Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

OPTIMIZING RESOURCE ALLOCATION IN NEXT-GENERATION OPTICAL ACCESS NETWORKS

by
Jingjing Zhang

To meet rapidly increasing traffic demands caused by the popularization of Internet and the spouting of bandwidth-demanding applications, Passive Optical Networks (PONs) exploit the potential capacities of optical fibers, and are becoming promising future-proof access network technologies. On the other hand, for a broader coverage area and higher data rate, integrated optical and wireless access is becoming a future trend for wireless access. This thesis investigates three next-generation access networks: Time Division Multiplexing (TDM) PONs, Wavelength Division Multiplexing (WDM) PONs, and WDM Radio-Over-Fiber (RoF) Picocellular networks.

To address resource allocation problems in these three networks, this thesis first investigates respective characteristics of these networks, and then presents solutions to address respective challenging problems in these networks. In particular, three main problems are addressed: arbitrating time allocation among different applications to guarantee user quality of experience (QoE) in TDM PONs, scheduling wavelengths optimally in WDM PONs, and jointly allocating fiber and radio resources in WDM RoF Picocellular networks. In-depth theoretical analysis and extensive simulations have been performed in evaluating and demonstrating the performances of the proposed schemes.

OPTIMIZING RESOURCE ALLOCATION IN NEXT-GENERATION OPTICAL ACCESS NETWORKS

by
Jingjing Zhang

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

March 2011

APPROVAL PAGE

OPTIMIZING RESOURCE ALLOCATION IN NEXT-GENERATION OPTICAL ACCESS NETWORKS

Jingjing Zhang

Nirwan Ansari, Dissertation Advisor — Date
Professor, Department of Electrical and Computer Engineering, NJIT

Roberto Rojas-Cessa, Committee Member — Date
Associate Professor, Department of Electrical and Computer Engineering, NJIT

Osvaldo Simeone, Committee Member — Date
Assistant Professor, Department of Electrical and Computer Engineering, NJIT

Guiling Wang, Committee Member — Date
Assistant Professor, Department of Computer Science, NJIT

Yanchao Zhang, Committee Member — Date
Associate Professor, School of Electrical, Computer, and Energy Engineering, Arizona State University

BIOGRAPHICAL SKETCH

Author: Jingjing Zhang

Degree: Doctor of Philosophy

Date: March 2011

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2011
- Master of Science in Electrical Engineering,
  Shanghai Jiao Tong University, Shanghai, China, 2006
- Bachelor of Arts in Electrical Engineering,
  Xi'an Institute of Posts and Telecommunications, Xi'an, China, 2003

Major: Electrical Engineering

Presentations and Publications:

J. Zhang and N. Ansari, "On OFDMA Resource Allocation and Wavelength Assignment in OFDMA-based WDM Radio-over-fiber Picocellular Systems," to appear in IEEE Journal on Selected Areas in Communications: Special Issue on Distributed Broadband Wireless Communications.

J. Zhang and N. Ansari, "Scheduling WDM EPON with Non-zero Laser Tuning Time," to appear in IEEE/ACM Transactions on Networking.

J. Zhang and N. Ansari, "Towards Energy-efficient 1G-EPON and 10G-EPON with Sleep-aware MAC Control and Scheduling," IEEE Communications Magazine, pp. s33-s38, No. 2, Feb. 2011.

J. Zhang and N. Ansari, "On the Capacity of WDM Passive Optical Networks (PONs)," IEEE Transactions on Communications, Vol. 59, No. 2, pp. 552-559, Feb. 2011.

J. Zhang and N. Ansari, "On Optimizing End-to-End QoE in Next Generation Networks: Challenges and Possible Solutions," to appear in IEEE Communications Magazine.

J. Zhang, N. Ansari, Y. Luo, and F. Effenberger, "Next-generation PONs: A Performance Investigation of Candidate Architectures for Next Generation Access Stage 1," IEEE Communications Magazine, Vol. 47, No. 8, pp. 49-57, August 2009.

J. Zhang and N. Ansari, "An Application-oriented Resource Allocation Scheme for EPON," IEEE Systems Journal, Vol. 4, No. 4, pp. 424-431, 2010.

J. Zhang and N. Ansari, "Design of WDM PON with Tunable Lasers: the Upstream Scenario," IEEE/OSA Journal of Lightwave Technology, Vol. 28, No. 2, pp. 228-236, 2010.

J. Zhang and N. Ansari, "On Optimizing the Tradeoff between the AWG Cost and Fiber Cost in Deploying WDM PONs," IEEE/OSA Journal of Optical Communications and Networking, Vol. 1, No. 5, pp. 352-365 Oct. 2009.

J. Zhang, N. Ansari, Y. Jin, and W. Hu, "Dichotomy Slot Allocation: A QoS Guaranteed Scheduling Algorithm for Input-Queued Switches," IEEE Systems Journal, Vol. 4, No. 1, pp. 74-83 March 2010.

J. Zhang, T. Wang, and N. Ansari, "An Efficient MAC Control Protocol for Asynchronous ONUs in OFDMA PON," to appear in OSA OFC, Los Angeles, CA, USA, 2011.

J. Zhang and N. Ansari, "Dynamic Time Allocation and Wavelength Assignment in Next Generation Multi-rate Multi-wavelength Passive Optical Networks," IEEE ICC, South Africa, 2010.

J. Zhang and N. Ansari, "On Analyzing the Capacity of WDM PONs," IEEE GLOBECOM, Honolulu, Hawaii, USA, 2009.

J. Zhang and N. Ansari, "Utility Max-min Fair Resource Allocation for Diversified Applications in EPON," AccessNets, Hongkong, China, 2009.

J. Zhang, Y. Jin, N. Ansari, and W. Hu, "Dichotomy Slot Allocation: A Low-Jitter Scheduling Scheme for Input-Queued Switches," IEEE Workshop on High Performance Switching and Routing (HPSR), Brooklyn, New York, USA, 2007.

To my beloved family

ACKNOWLEDGMENT

First and foremost I would like to thank my advisor, Professor Nirwan Ansari, for his advice, patience, guidance, encouragement, and support. His efforts in helping me to continually improve myself as a researcher are deeply appreciated. Without his help, this thesis can never be done.

To my committee members, Professor Roberto Rojas-Cessa, Professor Yanchao Zhang, Professor Osvaldo Simeone, and Professor Guiling Wang, I thank them for their time and advisement.

I want to express my heartfelt thanks to my dear parents, brother, grandparents, uncles and aunties. Their encouragement, sacrifices, and love have meant more to me than they can imagine. This thesis is dedicated to them.

Lastly, I want to thank my friends Chao Zhang, Jing Shi, Tao Han, Rui Zhang, Yunzhong Liu, Lin Cai, Ziqian Dong, Pitipatana Sakarindr, Amey Shevtekar, Mingzeng Cao, Renita Machado, Zhen Qin, Thomas Lo, and many others, who have given me many supports during the difficulties of my doctoral studies.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

As compared to the currently widely-deployed broadband access technologies such as DSL with various flavors, Cable modems, and hybrid fiber coaxial (HFC), passive optical networks (PON) and Fiber-to-the-Home cater for much higher speed access for new applications and services. As of the first quarter of 2009, there were over 30.8 million FTTH/FTTB subscribers in Asia-Pacific including South Korea, Hong Kong, Japan, Taiwan, Singapore, and China. The household penetrations in all of these countries are still growing.

As a point-to-multipoint network architecture, PON incurs a lower cost of optical cables and central office equipments as compared to the point to point architectures. PON contains two major components: the optical line terminal (OLT) at the central office and a number of optical network units (ONUs) near subscribers. OLT delivers its downstream bandwidth to ONUs via optical fibers in the optical distribution network (ODN). The upstream data traffic from ONUs are multiplexed and sent to OLT.

There are various flavors of PON technologies including A(ATM)PON, B(Broadband)PON, E(Ethernet)PON, and G(Gigabit)PON, among which GPON and EPON constitute two major flavors of currently commercially deployed PON systems. As standardized in IEEE 802.3ah, EPON uses symmetric 1 Gb/s upstream and downstream rates. GPON, as standardized in ITU-T G.984, provides 2.488 Gb/s downstream bandwidth and 1.244 Gb/s upstream bandwidth. However, with the popularization of Internet and spouting of bandwidth-demanding applications such as IPTV, both EPON and GPON may not be able to satisfy users' bandwidth requests. For future-proof, IEEE investigated and standardized 10G EPON in P.802.3av on

September 11, 2009. At the same time, Full Service Access Network (FSAN) working group and ITU-T are now studying next generation access (NGA) based on GPON. FSAN has planned two stages of NGA evolution: NGA1 and NGA2. NGA1 focuses on PON technologies compatible with GPON standards (ITU-T G.984 series) and the current Optical Distribution Network (ODN). In contrast, NGA2 is a long-term solution with an entirely new optical network type. The objective of NGA2 is to provision an independent PON scheme, without being constrained by the GPON standards and the currently deployed outside plant.

From the MAC layer's perspective, GPON, EPON, NGA1, and 10GEPON all can be considered as TDM systems, where the time allocation is a critical factor in achieving the system performance. Formerly, many scheduling algorithms have been proposed to guarantee QoS, ensure fairness, and maximize the link utilization in these networks. Currently, the access network has to accommodate many more classes of applications, each of which may have a specific requirement on quality of service parameters, such as delay, jitter, loss, and throughput. How to guarantee the specific QoS requirements of each application, and how to guarantee user Quality of Experience (QoE) are major issues, which are addressed in Chapter 2.

TDM systems increase their bandwidth provisioning by upgrading the data rate in the wavelength channel. An alternative scheme of increasing bandwidth is to use multiple wavelengths. The latter approach can be realized by WDM PONs such as NGA2. WDM PONs was proposed as early as 1989. However, the requirements of WDM devices make the network cost-prohibitive, thus hampering the deployment of WDM PONs. Recently, owing to the maturity of several WDM devices such as AWG, tunable transmitters, and RSOA, research on WDM PONs has been thriving. Owing to the nature of multi-wavelength support, wavelength scheduling constitutes another important issue in WDM PONs besides time allocation. In addition, tunable transmitters may see widespread usage in WDM PONs owing to their color-free

property. Tunable transmitters can be based on many technologies, and have different wavelength switching time depending on the specific adopted technology. The wavelength switching time of some technologies may be negligible and may also be non-negligible as compared to the duration of the resource allocation cycle. In Chapter 3, wavelength scheduling algorithms for various scenarios of wavelength switching time of tunable transmitters are proposed.

Integrated optical and wireless access network is another focus of this thesis as optical and wireless integration is becoming a clear trend in the years to come. This thesis specially focuses on Radio-over-fiber (RoF) picocellular networks, which distribute antennas over a cell to get closer to mobile users. By doing so, the signal to noise ratio (SNR) at the receiver can be increased, and thus the wireless access data rate can be increased accordingly. The coverage area of each antenna is greatly reduced as compared to the conventional cell, thus resulting in the sharing of wireless resources among a smaller number of users, and increasing the bandwidth share of each user. In Chapter 4, the joint optical and radio resource allocation problem in this network is addressed.

CHAPTER 2

RESOURCE ALLOCATION IN TDM PONS

In currently deployed TDM PONs such as Gigabit-capable PON (GPON) and Ethernet PON (EPON), upstream/downstream traffic are time division multiplexed (TDM) onto their respective dedicated wavelength channel. Owing to the sharing of the transmission media, proper schemes are required to arbitrate the bandwidth allocation among multiple ONUs, which may carry a variety of applications. To guarantee QoS for applications of ONUs, GPON [1] defines five types of traffic containers, each of which is characterized by four kinds of bandwidth components. A strict priority hierarchy of these four types of bandwidth components is employed in bandwidth allocation. Regarding EPON, although bandwidth allocation schemes are not specified, incorporating DiffServ framework into EPON scheduling schemes has been extensively studied [2–4].

These existing class-based bandwidth arbitration schemes in PONs are generally of coarse granularity, which, however, can hardly facilitate any particular QoS profile, required by newly emerging diversified applications, such as video streaming and e-science. These applications impose different QoS requirements as compared to those demanded by traditional video, voice, and data traffic. For example, large file transfer among e-science computing sites, on one hand, has strict throughput requirements, and hence possesses higher priorities over traditional data traffic. On the other hand, it is not delay sensitive as compared to voice and video traffic. For efficient QoS provisioning, the bandwidth allocation scheme is desired to consider these diversified QoS requirements and facilitate QoS profiles to provide a wide variety of services.

Driven by the desire of sharpening their competitive edges, service providers may regard user quality of experience (QoE) as the basis for making network management

and control decisions [5–7]. Network-level QoS performances are correlated to user QoE. Generally, better network-level QoS performances result in higher QoE. Formerly, to bridge QoE and QoS, experimental measurements and theoretical modeling have been broadly employed to characterize QoE as functions of QoS parameters for different applications [8–11].

In this chapter, QoE is considered as a function of network layer delay and loss performances, and QoE functions of applications are assumed to be known *a priori*. The following assumptions regarding user QoE are further made. First, the QoE of a user depends on the overall impacts of its online sessions. Usually, the subjective experience of a human being is significantly degraded by the poor performance of any online session of the user no matter how high the qualities of other sessions of the user are [11]. Therefore, the session with the lowest QoE score is used to characterize QoE of this particular user. Second, QoE of a user session depends on the performances of all network-level QoS parameters. Since degradation in any of QoS parameters may result in poor QoE, QoE of a user session is assumed to achieve a certain score only when both loss and delay satisfy certain criteria. With these assumptions, the problem of maximizing user QoE is recasted as the problem of finding a schedule which can guarantee corresponding QoS performances to guarantee the maximum QoE score for all user sessions for given incoming downstream or upstream traffic.

In TDM PON where multiple sessions from multiple users share one communication channel, there exists a tradeoff between performances of sessions. Moreover, for a given user session, there exists a tradeoff between its delay and loss performances too. A smaller delay may be achieved at the cost of a higher traffic loss ratio. Similarly, a smaller loss ratio may be obtained by further delaying the scheduling of the traffic. Therefore, when the best delay and loss performances cannot be simultaneously achieved for all sessions, optimizing the delay-loss tradeoff becomes a key issue in guaranteeing the maximum QoE for all users in TDM PONs.

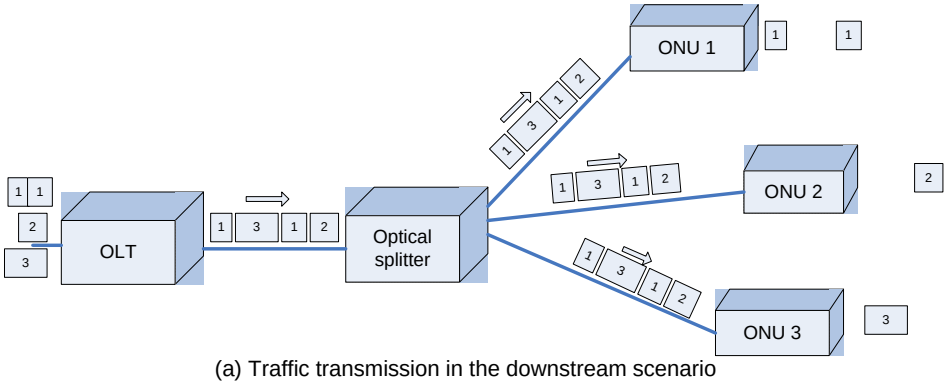
In this chapter, an optimal scheduling scheme which can guarantee QoE for all users in TDM PON is proposed. Formerly, Uysal *et al.* [12] proposed lazy schedules which minimize the energy consumption in transmitting packets over a wireless link. The authors first constructed an optimal offline schedule, and then proposed the online schedule based on the optimal offline schedule. Similarly, to address the problem of maximizing user QoE, the offline scheduling problem with the assumption that the arrival traffic over the entire time span is known *a priori* is first addressed. Based on the optimal offline scheduling scheme, an online scheduling scheme is first proposed to make its bandwidth allocation decision based on the information of traffic which arrives at ONU or OLT prior to the decision making time. After that, issues on the estimation of the request arrival time and request size in the upstream scenario are discussed. Extensive simulation results are included, and they show that the maximum QoE score can be achieved even when the traffic load is as large as 0.8585.

The rest of the chapter is organized as follows. Section 2.1 presents the system model and the definition of QoE functions. Section 2.2 discusses the scheduling scheme for the downstream scenario. Specifically, an optimal offline scheduling scheme is presented first followed by an online scheduling scheme. Section 2.3 discusses the estimation of the traffic arrival time and size in the upstream scenario. Section 2.4 presents simulation results in detail. The summary is included in Section 2.5.

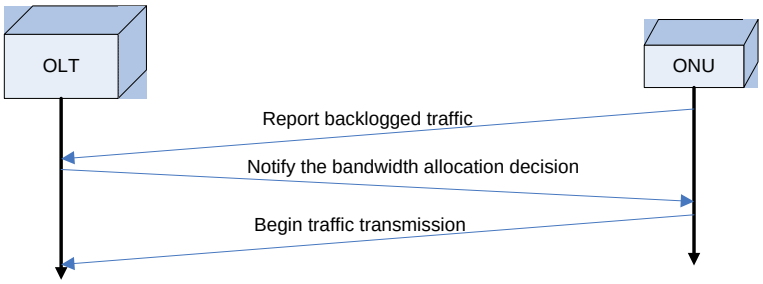
2.1 System Model

2.1.1 Network Architecture

Figure 2.1 shows the typical PON architecture, where a number of ONUs are connected to the OLT via optical fibers, and both the upstream and downstream transmissions are TDM based. This chapter considers the fiber-to-the-home (FTTH) scenario, where each ONU is a home user and has a number of online sessions.



(a) Traffic transmission in the downstream scenario



(b) MAC control in the upstream scenario

Figure 2.1 Illustration of the PON architecture.

For the downstream traffic transmission scenario as shown in Figure 2.1 (a), This chapter assumes OLT has virtual output queues, each of which corresponds to one user session at one ONU. Upon the arrival of the downstream traffic of each user session, OLT arbitrates the bandwidth allocation among different user sessions at different ONUs, and then dispatch the arrival traffic.

Consider deterministic entry of the downstream traffic into the network. Denote $\{\mathbf{r}_{i,j}^k\}_{i,j,k}$ as the kth request of session j at ONU i. Each request corresponds to some packets which arrives during a continuous time duration. $\mathbf{r}_{i,j}^k$ is associated with a double sequence $(a_{i,j}^k, x_{i,j}^k)$, where $x_{i,j}^k$ is the size expressed in time duration of request $\mathbf{r}_{i,j}^k$, and $a_{i,j}^k$ is the time at which request $\mathbf{r}_{i,j}^k$ arrives. For the downstream case, both $\{x_{i,j}^k\}_{i,j,k}$ and $\{a_{i,j}^k\}_{i,j,k}$ are known to the decision maker OLT.

For the upstream scenario, upstream traffic arrives at ONUs, and is not explicitly known to OLT. Since the decision maker OLT does not know the exact arrival time and

size of upstream arrival traffic, the upstream scenario in PONs is more complicated than the downstream scenario. Typically, as shown in Figure 2.1 (b), the life cycle of a packet in TDM PONs such as GPON and EPON can be described as follows.

- *At the ONU side*: The upstream data packet is queued in its corresponding buffer upon arrival, and waits to be reported to the OLT about its existence. The ONU reports its queue length at a proper time, which is taken after the data transmission in EPON and at the beginning of the frame in GPON.

- *At the OLT side*: OLT collects reports from ONUs, makes bandwidth allocation decisions, and then sends out its decisions to ONUs.

- *At the ONU side*: An ONU receives the bandwidth allocation decision sent from OLT, and then transmits packets queued in its buffer using the granted time duration.

Different from the downstream scenario where OLT can closely keep track of the traffic arrival information, OLT, in the upstream scenario, does not directly own the exact information of the traffic arrival time and size, i.e., $(a_{i,j}^k, x_{i,j}^k)$ needs to be estimated based on ONU reports.

2.1.2 QoE

QoE of an User: For user QoE, as aforementioned in the introduction, this chapter identifies QoE of a user based on the performance of the session with the lowest QoE score among all sessions of the user. That is to say, QoE of user i equals to $\min_j u_{i,j}$, where $u_{i,j}$ refers to QoE of session j of ONU i, and it can be regarded as the QoE score of user i when user i has session j only. Achieving a given QoE score v for user i implies that

$$u_{i,j} \geq v, \forall j \tag{2.1}$$

QoE of an User Session: As stated in the introduction, QoE of a session achieves QoE score v only when its loss and delay satisfy certain criteria. Denote $loss_{i,j}^{-1}(v)$ and $delay_{i,j}^{-1}(v)$ as the maximum allowable loss and delay for session j at ONU i to achieve QoE score v, respectively, i.e.,

$$\begin{cases} loss_{i,j}^{-1}(v) = \arg\max_{loss}\{u_{i,j} = v\} \\ delay_{i,j}^{-1}(v) = \arg\max_{delay}\{u_{i,j} = v\} \end{cases}$$

Then, the necessary and sufficient condition to guarantee QoE score v for user i is that

$$l_{i,j} \leq loss_{i,j}^{-1}(v) \text{ and } d_{i,j} \leq delay_{i,j}^{-1}(v), \forall j \tag{2.2}$$

where $l_{i,j}$ and $d_{i,j}$ denote the loss and delay performances of session j at ONU i, respectively.

In this chapter, QoE functions $u_{i,j}$, $loss_{i,j}^{-1}$, and $delay_{i,j}^{-1}$ is assumed to be known *a priori*. Investigating these QoE functions is rather challenging, and has received intensive research attention [8–11, 13, 14].

Delay and Loss: The subsequent question is how to define loss ratio and delay for user sessions. While many existing literatures employ average loss ratio as loss metric, the application level QoS perceived by end users is also affected by short-term loss patterns (loss burstiness and loss interval) [15, 16]. For delay, each traffic request is expected to have a bounded delay such that the delivery of this particular request is on time without degrading user's experience. The short term delay can also guarantee delay jitter performance [17]. Thereby, assume that $l_{i,j} \leq loss_{i,j}^{-1}(v)$ and $d_{i,j} \leq delay_{i,j}^{-1}(v)$. They further implying that delay and loss of every single request in the session are less than $loss_{i,j}^{-1}(v)$ and $delay_{i,j}^{-1}(v)$, respectively.

Denote $c_{i,j}^{k}$ and $y_{i,j}^{k}$ as the completion transmission time and the granted time duration of the kth request of session j at user i (request $\mathbf{r}_{i,j}^{k}$), respectively. Define

traffic loss ratio of request $\mathbf{r}_{i,j}^k$ as $(x_{i,j}^k - y_{i,j}^k)/x_{i,j}^k$, and the delay of request $\mathbf{r}_{i,j}^k$ as the difference between the request completion time and the request arrival time, i.e., $c_{i,j}^k - a_{i,j}^k$.

Then, mathematically, guaranteeing QoE score v of session j at user i implies that

$$\begin{cases} (x_{i,j}^k - y_{i,j}^k)/x_{i,j}^k \leq loss_{i,j}^{-1}(v), \forall k \\ c_{i,j}^k - a_{i,j}^k \leq delay_{i,j}^{-1}(v), \forall k \end{cases} \tag{2.3}$$

Further, guaranteeing QoE score v of user i implies that

$$\begin{cases} (x_{i,j}^k - y_{i,j}^k)/x_{i,j}^k \leq loss_{i,j}^{-1}(v), \forall k, \forall j \\ c_{i,j}^k - a_{i,j}^k \leq delay_{i,j}^{-1}(v), \forall k, \forall j \end{cases} \tag{2.4}$$

For ease of explanation, for any request $\mathbf{r}_{i,j}^k$, the largest QoE score v satisfying $(x_{i,j}^k - y_{i,j}^k)/x_{i,j}^k \leq loss_{i,j}^{-1}(v)$ and $c_{i,j}^k - a_{i,j}^k \leq delay_{i,j}^{-1}(v)$ is referred to as QoE of request $\mathbf{r}_{i,j}^k$ in the rest of the chapter.

Problem Formulation: Out of fairness concern, the objective is to achieve *max-min fairness* among QoE of all users in allocating resources [18, 19]. With the above definitions and assumptions, the problem of achieving max-min fairness can be formulated as:

Given traffic requests $\{\boldsymbol{r}_{i,j}^k\}$*, QoE functions* $u_{i,j}$*,* $loss_{i,j}^{-1}$*, and* $delay_{i,j}^{-1}, \forall i, \forall j$*, construct a schedule with the smallest delay* $l_{i,j}^k$ *and loss* $d_{i,j}^k$ *for all requests such that QoE score of any ONU* i *cannot be increased at the sacrifice of the decrease of QoE score of any other ONU whose QoE is already smaller than that of ONU* i*.*

This chapter first focuses on the downstream bandwidth allocation problem in which the decision maker OLT can track $\{(a_{i,j}^k, r_{i,j}^k)\}$ of all requests. Subsequently, the estimation of $\{(a_{i,j}^k, r_{i,j}^k)\}$ will be discussed in the upstream scenario.

2.1.3 Related Works

Formerly, many DBA algorithms have been proposed to ensure fairness and guarantee QoS for queues of ONUs in EPON and GPON. For example, the DiffServ framework was proposed to be incorporated into the DBA to provision QoS guarantees [2, 3, 20, 21]. However, the employed strict-priority discipline when incorporating the DiffServ framework into DBA raises the so-called *light-load penalty* problem [2]. To compensate for the light-load penalty, Kramer *et al.* [2] proposed a two-stage queueing system, where a proper local queue management scheme and a priority-based scheduling algorithm are employed. Kim *et al.* [22] adopted weighted fair queuing to give queues with different weights for their priorities. IPACT-LS [23] prevents ONUs from monopolizing the bandwidth by setting a predetermined maximum of the granted resources. Assi *et al.* [3] proposed to satisfy requests from light-load ONUs first, while penalizing heavily-loaded ONUs. Different from these existing works, from the perspective of user QoE, this chapter presents a scheme to achieve max-min QoE fairness for users in PONs.

Regarding related works on QoE-oriented scheduling, formerly, from the application-level QoS perspective, Cao *et al.* [24] and Wang *et al.* [25] modeled application utility as a function of available bandwidth, and employed four general utility shapes, namely, elastic utility, real-time utility, rate-adaptive utility, and step-wise utility, to model four classes of applications. Thakolsri *et al.* [26] maximized the sum of utilities by using a fast greedy algorithm which searches only the boundary of the utility space in High Speed Downlink Packet Access (HSDPA). Kuo *et al.* [27] investigated utility-based resource allocation for soft QoS traffic in infrastructure-based wireless networks, where soft QoS traffic refers to the traffic which demands a certain amount of bandwidth for normal operation but allows some flexibility when the given bandwidth is close to the preferred value. By modeling utilities as functions of available resources, the authors then proposed resource

allocation schemes to optimally schedule these soft QoS traffic. Rather than modeling QoE as a function of available bandwidth, this chapter models QoE directly as a function of delay and loss performances such that not only the amount of arrival traffic but also the traffic arrival time can be taken into account in arbitrating bandwidth allocations among users.

In [28, 29], application utilities are defined as functions of loss, delay, and jitter. Then, with the focus on the resource allocation in one cycle only, a heuristic resource allocation algorithm was proposed to guarantee high utilities for queues. In this chapter, without restricting the resource allocation in one cycle, the problem of allocating resources across the entire time span is formally formulated, and the optimal offline scheduling scheme is investigated first. Then, an online version of the offline scheduling scheme is derived.

2.2 The Downstream Scenario

This chapter starts from the offline scheduling which assumes that downstream traffic arrival during the whole time span, i.e., $\{(a_{i,j}^k, r_{i,j}^k)\}_{k=1}^{\infty}, \forall i, \forall j$, are known in advance. Then, inspired from the optimal offline scheduling scheme, an online scheduling scheme for real-time implementation is proposed. In the proposed scheme, the decision maker does not know the future incoming traffic at the decision making time.

2.2.1 Offline Scheduling

Maximize the Minimum QoE of all Users: First, the problem of achieving a given QoE score for all users is investigated. For a given QoE score v, based on Constraints (2.4), guaranteeing v for all users implies that

$$\begin{cases} (x_{i,j}^k - y_{i,j}^k)/x_{i,j}^k \leq loss_{i,j}^{-1}(v), \forall i, \forall j, \forall k \\ c_{i,j}^k - a_{i,j}^k \leq delay_{i,j}^{-1}(v), \forall i, \forall j, \forall k \end{cases} \tag{2.5}$$

Then, the problem is equivalent to the problem of constructing a schedule with $y^k_{i,j}$ and $c^k_{i,j}$ satisfying $y^k_{i,j} \geq x^k_{i,j} \cdot (1 - loss^{-1}_{i,j}(v))$ and $c^k_{i,j} \leq a^k_{i,j} + delay^{-1}_{i,j}(v)$, $\forall i, \forall j, \forall k$. The following Algorithm 1 addresses this problem.

Algorithm 1 Guarantee QoE score v for all users

1: Consider $x^k_{i,j} \cdot (1 - loss^{-1}_{i,j}(v))$ as the grant size $y^k_{i,j}$ for request $\mathbf{r}^k_{i,j}$.
2: Consider $a^k_{i,j} + delay^{-1}_{i,j}(v)$ as the scheduling deadline of request $\mathbf{r}^k_{i,j}$.
3: $t = 0$
4: **while** There exists unscheduled request **do**
5: Among unscheduled request $\mathbf{r}^k_{i,j}$ which arrive before t, select the one with the earliest deadline.
6: **if** request $\mathbf{r}^{k'}_{i',j'}$ arrives between t and $t + y^k_{i,j}$ and has deadline earlier than request $\mathbf{r}^k_{i,j}$ **then**
7: $c^k_{i,j} = a^{k'}_{i',j'}$
8: **else**
9: $c^k_{i,j} = t + y^k_{i,j}$ and denote request $\mathbf{r}^k_{i,j}$ as scheduled.
10: **end if**
11: Allocate the time between t and $c^k_{i,j}$ to request $\mathbf{r}^k_{i,j}$.
12: $t = c^k_{i,j}$
13: **end while**

Algorithm 1 is essentially a preemptive earliest-deadline-first (EDF) scheduling algorithm. Among all unscheduled requests, the one with the earliest deadline is scheduled with the highest priority. When a request is being scheduled, the scheduling can be preempted and resumed later if another request with an earlier deadline arrives.

Theorem 1. *If the schedule constructed by Algorithm 1 cannot guarantee all users with QoE score v, then, no schedule exists to guarantee all users with QoE score v.*

Proof. Assume the deadline of request $\mathbf{r}_{i,j}^k$ is violated in the schedule constructed by Algorithm 1, i.e., $c_{i,j}^k > a_{i,j}^k + delay_{i,j}^{-1}(v)$. The next will show that there does not exist a schedule which can schedule all requests prior to their respective deadlines.

It is not difficult to see that the scheduling policy described in Algorithm 1 is work-conservative. Then, the earliest time to schedule all requests with deadline earlier than $a_{i,j}^k + delay_{i,j}^{-1}(v)$ is $c_{i,j}^k$. If request $\mathbf{r}_{i,j}^k$ is scheduled earlier, there must exist some other request $\mathbf{r}_{i',j'}^{k'}$ with deadline earlier than $a_{i,j}^k + delay_{i,j}^{-1}(v)$ that completes its transmission at time $c_{i,j}^k$. In this case, the deadline request $\mathbf{r}_{i',j'}^{k'}$ is violated. □

If there exists a schedule guaranteeing QoE score v for all users, v is said to be achievable for all users. With the problem of guaranteeing QoE score v for all users being addressed, the minimum achievable QoE score for all users can be maximized by trying different v using the bisection method [30] which is described in Algorithm 2. The main idea is as follows: Let h and l be the highest and lowest value of QoE functions of all sessions. v is first let to be equal to h, and then whether v is achievable for all users is checked. If v is not achievable, h is updated to be v, and v is decreased to the midpoint between h and l; otherwise, h is increased to v, and v is increased to the midpoint between h and l. The above process is performed recursively until h and l are close enough to each other.

Further Increase of QoE of Some Sessions if Possible: In the above, the schedule with the maximum achievable QoE score for all users is obtained. Although there does not exist a better schedule which can increase QoE score of all sessions of all users at the same time, QoE score of some sessions of some users may be increased without decreasing those of other sessions. For example, assume user sessions can be classified into two classes, in which the two classes with the highest QoE scores are μ and ν, respectively. Without loss of generality, assume $\mu < \nu$. Then, with Algorithm 2, a QoE score higher than μ cannot be achieved for sessions in the second class whose QoE score can be as high as ν.

Algorithm 2 Maximize the minimum QoE among all users by employing the bisection method

1: Denote h and l as the highest and lowest value of QoE functions of all applications, respectively.
2: $v = h$
3: **while** h and l are not close enough **do**
4: **if** v is achievable for all users **then**
5: $l = v$
6: **else**
7: $h = v$
8: **end if**
9: $v = (h + l)/2$
10: **end while**

To further increase QoE scores of some sessions, this chapter proposes Algorithm 3 whose idea is similar to water-filling. In Algorithm 3, all sessions whose QoE scores have not reached their respective highest score are considered, and maximizing the minimum QoE of all these sessions is considered as the objective. The process is repeated until the QoE score cannot be further increased for any session. The detailed procedure involved in Line 3 of Algorithm 3 is similar to those described in Algorithm 2.

Algorithm 3 Further increase QoE of some user sessions

1: **while** QoE score can be increased for some session **do**
2: Decide the sessions whose QoE can be possibly increased, i.e., sessions which have not reach their respective highest QoE scores yet.
3: Maximize the minimum QoE score for these sessions.
4: **end while**

Further Improvement of Delay and Loss of Some Requests: Algorithm 2 and Algorithm 3 constructed a schedule with which QoE of any session i cannot be increased without the sacrifice of QoE of some other sessions whose QoE scores are already smaller than that of session i. Although QoE of a session cannot be increased, delay and loss of some requests in the session may be enhanced for two cases. First, the fact that QoE of a session cannot be increased implies that delay and loss performances of all requests in the session cannot be decreased simultaneously. However, for some requests in a session, their delay and loss performances may be possibly enhanced. This case is referred to *Case 1*. Second, to achieve the highest QoE score, usually, delay and loss do not need to be as small as zero owing to the human auditory and visual limitation. In the schedule produced by Algorithm 2 and Algorithm 3, if the highest QoE score of a session is achieved, delay and loss ratio of requests in the session equals to the maximum allowable value to achieve the highest QoE score. It is possible to further reduce delay and loss ratio from the maximum allowable value for some requests, or even all requests of a session. This case of further reducing delay and loss is referred to as *Case 2*.

Algorithm 4 describes the scheme of further enhancing delay and loss performances of some requests. In the real implementation, delaying the scheduling of a request is much easier than dropping some traffic of the request. Thus, delaying traffic is preferable over dropping traffic when either one has to be chosen.

In Algorithm 4, delay and loss performances are first enhanced for requests in Case 1. In the "while" loop described in Lines 2 - 5 of Algorithm 4, the minimum QoE score of requests which can be possibly increased is maximized. Then, the requests whose QoE score can be increased are updated, and the same process repeats until no request can have an increased QoE score without degrading QoE of others. In Lines 7 - 8 of Algorithm 4, the algorithm first obtains all requests which achieve their respective highest QoE scores. Then, the algorithm tries to gradually reduce loss ratio

Algorithm 4 Further increase QoE of some requests

1: /*Enhance delay and loss performances for Case 1*/
2: **while** QoE score can be increased for some request **do**
3: Among all requests of all sessions, decide requests whose QoE score cannot be further increased.
4: Obtain the maximum QoE score which can be guaranteed for all the other requests.
5: **end while**
6: /*Enhance delay and loss performances for Case 2*/
7: Obtain requests being scheduled with the highest QoE score.
8: Gradually decrease the loss ratio of these requests until no further loss can be made.
9: Obtain requests being scheduled with zero loss ratio.
10: Gradually decrease the delay of these requests until no further delay can be made.

of these requests until no further loss ratio can be made. Lines 9 - 10 of Algorithm 4, first obtain all requests with zero loss ratio, and then tries to gradually decrease delay of these requests until no further delay can be made.

2.2.2 Online Scheduling

The offline scheduling assumes that the arrival traffic across the entire time span is known to the decision maker. In the online scheduling for real implementation, the decision maker does not own the information of the future arrival traffic at the decision making instance.

In the online scheduling, at any given decision making time t, the decision maker decides the bandwidth to be allocated to unscheduled requests based on the information of requests with arrival time prior to t, i.e., $\{\mathbf{r}_{i,j}^k | a_{i,j}^k \leq t\}$. The scheduler optimistically assumes that there are no requests coming in the future to make the

best decision for the existing requests. The disability of foreseeing future incoming requests of the online scheduling will result in suboptimal solution as compared to the offline scheduling.

Algorithm 5 The online scheduling algorithm to achieve max-min QoE fairness

1: Denote the newly arrival request at time t as $\mathbf{r}_{i,j}^k$.
2: Calculate the deadline of scheduling request $\mathbf{r}_{i,j}^k$ with the highest QoE score.
3: Insert the request into the unscheduled request list, and then sort them in the ascending order of scheduling deadline.
4: Schedule request $\mathbf{r}_{i,j}^k$ according to the deadline order.
5: **if** The scheduling deadline of request $\mathbf{r}_{i,j}^k$ is violated **then**
6: Calculate the maximum allowable loss ratio to guarantee the highest QoE score of the request.
7: Let the loss ratio $l_{i,j}^k = 0$
8: **while** The deadline cannot be met and $l_{i,j}^k$ is no greater than the maximum allowable one **do**
9: Gradually increase $l_{i,j}^k$.
10: Update the grant bandwidth $y_{i,j}^k$ to request $\mathbf{r}_{i,j}^k$ as $x_{i,j}^k \cdot (1 - l_{i,j}^k)$, and check whether the request can be scheduled prior to the deadline
11: **end while**
12: **if** The scheduling deadline of the request is still not met **then**
13: Consider the current unscheduled requests and request $\mathbf{r}_{i,j}^k$, and employ Algorithm 2 to maximize the minimum QoE among all these requests.
14: **end if**
15: **end if**

Algorithm 5 describes the online version of the optimal offline scheduling scheme. In Lines 1-4, Algorithm 5 tries to schedule the newly incoming request $\mathbf{r}_{i,j}^k$ with zero loss ratio. If a schedule cannot be constructed to guarantee the request with zero

loss ratio as stated in Line 5, the request with some degraded loss performance but still achieve the highest QoE score will be scheduled. More specifically, the maximum allowable loss ratio which can guarantee the highest QoE score is first calculated. Then, the loss ratio is gradually increased until the traffic can be scheduled before the deadline or the loss ratio reaches the maximum allowable value (see Lines 6-11). If the highest QoE score cannot be guaranteed for this request as described in Line 12, the bisection method is applied to maximize the minimum QoE of all unscheduled requests as well as the newly arrival one.

The online version of the optimal offline scheduling scheme makes bandwidth allocation decisions every time a new request arrives. This imposes high requirement on the computational burden of the scheduler. To reduce the computational burden, instead of calculating a new schedule each time a request arrives, an alternative more practical method is to calculate the schedule until the channel is about to turn idle. This scheduling framework is referred to as just-in-time scheduling [31]. This scheme will be applied in the simulation section.

2.3 The Upstream Scenario

Different from the downstream scenario, the bandwidth allocation decision maker OLT does not own the exact information of the arrival time $a_{i,j}^k$ and size $x_{i,j}^k$ of each request. In this section, the estimation of $a_{i,j}^k$ and $x_{i,j}^k$ in the upstream scenario is discussed.

Several more notations to facilitate the estimation of the arrival upstream requests are first introduced. Denote $\alpha_{i,j}^k$ as the time that the kth report of session j at ONU i. Then, at time $\alpha_{i,j}^k + RTT_i/2$, OLT receives the kth report of session j at ONU i, where RTT_i is the round trip time between ONU i and OLT. Denote $\Delta_{i,j}^k$ as the interval between the $(k-1)$th and kth request sending time from session j of ONU i, i.e., $\Delta_{i,j}^k = [\alpha_{i,j}^{k-1}, \alpha_{i,j}^k]$. Denote $tr_{i,j}^k$, $dr_{i,j}^k$, and $ar_{i,j}^k$ as the transmitted

traffic, dropped traffic, and arrival traffic during interval $\Delta_{i,j}^k$. Then, at time $\alpha_{i,j}^k$, the backlogged traffic of session j at ONU i equals to

$$\sum_{p=1}^{k} (ar_{i,j}^p - tr_{i,j}^p - dr_{i,j}^p)$$

Denote $\gamma_{i,j}^k$ as the reported queue length (in time) contained in the kth report of session j at ONU i. Then, $\gamma_{i,j}^k = \sum_{p=1}^{k} (ar_{i,j}^p - tr_{i,j}^p - dr_{i,j}^p)$. The following can be further obtained.

$$ar_{i,j}^k = \gamma_{i,j}^k + \sum_{p=1}^{k} (tr_{i,j}^p + dr_{i,j}^p) - \sum_{p=1}^{k-1} ar_{i,j}^p \tag{2.6}$$

On the right side of Eqt. (2.6), $\gamma_{i,j}^k$ is the request reported to the access node. Both $tr_{i,j}^p$ and $dr_{i,j}^p$ are decided by OLT. Hence, by recursion, OLT can infer the arrival traffic $ar_{i,j}^k$ during time interval $\Delta_{i,j}^k$.

Besides the newly arrival traffic $ar_{i,j}^k$ during interval $\Delta_{i,j}^k$, OLT can estimate the arrival time of all backlogged traffic at time $\alpha_{i,j}^k$. Assume infinite buffer size at the user side, and the dropping is from the head of the queue. As compared to the scheme of dropping the latest arrival packets, dropping the oldest packets first can let precious resources be used for transmitting traffic with smaller delay, and hence larger QoE.

Then, among the total $\sum_{p=1}^{k} ar_{i,j}^p$ arrival traffic before time $\mathbf{a}_{i,j}^k$, the first $\sum_{p=1}^{k} (tr_{i,j}^p + dr_{i,j}^p)$ arrival traffic is either transmitted or dropped, and the latest arrival $\sum_{p=1}^{k} (ar_{i,j}^p - tr_{i,j}^p - dr_{i,j}^p)$ traffic remains in the queue and is reported to the central access node. Among the $\gamma_{i,j}^k$ request traffic, assume $\eta_{i,j}^k(p)$ traffic arrives during time interval $\Delta_{i,j}^p$. Then, it can be obtained that

$$\begin{cases} \eta_{i,j}^k(k) = \min\{ar_{i,j}^k, \gamma_{i,j}^k\} \\ \eta_{i,j}^k(p) = \min\{ar_{i,j}^p, (\gamma_{i,j}^k - \sum_{m=l+1}^{k} \eta_{i,j}^{k,m})^+\} \end{cases} \tag{2.7}$$

where $x^+ = \begin{cases} x \text{ if } x > 0 \\ 0 \text{ otherwise} \end{cases}$.

After the arrival time and size of each request have been estimated, the bandwidth allocation problem is boiled down to the problem discussed in the downstream scenario. Then, online Algorithm 5 can be employed to obtain the optimal solution.

2.4 Simulation Results and Analysis

In this section, taking the upstream transmission in EPON for example, the performance of our proposed online scheduling scheme is investigated.

IEEE 802.3ah has standardized the Multi-Point Control Protocol (MPCP) as the MAC layer control protocol for EPON [4]. Specifically, MPCP defines two 64-byte control messages REPORT and GATE for the bandwidth arbitration in the upstream. ONUs report its backlogged traffic to OLT by sending REPORT. After collecting REPORT from ONUs, OLT dynamically allocates bandwidth to ONUs and informs its grant decisions to ONUs via GATE. The cycle duration in EPON can be dynamically adaptive to the traffic. In the simulation, instead of letting OLT calculate schedules every time a REPORT from an ONU arrives, OLT keeps collecting REPORTs from different ONUs, and makes bandwidth allocation decision just before the upstream wavelength channel becomes idle.

The interval between two consecutive bandwidth allocation decision making time is referred to as a dynamic bandwidth allocation (DBA) cycle in this chapter. The DBA cycle is adapted to traffic variation. The maximum DBA cycle is set as $2ms$, and the data rate is set as 1.25 Gb/s. Then, the maximum traffic transmitted during a cycle is 2.5 Mbits. When the total requests are below 2.5 Mbits, every request is granted with the bandwidth equaling to its request size. When the total

requests increase beyond 2.5 Mbits, OLT allocates the bandwidth in the cycle with the duration of $2ms$ to ONUs.

For ease of explanation, three kinds of DBA cycles are defined according to the traffic load. For one particular DBA cycle, if the total requested traffic is within the capacity of the cycle, the cycle is referred to as a *low-load cycle*; if the requested traffic is greater than the capacity of the cycle but every ONU can still get the highest QoE score by delaying the scheduling of some traffic to the next cycle or dropping some traffic, the cycle is referred to as *medium-load cycle*; if the QoE score of some ONUs falls below the highest score, the cycle is referred to as a *high-load cycle*. In a low-load cycle, only Lines 1 - 4 in Algorithm 5 are performed; in a medium-load cycle, Lines 1 - 11 in Algorithm 5 are performed; when it comes to high-load cycles, the entire Algorithm 5 is performed. The operation in low-load cycles is the simplest, whereas the operation in medium-load cycles involves more calculations, and the operation in high-load cycles incurs the highest computational burden.

The number of ONUs is set to be 16, and the round trip time between ONUs and OLT is set to be $125\mu s$. The simulation model is developed on the OPNET platform. Since self-similarity is exhibited in many applications, Each user session is input with self-similar traffic. The Pareto parameter is set as 0.8. The packet length is uniformly distributed between 64 bytes and 1500 bytes. Assume each ONU has five sessions corresponding to five kinds of applications. Each of the five sessions is entered with traffic with the same statistical characteristics. The input traffic of all ONUs obey the same distribution.

First, distributions of low-load cycles, medium-load cycles, and high-load cycles are investigated under different traffic loads. The simulation time is set to be 2.5 seconds. Assume the maximum allowable delay to achieve the highest QoE score of the fiver sessions in each ONU are 3 ms, 4 ms, 5 ms, 6 ms, and 7 ms, respectively, no traffic loss is allowed, and the precise QoE functions are unknown. Figure 2.2

presents simulation results of the distribution of the three kinds of cycles. It was shown that when the network load is about 0.8585, the majority of the cycles are low-load cycles, and the high-load cycles are very few. That is to say, the maximum QoE can almost be guaranteed under this case. When the network load is increased to 0.9152, the medium-load cycles are increased to around 11% of the total number of cycles, and the high-load cycles takes around 5% of the total number of cycles. Then, in 5% of all cycles, QoE of some sessions are degraded in some degree. When the network load increases to 0.96, the majority of cycles are high-load cycles, implying that tremendous amount of computation is required.

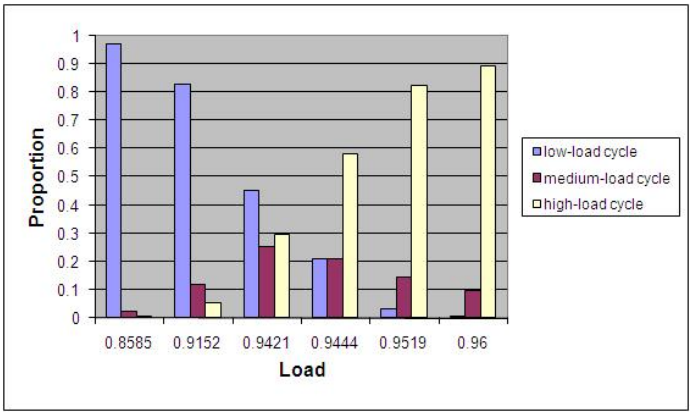


Figure 2.2 The proportion of three kinds of cycles under different traffic loads.

Figure 2.3 shows the throughput under different traffic loads, which is defined as the ratio of the sum of input traffic during the whole simulation time over the maximum traffic accommodated by the network. Throughput is defined as the ratio of the total amount of successfully transmitted traffic to the maximum traffic which can be accommodated by the network. Under low traffic load, all the requests can be successfully scheduled. Hence, the throughput increases with the increase of traffic load. When the traffic load is increased to a certain value, further increase will not increase the throughput of the network. As shown in Figure 2.3, the knee point happens when the network load is around 0.9421, where high-load cycles occupy less than 30% of the total cycles. In other words, most of the cycles will be either

low-load cycles or medium-load cycles, which do not involve much computation, and more importantly, the highest QoE score can be guaranteed in these low-load and medium load cycles.

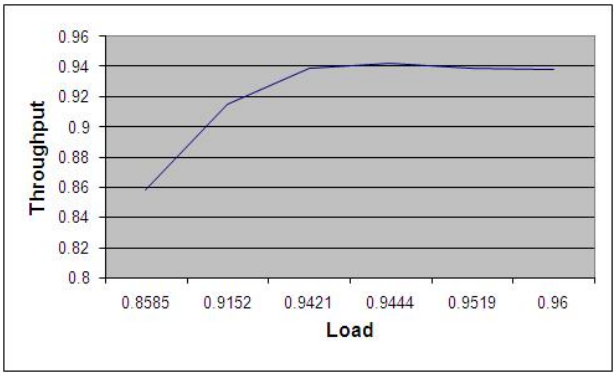


Figure 2.3 Throughput vs. traffic load.

Assume QoE functions are known. Two kinds of QoE functions will next be considered. One is a function of packet delay, and the other one is a function of loss ratio.

2.4.1 Packet Delay

First, QoE of a user session as a function of packet delay is considered, i.e., $u_{i,j}(loss, delay) = u^2_{i,j}(delay)$, defined as follows.

$$u^2_{i,1}(delay) = \begin{cases} 1 & delay \leq 3ms \\ e^{(delay-3)/3} & delay > 3ms \end{cases}, \forall i$$

$$u^2_{i,2}(delay) = \begin{cases} 1 & delay \leq 4ms \\ e^{(delay-4)/4} & delay > 4ms \end{cases}, \forall i$$

$$u^2_{i,3}(delay) = \begin{cases} 1 & delay \leq 5ms \\ e^{(delay-5)/5} & delay > 5ms \end{cases}, \forall i$$

$$u^2_{i,4}(delay) = \begin{cases} 1 & delay \leq 6ms \\ e^{(delay-6)/6} & delay > 6ms \end{cases}, \forall i$$

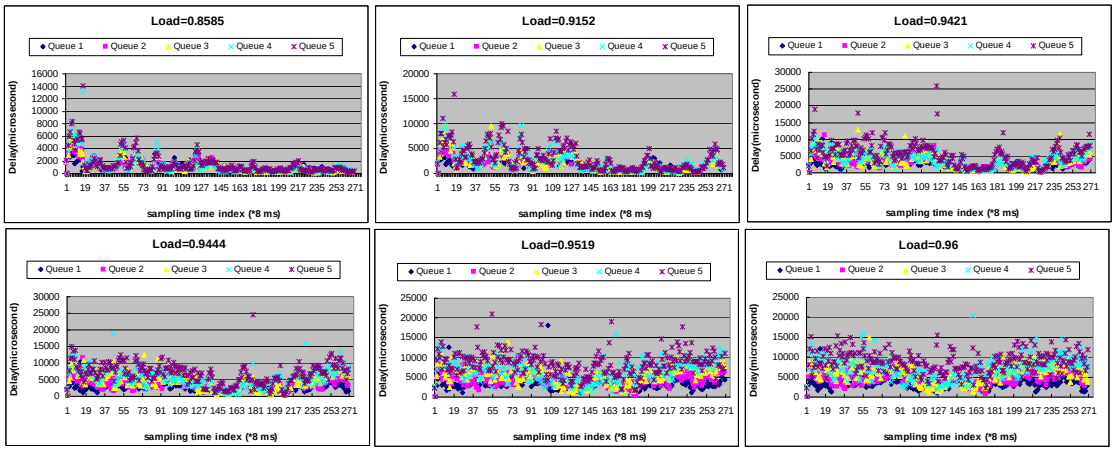


Figure 2.4 Delay vs. traffic load.

$$u_{i,5}^2(delay) = \begin{cases} 1 & delay \leq 7ms \\ e^{(delay-7)/7} & delay > 7ms \end{cases}, \forall i$$

Assume the traffic is delayed if it fails to be successfully transmitted. The buffer size of each queue is set as $25K$ bytes to avoid queues' build-up at high load. The network load is defined as the ratio of the total traffic admitted into the network to the capacity of the network.

Then, the objective is to show that QoS profiles received by the five kinds of session conform to the corresponding profiles derived from their application utilities. Fairness is achieved if application utilities obtained by sessions are equivalent to each other.

Figure 2.4 shows the average delay of packets received in cycles under six different network loads. Since the input traffic is self-similar, the delay of packets fluctuates cycle from cycle. Generally, with the increase of the traffic load, the packet delay increases. When the network load equals to 0.8585, most of the cycles are low-load cycles. Hence, almost all requests from these five kinds of sessions can be scheduled immediately. The five sessions experience similar delay performance. When the network load equals to 0.9162, around 15% cycles are medium-load cycles or high-load cycles, where requests from some of the sessions are delayed. It is

shown that session 5 experiences the largest delay while session 1 has the smallest delay, complying to their respective QoE profiles. With the increase of traffic load, delay of sessions increases but at different degrees, as determined by their respective application utilities. Delay of session 1 increases at the smallest degree, while that of session 5 increases at the largest degree. When the traffic load increases to 0.96, most of the cycles are high-load cycles. This implies that most of the requests have to be delayed before being successfully transmitted. Simulation shows that delay in this case is much higher than that in the case when the load is 0.8585.

Figure 2.5 shows QoE of the five kinds of sessions under different loads. It can be seen that QoE of all sessions are almost the same with small differences under a particular traffic load. For each session, there exists obvious differences in delay with different traffic loads. However, the difference in QoE is not that obvious, i.e., very small. The average QoE achieved when load equals to 0.96 is slightly lower than that achieved when load is 0.8585. This can be attributed to the fact that QoE is set as the same value when the delay is below a certain value.

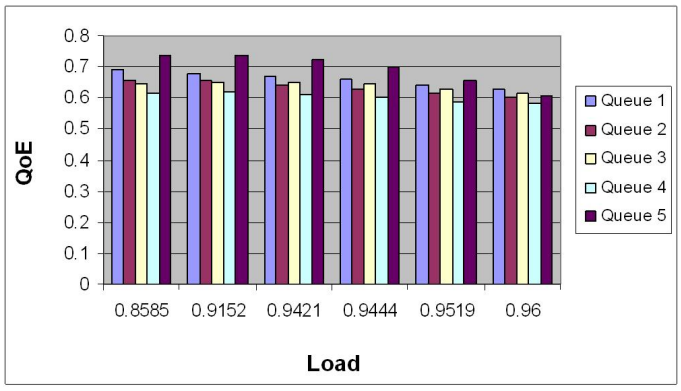


Figure 2.5 QoE vs. traffic load.

2.4.2 Loss Ratio

Consider QoE as a function of packet loss ratio, i.e.,$u_{i,j}(loss, delay) = u^1_{i,j}(loss)$. In a particular cycle, if the request is greater than the capacity of the cycle, the extra

requested traffic is dropped rather than delayed. The buffer size for each queue for a user session is set as infinity. For the five sessions in each ONU, $u^1_{i,j}(loss)$ is defined as follows.

$$u^1_{i,1}(loss) = \begin{cases} 1 & loss \leq 0.01 \\ (1-loss)/0.99 & loss \in [0.01, 1] \end{cases}, \forall i$$

$$u^1_{i,2}(loss) = \begin{cases} 1 & loss \leq 0.1 \\ (1-loss)/0.9 & loss \in [0.1, 1] \end{cases}, \forall i$$

$$u^1_{i,3}(loss) = \begin{cases} 1 & loss \leq 0.2 \\ (1-loss)/0.8 & loss \in [0.2, 1] \end{cases}, \forall i$$

$$u^1_{i,4}(loss) = \begin{cases} 1 & loss \leq 0.3 \\ (1-loss)/0.7 & loss \in [0.3, 1] \end{cases}, \forall i$$

$$u^1_{i,5}(loss) = \begin{cases} 1 & loss \leq 0.4 \\ (1-loss)/0.6 & loss \in [0.4, 1] \end{cases}, \forall i$$

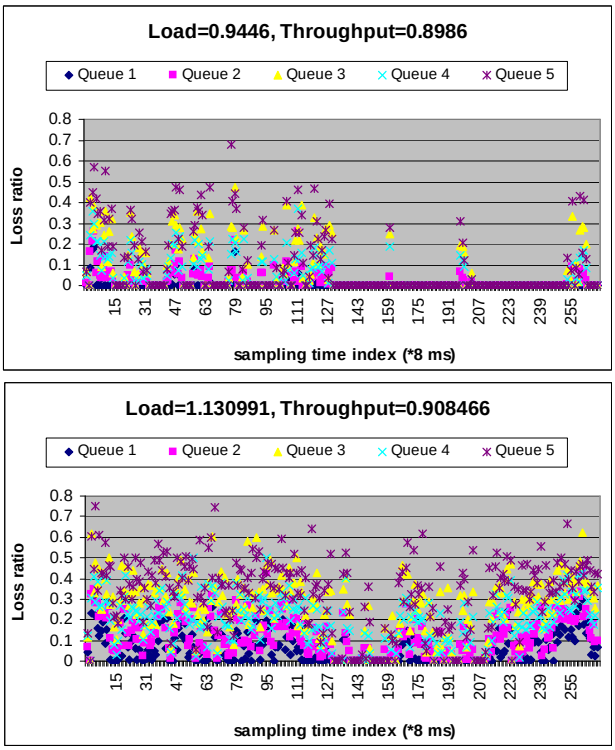


Figure 2.6 Sampled packet loss ratio.

Figure 2.6 shows the sampled packet loss ratio of five kinds of sessions, each of which assumes one of the above five different QoE functions. The sampling is taken every 8 ms. Simulations show that packet loss happens during some of the cycles when the network load equals to 0.9446, whereas packet loss happens during most of the cycles when the network load is 1.3099. It is also shown that five kinds of sessions experience different packet loss ratios during heavily-loaded cycles. From application functions, QoE of the five sessions equal to the highest value of 1 when the packet loss ratios of session 1, 2, 3, 4, and 5 are below 0.01, 0.1, 0.2, 0.3, and 0.4, respectively. From Figure 2.6, it can be seen that almost all points comply with this rule. On the other hand, when the network is heavily loaded and the highest QoE score cannot be guaranteed for sessions, the packet loss ratio of session 1, 2, 3, 4, and 5 will be increased to be higher than 0.01, 0.1, 0.2, 0.3, and 0.4, respectively. For fairness, this increase should enable the five sessions achieve the same QoE. This is also substantiated in the simulation results. Therefore, in terms of the packet loss ratio, our algorithm can guarantee fairness among the five sessions.

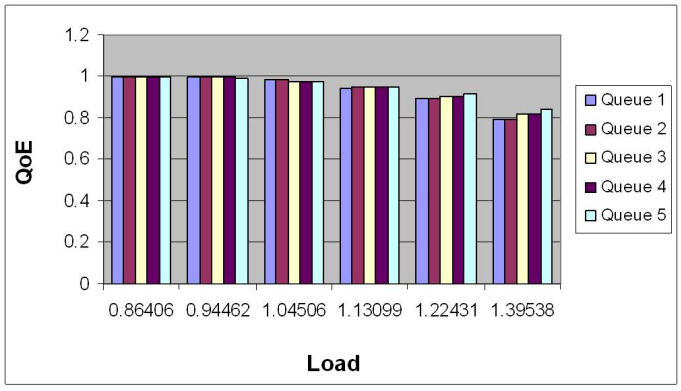


Figure 2.7 QoE vs. traffic load.

Figure 2.7 shows QoE scores of the five kinds of sessions under different loads. It shows that, under a particular traffic load, sessions of all sessions are almost the same with slight differences. QoE of sessions decreases with the increase of traffic load because of the increased packet loss ratio. When the traffic load is less than 1,

QoE scores of all sessions approach the maximum value of 1. When traffic load is greater than 1, QoE of sessions decrease slightly at nearly equal degrees.

2.5 Summary

From the perspective of optimizing user QoE, scheduling schemes to optimize the tradeoff between delay and loss performances of user sessions in TDM PONs have been proposed, where all user sessions share one upstream/downstream wavelength channel. First, with the assumption that the arrival traffic information across the entire time span is known *a priori*, an optimal offline scheduling scheme to achieve max-min QoE fairness among all users have been proposed. Then, based on the optimal offline scheduling scheme, the online scheduling scheme have been derived. Extensive simulations have been performed to show that the proposed scheme can guarantee the maximum utility even when the traffic load is as large as 0.8585 in EPON, and achieves equal utilities when the network is highly loaded.

CHAPTER 3

MULTI-WAVELENGTH SCHEDULING IN WDM PONS

3.1 Introduction

Hybrid Wavelength Division Multiplexing (WDM)/Time Division Multiplexing (TDM) Passive Optical Network (PON) is becoming a promising solution for next-generation broadband optical access [32, 33]. Instead of using only one wavelength to provision bandwidth in upstream and downstream as TDM PON does, hybrid WDM/TDM PON increases the number of working wavelengths in each stream to exploit the high bandwidth of optical fibers. On the other hand, hybrid WDM/TDM PON bridges the gap between TDM PON and pure WDM PON, and can be deployed by smoothly migrating from the currently deployed TDM PON [34–36].

In hybrid WDM/TDM PONs, an important optical device is the optical laser used for generating optical signals with multiple wavelengths. Depending on the wavelength generation capability, there are three major classes of lasers available for use, namely, multi-wavelength lasers, wavelength-specified lasers, and wavelength-tunable lasers [37]. A multi-wavelength laser is able to generate multiple WDM wavelengths simultaneously, including multi-frequency laser, Gain-Coupled DFB LD Array, and Chirped-Pulse WDM. Multi-wavelength lasers are usually used at the OLT to generate downstream traffic or seed ONUs with optical signals for their upstream data transmission [38, 39]. Instead of generating multiple wavelengths, a wavelength-specified laser, e.g., the common distributed feedback (DFB), can only emit one specific wavelength. Wavelength-specified sources have been extensively used in BPON, EPON [4], GPON [40], and next-generation access stage 1 (NGA1) [1]. However, with wavelength-specified lasers, no statistical gain can be exploited among ONUs which can support different wavelengths. Wavelength-tunable lasers are able

to generate multiple wavelengths, but only one wavelength at a time [41, 42]. As compared to wavelength-fixed lasers, tunable lasers possess advantages in two major aspects. First, from the perspective of the MAC layer, the wavelength tunability of tunable lasers facilitates the statistical multiplexing of traffic from all ONUs, thus potentially yielding better system performance [43, 44]. Second, from the perspective of network operators, tunable lasers enable the color-free property of ONUs, which further facilitates the simplified inventory management, reduced sparing cost, and automated wavelength provisioning [35]. Owing to these advantages, wavelength-tunable lasers are promising light source generators for hybrid WDM/TDM PONs.

One typical example of hybrid WDM/TDM PON architectures employing tunable lasers is SUCCESS [34]. SUCCESS equips OLT with tunable lasers to generate downstream data traffic and provides ONUs with optical CW bursts for their upstream data transmission. The wavelength tunability of tunable lasers was exploited to provision high bandwidth and realize a smooth migration path from current TDM PONs to WDM PONs. Bock *et al.* [45] also proposed an architecture of using tunable lasers at the OLT. In addition to equipping OLT with tunable lasers, the network equips ONUs with tunable lasers as well. Das *et al.* [46] proposed to equip each ONU with a tunable laser to facilitate a fully flexible dynamic bandwidth allocation in the upstream direction. In Reference [47], an architecture which equips ONUs with tunable lasers is proposed. With the focus on the laser tuning range which is an important parameter of tunable lasers, the impact of the laser tuning range on the network capacity is investigated, and WDM PONs is designed by selecting lasers with proper tuning ranges to minimize the capital investment of the PON. In Reference [43], capacities of hybrid WDM/TDM PONs are theoretically analyzed.

Currently, tunable lasers can be manufactured by various technologies such as mechanical, acousto-optic, or electro-optic tunability. Despite the many kinds of options, tunable lasers are still costly, thus inhibiting their wide deployment

in networks. The important cost elements of tunable lasers include "non-optical" specifications such as package dimension, output power, power variation over wavelength, and electrical power dissipation, and "optical" specifications such as the tuning speed and the tuning range [42]. According to the adopted technology, the tuning time may range from a few tens of nano-seconds (electro-optic) to a few tens of milli-seconds (mechanical), or even seconds or minutes. Generally, a higher tuning speed can yield a better system performance. Bock *et al.* [45] claimed that tuning times in the range of microseconds offer good network performance at data rates of 2.5 Gb/s. However, the higher the tuning speed, the more sophisticated the technology is needed, and consequently the higher the tunable laser cost.

This chapter focuses on investigating the impact of laser tuning (wavelength switching) time on dynamic bandwidth allocation (DBA) algorithms, and consequently the system performance. The bandwidth allocation refers to either downstream or upstream bandwidth allocation depending on the placement and usage of tunable lasers in the network. Laser tuning time constitutes an important consideration factor in designing DBA algorithms. When the laser tuning time is infinite, lasers have to stay on the same wavelengths all the time, and requests from ONUs can only be scheduled on the wavelengths their respective corresponding lasers stayed. No statistical gain can be exploited among ONUs which can support different wavelengths. When the laser tuning time is zero, requests can be scheduled on any wavelength any time. Then, statistical gain among all requests can be exploited. When the laser tuning time is between zero and infinity, proper DBA algorithms are desired to exploit the statistical gain among requests to the best under the condition that lasers are given enough time to switch wavelengths. To the best of our knowledge, this is the first time that the DBA problem with the consideration of laser tuning time is reported.

In this chapter, the DBA problem under the condition of non-zero laser tuning time is mapped into a multi-processor scheduling problem, with wavelength channels as machines, and requests from ONUs as jobs. The objective is to minimize the latest ONU request (job) completion time for the sake of small delay, fairness, and load balancing. It will be shown that when the laser tuning time is non-zero, both preemptive and non-preemptive scheduling problems with the objective of minimizing the latest ONU request (job) completion time are NP-hard. Heuristic preemptive and non-preemptive scheduling algorithms are then presented to address the scheduling problems, respectively. Theoretical analyses show that the heuristic preemptive scheduling algorithm achieves an approximation ratio of at most 2 and the heuristic non-preemptive scheduling algorithm achieves an approximation ratio of at most $2 - 1/m$, where m is the number of wavelengths. Simulation results show that our proposed algorithms with the consideration of the laser tuning time have made significant performance improvement as compared to previous algorithms without considering the laser tuning time. It is also shown that the preemptive scheduling scheme has some advantages over non-preemptive scheduling in terms of average delay and average throughput when the number of wavelengths is large and the number of ONUs is small. The advantages diminish with the decrease of the number of wavelengths and the increase of the number of ONUs. Note that this chapter assumes that all lasers have the same optical specifications including tuning time and tuning range for the color-free purpose.

The rest of the chapter is organized as follows. Section 3.2 describes the media access control, the scheduling framework, and the scheduling policy in hybrid WDM/TDM PON. Section 3.3 maps the DBA problem into a multi-processor scheduling problem, and presents the formal problem formulation. Section 3.4 presents preemptive scheduling schemes in a single cycle, and Section 3.5 describes non-preemptive scheduling schemes in a single cycle. Section 3.6 discusses the

scheduling problem in multiple DBA cycles. Section 3.7 presents simulation results and analyses. Section 3.8 presents concluding remarks.

3.2 Media Access Control, Scheduling Framework, and Scheduling Policy

For the downstream transmission in hybrid WDM/TDM PONs, the downstream incoming packets are queued in buffers at the OLT upon arrivals. Then, OLT determines the downstream bandwidth allocated to ONUs, and sends the downstream packets out to ONUs. Different from the downstream transmission, the upstream transmission in hybrid WDM/TDM PONs needs a proper MAC protocol to avoid data collision among ONUs. For backward compatibility, the MAC layer control protocol of hybrid WDM/TDM PONs inherits some characteristics from those of EPON and GPON, two major flavors of the existing TDM PONs. The data transmission processes of the two PONs are similar and can be generalized as follows: ONUs report their queue lengths and send their data packets to OLT using time slots allocated by OLT; OLT collects queue requests, makes bandwidth allocation decisions, and then notifies ONUs when and on which channel they can transmit packets. Such a request-grant based transmission mechanism is highly likely to be adopted in hybrid WDM/TDM PONs for consistency [48–50]. Following the assumption of a request-grant based MAC control mechanism, OLT gathers most of the intelligence and control of the network, and its functions determine the performance of the network.

Formerly, McGarry *et al.* [31] introduced the concept of the scheduling framework and scheduling policy to address the issues on when and how OLT performs DBA, respectively. Three scheduling frameworks were defined, i.e., on-line scheduling, off-line scheduling, and just-in-time scheduling. On-line scheduling refers to the operation that OLT determines bandwidth allocated to an ONU immediately after

receiving this ONU's request; off-line scheduling refers to the operation that OLT performs DBA after receiving queue requests from all ONUs. Both on-line scheduling and off-line scheduling have their advantages and disadvantages. On-line scheduling enables ONUs get immediate grants. However, the bandwidth allocation decision is made based on only one ONU's request. This may result in unfairness for other ONUs with upcoming requests. Off-line scheduling achieves better fairness by making decisions based on the requests of all ONUs. However, it incurs delays for ONUs to receive grants, and underutilizes the gap between the time that OLT sends out grant and the time that OLT receives the report from the first ONU. To overcome the near-sight problem of on-line scheduling and the underutilization problem of off-line scheduling, McGarry *et al.* [31] proposed just-in-time scheduling, where OLT postpones the decision making time until one channel is about to become idle. The decision making time in just-in-time scheduling is later than that in on-line scheduling, and is earlier than that in off-line scheduling. These three scheduling frameworks show similar advantages and disadvantages when they are applied in downstream scheduling.

The scheduling policy addresses the problem of how to perform DBA. It involves two problems: wavelength assignment and time allocation. For wavelength assignment, the earliest-channel-available-first rule was proposed to be employed with the assumption that each ONU can support all wavelengths [48, 50]. To make the algorithm applicable to the case that ONUs may support only a subset of the wavelengths, McGarry *et al.* [49] modified the earliest-channel-available-first rule into next-available-supported-channel-first. In [31], McGarry *et al.* converted the wavelength assignment problem into a matching problem between wavelengths and ONUs, and proposed Weighted Bipartite Matching to solve the matching problem. In [51], McGarry *et al.* modeled the problem into a multiprocessor scheduling problem and proposed to use longest processing time (LPT) first rule to address the minimizing

makespan problem for the case that ONUs can access all the wavelengths. When ONUs can access a limited set of wavelengths, they schedule ONUs with the least flexible job (LFJ) first rule. Meng *et al.* [52] studied the joint grant scheduling and wavelength assignment problem. They formulated it into a mixed integer linear programming (MILP) problem, and employed tabu search to obtain the optimal solution. For the time allocation problem, the time allocated to ONUs usually equals to its corresponding request when the on-line scheduling framework is adopted. In off-line scheduling, Dhaini *et al.* [50] proposed three time allocation algorithms, whereby low-load ONUs can always have their requests satisfied and high-load ONUs share the excess bandwidth by using different methods.

This chapter focuses on investigating the scheduling policy for off-line scheduling and just-in-time scheduling frameworks. Our objective is to consider laser tuning time and propose proper scheduling policy to exploit the benefit introduced by laser tunability to the best. The proposed scheduling policy can be applied in both downstream and upstream scheduling where tunable lasers are used to generate optical signals. To the best of our knowledge, this is the first time that the scheduling problem with the consideration of laser tuning time is investigated in WDM PONs.

Formerly, scheduling schemes with the consideration of the laser tuning time have been proposed for WDM broadcast-and-select networks [53], where each network node is configured with one tunable transmitter (TT) or fixed transmitter (FT), and one tunable receiver (TR) or fixed receiver (FR). The scheduling problem under the "FT-TR" configuration is the dual problem of that under the "TT-FR" configuration. The scheduling problems under both configurations have received extensive research attention [54–57]. Consider the "TT-FR" configuration. Since the receiver at each node is fixed tuned, the traffic to be scheduled on each wavelength is determined. The scheduling problem is reduced to the problem of sequencing requests on each wavelength channel such that enough time durations are left for

two requests originating from the same node and the schedule length is minimized. This is different from the DBA problem to be addressed in this chapter, which involves both wavelength assignment and time allocation problems. For the "TT-TR" configuration, Rouskas *et al.* [58] decomposed it into two subproblems: determine the wavelength tuned by each receiver, and determine the scheduling time of the request of each node pair. The latter problem is equivalent to the problem under the "TT-FR" and "FT-TR" configurations. The former problem is similar to the DBA problem in WDM PONs, which is going to be addressed in this chapter. For the former problem, Baldine *et al.* [59] considered the variation of traffic patterns, and tried to adjust the wavelengths tuned by receivers to accommodate the new traffic pattern such that the total number of retunings is minimized. In this way, a small total amount of time is spent in the tuning process, and thus high bandwidth utilization may be achieved since the wavelength channel may be idle during the tuning. This chapter assumes that, during the retuning time that a receiver retunes to a wavelength, the objective wavelength can be used by other receivers, and thus a higher and even 100% time utilization can be achieved. Constructing a schedule which can fully utilize these retuning time durations can be considered as the objective of this chapter. As will be discussed next, the problem can be considered as one multiprocessor scheduling problem where jobs take nonnegligible time to switch wavelengths.

3.3 Modeling and Problem Formulation

The bandwidth allocation problem in hybrid WDM/TDM PONs in a single DBA cycle discussed in this chapter can be described as:

Given the laser tuning time, requests from n ONUs, the available time of m wavelength channels, and the wavelength initially tuned by each laser, construct a schedule of the minimum length such that all requests can be accommodated and lasers are given enough time to switch wavelengths.

The bandwidth allocation problem in WDM/TDM PONs can be mapped into a multi-processor scheduling problem [60], with ONU requests as jobs, and wavelength channels as machines. Jobs and machines in this particular problem possess their respective unique characteristics.

3.3.1 Wavelengths → Machines

Assume data rates are the same on all wavelengths. Wavelength channels are modeled as parallel machines. Note that these wavelength channels may not be simultaneously available.

3.3.2 ONU Requests → Jobs

There are two options to model jobs. The first one is to model each queue request of an ONU as an individual job. However, owing to the laser on/off time, some guard time is needed between scheduling of jobs from different ONUs. To save the guard time, jobs from the same ONU should be scheduled consecutively, and thus can be grouped together as a single job for simplicity. This chapter regards the total requests of an ONU as a single job. Then, jobs possess two main properties. First, owing to the laser tuning time, a certain time gap is needed between the scheduling of jobs from the same ONU on different wavelength channels. Second, a job can be divided into subjobs, corresponding to requests of queues in the ONU, and each subjob can be further divided into subjobs, corresponding to requests of packets in the queue. In GPON with the allowance of packet fragmentation, scheduling of a packet can even be divided into scheduling of its partial packets, while in EPON without packet fragmentation, scheduling of a packet cannot be further divided. Therefore, jobs are preemptable in hybrid WDM/TDM GPON, and preemptable in a certain degree in hybrid WDM/TDM EPON.

In multi-processor scheduling, preemption enables jobs to be scheduled more flexibly, and thus yielding better system performances as compared to non-preemption. However, when preemption is allowed, jobs may be divided and scheduled in non-continuous time periods, which incur some extra time gap for laser on/off. It is not easy to tell whether the extra cost introduced by the guard time outweighs the extra performance improvement introduced by flexibility. This chapter simply assumes zero guard time for laser on/off, and investigates and compares the performances of preemptive and non-preemptive scheduling.

3.3.3 Scheduling Objective

In assigning wavelengths to ONUs, the objective is to minimize the latest job completion time among all requests and equalize the usage of all wavelength channels for two main reasons. First, assume one wavelength is more loaded than another one. ONUs assigned with the over-loaded wavelength may experience longer waiting time than those using the other wavelength. Equalizing the wavelength usage can ensure fairness among ONUs. Second, in terms of the just-in-time scheduling framework, OLT makes bandwidth allocation decisions before any of the wavelengths becomes idle. If all wavelengths become idle simultaneously, the scheduler can collect the requests from most of the ONUs, and thus make a fair decision. If one wavelength turns idle much earlier than the others, few requests arrive at the scheduler before the decision making time. In the worst case, just-in-time scheduling may be degraded into on-line scheduling, which makes the decision for one ONU request only. This will result in unfairness and increase the frequency of calculating bandwidth allocation.

Therefore, for the sake of small delay, fairness, and load balancing, minimizing the latest job completion time is considered as the scheduling objective.

3.3.4 Formal Problem Formulation

To mathematically formulate our scheduling problem, the definition of a *DBA cycle* is introduced. A DBA cycle in hybrid WDM/TDM PON refers to the time difference between two consecutive DBA decision making instances.

Formally, our problem of DBA in a single DBA cycle can be stated as follows:

Given:

1. n : The number of ONUs.
2. m : The number of wavelengths.
3. τ : The laser tuning time.
4. t : The decision making time of the current cycle.
5. $\mathbf{r} = \{r_1, r_2, ..., r_n\}$: The time durations of requests from n ONUs.
6. $\{\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_n^{-1}\}$: The wavelengths tuned by lasers at respective ONUs at the decision making time t.
7. $\{C_1^{-1}, C_2^{-1}, ..., C_m^{-1}\}$: The latest job completion time on m wavelengths in the last cycle.

Define:

1. γ_w: The sum of all requests with $\lambda_i^{-1} = w$, i.e., $\gamma_w = \sum_{\{i|\lambda_i^{-1}=w\}} r_i$.
2. $S^p(\mathbf{r}, \tau)$, $S^{\bar{p}}(\mathbf{r}, \tau)$: The preemptive and non-preemptive schedules with respect to requests $\mathbf{r}$ and the laser tuning time τ.
3. $C_w^{S,p}(\mathbf{r}, \tau)$, $C_w^{S,\bar{p}}(\mathbf{r}, \tau)$: The latest job completion time on wavelength w in schedule S^p and $S^{\bar{p}}$, respectively.

4. $C_{max}^{S,p}(\mathbf{r},\tau)$, $C_{max}^{S,\bar{p}}(\mathbf{r},\tau)$: The latest job completion time among all wavelengths in schedule S^p and $S^{\bar{p}}$, respectively. $C_{max}^{S,p}(\mathbf{r},\tau) = \max_{w=1}^{m} C_w^{S,p}(\mathbf{r},\tau)$. $C_{max}^{S,\bar{p}}(\mathbf{r},\tau) = \max_{w=1}^{m} C_w^{S,\bar{p}}(\mathbf{r},\tau)$.

Objective:

Find the optimal preemptive schedule $\mathfrak{S}^{\mathfrak{p}}$ and non-preemptive schedule $\mathfrak{S}^{\bar{\mathfrak{p}}}$ such that $C_{max}^{\mathfrak{S},\mathfrak{p}}(\mathbf{r},\tau) \leq C_{max}^{S,p}(\mathbf{r},\tau)$ for all other S^p, and $C_{max}^{\mathfrak{S},\bar{\mathfrak{p}}}(\mathbf{r},\tau) \leq C_{max}^{S,\bar{p}}(\mathbf{r},\tau)$ for all other $S^{\bar{p}}$.

Subject to:

1. Each request is allocated with sufficient time duration to be transmitted.
2. Each laser is given sufficient time to switch wavelengths if necessarily.
3. One laser cannot transmit on multiple wavelengths simultaneously.

To describe our proposed schemes, the following notations are further introduced.

Denote $\alpha_{i,w}$ as the earliest time that wavelength w can be allocated to the request from ONU i.

- For the downstream transmission, lasers can get the time allocation decision from the decision maker as early as the decision making time t. Consider the tuning time for lasers to switch wavelengths; laser i can tune to wavelength w at time t if $\lambda_i^{-1} = w$ and time $t+\tau$ if $\lambda_i^{-1} \neq w$. On the other hand, the latest job completion time on m wavelengths are $\{C_1^{-1}, C_2^{-1}, ..., C_n^{-1}\}$ in the former cycle. Therefore,

$$\alpha_{i,w} = \begin{cases} \max\{C_w^{-1}, t\} & \text{if } \lambda_i^{-1} = w \\ \max\{C_w^{-1}, t+\tau\} & \text{otherwise} \end{cases}$$

- For the upstream transmission, laser i needs to wait for $RTT_i/2$ time duration to receive the decision sent from the OLT, and the upstream traffic needs another

$RTT_i/2$ time to arrive at OLT. Considering both the laser tuning time and the latest job completion time in the last cycle,

$$\alpha_{i,w} = \begin{cases} \max\{C_w^{-1}, t + RTT_i\} & \text{if } \lambda_i^{-1} = w \\ \max\{C_w^{-1}, t + \tau + RTT_i\} & \text{otherwise} \end{cases}$$

For simplicity, assume $RTT_i = RTT_i', \forall i \neq i'$, in this chapter. Then, for both upstream and downstream transmissions, $\alpha_{i,w} = \alpha_{i',w}$, $\forall i$ with $\lambda_i^{-1} \neq w$, and $\forall i'$ with $\lambda_{i'}^{-1} \neq w$. For notational convenience, $\alpha_{i,w}$ for request i with $\lambda_i^{-1} = w$ is denoted as a^l, and $\alpha_{i,w}$ for request i with $\lambda_i^{-1} \neq w$ is denoted as a^u. $a^l \leq a^u \leq a^l + \tau$, i.e., $\alpha_{i,w} = \begin{cases} a_w^l & \text{if } \lambda_i^{-1} = w \\ a_w^u & \text{otherwise} \end{cases}$.

Next, the preemptive and nonpreemptive scheduling algorithms are investigated, respectively.

3.4 Preemptive Scheduling in a Single Cycle

The problem is equivalent to the preemptive multiprocessor scheduling problem with the objective of minimizing makespan subject to the constraints that machines are non-simultaneously available and jobs take non-negligible time to switch machines.

When the laser tuning time $\tau = 0$, and all wavelengths channels are available from the same time, i.e., $a_w^l = a_{w'}^l, \forall w \neq w'$, the problem is equivalent to the $p|pmtn|C_{max}$ multiprocessor scheduling problem [61], which can be easily solved. When $\tau = 0$ and wavelength channels are not simultaneously available, i.e., $\exists w \neq w', a_w^l \neq a_{w'}^l$, the problem can be solved by slightly modifying the algorithm for the $p|pmtn|C_{max}$ problem.

When the laser tuning time $\tau = +\infty$, the request from ONU i can only be scheduled on the original wavelength λ_i^{-1} tuned by ONU i. The latest job completion time on wavelength w equals to $a_w^l + \sum_{\{i|\lambda_i^{-1}=w\}} r_i$. Among all wavelengths, the latest

job completion time $C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$ equals to $\max_{w=1}^{m}(a_w^l+\sum_{\{i|\lambda_i^{-1}=w\}}r_i)$. Therefore,

$$C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)=\max_{w=1}^{m}(\sum_{\{i|\lambda_i^{-1}=w\}}r_i+a_w^l)$$

When the laser tuning time τ is an arbitrary value, this is the first time that the problem is investigated. The preemptive scheduling problem is shown to be NP-hard.

Theorem 2. *When the laser tuning time τ is arbitrary, the preemptive scheduling problem with the objective of minimizing the latest request completion time is NP-hard.*

Proof. Consider the following downstream traffic scheduling problem with

$$C_w^{-1}=\begin{cases}t+\tau & \text{if } w=1\\ t & \text{otherwise}\end{cases}$$

$\lambda_i^{-1}=1\ \forall i$, and $r_i\in[\ell-t-2\tau,\ell-t-\tau]$, $\forall i$. Then, after checking all i and w, $\alpha_{i,w}=t+\tau,\forall i,w$. Since $\alpha_{i,w}+r_i\leq\ell$ and $\alpha_{i,w}+r_i+\tau\geq\ell$, any request can be scheduled on any wavelength, but cannot be divided into parts and scheduled on multiple wavelengths. The time duration which can be allocated on any wavelength equals to $(\ell-t-\tau)$. The problem of determining whether all requests can be scheduled before ℓ is equivalent to the problem of deciding whether all these given requests can be divided into m groups, in which the sum of requests in each group is no greater than $\ell-t-\tau$. The latter problem is equivalent to the bin packing problem, which is NP-hard [62]. Hence, the preemptive scheduling problem with the objective of minimizing the latest request completion time is NP-hard when the laser tuning time τ is arbitrary. □

Because of the NP-hard property, heuristic algorithms is next proposed to solve the problem.

3.4.1 Naive Preemptive Scheduling

The first preemptive scheduling algorithm to be proposed, referred to as *naive preemptive scheduling*, is based on the schedules constructed for $\tau = 0$ and $\tau = +\infty$, respectively.

Since lasers with smaller tuning time yield a smaller latest job completion time, for any given requests $\mathbf{r}$, $C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau) \leq C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau + \epsilon), \forall \epsilon > 0$. Hence,

$$C_{max}^{\mathfrak{S},p}(\mathbf{r}, 0) \leq C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau) \leq C_{max}^{\mathfrak{S},p}(\mathbf{r}, +\infty)$$

The main idea of naive preemptive scheduling is to first construct a schedule $\mathfrak{S}^p(\mathbf{r}, 0)$ assuming that the laser tuning time is zero, and a schedule $\mathfrak{S}^p(\mathbf{r}, +\infty)$ assuming that the laser tuning time is $+\infty$. Then, naive preemptive scheduling adjusts the schedule $\mathfrak{S}^p(\mathbf{r}, 0)$ to give all lasers enough time to switch wavelengths. If the schedule length is less than the length of $\mathfrak{S}^p(\mathbf{r}, +\infty)$, the adjusted schedule based on $\mathfrak{S}^p(\mathbf{r}, 0)$ is considered as the final schedule; otherwise, $\mathfrak{S}^p(\mathbf{r}, +\infty)$ is considered as the final schedule. Algorithm 6 details the proposed naive preemptive scheduling.

The part between Line 2 and Line 10 in Algorithm 6 is to construct a schedule assuming that the laser tuning time is zero. For the scheduling algorithm with zero laser tuning time, the main idea is to try different number ℓ and decide whether there exists a schedule whose latest job completion time is ℓ. Finally, the schedule with the minimum latest job completion time can be obtained.

For a given ℓ, requests are first sorted in the descending order of their sizes, and wavelengths are sorted in the ascending order of their available time as described in Line 3 and Line 4. The sorting is to make sure that large requests receive enough allocations of nonoverlapping time durations. Then, the time resource on a wavelength is assigned to requests one by one from the back of the time span until the time on that wavelength is used out. If the remaining time on a wavelength

Algorithm 6 Naive preemptive scheduling

1: $\ell = (\sum_i r_i + \sum_w a_w)/m$
2: **while** The smallest latest request completion time ℓ has not been found **do**
3: Index ONU requests such that $r_1 \geq r_2 \geq ... \geq r_n$
4: Index wavelengths such that $a_1^l \leq a_2^l \leq ... \leq a_m^l$
5: Select an ONU request and a wavelength
6: Schedule the request on the back of the wavelength. If the remaining time on a wavelength is not enough for the request, schedule the remaining unscheduled part of the request to another wavelength.
7: **if** Not all requests can be scheduled before ℓ **then**
8: Find a proper ℓ
9: **end if**
10: **end while**
11: Postpone the scheduling of all requests on a wavelength by τ
12: Postpone the scheduling of the last request on a wavelength by τ
13: If the length of the constructed schedule is longer than $C_{max}^{\mathfrak{S},p}(\mathbf{r}, +\infty)$, $\mathfrak{S}^p(\mathbf{r}, +\infty)$ is considered as the final schedule.

is not enough to satisfy a request, the unscheduled part will be moved to the next wavelength as described in Line 6.

To generate a feasible schedule which gives lasers sufficient time to switch wavelengths, some further adjustments are performed. The first step is to postpone the scheduling of all requests by τ as described in Line 11. The second step is to postpone the scheduling of the last request on each wavelength by τ as described in Line 12.

Next will prove that the schedule produced by naive preemptive scheduling is a feasible schedule under the condition that the laser tuning time equals to τ.

Lemma 1. *The schedule produced by Algorithm 6 is a feasible schedule for the case that the laser tuning time equals to* τ.

Proof. Since all requests are postponed by time τ, the corresponding laser for request i is idle during $[a_w^l, a_w^l + \tau]$, and laser i is given enough time to schedule the first request. Besides, in the schedule $\mathfrak{S}^p(\mathbf{r}, 0)$, only requests scheduled at the beginning or the end of the time span of a wavelength may be preempted. In Line 12, the last scheduled request on each wavelength is postponed by τ, and hence lasers are given sufficient time to schedule the last scheduled request.

Therefore, in the schedule produced by Algorithm 6, all requests have been scheduled, and lasers are all given enough time to switch wavelengths. □

The following theorem can be derived regarding to $C_{max}^{S,p}(\mathbf{r}, \tau)$ produced by naive preemptive scheduling.

Theorem 3. *For a given* $\mathbf{r}$*, the schedule* S *produced by Algorithm 6 has the property that* $C_{max}^{S,p}(\mathbf{r}, \tau) = \min\{C_{max}^{\mathfrak{S},p}(\mathbf{r}, 0) + 2\tau, C_{max}^{\mathfrak{S},p}(\mathbf{r}, +\infty)\}$*, and consequently the optimal schedule* $\mathfrak{S}(\mathbf{r}, \tau)$ *has the property that* $C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau) \leq \min\{C_{max}^{\mathfrak{S},p}(\mathbf{r}, 0) + 2\tau, C_{max}^{\mathfrak{S},p}(\mathbf{r}, +\infty)\}$

Proof. The "while" loop between Line 2 and Line 10 optimally solves the scheduling problem under the condition of zero laser tuning time. As compared to the schedule

with zero laser tuning time, the latest job completion time on each wavelength in the final schedule is increased by 2τ. Thus, $C^S_{max}(\mathbf{r},\tau)$ obtained by naive preemptive scheduling equals to $\min\{C^{\mathfrak{S}}_{max}(\mathbf{r},0)+2\tau, C^{\mathfrak{S}}_{max}(\mathbf{r},+\infty)\}$. Since $C^{\mathfrak{S},p}_{max}(\mathbf{r},\tau) \leq C^{S,p}_{max}(\mathbf{r},\tau)$, $C^{\mathfrak{S},p}_{max}(\mathbf{r},\tau) \leq \min\{C^{\mathfrak{S}}_{max}(\mathbf{r},0)+2\tau, C^{\mathfrak{S}}_{max}(\mathbf{r},+\infty)\}$. □

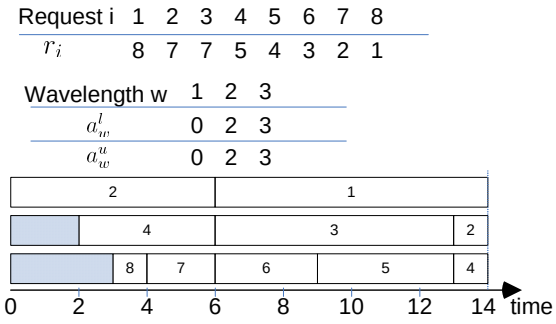
Computational Complexity: The complexities of the two ordering processes are $O(n\log(n))$ and $O(m\log(m))$, respectively. The complexity of the "for" loop in Algorithm 6 is $O(n)$. Lines 11, 12, and 13 are all of complexity of $O(n)$. Hence, the total complexity of Algorithm 6 is $O(n\log(n)+m\log(m))$.

The example with 8 ONUs and 3 wavelengths as shown in Figure 3.1 illustrates Algorithm 6. Figure 3.1 (a) shows the constructed schedule assuming that the laser tuning time equals to zero. Request 1 is allocated with the time duration [6,14] on wavelength 1. The remaining time duration on wavelength 1 is not enough to satisfy request 2. Part of request 2 is scheduled in time duration [0,6] on wavelength 1 and the other part is scheduled in time duration [13,14] on wavelength 2. Similarly, part of request 4 is scheduled on wavelength 2, and the other part is scheduled on wavelength 3. All the requests can be scheduled before time 14. Figure 3.1 (b) shows the final schedule after adjustment. When $\tau = 1.5$, the latest job completion time is increased from 14 to 17.

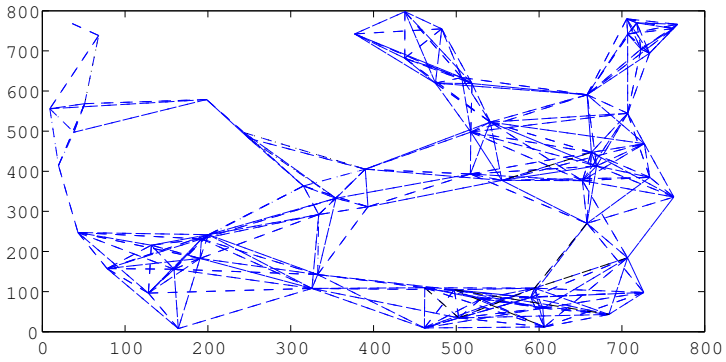
3.4.2 Heuristic Preemptive Scheduling

In the schedule produced by Algorithm 6, there are two idle time gaps of duration τ on each wavelength. One is the time gap between a^l_w and $a^l_w+\tau$, and the other one is the time gap before the scheduling of the last request on the wavelength. To produce a schedule S with $C^S_{max}(\mathbf{r},\tau)$ smaller than $\min\{C^{\mathfrak{S}}_{max}(\mathbf{r},0)+2\tau, C^{\mathfrak{S}}_{max}(\mathbf{r},+\infty)\}$, these idle time gaps need to be filled to the best. To this end, a heuristic preemptive scheduling algorithm can be proposed as described in Algorithm 7.

Algorithm 7 constructs the schedule according to three basic rules.



(a) The preemptive schedule when $\tau = 0$



(b) The preemptive schedule when $\tau = 1.5$

Figure 3.1 An example of naive preemptive scheduling when $\tau = 1.5$.

- Since the time gap between a_w^l and a_w^u on wavelength w can only be filled by requests with $\lambda_i^{-1} = w$, Algorithm 7 fills this time gap with requests with $\lambda_i^{-1} = w$ to the best. Request i is scheduled on a wavelength other than its originally tuned wavelength λ_i^{-1} only if the gap filling between $a_{\lambda_i^{-1}}^l$ and $a_{\lambda_i^{-1}}^u$ is not affected after the scheduling.

- Similar to Algorithm 6, to guarantee that large requests receive enough bandwidth, the resource in the wavelength with the smallest a_w^l is allocated first, and the largest request is scheduled first.

- Similar to Algorithm 6, preemption is disallowed in the middle of the time span on a wavelength. If one request is preempted and scheduled during $[\mu, \nu]$ on a wavelength, this request cannot be scheduled during $\mu - \tau$ and $\nu + \tau$ on any wavelength, thus resulting in smaller chances of scheduling the remainder of the request in other wavelengths.

More specifically, Algorithm 7 divides the resource allocation process on a wavelength into two steps. The first step is to allocate the time duration between a_w^u and ℓ. The second step is to allocate the time duration between a_w^l and a_w^u. The time duration between a_w^u and ℓ on wavelength w can be allocated to any request, while the time duration between a_w^l and a_w^u on wavelength w can only be allocated to requests with $\lambda_i^{-1} \neq w$.

Step 1: The Allocation Between a_w^u and ℓ: When allocating the resource between a_w^u and ℓ, the largest request is considered first. To avoid preemption in the middle of the schedule, a request is scheduled on wavelength w only if the remaining available time duration after a_w^u is enough to accommodate the request. On the other hand, since allocating request i to wavelength w will decrease the total traffic $\gamma_{\lambda_i^{-1}}$ which can be used to fill the gap between $a_{\lambda_i^{-1}}^l$ and $a_{\lambda_i^{-1}}^u$ on wavelength λ_i^{-1}, request

Algorithm 7 Heuristic Preemptive Scheduling

Initialize $x_w = \gamma_w$, $y_w = \ell$

Index wavelengths such that $a_1^l \leq a_2^l \leq ... \leq a_m^l$

for $w = 1 : m$ **do**

  */*Step 1: The allocation between a_w^u and ℓ*/*

  Index unscheduled requests such that $r_1 \geq r_2 \geq ...$

  for $i = 1 : n$ **do**

    if $r_i \leq y_w - a_w^u$ & $x_{\lambda_i^{-1}} - r_i \geq (a_{\lambda_i^{-1}}^u - a_{\lambda_i^{-1}}^l)$ **then**

      Allocate time duration $[y_w - r_i, y_w]$ to request i

      $y_w = y_w - r_i$, $x_{\lambda_i^{-1}} = x_{\lambda_i^{-1}} - r_i$

    end if

  end for

  Denote x_w and y_w as x_w^* and y_w^*, respectively.

  */*Step 2: The allocation between a_w^l and y_w^**/*

  Index unscheduled requests with $\lambda_i^{-1} = w$ such that $r_1 \geq r_2 \geq ...$

  $i = 1$

  while there is available time on wavelength w and there are unscheduled requests with $\lambda_i^{-1} = w$ **do**

    if $r_i \leq y_w - a_w^l$ **then**

      Allocate time $[y_w - r_i, y_w]$ on wavelength w to request i

    else

      Allocate time $[a_w^l, y_w]$ on wavelength w and time $[\ell - (r_i - y_w + a_w^l), \ell]$ on wavelength $w + 1$ to request i

      $y_{w+1} = \ell - (r_i - y_w + a_w^l)$

    end if

    $i = i + 1$

  end while

end for

i is allocated onto wavelength w only if the gap filling between $a^l_{\lambda_i^{-1}}$ and $a^u_{\lambda_i^{-1}}$ is not affected. The conditions are described in Line 7 of Algorithm 7.

x_w is used to track the total traffic which can fill the gap between a^l_w and a^u_w, and y_w to track the last available time stamp on wavelength w. x_w and y_w are initialized to be γ_w and ℓ, respectively. The time duration between a^l_w and y_w is the resource which is still available on wavelength w. Denote x_w and y_w after performing the first step as x^*_w and y^*_w, respectively. The first step tries to let y^*_w approach a^u_w without affecting the filling of the gap between a^l_w and $a^u_w, \forall w$.

Step 2: The Allocation Between a^l_w and a^u_w: After performing the first step, the available time duration on wavelength w is actually between a^l_w and y^*_w. The second step of allocating the time between a^l_w and y^*_w considers the largest unscheduled request with $\lambda_i^{-1} = w$ first. If the remaining time is not enough to schedule the request, the remaining unscheduled part of the request is scheduled onto the next wavelength as described in Line 20.

It can be seen that if $x^*_w \geq y^*_w - a^l_w$, the time duration between a^l_w and y^*_w on wavelength w can be allocated, and there is no idle time gap on wavelength w. If $x^*_w < y^*_w - a^l_w$, the time duration between a^l_w and $y^*_w - x^*_w$ on wavelength w is idle. Algorithm 7 takes two measures to reduce the duration of the idle time gap. One is to let y^*_w approach a^u_w to the best, and the other one is to make sure that x^*_w is always above $a^u_w - a^l_w$ if $\gamma_w \geq a^u_w - a^l_w$.

Analysis: Theorem 4 describes the upper bound of $C^{S,p}_{max}(\mathbf{r}, \tau)$ produced by Algorithm 7.

Theorem 4. *For given* $\mathbf{r}$ *and* τ*, schedule* S *produced by Algorithm 7 has the property that* $C^{S,p}_{max}(\mathbf{r}, \tau) \leq C^{\mathfrak{S},p}_{max}(\mathbf{r}, \tau) + \max^n_{i=1} r_i$*, where* $\mathfrak{S}$ *is the optimal schedule with the minimum latest request scheduling time.*

Proof. This theorem is proven by showing that when $\ell = C^{\mathfrak{S},p}_{max}(\mathbf{r}, \tau) + \max^n_{i=1} r_i$, Algorithm 7 can schedule all requests before ℓ.

Denote gap_w and $\mathfrak{gap}_w$ as the duration of idle time on wavelength w in schedule S produced by Algorithm 7 and that in the optimal schedule $\mathfrak{S}$. As described above, when $x_w^* \geq y_w^* - a_w^l$, there is no idle time on wavelength w; otherwise, the time between a_w^l and $y_w^* - x_w^*$ on wavelength w is idle, and gap_w equals to $y_w^* - x_w^* - a_w^l$. Also, it is known from the above that $x_w^* \begin{cases} \geq a_w^u - a_w^l & \text{if } \gamma_w \geq a_w^u - a_w^l \\ = \gamma_w & \text{otherwise} \end{cases}$. Thus,

$$gap_w \begin{cases} \leq y_w^* - a_w^u & \text{if } \gamma_w \geq a_w^u - a_w^l \\ = y_w^* - \gamma_w - a_w^l & \text{otherwise.} \end{cases}$$

On the other hand, in the optimal schedule $\mathfrak{S}$, when $\gamma_w < a_w^u - a_w^l$, there must be some idle time duration between a_w^l and a_w^u, and $\mathfrak{gap}_w = a_w^u - a_w^l - \gamma_w$. Accordingly, it can be further obtained that $gap_w \leq \mathfrak{gap}_w + y_w^* - a_w^u$ for all w. In Step 1, y_w^* is let to approach a_w^u to the best by trying every unscheduled job. It can be easily obtained that $y_w^* - a_w^u \leq \max_{i=1}^n r_i$ if there is still an unscheduled job which does not have to be scheduled on its originally tuned wavelength. Then,

$$gap_w \leq \mathfrak{gap}_w + y_w^* - a_w^u \leq \mathfrak{gap}_w + \max_{i=1}^n r_i$$

Therefore, all requests can be scheduled before $C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau)+\max_{i=1}^n r_i$ by using Algorithm 7. □

Corollary 1. *The approximation ratio of the heuristic preemptive scheduling algorithm is at most* 2.

Proof. Based on Theorem 4, $C_{max}^{S,p}(\mathbf{r},\tau) \leq C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau) + \max_{i=1}^n r_i$. On the other hand, $C_{max}^{S,p}(\mathbf{r},\tau) \leq \max_{i=1}^n r_i$. Therefore,

$$C_{max}^{S,p}(\mathbf{r},\tau)/C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau) \leq 2$$

That is to say, the approximation ratio is at most 2. □

Computational Complexity: The ordering process in Line 2 of Algorithm 7 has the complexity of $O(mlog(m))$. The "for" loop in Step 1 has the complexity of $O(n)$. The total complexity of performing Step 1 is $O(mn)$. The complexity of Step 2 is $O(nlog(n))$. Hence, Algorithm 7 has the complexity of $O(mn+mlog(m)+nlog(n))$.

Taking the example with 12 ONUs and 4 wavelengths, Figure 3.2 illustrates the algorithm. The laser tuning time $\tau = 5$. The decision making time $t = 0$. The latest job completion time on wavelengths in the last cycle are 0, 1, 1, and 2, respectively. Then, $a^l = \{0, 1, 1, 2\}$, $a^u = \{5, 5, 5, 5\}$, and $\ell = 15$. Request 1 with the largest size cannot be scheduled on wavelength 1 because the bandwidth from a_1^u to ℓ is not enough to satisfy Request 1. Request 2 satisfies all conditions and is scheduled on wavelength 1. After scheduling Request 2, the remaining time duration between a_1^u and ℓ can only accommodate requests with sizes no greater than 3. Request 8 with size 3 is not scheduled between a_1^u and ℓ on wavelength 1 because the gap between a_1^l and a_1^u cannot be filled without Request 8. After scheduling Request 9 , the time between a_1^u and ℓ on wavelength 1 has been all allocated. The scheduling enters into the second step of allocating $[a_1^l, a_1^u]$. After scheduling Request 6 which is the largest among all requests with $\lambda_i^{-1} = 1$, the remaining time duration is not enough to schedule the next largest request, i.e., Request 8 with 3. Hence, Request 8 is divided into two parts, among which the first part of size 1 is scheduled on wavelength 1 and the second part of size 2 is scheduled on wavelength 2. After repeating this process, all requests can be scheduled before $\ell = 15$.

3.5 Non-preemptive Scheduling in a Single Cycle

When the laser tuning time $\tau = 0$, the non-preemptive scheduling problem was proved NP-hard [61]. For the case that all wavelength channels are available at the same time, i.e., $a_w^l = a_{w'}^l, \forall w \neq w'$, the problem is equivalent to the $p||C_{max}$ multiprocessor scheduling problem. LPT was shown to have $4/3 - 1/n$ approximation ratio, and

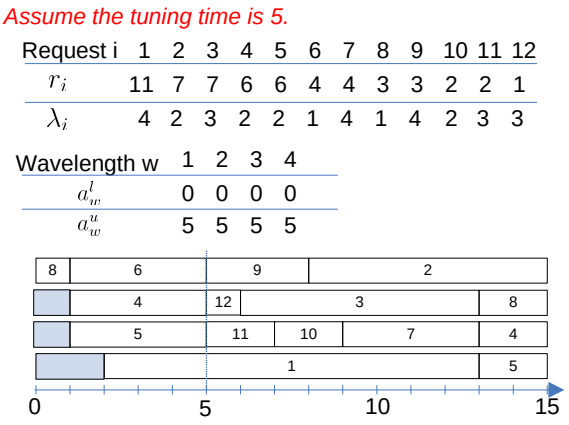


Figure 3.2 One example of heuristic preemptive scheduling when $\tau = 5$.

the MULTIFIT algorithm is with a smaller approximation ratio of 72/61 [63]. When the wavelength channels are non-simultaneously available, Lee *et al.* [64] proposed modified LPT (MLPT) to achieve 4/3 approximation ratio. Lin *et al.* [65] showed that 4/3 is the exact bound for MLPT when the number of processors is greater than two, and the approximation ratio is 5/4 when the number of processors equals to two. Chang *et al.* [66] showed that the approximation ratio of the MULTIFIT algorithm is $9/7 + 2^{-k}$, where k is the selected number of the major iterations in the MULTIFIT. This is the smallest one known so far.

When $0 < \tau < +\infty$, the problem is NP-hard since it is not easier than the problem under the case that $\tau = 0$. Two heuristic algorithms are then proposed to address it.

3.5.1 Naive Nonpreemptive Scheduling

As described in Algorithm 8, naive nonpreemptive scheduling is based on the algorithm proposed for the $\tau = 0$ case. This chapter uses the MULTIFIT algorithm to construct the nonpreemptive schedule for the $\tau = 0$ case.

In Algorithm 8, schedule S with the assumption of zero laser tuning time is first constructed by using the MULTIFIT algorithm. Then, the scheduling of all requests are postponed by time τ to give lasers sufficient time to switch wavelengths.

Algorithm 8 Naive Nonpreemptive Scheduling

1: Construct schedule S by using MULTIFIT.

2: Postpone scheduling of all requests in S by time τ

Since preemption is disallowed in schedule S, each request is scheduled on one wavelength only in schedule S. By postponing all requests in $S(\mathbf{r}, 0)$ with a time duration of τ, the time period between a_w^l and $a_w^l + \tau$ is idle, and hence lasers can have sufficient time to switch wavelengths.

From Algorithm 8, the following theorem can be derived regarding to $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau)$.

Theorem 5. *For a given* $\mathbf{r}$, $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau) \leq C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, 0) + \tau$.

Proof. For a given $\mathbf{r}$, assume $\mathfrak{S}$ is the optimal schedule for the case that $\tau = 0$. If all requests in $\mathfrak{S}$ are postponed by τ time duration, the newly obtained schedule is a feasible schedule for the case that $\tau \neq 0$. Hence, $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau) \leq C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, 0) + \tau$. □

3.5.2 Heuristic Nonpreemptive Scheduling

In the schedule produced by Algorithm 8, the time duration between a_w^l and $a_w^l + \tau$ is unoccupied. To generate a schedule with a smaller latest job completion time, Algorithm 9 is proposed by filling idle time durations on all wavelengths. Algorithm 9 contains two steps. The first step is to allocate the time period between a_w^l and a_w^u on wavelength w for requests with $\lambda_i^{-1} = w$. Large jobs are given higher priorities over small jobs since they are not easy to switch wavelengths. The second step is to allocate the time period between a_u^l and ℓ. The MULTIFIT algorithm is applied directly in the second step.

Theorem 6. *The approximation ratio of Algorithm 9 is at most* $2 - 1/m$.

Algorithm 9 Heuristic Nonpreemptive Scheduling

1: */*Step 1: Schedule between a_w^l and a_w^u */*

2: **for** $w = 1 : m$ **do**

3: Index requests with $\lambda_i^{-1} = w$ such that $r_1 \geq r_2 \geq r_3$....

4: Allocate time to requests with $\lambda_i^{-1} = w$ until the time exceeds a_w^u

5: **end for**

6: */*Step 2: Schedule between a_w^u and ℓ */*

7: Run the MULTIFIT algorithm to allocate bandwidth to the remaining unscheduled requests

Proof. Let job x be the last job assigned to all m wavelengths. The processing time of job x is r_x. Also, let s be the start time that job x is processed. Then, the latest job completion time equals to $r_x + s$. Let $\beta_w = \max\{\tau - \sum_{\{i|\lambda_i^{-1}=w\}} r_i, 0\}$. β_w denotes the time duration between time α_w^l and α_w^u which cannot be filled anyway. It is easy to see that, with Algorithm 9, $s < (\sum_{i=1}^{x-1} r_i + \sum_{i=x+1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m$. $s + r_x/m < (\sum_{i=1}^{x-1} r_i + \sum_{i=x+1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m + r_x/m = (\sum_{i=1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m + (m-1)r_x/m$. On the other hand, the optimal value $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau)$ must not be less than $(\sum_{i=1}^{n} r_i + \sum_{w=1}^{m} (\alpha_w^l + \beta_w))/m$. Therefore, $s + r_x < C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau) + (1 - 1/m)r_x$. Also, it is known that $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau) > r_x$. Consequently, it can be obtained that $(s + r_x)/C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau) < 2 - 1/m$. So, the approximation ratio is at most $2 - 1/m$. □

Computational Complexity: The ordering process in Step 1 of Algorithm 9 has the complexity of $O(nlog(n))$. The allocation process in Step 1 has the complexity of $O(n)$. Hence, the complexity of Step 1 in Algorithm 9 is $O(nlog(n))$. Algorithm 8 has the complexity of $O(nm)$. Therefore, Algorithm 9 has the complexity of $O(nlog(n) + nm)$.

Taking the same example as shown in Figure 3.2, Figure 3.3 illustrates Algorithm 9. Assume the tuning time is 5 and $\ell = 13$. After the first step, jobs 2, 3, 4, 8, and 10

Figure 3.3 An example of heuristic nonpreemptive scheduling when $0 < \tau < +\infty$.

still use their last tuned wavelengths. After the second step, all the remaining jobs are successfully scheduled before time 13. Note that ℓ cannot be further decreased. The latest job completion time is thus 13.

3.6 Discussions on the Scheduling in the Multi-cycle Scenario

As discussed before, besides the decision making time t, the laser tuning time τ, and requests $\mathbf{r}$, the latest job completion time also depends on $\{C_1^{-1}, C_2^{-1}, ..., C_w^{-1}\}$ and $\{\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_m^{-1}\}$, which are determined by the schedule in the last cycle. The relations between $C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau)$, $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau)$ and $\{C_1^{-1}, C_2^{-1}, ..., C_w^{-1}\}$, and $\{\lambda_1^{-1}, \lambda_2^{-1}, ..., \lambda_m^{-1}\}$ will be next respectively discussed.

3.6.1 $C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau)$ vs. $\{C_1^{-1}, C_2^{-1}, ..., C_m^{-1}\}$

To obtain small $C_{max}^{\mathfrak{S},p}(\mathbf{r}, \tau)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r}, \tau)$, the earliest time $\alpha_{i,w}$ that wavelength w can be allocated to request i is desired to be small for all i, w; this consequently requires small $\{C_1^{-1}, C_2^{-1}, ..., C_m^{-1}\}$. Thus, minimizing the latest job completion time in the last cycle can help produce a small latest job completion time in the current cycle. On the other hand, as discussed in Section 3.3.3, minimizing the latest job completion time in a cycle can achieve small delay, fairness, and load balancing of

traffic in that particular cycle. Therefore, minimizing the latest job completion time yields good performances not only for traffic in the current cycle, but also for traffic in future cycles. The latest job completion time in each cycle should be minimized when considering the traffic in multiple cycles.

3.6.2 $C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},\tau)$ vs. $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$

$C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},\tau)$ are closely related to $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$. The following two lemmas can be derived.

Lemma 2. *When* $\tau = 0$, $C_{max}^{\mathfrak{S},p}(\mathbf{r},0)$ *and* $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)$ *are independent of* $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$.

Proof. When $\tau = 0$, $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_w^{-1}\}$ can be changed to any other $\{\lambda_1',\lambda_2',...,\lambda_w'\}$ in no time at the decision making time t. Therefore, $C_{max}^{\mathfrak{S},p}(\mathbf{r},0)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)$ are independent of $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$. □

Lemma 5 states that, when $\tau = 0$, the scheduling in the current cycle does not depend on the scheduling in the last cycle, and does not need to consider the scheduling in future cycles.

Lemma 3. *When* $\tau = +\infty$, *if* $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$ *satisfies the condition that no other* $\{\lambda_1',\lambda_2',...,\lambda_m'\}$ *can yield a smaller* $\max_{w=1}^{m}(a_w^l+\gamma_w)$, $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$ *can produce the smallest* $C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$ *and* $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty)$.

Proof. As described above, $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty) = C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty) = \max_{w=1}^{m}(a_w^l+\gamma_w)$. If no other $\{\lambda_1',\lambda_2',...,\lambda_m'\}$ yields a smaller $\max_{w=1}^{m}(a_w^l+\gamma_w)$ than $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$, $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$ can produce the smallest $C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty)$. □

Lemma 6 states that, when $\tau = +\infty$, the scheduling in the current cycle is closely related to $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$ which is determined in the last cycle. For given requests $\mathbf{r}$, $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$ is desired to achieve equal $(a_w^l+\gamma_w)$ among all wavelengths.

When $0 < \tau < +\infty$, both preemptive and non-preemptive scheduling with minimizing the latest job completion time as the objective are NP-hard. $C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},\tau)$ cannot be expressed as a closed-form function of $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$. It can be estimated that, the smaller the τ, the less dependency of $C_{max}^{\mathfrak{S},p}(\mathbf{r},\tau)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},\tau)$ on $\{\lambda_1^{-1},\lambda_2^{-1},...,\lambda_m^{-1}\}$.

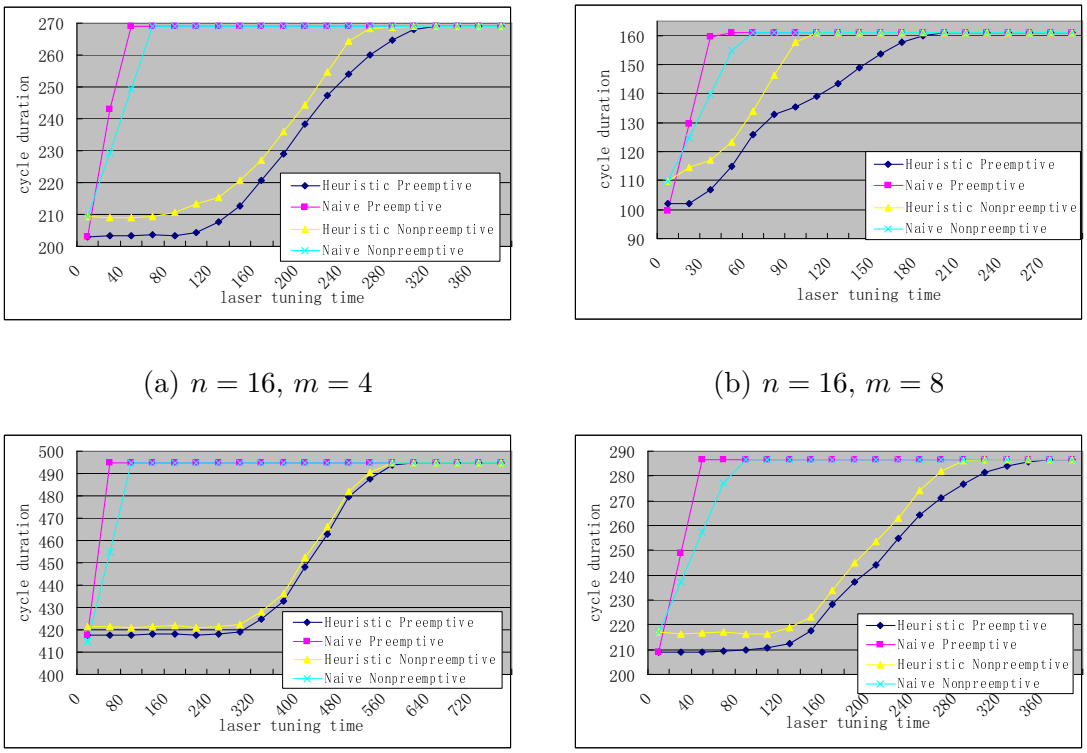
3.7 Simulation Results and Analysis

In this section, the cycle duration in the single DBA cycle case is first investigated, and then the performances in the multiple-cycle case are discussed.

3.7.1 The Single-Cycle Case

In the simulation, requests are expressed in terms of time durations, i.e., requested time durations. The number of ONUs n is set as 16 or 32, and the number of wavelengths m is set as 4 or 8. All the channels are available from time 0. The request sizes are uniformly distributed between 0 and 100, and the originally tuned wavelength λ_i^{-1} of ONU i at time 0 is set as $\lfloor i/m \rfloor$ such that each wavelength channel was tuned to by the same amount of ONUs at time 0. The cycle duration equals to the latest request completion time in this case. 200 sets of requests are randomly generated and their average performances are investigated.

Figure 3.4 shows the relation between the cycle duration and the laser tuning time. From the figure, the gap between $C_{max}^{\mathfrak{S},p}(\mathbf{r},0)$ and $C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$, and the gap between $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)$ and $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty)$ can also be observed. Since the cycle duration increases with the increase of the laser tuning time, $1 - C_{max}^{\mathfrak{S},p}(\mathbf{r},0)/C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$ can be interpreted as the maximum relative saving of the cycle duration benefited from the laser tunability. Simulation results in Figure 3.4 show that $1-C_{max}^{\mathfrak{S},p}(\mathbf{r},0)/C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$ increases with the increase of the number of ONUs and the decrease of the number



(a) $n = 16$, $m = 4$

(b) $n = 16$, $m = 8$

(c) $n = 32$, $m = 4$

(d) $n = 32$, $m = 8$

Figure 3.4 The cycle duration vs. laser tuning time.

of wavelengths. This is reasonable because large multiplexing gain can be exploited when the number of ONUs is large and the number of wavelength is small.

Owing to the NP-hard property of the nonpreemptive scheduling problem, the exact value of $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)$ cannot be obtained. However, $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)$ is between $C_{max}^{\mathfrak{S},p}(\mathbf{r},0)$ and the schedule length $C_{max}^{S,\bar{p}}(\mathbf{r},0)$ produced by naive preemptive scheduling. Simulation results in Figure 3.4 show that, in the four cases of combinations of ONU number and wavelength number, $C_{max}^{S,\bar{p}}(\mathbf{r},0)$ exceeds $C_{max}^{\mathfrak{S},p}(\mathbf{r},0)$ by no more than 10%. Hence, it can be inferred that $1 - C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)/C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty)$ also increases with the increase of the number of ONUs and the decrease of the number of wavelengths.

For naive preemptive scheduling, Section 3.4.1 shows that, for a given $\mathbf{r}$, $C_{max}^{S,p}(\mathbf{r},\tau) = \min\{C_{max}^{\mathfrak{S},p}(\mathbf{r},0) + 2\tau, C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)\}$. Therefore, when τ is not greater than
$(C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty) - C_{max}^{\mathfrak{S},p}(\mathbf{r},0))/2$, the cycle duration increases with the increase of τ by 2τ. When the laser tuning time increases beyond $(C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty) - C_{max}^{\mathfrak{S},p}(\mathbf{r},0))/2$, the cycle duration equals to $C_{max}^{\mathfrak{S},p}(\mathbf{r},+\infty)$. For naive nonpreemptive scheduling, $C_{max}^{S,\bar{p}}(\mathbf{r},\tau) = \min\{C_{max}^{S,\bar{p}}(\mathbf{r},0)+\tau, C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty)\}$. Therefore, the cycle duration remains constant as $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},+\infty)$ when the laser tuning time increases beyond $C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},\infty) - C_{max}^{\mathfrak{S},\bar{p}}(\mathbf{r},0)$.

For heuristic preemptive scheduling and heuristic non-preemptive scheduling, simulation results shown in Figure 3.4 demonstrate that they yield significant performance improvement as compared to naive preemptive scheduling and naive non-preemptive scheduling, respectively. The general trend of the relation between the cycle duration and the laser tuning time τ is as follows. When τ is below some value, referred to as "knee point 1", the cycle duration almost keeps as low as the case that $\tau = 0$. Beyond "knee point 1", the cycle duration increases with the increase of τ. Further increasing of τ to another value, referred to as "knee point 2", the cycle duration almost equals to the value obtained in the case that $\tau = +\infty$. When

$n = 16$ and $n = 4$, Figure 3.4(a) shows that, in the curve describing the relation between cycle durations produced by heuristic preemptive scheduling and the laser tuning time, "knee point 1" happens when τ is around 120 and $C_{max}^{\mathfrak{S},p}(\mathbf{r}, 0)$ is around 202. So, "knee point 1" is even larger than $C_{max}^{\mathfrak{S},p}(\mathbf{r}, 0)/2$. In addition, from Figure 3.4 it can be seen that "knee point 1" increases with the increase of the ONU number, and increases with the decrease of the wavelength number. This is again due to the fact that the multiplexing gain is large when the number of ONUs is large and the number of wavelengths is small. On the other hand, "knee point 2" is always around $C_{max}^{\mathfrak{S},p}(\mathbf{r}, +\infty)$ in all cases of n and m.

Besides, heuristic preemptive scheduling performs better than heuristic nonpreemptive scheduling because of the scheduling flexibility benefited from the allowance of preemption. For a given n and m, "knee point 2" of heuristic nonpreemptive scheduling is smaller than that of heuristic preemptive scheduling, and "knee point 2" of two algorithms are similar. The outperformance of heuristic preemptive scheduling over heuristic nonpreemptive scheduling is not obvious when $n = 32$ and $m = 4$, but is significant when $n = 16$ and $m = 8$. When $n = 16$ and $m = 4$, the outperformance is less than 5%.

3.7.2 The Multiple-cycle Case

For the multiple cycle case, the configuration of $n = 16$ and $m = 8$ is taken for example to investigate the performance. The simulation setup is as follows. The data rate on each wavelength channel is set as $1Gb/s$. A finite-time horizon with the time duration of 2 seconds is chosen. Assume the traffic of an ONU arrives in bursts. The burst size obeys Pareto distribution with the Pareto index $\alpha = 1.4$ and the mean equals to $31.25k$ bytes, which takes about 0.25ms to transmit. The burst inter-arrival time also obeys the Pareto distribution with $\alpha = 1.4$ and the mean equals to x. x is varied to obtain different traffic loads. The traffic load is defined as the

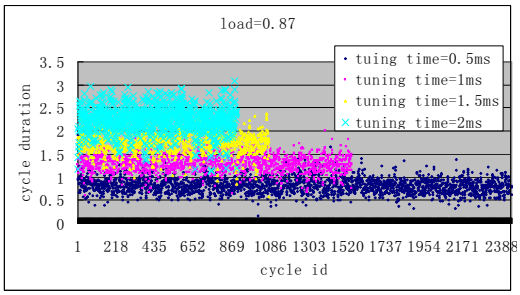
ratio between the total size of the arrival bursts and the maximum value which can be accommodated, which is $4 * 1Gb/s * 2s = 8G$ bits.

Consider the off-line scheduling framework, in which OLT performs DBA after receiving requests from all ONUs. Since naive preemptive scheduling and naive nonpreemptive scheduling perform much worse than heuristic preemptive scheduling and heuristic nonpreemptive scheduling, respectively, performances of heuristic preemptive scheduling and heuristic nonpreemptive scheduling in the multiple-cycle case are only investigated.

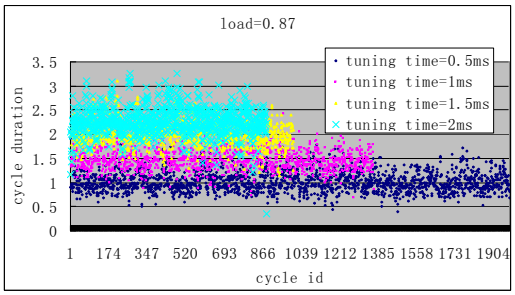
The cycle duration is adaptive to the incoming traffic. Therefore, the cycle duration varies over time. Different scheduling schemes produce different cycle durations, and thus result in different total number of cycles during the 2 seconds.

Figure 3.5(a) shows the cycle duration of heuristic preemptive scheduling under the condition that the traffic load equals to 0.87. When the laser tuning time equals to 0.5ms, the cycle duration is around 0.75ms, and the total number of cycles in the 2 seconds time period is around 2388. With the increase of the laser tuning time, the cycle duration increases, and the total number of cycles decreases. When the laser tuning time equals to 2ms, most of the cycles last between 1.75ms and 2.75ms, and the total number of cycles in the 2 seconds time period is decreased to around 869. The cycle durations produced by heuristic nonpreemptive scheduling are shown in Figure 3.5(b). As compared to Figure 3.5(a), the cycle durations produced by nonpreemptive scheduling algorithm in each case of laser tuning time increase slightly.

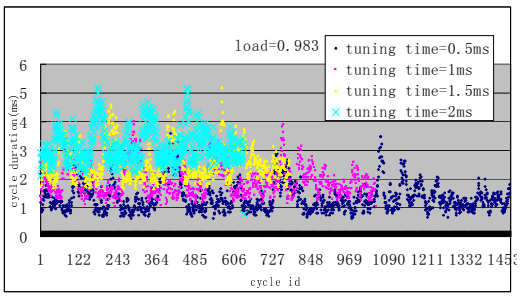
Figure 3.5(c) and Figure 3.5(d) show the cycle durations when the traffic load increases to 0.983. In comparing Figure 3.5(c) and Figure 3.5(a), the cycle duration greatly increases although there is only 0.113 increase of the traffic load. The cycle durations as shown in Figure 3.5(c) fluctuate tremendously when the load is 0.983. When the laser tuning time equals to 0.5ms, the cycle duration varies between 0.5ms and 3.75ms. Although the cycle duration fluctuates over time, its average value



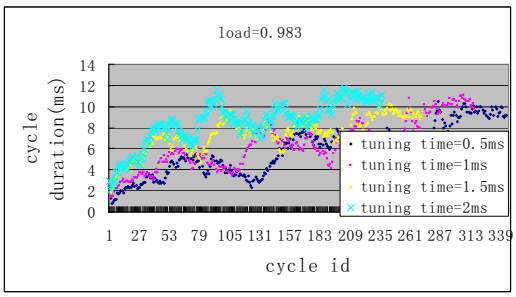
(a) Preemptive scheduling, load=0.87



(b) Non-preemptive scheduling, load=0.87



(c) Preemptive scheduling, load=0.983



(d) Non-preemptive scheduling, load=0.983

Figure 3.5 Variation of the cycle duration over time ($n = 16$, $m = 4$).

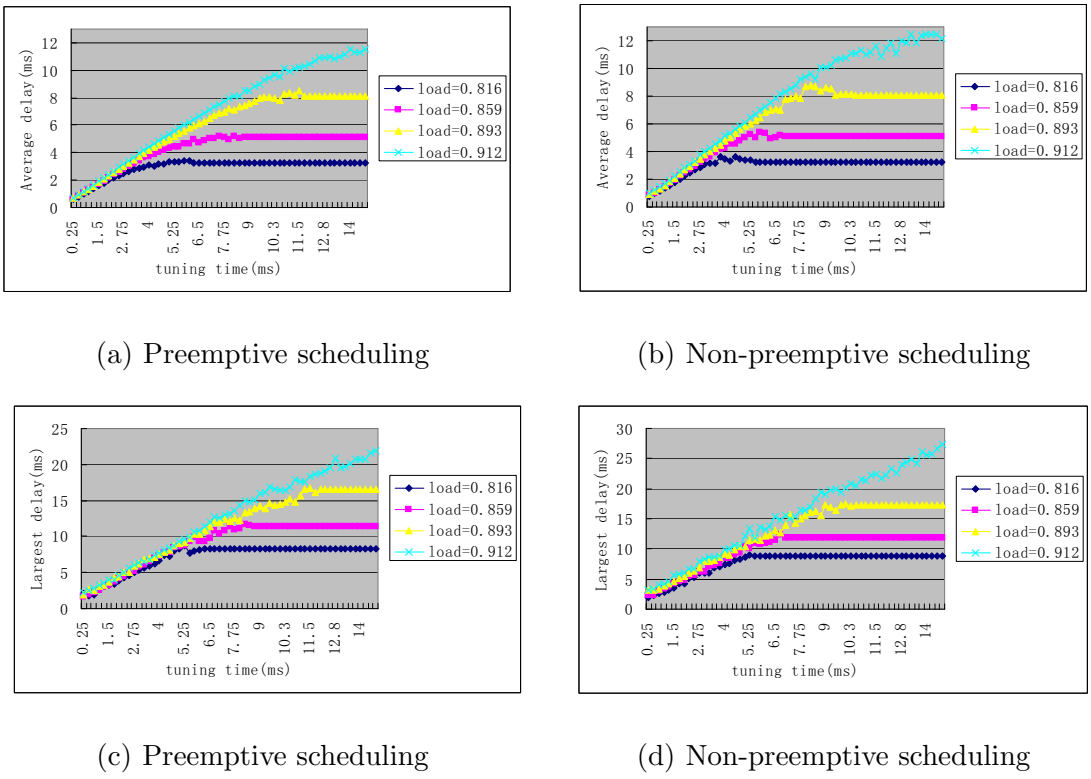
remains almost constant over time. This implies that heuristic preemptive scheduling can achieve throughput as high as 0.983. Figure 3.5(d) shows the cycle durations produced by heuristic nonpreemptive scheduling. The cycle duration fluctuation is more severe than that produced by heuristic preemptive scheduling. Also, cycle durations are much greater than those produced by heuristic preemptive scheduling. Moreover, the average values of the cycle durations keep increasing over time, implying that heuristic nonpreemptive scheduling is not able to admit traffic load as high as 0.983. Therefore, when the network is heavily loaded, heuristic preemptive scheduling can achieve significant better performance than heuristic nonpreemptive scheduling. However, when the network is lightly loaded, heuristic nonpreemptive scheduling yields similar performance as heuristic preemptive scheduling.

Figure 3.6 shows the average delay and the largest delay performance of the heuristic preemptive scheduling and the heuristic nonpreemptive scheduling. In offline scheduling of WDM PON, the arrival traffic in the current cycle will be transmitted in the next cycle. So, the average delay should be around the average cycle duration, and the largest delay should be around two times that of the largest cycle duration. The largest delay as shown in Figure 3.6(c) is around twice of the average delay as shown in Figure 3.6(a); it agrees with the analysis. In addition, for a given traffic load, delay generally increases with the increase of the laser tuning time. This is due to the fact that large laser tuning time results in large cycle duration, which further introduces large delay. When the laser tuning time increases to a certain value that the laser tunability cannot help improve the performance, delay will become constant.

Since heuristic preemptive scheduling produces a shorter cycle duration than heuristic nonpreemptive scheduling, delay performance of heuristic preemptive scheduling is better than that of heuristic nonpreemptive scheduling. However, the outperformance of heuristic preemptive scheduling over heuristic nonpreemptive scheduling is not obvious in four cases of traffic loads. When the laser tuning time is too large that tunability cannot help improve the system performance, delay performance only depends on the incoming traffic profile, and thus both heuristic preemptive scheduling and heuristic nonpreemptive scheduling achieve the same performance. However, when the traffic load is too large (such as 0.983), preemptive scheduling may still be able to achieve relatively stable delay performance, while non-preemptive scheduling may result in the phenomenon that delay keep increasing over time.

3.8 Summary

This chapter has investigated the dynamic wavelength assignment and time allocation problem in a single resource allocation cycle in hybrid WDM/TDM PONs with



(a) Preemptive scheduling

(b) Non-preemptive scheduling

(c) Preemptive scheduling

(d) Non-preemptive scheduling

Figure 3.6 Average delay and the largest delay vs. laser tuning time.

tunable lasers as optical light generators. The scheduling problem is mapped into a multi-processor scheduling problem, with wavelength channels as machines and ONU requests as jobs. Owing to the laser tuning time, jobs in this particular problem possess the property that sufficient guard time should be given when scheduling one job on two machines. It has been shown that both preemptive and non-preemptive scheduling problems with the objective of minimizing the schedule length are NP-hard when the laser tuning time is nonzero. Thus, heuristic scheduling schemes have been proposed for the case of arbitrary laser tuning time. Theoretical analyses show that the approximation ratios of the heuristic preemptive scheduling algorithm and the heuristic nonpreemptive scheduling algorithm are at most 2 and $2-1/m$, respectively, where m is the number of wavelengths. Simulation results show that our proposed heuristic scheduling algorithms achieve significantly better performances than naive algorithms which are directly derived from existing algorithms. It is also shown that

the preemptive scheduling algorithm achieves slightly better performances than the nonpreemptive scheduling algorithm when the network is lightly loaded. However, the preemptive scheduling algorithm performs significantly better when the traffic load exceeds some value.

CHAPTER 4

RESOURCE ALLOCATION IN INTEGRATED OPTICAL AND WIRELESS ACCESS NETWORKS

Radio-over-fiber (RoF) picocellular networks [67,68] are becoming promising options for delivering high speed wireless access services to accommodate bandwidth-demanding applications, such as HDTV. Instead of centrally locating antennas at the base station in conventional wireless networks, the RoF picocellular network distributes antennas over the cell to get closer to mobile users. This can increase the signal to noise ratio (SNR) at the receiver and thereby increase the wireless access data rate. The coverage area of each antenna is greatly reduced as compared to the conventional cell, thus resulting in the sharing of wireless resources among a smaller number of users, and increasing the bandwidth share of each user.

Typically, in RoF picocellular networks, upstream wireless signals are first sent to distributed antennas, and then converted to optical signals and further transmitted to the base station which is usually located at the central office. The downstream signal transmits in the opposite direction. In the physical layer, radio signals are usually delivered directly at high frequencies to/from the base station by utilizing RoF transmission technology. The simple structure of antennas makes the RoF network a promising cost-effective wireless access solution especially for in-building environment such as airports, conference centers, shopping malls, stadiums, and subways [69]. Owing to the high bandwidth provisioning, RoF enables promising applications in many network scenarios such as fourth-generation (4G) wireless systems and wireless local area networks (WLANs) [70].

RoF picocellular networks can be considered as the integration of wireless access and optical access. The wireless access refers to the communication between mobile

users and antennas, whereas the optical access refers to the communication between antennas and the base station. In this chapter, orthogonal frequency-division multiple access (OFDMA) is considered as the wireless modulation and access method [71,72]. OFDMA, well known for its immunity to multipath interference, has been adopted by both LTE and WiMAX as the downlink access scheme [73]. OFDMA divides the frequency band into non-overlapping orthogonal OFDMA subcarriers. These subcarriers can be flexibly allocated to individual mobile users at different time slots based on the real-time incoming user traffic demands and wireless channel status.

For the optical access, one solution is to use one single upstream/downstream wavelength to carry traffic of all mobile users. Then, OFDMA subcarriers carried over the wavelength are shared by all users in the picocellular network. However, the single wavelength may not be able to accommodate future bandwidth-demanding multimedia applications. To meet the ever increasing bandwidth requirement, 4G wireless systems are being rapidly developed and deployed, and the optical access systems in particular passive optical networks (PONs) are increasing upstream or downstream wavelengths [1, 47, 74–79]. In order to accommodate the growing traffic in wireless networks especially 4G systems, wavelength-division multiplexing (WDM) is adopted in the optical access. In WDM optical access networks, the downstream optical signals are usually demultiplexed into individual wavelengths and delivered to picocells by using demultiplexing devices such as arrayed waveguide gratings (AWG), optical add-drop multiplexer (OADM), or wavelength filters, and the upstream optical signals modulated onto certain wavelengths are first generated by lasers at antennas, and then multiplexed onto a single fiber by using multiplexing devices such as couplers [68, 80].

For high bandwidth provisioning, one or more wavelengths is expected to be dedicated for each picocell. However, a large quantity of wavelengths are needed for a large-scale picocellular network; this further incurs high network cost since the prices

of WDM optical devices are usually high when the number of supporting wavelengths is large. For a reasonable network cost, the number of wavelengths may fall below the number of picocells in large picocellular networks. In this case, multiple picocells need to share the same wavelength.

In consideration of the scenario that one WDM wavelength is shared among multiple picocells, the chapter investigates the wavelength assignment and OFDMA resource block (RB) allocation problems in the OFDMA-based WDM RoF network. For a better description, the area covered by picocells sharing the same wavelength is referred to as a *nanocell*. Figure 4.1 shows one example of the RoF picocellular network. The base station which is typically located at the central office is connected with multiple antennas via optical fibers [70]. In the example shown in Figure 4.1, the base station connects with 36 antennas, among which each set of 9 antennas covers a nanocell. Note that a nanocell may not cover a continuous geographic area. Since one OFDMA RB in an OFDMA symbol can only be allocated to one picocell in a nanocell at a time, the inter-nanocell interference is eliminated. However, interferences between picocells in different nanocells may still exist when those picocells are allocated with OFDMA RBs of the same frequency. Such inter-nanocell interference can be minimized by assigning RBs of the same frequency to picocells which pose the least interference to each other. However, this may result in some picocells being allocated with a large number of OFDMA RBs, while some other picocells may be allocated with few OFDMA RBs. In picocells which are allocated with many OFDMA RBs, each OFDMA RB may receive a very limited power share owing to the power constraint of the picocell. Thereby, the signal to interference plus noise ratio (SINR) perceived by users allocated with the RB is small even if the inter-picocell interference can be avoided.

This chapter considers the power constraints of picocells and investigates the problem of minimizing the inter-nanocell interference in allocating OFDMA RBs

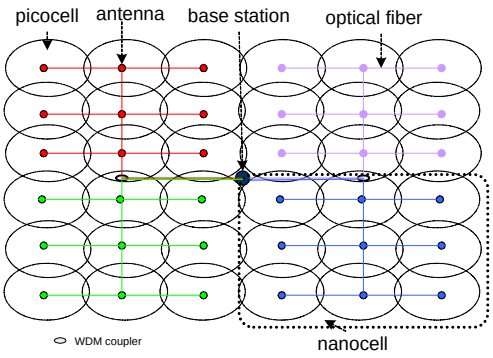


Figure 4.1 WDM Radio-over-fiber picocellular network architecture.

and assigning wavelengths. Specifically, conflict graphs are employed to characterize constraints of OFDMA RB allocation and wavelength assignment. By using conflict graphs, the problem of maximizing the number of allocated OFDMA RBs is proved to be strong NP-hard when no inter-picocell interference is allowed and the power constraints of picocells are considered. It is also shown that the problem of allocating OFDMA resources at a time instance can be polynomially reducible to graph problems. Finally, the wavelength assignment problem with the objective of maximizing the number of assigned RBs is heuristically mapped into graph partitioning problems, and algorithms are proposed to address these problems.

The rest of the chapter is organized as follows. Section 4.1 describes the system model, presents the formal formulations of the problems, and discusses related works. Section 4.2 discusses the OFDMA resource allocation problem. Section 4.3 investigates the wavelength assignment problem. Section 4.4 presents and analyzes extensive simulation results. Section 4.5 presents concluding remarks.

4.1 System Model, Problem Formulation, and Related Works

4.1.1 System Model

Similar to WiMAX, the OFDMA radio resource is assumed to be partitioned in both time and frequency domains [81–83]. Specifically, the frequency resource is

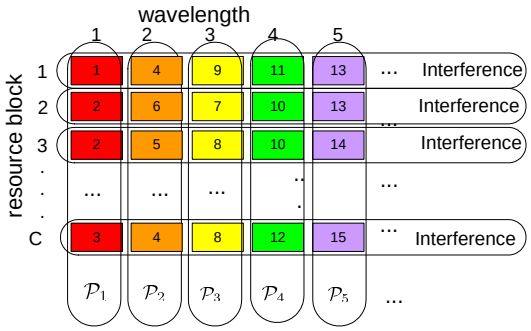


Figure 4.2 One example of OFDMA RB allocation at a time instance.

divided into multiple non-overlapping OFDMA RBs, each of which contains a subset of OFDMA subcarriers. OFDMA RB in a time slot serves as the minimum unit of resource allocation. Picocells in the same nanocell share OFDMA resources in an OFDMA symbol. Since picocells in the same nanocell are not allocated with the same RB at a time, they do not pose interferences to each other. However, picocells in different nanocells may interfere with each other. Figure 4.2 shows one example of the wavelength assignment and RB allocation at a time instance. Let set $\mathcal{P}_w$ contain picocells in the nanocell assigned with wavelength w. In the example, $\mathcal{P}_1 = \{1, 2, 3\}$, $\mathcal{P}_2 = \{4, 5, 6\}$, $\mathcal{P}_3 = \{7, 8, 9\}$, etc., and Picocells 1, 4, 9, 11, and 13 are allocated with RB 1, and interfere each other.

To minimize the inter-nanocell interference, picocells which pose the least interferences to each other should be selected and assigned with the same RB. However, considering the interferences only may result in some picocells being over-allocated with many RBs while some others under-allocated with few RBs. Owing to the power constraint at each picocell, the RB will get small power share if the picocell is allocated with many RBs, thereby reducing the signal power and limiting the user data rate. Therefore, both the inter-nanocell interference and power constraints of picocells need to be considered so as to maximize the total delivered data rates at a time.

The following assumptions about the interference and user data rate are invoked.

- *Assumption 1*: Assume that wireless channel interference dominates the optical wavelength channel interference, and consider the wireless channel interference only.

- *Assumption 2*: Similar to the interference model in [84–87], consider the binary case of the interference between picocells and assume that the wireless channel is RB inselective. Denote $I_{p,p'}$ as the interference between picocell p and picocell p'. If the transmission of RBs in picocell p interfere that in picocell p', $I_{p,p'} = 1$; otherwise, $I_{p,p'} = 0$. Assume the interference is symmetric, i.e., $I_{p,p'} = I_{p',p}$. Since the wireless channel condition is dynamically changing, the interference $I_{p,p'}$ between two picocells changes over time.

- *Assumption 3*: Considering the power constraint of each picocell, each picocell is assumed to be allocated with at most C/P RBs at a time, where C is the number of RBs and P is the number of picocells in a nanocell. Each RB is also assumed to be allocated with the same amount of power.

- *Assumption 4*: This chapter does not investigate the problem of further allocating OFDMA RB to mobile users, but assume that the maximum rate delivered by any RB at any picocell is the same when there is no inter-nanocell interference.

4.1.2 Mathematical Formulation

This chapter investigates a slot-based wavelength assignment and OFDMA RB allocation scheme. To achieve high throughput, the total transmitted data rates of all RBs in a time slot is maximized.

Let W be the number of WDM wavelengths, and set $\mathcal{Q}$ contain all the picocells which have backlogged traffic in the time slot. The wavelength assignment problem is to divide set $\mathcal{Q}$ into W subsets, each of which is assigned with one wavelength.

As defined earlier, $\mathcal{P}_w$ contains picocells in the nanocell assigned with wavelength w. Then, $\cup_{w=1}^{W}\mathcal{P}_w = \mathcal{Q}$ and $\mathcal{P}_w \cap \mathcal{P}_{w'} = \emptyset, \forall w \neq w'$. Each nanocell is assumed to contain the same amount of picocells. Denote P as the number of picocells in a nanocell. Then, $P = |\mathcal{P}_w| = |\mathcal{Q}|/W, \forall w$.

Denote $x_{w,c}$ as the picocell to which RB c carried by wavelength w is assigned at a time instance. $x_{w,c} \in 0 \cup |\mathcal{Q}|$. $x_{w,c} = 0$ if RB c carried by wavelength w is not assigned to any picocell. Denote $y_{w,c}$ as the indicator of whether RB c on wavelength w is allocated. $y_{w,c} \in \{0, 1\}$. $y_{w,c} = 1$ if $x_{w,c} > 0$; $y_{w,c} = 0$, otherwise. Since the rate delivered by any RB at any picocell is assumed to be the same (see Assumption 4), the data rate delivered to picocell p is proportional to the number of RBs allocated to picocell p, i.e., $\sum_{\{c|x_{w,c}=p\}} y_{w,c}$.

The total transmitted data rates of all RBs in the network is proportional to $\sum_p \sum_{\{c|x_{w,c}=p\}} y_{w,c} = \sum_w \sum_c y_{w,c}$. Then, the joint wavelength assignment and OFDMA RB allocation problem with the objective of maximizing the total transmitted data rates in the network subject to the constraints that no interferences are allowed can be described as follows.

Given: $I_{WP\times WP}$ and set $\mathcal{Q}$

Determine: $\mathcal{P}_w, \forall w$ and $x_{w,c}, \forall w, c$

Objective: maximize $\sum_w \sum_c y_{w,c}$

Subject to:

$$\cup_{w=1}^{W}\mathcal{P}_w = \mathcal{Q} \tag{4.1}$$

$$\mathcal{P}_w \cap \mathcal{P}_{w'} = \emptyset, \forall w \neq w' \tag{4.2}$$

$$x_{w,c} \in \mathcal{P}_w \tag{4.3}$$

$$I_{x_{w,c},x_{w',c}} = 0, \forall c, \forall w \neq w' \tag{4.4}$$

$$\sum_{\{c|x_{w,c}=p\}} y_{w,c} \leq C/P, \forall w, \forall p \in \mathcal{P}_w \tag{4.5}$$

Constraints (4.1) and (4.2) describe the wavelength assignment constraints. Constraint (4.4) states that RB c cannot be allocated to two picocells posing interferences to each other. Constraint (4.5) limits that the number of RBs allocated to any picocell p in any nanocell w cannot be greater than C/P.

The problem is decomposed into two subproblems, i.e., wavelength assignment and OFDMA RB allocation. The OFDMA RB allocation problem can be formulated as:

Given $I_{WP\times WP}$ and $\mathcal{P}_w, \forall w$, determine $x_{w,c}, \forall w, c$ subject to constraints (4.3 - 4.5).

A different wavelength assignment scheme $\{\mathcal{P}_w\}_{w=1}^{W}$ may result in different $f(\{\mathcal{P}_w\}_{w=1}^{W})$. Let $f(\{\mathcal{P}_w\}_{w=1}^{W})$ be the maximum number of assigned RBs with respect to a wavelength assignment scheme $\{\mathcal{P}_w\}_{w=1}^{W}$. The wavelength assignment problem can be formulated as:

Given the interference $I_{WP\times WP}$ and set $\mathcal{Q}$, find $\{\mathcal{P}_w\}_{w=1}^{W}$ such that $f(\{\mathcal{P}_w\}_{w=1}^{W})$ is maximized subject to constraints (4.1) and (4.2).

Since $I_{WP\times WP}$ is time varying (see Assumption 2), the optimal wavelength assignment changes over time. To dynamically assign wavelengths, antennas need to be equipped with wavelength tunable transceivers, which are currently still cost-prohibitive. If wavelength-fixed devices are employed in the network, the problem of determining the wavelength supported by each optical transceiver can be similarly formulated by replacing the real-time interference matrix $I_{WP\times WP}$ with statistical interference $I_{WP\times WP}$.

The OFDMA RB allocation problem and wavelength assignment problem will be next respectively addressed. Table 4.1 lists notations used in the chapter.

Table 4.1 Notations

Symbol	Definition
W	The number of wavelengths or nanocells
P	The number of picocells in a nanocell
C	The number of OFDMA RBs in an OFDMA symbol
$\mathcal{Q}$	The set of all picocells
$\mathcal{P}_w$	The set which contains picocells in the nanocell assigned with wavelength w
$x_{w,c}$	The picocell to which RB c carried by wavelength w is assigned
$y_{w,c}$	The indicator of whether RB c carried by wavelength w is allocated
$I_{p,p'}$	The binary interference between picocell p and picocell p'
$f(\{\mathcal{P}_w\}_{w=1}^{W})$	The maximum number of assigned RBs with respect to a wavelength assignment scheme $\{\mathcal{P}_w\}_{w=1}^{W}$
$\mathcal{G}(V,E)$	The constructed conflict graph
$\mathcal{G}^{\alpha}(V,E^{\alpha})$	The conflict graph containing interference edges only
$\mathcal{G}^{\beta}(V,E^{\beta})$	The conflict graph containing co-nanocell edges only
$\mathcal{N}(\mathcal{G}(V,E))$	The maximum number of vertices in graph $\mathcal{G}(V,E)$ which can be colored by using P colors

4.1.3 Related Works

Formerly, dynamic power and resource allocation have been proposed to maximize the sum of throughput over all users or equalize user throughput in OFDMA-based cellular networks [88, 89]. Zhu *et al.* [90] presented chunk-based OFDMA subcarrier allocation schemes to simplify the subcarrier allocation algorithm and reduce the overhead. From the combinatorial optimization perspective, Reuven *et al.* [91] investigated the issue of properly selecting packets to be transmitted, determining Phy-profiles for each packet, and constructing OFDMA frame matrix such that the profit gained by the transmitted traffic can be maximized. Lee *et al.* [83] tried to optimally select the MIMO mode (multiplexing or diversity) so as to maximize the proportional fairness criterion with the constraints that only one mode can be selected per user per time interval. For multicell wireless networks, Wang *et al.* [92] investigated the direct sequence code division multiple access (DS-CDMA) microcellular network operating over a multipath Rician fading channel and sharing common spectrum with various narrowband waveforms. To reject the intra-cell as well as inter-cell interference, a suppression filter was equipped at each CDMA receiver and its performance was investigated. Sang *et al.* [93] proposed a scalable cross-layer framework to coordinate the packet-level scheduling, call-level cell selection, and system-level cell coverage for CDMA systems. Gault *et al.* [94] investigated the power and subcarrier allocation issue with the objective of minimizing the total transmitted power based on the statistical knowledge of the user channels.

Resource allocation in WDM access networks also received intensive attention in the past. McGarry *et al.* [51] modeled the wavelength assignment problem into a multiprocessor scheduling problem and proposed to use the longest processing time (LPT) first rule to address the minimizing makespan problem for the case that ONUs can access all the wavelengths. Meng *et al.* [52] studied the joint grant scheduling and wavelength assignment problem. They formulated it into a mixed

integer linear programming (MILP) problem, and employed tabu search to obtain the optimal solution. In [43, 44], the capacity of WDM passive optical networks is theoretically analyzed. In [95], with consideration of the laser tuning time, wavelength scheduling schemes are proposed to schedule ONU traffic as early as possible in hybrid WDM/TDM PONs.

Regarding the optical and wireless integration, Sarkar *et al.* [96, 97] proposed a hybrid wireless-optical broadband access network (WOBAN) and employed the Lagrangian relaxation technique to address the problem of optimal placement of ONUs and BSs. In WOBAN, mobile users communicate with a wireless BS, which is connected to the ONU. Koonen *et al.* [98] proposed a fiber-wireless network which uses a flexible wavelength router at a local spitting center to adjust wavelength routing between OLT and ONUs. In this case, the wavelength can be dynamically assigned to each ONU/cell. For the wireless access part, the radio access function is integrated with ONUs [98]. The two integrated optical-wireless networks share one common characteristic, that is, the radio access controller is responsible for the wireless resource allocation of a single cell only, which is different from our case that the base station controls wavelength assignment and OFDMA resource allocation in all picocells.

To the best of our knowledge, our proposal is the first attempt to tackle the wavelength assignment and OFDMA resource allocation problem in OFDMA-based WDM RoF networks, in which wavelength assignment and OFDMA resource allocation need to be properly tackled in consideration of the inter-nanocell interference.

4.2 OFDMA Resource Allocation

In this section, the OFDMA resource allocation problem is first transformed into graph problems, then their complexities are analyzed. Finally, solutions are proposed to address them.

4.2.1 Conflict Graph

Following the idea of modeling binary interferences among nodes in wireless networks [84–87, 99], conflict graph is used to model the interferences in this chapter. Besides the interference, the co-nanocell scheduling constraints are characterized by using the conflict graph as well. Denote $\mathcal{G}(V, E)$ as the conflict graph. In $\mathcal{G}(V, E)$, vertices represent picocells, and $|V| = WP$. Edges characterize the scheduling constraints among picocells. Two vertices are connected if they cannot be allocated with the same RB at a time.

There are two kinds of edges in $\mathcal{G}(V, E)$. When $I_{p,p'} = 1$, picocell p interferes with picocell p', and hence vertices p and p' are connected. These edges are referred to as *interference edges*. When picocell p and picocell p' are within the same nanocell, they cannot be allocated with the same RB at a time, and hence are connected. These edges are referred to as *co-nanocell edges*. Note that an edge can be both co-nanocell edge and interference edge. The graph containing interference edges only is denoted as $\mathcal{G}^{\alpha}(V, E^{\alpha})$, and that containing co-nanocell edges only is denoted as $\mathcal{G}^{\beta}(V, E^{\beta})$. Then, $E = E^{\alpha} \cup E^{\beta}$.

Figure 4.3 shows one example of the conflict graph. The network contains 16 picocells, among which four nearby picocells constitute a nanocell. Figure 4.3 (b) shows the conflict graph with the interference edges only, and Figure 4.3 (a) shows the conflict graph with the co-channel edges only. Some edges are both co-nanocell edges and interference edges as shown in Figure 4.3 (c).

By using conflict graphs, the OFDMA RB allocation problem is transformed into the problem of labeling the vertices by RB id such that no two adjacent vertices are labeled with the same RB id. The objective of maximizing the number of allocated RBs is equivalent to that of maximizing the sum of labels labeled on all vertices. Note that one vertex can be labeled with more than one RB id since one picocell can be

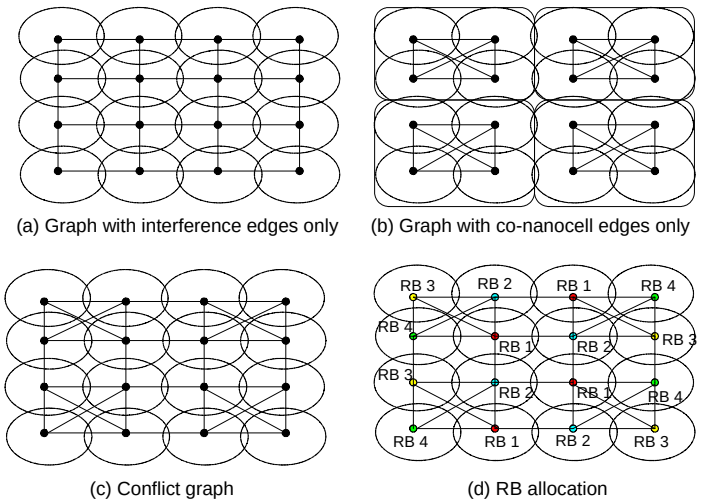


Figure 4.3 An example of conflict graph and its coloring.

allocated with more than one RB at a time. Figure 4.3 (d) shows one RB labeling scheme with four RBs for the conflict graph as shown in Figure 4.3 (c).

4.2.2 Computational Complexity

The OFDMA RB allocation problem is shown to be NP-hard in the strong sense.

Theorem 7. *The OFDMA RB allocation problem with the objective of maximizing the number of allocated RBs at a time is strong NP-hard.*

Proof. The strong NP-hardness property of the OFDMA RB allocation problem is proved by showing that the maximum independent set problem is reducible to this problem.

Given a graph $G(V, E)$, the independent set is a set containing vertices of which no two vertices are adjacent. The maximum independent set problem is to find the independent set with the largest size.

Consider an arbitrary instance of the maximum independent set problem for graph $G(V, E)$. An equivalent OFDMA RB problem is constructed. Let both the total number of picocells PW and the number of wavelengths W be $|V|$, and graph

$G(V, E)$ be the conflict graph. Then, $P = 1$, each picocell has a dedicated wavelength, and each vertex v in the conflict graph can be labeled with at most C RBs.

It is shown that the optimal labeling scheme is to label all vertices in the maximum independent set of graph $G(V, E)$ with all RBs, and leave all the other vertices unlabeled. The vertices labeled by any RB id must be in an independent set. So, the maximum number of vertices a RB can be labeled equals to to the size of the largest independent set. This scheme achieves the maximum number. Therefore, finding the optimal labeling is equivalent to finding the maximum independent set. The independent set problem is known to be strong NP-hard. Thus, the RB allocation problem is strong NP-hard. □

Owing to the NP-hardness property of the problem, the brute force search may be employed to find the optimal solution. To examine whether or not the brute-force search is practical, the running time of the brute force search is evaluated for this problem.

Lemma 4. *The running time of the brute-force search for the optimal solution to the OFDMA resource allocation problem is* $O(P^{CW})$.

Proof. Each RB can be allocated to any picocell in a nanocell, and thus the number of choices is P. For the total of C resource blocks, the number of choices is P^C in a nanocell. For the total of W nanocells, the total choices is P^{CW}. It is exponential both in C and W. □

Typically, the number of resource blocks C is 25, 50, 75, 100 in 3GPP LTE; the number of WDM wavelengths W in PONs is 2, 4, 8, 16, 32; the number of picocells P in a nanocell can be in the order of tens. Therefore, the brute force scheme is highly impractical.

4.2.3 OFDMA RB Allocation Algorithms

Here, optimal and heuristic algorithms are developed to address the OFDMA RB allocation problem.

Vertex-coloring-based RB Allocation: The first algorithm proposed is vertex-coloring-based RB allocation. First, the problem under two extreme cases of the nanocell size P are considered.

Based on the proof of Theorem 7, the following Lemma 5 is derived.

Lemma 5. *When the number of picocells P in a nanocell equals to 1, the RB allocation problem is equivalent to the maximum independent set problem.*

Proof. See the proof of Theorem 7. □

Lemma 6. *When the number of picocells P in a nanocell equals to the number of RBs, i.e., C, the RB allocation problem is polynomially reducible to the vertex coloring problem.*

Proof. When $P = C$, each picocell can be allocated with at most one RB based on Assumption 3. The objective of maximizing the number of labels labeled on all vertices is equivalent to that of maximizing the number of labeled vertices. If the conflict graph is C-colorable, by regarding each color as a RB, a labeling is obtained to achieve PW labeled vertices. When $n < PW$, for any feasible labeling with n labeled vertices, the graph after removing $PW - n$ unlabeled vertices along with their connecting edges is C-colorable. There are $\binom{PW}{n}$ choices of choosing n vertices from the total of PW vertices. That is to say, the decision problem of determining whether n is achievable is polynomially reducible to the vertex coloring problem. Therefore, the RB allocation problem is polynomially reducible to the vertex coloring problem when $P = C$. □

The maximum independent set problem can be considered as a special case of vertex-coloring problem, where the number of colors is one. Therefore, problems

under both of these two extreme cases are polynomially reducible to the vertex-coloring problem. Thereby, a vertex-coloring-based RB allocation approach is proposed as described in Algorithm 10.

Algorithm 10 Vertex-coloring-based RB allocation

1: Divide RBs evenly into P groups, and include RBs $(j-1)C/P+1, ..., jC/P$ into the jth group $(1 \leq j \leq P)$.
2: $n = PW$
3: $ind = 0$
4: **while** $ind = 0$ **do**
5: Determine whether P colors can color n vertices
6: **if** Yes, **then**
7: Color these n vertices, and include vertices colored by color j into set ψ^j
8: Label RBs in group j onto vertices in ψ^j
9: $ind = 1$
10: **else**
11: $n = n - 1$
12: **end if**
13: **end while**

The main idea of Algorithm 10 is to color vertices as much as possible using P colors. First, the algorithm tries to color all vertices by using P colors. If it cannot be achieved, the algorithm removes one vertex, and tries to color all the remaining vertices by using P colors. The process repeats until the maximum number of vertices colored by P colors is found. Let n equal to the maximum number of colored vertices. For vertices colored by the same color, Algorithm 10 labels each of them with C/P RBs among all RBs. Let ψ^j include vertices colored by color j. The total number of labels equals to $\sum_{j=1}^{P} |\psi^j| C/P$.

Besides the above two extreme cases, Algorithm 10 can achieve the optimal value in every scenario of P and C.

Theorem 8. *The optimal solution to the OFDMA RB allocation problem can be obtained by using Algorithm 10*

Proof. Let n be the maximum number of vertices colored by P colors. That is, there are at most n vertices contained in the union of P independent sets of the conflict graph. In Algorithm 10, any P of these C RBs are allocated to n vertices. The total number of labels labeled on vertices equals to nC/P. Assume there exists a scheme that achieves $m(m > nC/P)$ labels, then there must exist P RBs being allocated to more than n vertices. Vertices allocated with the same RB constitute an independent set. Then, there exist P independent sets whose union is of size greater than n. This contradicts the fact that n is the maximum number of vertices colored by P colors. □

The following corollaries pertain to the optimal value and the graph which can achieve the upper bound CW.

Corollary 2. *The maximum number of allocated RBs at a time equals to $\mathcal{N}(\mathcal{G}(V,E))C/P$, where $\mathcal{G}(V,E)$ is the conflict graph, and $\mathcal{N}(\mathcal{G}(V,E))$ is the maximum number of vertices in graph $\mathcal{G}(V,E)$ which can be colored by using P colors.*

Proof. According to Theorem 8, the maximum number of RBs allocated at a time equals to $n \cdot C/P$, where n is the maximum number of vertices which can be colored by P colors in graph $\mathcal{G}(V,E)$. Thus, this corollary has been proved. □

Corollary 3. *The total number of RBs which can be allocated at a time achieves the upper bound CW if and only if the conflict graph is P-colorable.*

Proof. If the conflict graph is P-colorable, $\mathcal{N}(\mathcal{G}(V,E)) = PW$, and the total number of allocated RBs equals to $PW \cdot C/P = WC$; otherwise, $\mathcal{N}(\mathcal{G}(V,E))$ is less than PW, and thus the total number of allocated RBs is less than CW. □

Computational Analysis: In Algorithm 10, the vertex coloring problem, which is known to be strong NP-hard, needs to be addressed. The brute force search scheme for a graph with $|V|$ vertices and P colors runs in time $O(P^{|V|})$. Line 4 of Algorithm 10 involves checking whether n among PW vertices can be colored by P colors. For a given n, the running time from Line 3 to Line 11 is $O(\binom{PW}{n}P^n)$. Thus, the running time of Algorithm 11 is $O(\sum_{n=1}^{PW}\binom{PW}{n}P^n) = O((1+P)^{PW})$. By eliminating the dependence on C which can be up to 100, the vertex-coloring-based scheme has a smaller running time as compared to the brute-force search solution to the original problem, which is $O(P^{CW})$. However, it is still impractical since the running time is exponential in both P and W.

Independent-set-based RB Allocation: An independent-set-based RB allocation scheme as described in Algorithm 11 is proposed to obtain a more efficient algorithm.

Algorithm 11 Independent-set-based RB allocation

1: Divide RBs evenly into P groups, and include RBs $(i-1)\frac{C}{P}+1, ..., i\frac{C}{P}$ into the ith RB group $(1 \leq i \leq P)$.
2: Let $G = \mathcal{G}(V, E)$,
3: **for** $i = 1 : P$ **do**
4: Find the maximum independent set in graph G, and denote the set as ϕ^i
5: Label all vertices in ϕ^i with RBs in group i.
6: Remove vertices in ϕ^i along with their connecting edges from graph G.
7: **end for**

In Algorithm 11, RBs are first divided into P groups, where group i contains RB $\{(i-1)\frac{C}{P}+1, (i-1)\frac{C}{P}+2, ..., i\frac{C}{P}\}$. Graph G is initialized as $\mathcal{G}(V, E)$. Then, the maximum independent set is found in graph G, and all vertices in the independent set are labeled with RBs in a RB group. After that, graph G is updated by removing

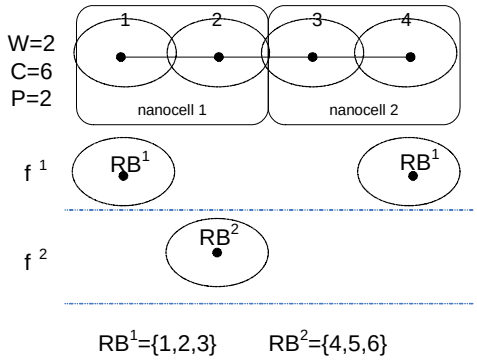


Figure 4.4 One example of Heuristic-1 for $P = 2$ and $W = 2$.

all vertices in the independent set along with their connecting edges. The process is repeated until all RBs are labeled.

Denote ϕ^i as the maximum independent set in the ith iteration. The number of vertices labeled with RB j with $(i-1)\frac{C}{P}+1 \leq j \leq i\frac{C}{P}$ equals to the size of ϕ^i. The total number of labels labeled on all vertices equals to $\sum_{i=1}^{P} |\phi^i|\frac{C}{P}$.

In the ideal case, $|\phi^i| = W, \forall i$. Then, each RB is labeled on W vertices, and the number of total labels labeled on vertices equals to $P \cdot (W\frac{C}{P}) = WC$, which is the upper bound of the optimal value. However, the size of the independent set may decrease iteration by iteration. This happens for conflict graphs with optimal values below the upper bound WC. Another reason may be due to the greedy nature of Algorithm 11. Algorithm 11 greedily selects the maximum independent set in each iteration. This may decrease the size of the maximum independent set in the subsequent iterations.

Figure 4.4 shows one simple example with two nanocells and four picocells. In iteration 1, the independent set contains picocell 1 and 4, whereas the independent set in iteration 2 can only contain either picocell 2 or 3. Let RB^i denote the ith RB group. Then, RBs in set RB^2 can only be labeled on one vertex. It is not difficult to see that the optimal solution is to let $\phi^1 = \{1, 3\}$ and $\phi^2 = \{2, 4\}$.

Computational Analysis: Algorithm 11 involves addressing the maximum independent set problem, which is known to be NP-hard. The brute force approach of checking every vertex subset for a graph with $|V|$ vertices runs in time $O(2^{|V|})$. The problem can be solved by more efficient exact algorithms, for example, the algorithm with time bound of $O(2^{|V|/3})$ proposed by Tarjan [100], and the measure and conquer approach with time bound of $O(2^{0.287|V|})$ [101]. The best known is the one with time bound of $O(2^{0.276|V|})$ proposed by Robson [102]. In Algorithm 11, the graph sizes in these P iterations are PW, $PW - |\phi_1|$, $PW - |\phi_1| - |\phi_2|$,..., $PW - \sum_{i=1}^{P-1} |\phi_i|$. Thus, the running time of Algorithm 11 is $O(P2^{0.276PW})$ if Robson's algorithm is used. The running time is approximately $P(1.21/(1+P))^{PW}$ of that of Algorithm 10.

Greedy RB Allocation: Although Algorithm 11 has a reduced running time as compared to that of Algorithm 10 and the brute force approach to the original problem, it is still exponential in P and W, and becomes impractical when P and W are large. To further reduce the running time, heuristic graph coloring or maximum independent set algorithms can be employed.

There are numerous heuristic graph coloring and maximum independent set algorithms. This chapter by no means applies each of them into Algorithm 10 and Algorithm 11, and discusses their performances. For computational efficiency, the following greedy graph coloring algorithm and greedy maximum independent set algorithm are incorporated into Algorithm 10 and Algorithm 11, respectively.

Greedy maximum independent set algorithm: include the vertex with the least degree in the independent set, and remove vertices connected to the vertex from the graph. This process repeats until no more vertex can be included.

Greedy vertex coloring algorithm: order the vertices in the ascending order of their degrees, and assign vertex v with the smallest available color which is not used by adjacent vertices of vertex v, and add a fresh color if needed.

After applying the above greedy graph coloring algorithm and greedy maximum independent set algorithm, both Algorithms 10 and 11 are reduced to the following greedy Algorithm 12. In Algorithm 12, vertices are first ordered in the ascending

Algorithm 12 Greedy RB allocation

1: Divide RBs evenly into P groups, and include RBs $(j-1)C/P+1, ..., jC/P$ into the jth group $(1 \leq j \leq P)$.
2: Sort vertices in the conflict graph in the ascending order of their degrees.
3: **for** $i = 1 : PW$ **do**
4: $j = 1$
5: **while** vertex i has not been colored & $j \leq P$ **do**
6: **if** vertex can be colored with color j **then**
7: color it
8: **else**
9: $j = j + 1$
10: **end if**
11: **end while**
12: **end for**
13: Label vertices colored by color j with RBs in group j

order of their degrees, and then colored by one of these P available colors. When there are multiple colors available, the one with the smallest index is selected. Good performance requires the number of uncolored vertices to be as small as possible.

Computational Analysis: By using a proper ordering algorithm, the complexity of the ordering process in Line 1 of Algorithm 12 is $O(PW \log(PW))$. For each vertex, the process of selecting colors is of complexity $O(P)$. Thus, the complexity of the greedy RB allocation is $O(PW \log(PW) + P^2W)$.

Here, the performance of Algorithm 12 is analyzed. In Algorithm 12, if the condition "$j \leq P$" in the "while" loop is removed, the algorithm becomes a greedy

vertex coloring algorithm. For the greedy vertex coloring algorithm, denote $\mathcal{X}$ as the number of colors required to color all vertices, and $\psi^i(1 \leq i \leq \mathcal{X})$ as the set containing all vertices colored by color i. Then, $\sum_{i=1}^{\mathcal{X}} |\psi^i| = PW$. It can be easily obtained that the total number of vertices which can be colored by P colors using Algorithm 12 equals to $\sum_{i=1}^{P} |\psi^i|$. If the conflict graph is a clique, each vertex needs to be colored by a distinct color, and P colors can only color P vertices. In this case, Algorithm 12 is the optimal solution. If the conflict graph is not a clique, according to Brooks' theorem [103], $\mathcal{X} \leq \Delta$, where Δ is the maximum degree of vertices in the conflict graph. Then,

- When $\Delta \leq P$, Algorithm 12 can color all vertices by P colors, and it achieves the optimal solution.
- When $\Delta > P$,

$$\sum_{i=1}^{P} |\psi^i| \geq P/\mathcal{X} \cdot \sum_{i=1}^{\mathcal{X}} |\psi^i| \tag{4.6}$$

$$= P/\mathcal{X} \cdot PW \geq P/\Delta \cdot PW \tag{4.7}$$

Condition (4.6) holds since $|\psi^i| > |\psi^j|$ if $i < j$. Thus, the total number of allocated RBS is lower bounded by $P/\Delta \cdot PW \cdot C/P = CW \cdot P/\Delta$, where CW is the upper bound of the number of allocated RBs.

4.3 Wavelength Assignment

The above discusses the OFDMA RB allocation problem for a given conflict graph.

As stated in Corollary 2, the maximum number of allocated RBs at a time equals to $\mathcal{N}(\mathcal{G}(V, E))C/P$, where $\mathcal{N}(\mathcal{G}(V, E))$ is the maximum number of vertices that can be colored by P colors in conflict graph $\mathcal{G}(V, E)$. $\mathcal{N}(\mathcal{G}(V, E))$ depends on the connectivity of the conflict graph, i.e., the edges in graph $\mathcal{G}(V, E)$.

The edges in the conflict graph are contained in set $E = E^{\alpha} \cup E^{\beta}$, where E^{α} and E^{β} refer to interference edges and co-nanocell edges, respectively. The wavelength assignment can be further formulated as deciding E^{β} for given interference edges E^{α} such that $\mathcal{N}(\mathcal{G}(V, E))$ is maximized. However, owing to the NP-hardness property, $\mathcal{N}(\mathcal{G}(V, E))$ cannot be explicitly expressed as a function of the edge set E.

Intuitively, the more the connecting edges in graph $\mathcal{G}(V, E)$, the smaller the $\mathcal{N}(\mathcal{G}(V, E))$. Based on this intuition, minimizing $|E|$ is heuristically treated as the objective in assigning wavelengths. The problem is further transformed into minimizing $|E^{\alpha} \cup E^{\beta}|$ for given $|E^{\alpha}|$.

$$
\begin{aligned}
& |E^{\alpha} \cup E^{\beta}| \\
&= |E^{\alpha} + E^{\beta} - E^{\alpha} \cap E^{\beta}| \\
&= |E^{\beta}| + |E^{\alpha} - E^{\alpha} \cap E^{\beta}| \\
&= WP(P-1)/2 + |E^{\alpha} - E^{\alpha} \cap E^{\beta}|
\end{aligned}
$$

$|E^{\beta}| = WP(P-1)/2$ follows from the fact that the graph with E^{β} only contains W fully connected subgraphs of sizes P. Again, owing to this property of E^{β}, minimizing $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ for given E^{α} is equivalent to the problem of partitioning graph into parts such that parts are of the same sizes with few connections among them, i.e., the graph partitioning problem.

The graph partitioning problem is also NP hard. The brute force search approach involves checking every partitioning choice; the total number of choices can be as large as $\prod_{w=1}^{W} \binom{w \cdot P}{P}$. Many heuristic algorithms have been proposed, among which Kernighan-Lin Algorithm has running time of $O(|V|^2 \log |V|)$ [104].

4.4 Simulation Results and Analysis

For the OFDMA RB allocation, the above presents three algorithms: vertex-coloring based approach, independent-set based approach, and greedy algorithm. With optimal

vertex coloring, the vertex-coloring based approach can produce the maximum number of allocated RBs with running time of $O((1+P)^{PW})$. At the sacrifice of the performance in some degree, the independent-set based approach reduces the running time to $O(P2^{0.267PW})$ by using Robson's maximum independent set algorithm. The greedy algorithm is the most efficient with running time of $O(PW\log(PW) + P^2W)$ with the most compromised performance.

Assume each operation takes around $1ns$. Table 4.2 compares the running time of the three algorithms for some P and W. The overall frame length in 3GPP LTE is around 10 ms, and the typical WiMAX frame length ranges from 2.5ms to 20 ms. Hence, it is usually impractical if the resource allocation algorithm takes longer than 1 ms. Table 4.2(a) shows that the vertex-coloring based approach is impractical even with two wavelengths and five picocells per nanocell. The independent-set based approach can be employed in real systems when the number of wavelengths and the nanocell size are below some thresholds, as indicated red in Table 4.2(b). The greedy algorithm takes less than $10\mu s$ even with 16 wavelengths and 20 picocells per nanocell, as shown in Table 4.2(c).

As shown in Corollary 2, the maximum number of allocated RBs at a time equals to $\mathcal{N}(\mathcal{G}(V,E))C/P$, which determines the system performance. The simulation assumes $C = P$, and investigates the relationship between $\mathcal{N}(\mathcal{G}(V,E))$ and the conflict graph $\mathcal{G}(V,E)$.

Consider a topology with n antennas uniformly distributed in an $800m \times 800m$ square area. Assume the communication range is $r/2$, and then the interference range is r. Figure 4.5 shows one example of 64 picocells and 4 wavelengths. The communication range is 100 m. Figure 4.5 (a) illustrates the geographical distribution of these distributed antennas. Figure 4.5 (b) is the conflict graph $\mathcal{G}^{\alpha}(V,E^{\alpha})$ containing interference edges only. In Figure 4.5 (c), picocells are grouped into four groups as indicated by four different colors. Picocells in the same group constitute a nanocell.

Table 4.2 The Running Time of Three OFDMA RB Allocation Algorithms

(a) Vertex-coloring-based approach

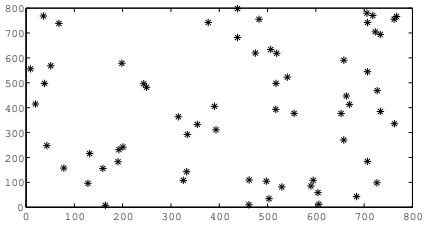
W \ P	5	10
2	0.06 s	6.7×10^{11} s
4	3.65×10^{6} s	

(b) Independent-set-based approach

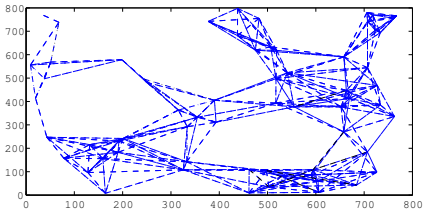
W \ P	5	10	15	20
2	31.82 ns	405 ns	3.87 μs	32.8 μs
4	202 ns	16.4 μs	1 ms	53.8 ms
8	8.2 μs	26.5 ms	66.2 s	1.45×10^{5}s
16	13.4 ms	7.2×10^{4}s	2.9×10^{11}s	1×10^{18}s

(c) Greedy algorithm

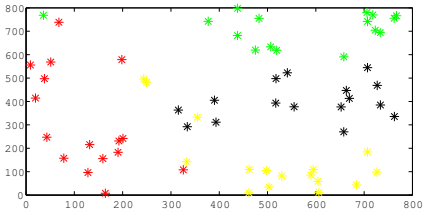
W \ P	5	10	15	20
2	0.0964μs	0.3329μs	0.6844μs	1.1458μs
4	0.1929μs	0.6658μs	1.3688μs	2.2915μs
8	0.3858μs	1.3315μs	2.7377μs	4.5830μs
16	0.7715μs	2.6630μs	5.4753μs	9.1660μs



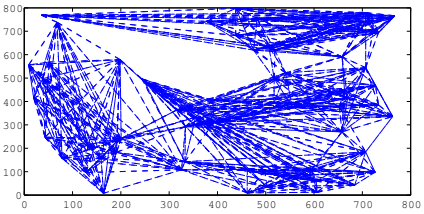
(a) Distribution of antennas



(b) $\mathcal{G}^{\alpha}(V, E^{\alpha})$



(c) Wavelength assignment



(d) $\mathcal{G}(V, E)$

Figure 4.5 $n = 64$, $W = 4$, $r = 200$ m.

In the simulation, Kernighan-Lin Algorithm is used to partition the graph. The final conflict graph is shown in Figure 4.5 (d).

In Figure 4.6, the interference range is varied so as to observe its impact on $\mathcal{N}(\mathcal{G}(V, E))$ and $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ for $n = 64$ and $W = 4$. The displayed results are the average values of 10 simulations. The greedy RB allocation algorithm is performed. When the interference range are small, the number of interference edges and $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ are small. In this case, almost all these 64 vertices can be colored by $P = 16$ colors. With the increase of the interference range, the number of interference edges increases, and the less likely a vertex can be colored. When the number of interference range equals to 800 meters, $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ increases to around 800, and $\mathcal{N}(\mathcal{G}(V, E))$ is reduced to around 20.

In Figure 4.7, the interference range is fixed to be $r = 200$ meters. The number of picocells in the area and the wavelength number are varied to observe the variation of $\mathcal{N}(\mathcal{G}(V, E))$. The displayed value is the average results of 10 simulations. For a given n, small W implies large P, large $|E^{\beta}|$, and small $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$. In the extreme

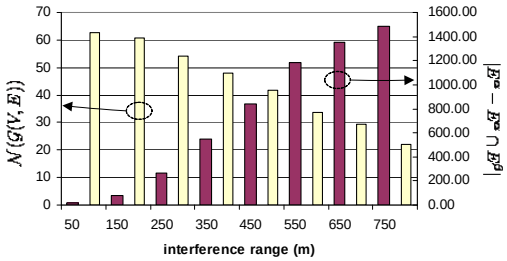


Figure 4.6 $\mathcal{N}(\mathcal{G}(V,E))$ and $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ vs. r.

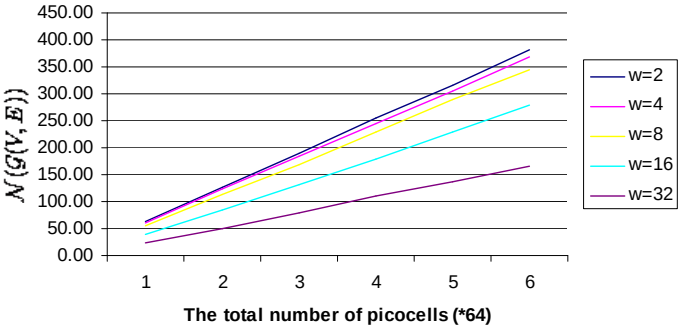


Figure 4.7 $\mathcal{N}(\mathcal{G}(V,E))$ vs. W and n.

case of $W = 1$, $|E^{\alpha} - E^{\alpha} \cap E^{\beta}| = |E^{\beta}|$, and $P = n$. The conflict graph with n vertices is a fully connected graph, and it is P-colorable since $P = n$. Simulations show that, when $W = 2$, $\mathcal{N}(\mathcal{G}(V,E))$ almost equals to n, which agrees with the theoretical analysis. When W is large, P is small, and $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ is large. Then, the number of colored vertices becomes small. Figure 4.7 shows $\mathcal{N}(\mathcal{G}(V,E))$ decreases with the increase of the wavelength number.

When discussing the wavelength assignment problem in Section 4.3, the wavelength assignment problem is transformed into the graph partition problem based on the assumption that the larger the $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$, the smaller the $\mathcal{N}(\mathcal{G}(V,E))C/P$. Here, the assumption is tested by simulations. In Figure 4.8, $P = 16$, $W = 8$, and $r = 200$, run 1000 simulations, and plot $\mathcal{N}(\mathcal{G}(V,E))$ vs. $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ in each

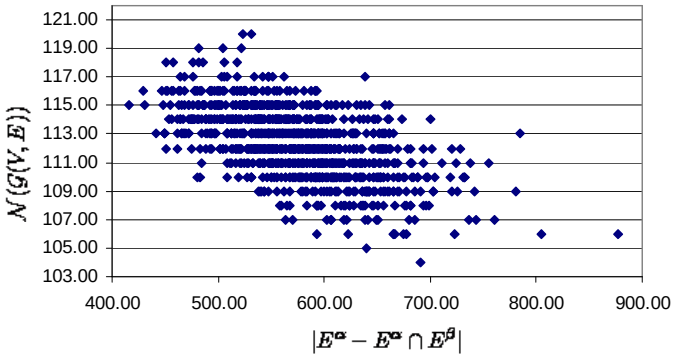


Figure 4.8 $\mathcal{N}(\mathcal{G}(V, E))$ vs. $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$ in 1000 simulations.

simulation. Although $\mathcal{N}(\mathcal{G}(V, E))$ fluctuates for a given $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$, the general trend is that $\mathcal{N}(\mathcal{G}(V, E))$ decreases with the increase of $|E^{\alpha} - E^{\alpha} \cap E^{\beta}|$.

4.5 Summary

This chapter investigates the OFDMA resource allocation and wavelength assignment problems in WDM radio-over-fiber picocellular networks. With the assumption that the data rate delivered by each resource block in each picocell is the same, the problem of maximizing the sum of data rates is reduced to the problem of maximizing the total number of allocated OFDMA resource blocks. It has been shown that the problem of maximizing the total number of allocated RBs is strong NP-hard. Then, three algorithms have been proposed to address it: the vertex-coloring based approach, the independent-set based approach, and the greedy algorithm. Vertex-coloring based algorithm can obtain the optimal result, but is computational intensive. The independent-set based approach reduces the complexity at minor expense of performances. The greedy algorithm, though has the worst performance among the three, is efficient and scalable. The wavelength assignment problem is heuristically formulated into a connectivity minimization problem, and employ graph partitioning algorithms to address it. This assumption is shown to be reasonable by simulations.

Simulation results also show that the performances of the greedy resource allocation algorithm conform closely with the theoretical analysis.

REFERENCES

[1] J. Zhang, N. Ansari, Y. Luo, F. Effenberger, and F. Ye, "Next-generation PONs: a performance investigation of candidate architectures for next-generation access stage 1," *IEEE Communications Magazine*, vol. 47, no. 8, pp. 49–57, August 2009.

[2] G. Kramer, B. Mukherjee, S. Dixit, Y. Ye, and R. Hirth, "Supporting differentiated classes of service in Ethernet passive optical networks," *Journal of Optical Networking*, vol. 1, no. 8, pp. 280–298, 2002.

[3] C. Assi, Y. Ye, S. Dixit, and M. Ali, "Dynamic bandwidth allocation for quality-of-service over Ethernet PONs," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 9, pp. 1467–1477, Nov. 2003.

[4] Y. Luo and N. Ansari, "Bandwidth allocation for multiservice access on EPONs," *IEEE Communications Magazine*, vol. 43, no. 2, pp. S16–S21, Feb. 2005.

[5] P. Brooks and B. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," *IEEE Network*, vol. 24, no. 2, pp. 8 –13, march-april 2010.

[6] A. Takahashi, D. Hands, and V. Barriac, "Standardization activities in the ITU for a QoE assessment of IPTV," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 78–84, 2008.

[7] J. Zhang and N. Ansari, "On optimizing end-to-end QoE in next generation networks: Challenges and possible solutions," *to appear in IEEE Communications Magazine*, 2011.

[8] S. Tasaka and Y. Ishibashi, "Mutually compensatory property of multimedia QoS," *IEEE International Conference on Communications*, vol. 2, pp. 1105–1111 vol.2, 2002.

[9] P. Quax, P. Monsieurs, W. Lamotte, D. De Vleeschauwer, and N. Degrande, "Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game," in *3rd ACM SIGCOMM workshop on Network and system support for games*, 2004, pp. 152–156.

[10] T. Hoßfeld and A. Binzenhöfer, "Analysis of Skype VoIP traffic in UMTS: End-to-end QoS and QoE measurements," *Computer Networks*, vol. 52, no. 3, pp. 650–666, 2008.

[11] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *IEEE Network*, vol. 24, no. 2, pp. 36 –41, march-april 2010.

[12] E. Uysal-Biyikoglu, B. Prabhakar, and A. El Gamal, "Energy-efficient packet transmission over a wireless link," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 4, pp. 487–499, 2002.

[13] ITU-T, "Recommendation G. 1010: End-user multimedia QoS categories," 2001.

[14] P. Reichl, B. Tuffin, and R. Schatz, "Economics of logarithmic quality-of-experience in communication networks," in *The 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE)*, june 2010, pp. 1 –8.

[15] H. Sanneck, G. Carle, and R. Koodli, "Framework model for packet loss metrics based on loss runlengths," in *SPIE/ACM SIGMM Multimedia Computing and Networking*, 2000.

[16] A. Raake, "Short-and long-term packet loss behavior: towards speech quality prediction for arbitrary loss distributions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1957–1968, 2006.

[17] D. Verma, H. Zhang, and D. Ferrari, "Delay jitter control for real-time communication in a packet switching network," *IEEE TRICOMM*, pp. 35–43, Apr 1991.

[18] B. Radunovic and J.-Y. Le Boudec, "A unified framework for max-min and min-max fairness with applications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1073–1083, Oct. 2007.

[19] D. Nace and M. Pioro, "Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 5–17, Quarter 2008.

[20] Y. Luo and N. Ansari, "Limited sharing with traffic prediction for dynamic bandwidth allocation and QoS provisioning over Ethernet passive optical networks," *Journal of Optical Networking*, vol. 4, no. 9, pp. 561–572, 2005.

[21] H. Naser and H. Mouftah, "A joint-ONU interval-based dynamic scheduling algorithm for Ethernet passive optical networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 889–899, Aug. 2006.

[22] C. Kim, T.-W. Yoo, and B.-T. Kim, "A hierarchical weighted round robin EPON DBA scheme and its comparison with cyclic water-filling algorithm," *IEEE International Conference on Communications*, pp. 2156–2161, June 2007.

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

[24] Z. Cao and E. Zegura, "Utility max-min: an application-oriented bandwidth allocation scheme," *IEEE INFOCOM*, vol. 2, pp. 793–801 vol.2, Mar 1999.

[25] W. Wang, M. Palaniswami, and S. Low, "Application-oriented flow control: fundamentals, algorithms and fairness," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 6, p. 1291, 2006.

[26] S. Thakolsri, S. Khan, E. Steinbach, and W. Kellerer, "QoE-driven cross-layer optimization for high speed downlink packet access," *Journal of Communications*, vol. 4, no. 9, pp. 669–680, 2009.

[27] W. Kuo and W. Liao, "Utility-based radio resource allocation for QoS traffic in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2714–2722, 2008.

[28] J. Zhang and N. Ansari, "Utility max-min fair resource allocation for diversified applications in EPON," in *AccessNets*, Hongkong, China, 2009.

[29] J. Zhang and N. Ansari, "An application-oriented resource allocation scheme for EPON," *IEEE Systems Journal*, vol. 4, no. 4, pp. 424–431, 2010.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[31] M. P. McGarry, M. Reisslein, C. J. Colbourn, M. Maier, F. Aurzada, and M. Scheutzow, "Just-in-time scheduling for multichannel EPONs," *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 10, pp. 1204–1216, 2008.

[32] F. Effenberger, D. Clearly, O. Haran, G. Kramer, R. D. Li, M. Oron, and T. Pfeiffer, "An introduction to PON technologies," *IEEE Communications Magazine*, vol. 45, no. 3, pp. S17–S25, March 2007.

[33] L. Hutcheson, "FTTx: Current status and the future," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 90–95, July 2008.

[34] F.-T. An, D. Gutierrez, K. S. Kim, J. W. Lee, and L. Kazovsky, "SUCCESS-HPON: A next-generation optical access architecture for smooth migration from TDM-PON to WDM-PON," *IEEE Communications Magazine*, vol. 43, no. 11, pp. S40–S47, Nov. 2005.

[35] K. Grobe and J.-P. Elbers, "PON in adolescence: from TDMA to WDM-PON," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 26–34, January 2008.

[36] F. Ponzini, F. Cavaliere, G. Berrettini, M. Presi, E. Ciaramella, N. Calabretta, and A. Bogoni, "Evolution scenario toward WDM-PON," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 1, no. 4, pp. C25–C34, September 2009.

[37] C.-H. Lee, S.-M. Lee, K.-M. Choi, J.-H. Moon, S.-G. Mun, K.-T. Jeong, J. H. Kim, and B. Kim, "WDM-PON experiences in korea," *Journal of Optical Networking*, vol. 6, no. 5, pp. 451–464, 2007.

[38] S. Oh, J. Shin, K. Kim, D. Lee, S. Park, H. Sung, Y. Baek, and K. Oh, "200 GHz-spacing 8-channel multi-wavelength lasers for WDM-PON optical line terminal sources," *Optics Express*, vol. 17, no. 11, pp. 9401–9407, 2009.

[39] M. Yoshino, N. Miki, N. Yoshimoto, and K. Kumozaki, "Multiwavelength Optical Source for OCDM Using Sinusoidally Modulated Laser Diode," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, pp. 4524–4529, 2009.

[40] J. Angelopoulos, H.-C. Leligou, T. Argyriou, S. Zontos, E. Ringoot, and T. Van Caenegem, "Efficient transport of packets with QoS in an FSAN-aligned GPON," *IEEE Communications Magazine*, vol. 42, no. 2, pp. 92–98, Feb 2004.

[41] L. Coldren, G. Fish, Y. Akulova, J. Barton, L. Johansson, and C. Coldren, "Tunable semiconductor lasers: a tutorial," *IEEE/OSA Journal of Lightwave Technology*, vol. 22, no. 1, pp. 193–202, Jan. 2004.

[42] J. Buus and E. J. Murphy, "Tunable lasers in optical networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 24, no. 1, p. 5, 2006.

[43] J. Zhang and N. Ansari, "On analyzing the capacity of WDM PONs," in *proc. IEEE GLOBECOM*, Honolulu, Hawaii, USA, 2009.

[44] J. Zhang and N. Ansari, "On the capacity of WDM passive optical networks," *to appear in IEEE Transactions on Communications*.

[45] C. Bock, J. Prat, and S. Walker, "Hybrid WDM/TDM PON using the AWG FSR and featuring centralized light generation and dynamic bandwidth allocation," *IEEE/OSA Journal of Lightwave Technology*, vol. 23, no. 12, pp. 3981–3988, Dec. 2005.

[46] G. Das, B. Lannoo, H.-D. Jung, T. Koonen, D. Colle, M. Pickavet, and P. Demeester, "A new architecture and MAC protocol for fully flexible hybrid WDM/TDM PON," in *35th European Conference on Optical Communication (ECOC)*, Sept. 2009.

[47] J. Zhang and N. Ansari, "Design of WDM PON with tunable lasers: The upstream scenario," *IEEE/OSA Journal of Lightwave Technology*, vol. 28, no. 2, pp. 228–236, 2010.

[48] K. H. Kwong, D. Harle, and I. Andonovic, "Dynamic bandwidth allocation algorithm for differentiated services over WDM EPONs," in *The Ninth International Conference on Communications Systems*, Sept. 2004, pp. 116–120.

[49] M. McGarry, M. Reisslein, and M. Maier, "WDM Ethernet passive optical networks," *IEEE Communications Magazine*, vol. 44, no. 2, pp. 15–22, Feb. 2006.

[50] A. R. Dhaini, C. M. Assi, M. Maier, and A. Shami, "Dynamic wavelength and bandwidth allocation in hybrid TDM/WDM EPON networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 1, pp. 277–286, 2007.

[51] M. P. McGarry, M. Reisslein, M. Maier, and A. Keha, "Bandwidth management for WDM EPONs," *OSA Journal Optical Networking*, vol. 5, no. 9, pp. 637–654, 2006.

[52] L. Meng, J. El-Najjar, H. Alazemi, and C. Assi, "A joint transmission grant scheduling and wavelength assignment in multichannel SG-EPON," *Journal of Lightwave Technology*, vol. 27, no. 21, pp. 4781–4792, 2009.

[53] G. Pieris and G. Sasaki, "Scheduling transmissions in WDM broadcast-and-select networks," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 105–110, 1994.

[54] A. Bianco, M. Guido, E. Leonardi, and D. di Elettronica, "Incremental scheduling algorithms for WDM/TDM networks with arbitrary tuning latencies," *IEEE transactions on communications*, vol. 51, no. 3, pp. 464–475, 2003.

[55] M. Borella and B. Mukherjee, "Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 923–934, 1996.

[56] G. Rouskas and V. Sivaraman, "Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, p. 370, 1997.

[57] Z. Ortiz, G. Rouskas, and H. Perros, "Scheduling of multicast traffic in tunable-receiver WDM networks with non-negligible tuning latencies," *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 4, pp. 301–310, 1997.

[58] G. Rouskas and V. Sivaraman, "On the design of optimal TDM schedules for broadcast WDM networks with arbitrary transceiver tuning latencies," in *IEEE INFOCOM*, vol. 96. Citeseer, pp. 1217–1224.

[59] I. Baldine and G. Rouskas, "Dynamic load balancing in broadcast WDM networks with tuning latencies," in *IEEE INFOCOM*, 1998, pp. 78–85.

[60] E. Hou, N. Ansari, and H. Ren, "A genetic algorithm for multiprocessor scheduling," *IEEE Transactions on Parallel and Distributed Systems*, vol. 5, no. 2, pp. 113–120, 1994.

[61] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, 2002.

[62] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[63] D. Friesen and M. Langston, "Evaluation of a MULTIFIT-based scheduling algorithm," *Journal of Algorithms*, vol. 7, no. 1, pp. 35–59, 1986.

[64] C. Lee, "Parallel machines scheduling with nonsimultaneous machine available time," *Discrete Applied Mathematics*, vol. 30, no. 1, pp. 53–61, 1991.

[65] G. Lin, Y. He, Y. Yao, and H. Lu, "Exact bounds of the modified LPT algorithm applying to parallel machines scheduling with nonsimultaneous machine available times," *Applied Mathematics J. Chinese Univ. B*, vol. 12, no. 1, 1997.

[66] S. Chang and H. Hwang, "The worst-case analysis of the MULTIFIT algorithm for scheduling nonsimultaneous parallel machines," *Discrete Applied Mathematics*, vol. 92, no. 2-3, pp. 135–147, 1999.

[67] M. Sauer, A. Kobyakov, and J. George, "Radio over fiber for picocellular network architectures," *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 11, pp. 3301 –3320, nov. 2007.

[68] C. Lim, A. Nirmalathas, M. Bakaul, P. Gamage, K.-L. Lee, Y. Yang, D. Novak, and R. Waterhouse, "Fiber-wireless networks and subsystem technologies," *IEEE/OSA Journal of Lightwave Technology*, vol. 28, no. 4, pp. 390 –405, feb.15, 2010.

[69] Z. Jia, J. Yu, G. Ellinas, and G. Chang, "Key enabling technologies for optical–wireless networks: Optical millimeter-wave generation, wavelength reuse, and architecture," *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 11, pp. 3452–3471, 2007.

[70] G. Singh and A. Alphones, "OFDM modulation study for a radio-over-fiber system for wireless LAN (IEEE 802.11 a)," in *ICICS-PCM*, vol. 3. IEEE, 2003, pp. 1460–1464.

[71] H. Kim, J. Cho, S. Kim, K. Song, H. Lee, J. Lee, B. Kim, Y. Oh, J. Lee, and S. Hwang, "Radio-over-fiber system for TDD-based OFDMA wireless communication systems," *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 11, pp. 3419–3427, 2007.

[72] J. Song and A. Islam, "Distortion of OFDM signals on radio-over-fiber links integrated with an RF amplifier and active/passive electroabsorption modulators," *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 5, pp. 467–477, 2008.

[73] M. Ergen, *Mobile Broadband: Including WiMAX and LTE*. Springer Verlag, 2009.

[74] X. Zhang, B. Liu, J. Yao, K. Wu, and R. Kashyap, "A novel millimeter-wave-band radio-over-fiber system with dense wavelength-division multiplexing bus architecture," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 2, pp. 929–937, 2006.

[75] J. Vegas Olmos, T. Kuri, and K. Kitayama, "Dynamic reconfigurable WDM 60-GHz millimeter-waveband radio-over-fiber access network: Architectural considerations and experiment," *IEEE/OSA Journal of Lightwave Technology*, vol. 25, no. 11, pp. 3374–3380, 2007.

[76] K. Prince, J. Jensen, A. Caballero, X. Yu, T. Gibbon, D. Zibar, N. Guerrero, A. Osadchiy, and I. Monroy, "Converged wireline and wireless access over a 78-km deployed fiber long-reach WDM PON," *IEEE Photonics Technology Letters*, vol. 21, no. 17, pp. 1274 –1276, sept.1, 2009.

[77] L. Chen, J. Yu, S. Wen, J. Lu, Z. Dong, M. Huang, and G. Chang, "A novel scheme for seamless integration of ROF with centralized lightwave OFDM-WDM-PON system," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 14, pp. 2786 –2791, july15, 2009.

[78] Y. Won, H. Kim, Y. Son, and S. Han, "Full colorless WDM-radio over fiber access network supporting simultaneous transmission of millimeter-wave band and baseband gigabit signals by sideband routing," *IEEE/OSA Journal of Lightwave Technology*, vol. PP, no. 99, pp. 1 –1, 2010.

[79] Z. Cao, J. Yu, H. Zhou, W. Wang, M. Xia, J. Wang, Q. Tang, and L. Chen, "WDM-RoF-PON architecture for flexible wireless and wire-line layout," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 2, pp. 117 –121, february 2010.

[80] J. Zhang and N. Ansari, "Minimizing the arrayed waveguide grating cost and the optical cable cost in deploying WDM passive optical networks," *IEEE/OSA Journal of Optical Communication and Networking*, vol. 1, no. 5, pp. 352–365, 2009.

[81] G. Berardinelli, L. Ruiz de Temino, S. Frattasi, M. Rahman, and P. Mogensen, "OFDMA vs. SC-FDMA: performance comparison in local area imt-a scenarios," *IEEE Wireless Communications*, vol. 15, no. 5, pp. 64 –72, october 2008.

[82] J. Leinonen, J. Hamalainen, and M. Juntti, "Performance analysis of downlink OFDMA resource allocation with limited feedback," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, pp. 2927 –2937, june 2009.

[83] S. Lee, S. Choudhury, A. Khoshnevis, S. Xu, and S. Lu, "Downlink MIMO with frequency-domain packet scheduling for 3GPP LTE," in *IEEE INFOCOM*, 2009.

[84] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless networks*, vol. 11, no. 4, pp. 471–487, 2005.

[85] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *ACM MobiCom*, 2005.

[86] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *IEEE INFOCOM*, vol. 6, 2006.

[87] A. Subramanian, H. Gupta, S. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *IEEE transactions on mobile computing*, pp. 1459–1473, 2008.

[88] G. Li and H. Liu, "Downlink radio resource allocation for multi-cell OFDMA system," *IEEE transactions on wireless communications*, vol. 5, no. 12, pp. 3451–3459, 2006.

[89] M. Pischella and J. Belfiore, "Power control in distributed cooperative OFDMA cellular networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1900–1906, 2008.

[90] H. Zhu and J. Wang, "Chunk-based resource allocation in OFDMA systems-part I: chunk allocation," *Communications, IEEE Transactions on*, vol. 57, no. 9, pp. 2734–2744, 2009.

[91] L. K. Reuven Cohen, "Computational analysis and efficient algorithms for micro and macro OFDMA scheduling," in *IEEE Infocom*, 2008.

[92] J. Wang and L. Milstein, "CDMA overlay situations for microcellular mobile communications," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 603–614, 1995.

[93] A. Sang, X. Wang, M. Madihian, and R. Gitlin, "Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems," *Wireless Networks*, vol. 14, no. 1, pp. 103–120, 2008.

[94] S. Gault, W. Hachem, and P. Ciblat, "Performance analysis of an OFDMA transmission system in a multicell environment," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 740–751, 2007.

[95] J. Zhang and N. Ansari, "Scheduling hybrid WDM/TDM passive optical networks with non-zero laser tuning time," *to appear in IEEE/ACM Transactions on Networking*.

[96] S. Sarkar, H. Yen, S. Dixit, and B. Mukherjee, "A mixed integer programming model for optimum placement of base stations and optical network units in a hybrid wireless-optical broadband access network (WOBAN)," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2007, pp. 3907–3911.

[97] S. Sarkar, H. Yen, S. Dixit, and B. Mukherjee, "Hybrid wireless-optical broadband access network (WOBAN): network planning using Lagrangean relaxation," *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1094–1105, 2009.

[98] T. Koonen, K. Steenbergen, F. Janssen, and J. Wellen, "Flexibly reconfigurable fiber-wireless network using wavelength routing techniques: The ACTS project AC349 PRISMA," *Photonic Network Communications*, vol. 3, no. 3, pp. 297–306, 2001.

[99] P. Djukic and S. Valaee, "Delay aware link scheduling for multi-hop TDMA wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 3, pp. 870–883, 2009.

[100] R. Tarjan and A. Trojanowski, "Finding a maximum independent set," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 537–546, 1977.

[101] F. Fomin, F. Grandoni, and D. Kratsch, "A measure & conquer approach for the analysis of exact algorithms," *Journal of the ACM (JACM)*, vol. 56, no. 5, p. 25, 2009.

[102] J. Robson, "Algorithms for maximum independent sets," *Journal of Algorithms*, vol. 7, no. 3, pp. 425–440, 1986.

[103] D. West *et al.*, *Introduction to graph theory*. Prentice Hall Upper Saddle River, NJ, 2001.

[104] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.