Design and analysis of medium access control protocols in WDM optical networks

Huang, Xiaohong

2005

Huang, X. H. (2005). Design and analysis of medium access control protocols in WDM optical networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4400>

<https://doi.org/10.32657/10356/4400>

Nanyang Technological University

Downloaded on 26 Apr 2023 09:22:03 SGT

Design and Analysis of Medium Access Control Protocols in WDM Optical Networks

Huang Xiaohong

School of Electrical & Electronic Engineering

**A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy**

2005

Statement of Originality

I hereby certify that the content of this thesis is the result of work done by me and has not been submitted for a higher degree to any other University or Institution.

20/05/2005

Date

Huang Xiaohong

Acknowledgement

First and foremost, I wish to thank my supervisor, A/P Ma Maode. He has led me into WDM research area and has taught me almost hand-by-hand. Without his support, guidance, and constant encouragement, this dissertation would not have been possible. I truly appreciate all that he has done.

I also gratefully acknowledge many professors and researchers whom I have the honor to know. Among them, special thanks go to A/P Zhong Wende, A/P Qin Yang and A/P Lu Chao, for their encouragement and valuable help in my research. Moreover, I am highly thankful for the exciting discussions and countless refreshing social activities with all the members of the Communication Lab IV making my daily academic work and private life an immeasurable pleasure.

It is also a great pleasure to thank my fellow graduate students and friends. I have enjoyed many technical discussions with Leng Supeng, Wang Haibo, Zhang Yan, Zhu Yongqing, and Lin Guochun with whom I work most closely. I also wish to thank Hou Fang, Ouyang Yue, Hou Zhihua, Zhang Keli, Song Zhefeng, Wang Wen, Luo Xingfang, Xiao Shi, and Zhong Xuxia for their technical help and warmest friendships during my graduate career.

Last, but not least, my boundless gratitude is devoted to my family for their love and support during my graduate school career. The path to complete this dissertation was far from clear during some difficult times, and I would like to especially thank my mother, father and brothers for their understanding and many kind words of encouragement. Without their encouragement, this work could not have been completed.

Summary

The already enormous and still increasing bandwidth demand due to the incomparable popularity of Internet requires new high performance networking technologies. At the same time, emerging multimedia applications, such as video conferencing, interactive television, and video-on-demand are expected to further test the limits of current network infrastructures. As such, Wavelength Division Multiplexing (WDM) optical networks are playing a pivotal role. This dissertation investigates the problem of scheduling and managing variable-length message transmission on single-hop passive-star coupled WDM optical Local Area Networks (LANs)/Metropolitan Area Networks (MANs) subject to the limited number of WDM channels and a large number of user population.

First of all, we have proposed and investigated three channel assignment algorithms and two message sequencing algorithms for the specified WDM optical networks. The Latency Minimizing Scheduling (LMS) algorithm, the Continuous Channel Scheduling (CCS) algorithm, and the Continuous Channel-Minimum Scheduling Latency (CC-MSL) algorithm are channel assignment algorithms that address the problem of selecting an appropriate channel and a time slot on that channel to transmit a message. They try to reduce the scheduling latency and the negative impact of the tuning overheads of transceivers. The Minimum Flow Time Scheduling (MFTS) algorithm and the Scheduling Minimizing Idle Time (SMIT) algorithm are message sequencing algorithms that address the order in which messages are sent. The purpose of them is to improve the network performance in terms of average message delay by taking into

consideration the physical characteristic of WDM optical network and node equipment adequately. These scheduling algorithms have been proved to be efficient scheduling algorithms in the specified WDM optical networks.

Second, we have put forward several scheduling mechanisms to support Quality of Service (QoS) of a variety of applications. The first algorithm is called Differentiated Dropping Scheduling (DDS), which is to handle real-time applications. This algorithm takes advantage of the special properties of the specified WDM network environment to reduce the number of messages violating the time constraints. The second algorithm we put forward is Cost-Based Priority Scheduling (CBPS), which is to provide differentiated service to the messages with and without time constraints. This algorithm is shown to be able to reduce the messages loss rate as well as the average message delay when an integrated traffic is applied to the network so that the transmission of both types of messages could be benefited. Last, based on CBPS algorithm, a novel framework for QoS prediction is set up, which aims to predict whether the QoS requirements of different traffic can be satisfied simultaneously. Moreover, this structure can be regarded as a part of admission control. That is, when a new traffic with new QoS requirement requests for the connection, the QoS prediction structure can be performed at real-time to estimate whether the new QoS can be adapted to the current QoS. If not, an estimation of the lower level by which the new traffic can be accepted will be given.

Finally, two Mixed Integer Linear Programming (MILP) based off-line scheduling algorithms have been proposed to obtain the optimal scheduler in terms of average delay or loss rate respectively. This scheme is able to include the system constraints of single-hop passive-star coupled WDM optical networks, i.e., channel availability, receiver availability and tuning overhead, in mathematical formulations. With the mathematical constraints and objective functions, the optimal solutions of MILP models can be obtained.

Table of Contents

ACKNOWLEDGEMENT	I
SUMMARY	II
TABLE OF CONTENTS	IV
LIST OF ABBREVIATIONS	VII
LIST OF FIGURES	X
LIST OF TABLES	XIV
CHAPTER 1 . INTRODUCTION.....	1
1.1 Motivation.....	1
1.2 Objectives	6
1.3 Major Contribution of the Thesis.....	8
1.4 Organization of the Thesis	11
CHAPTER 2 . OVERVIEW OF MAC PROTOCOLS ON SINGLE-HOP PASSIVE-STAR COUPLED WDM OPTICAL NETWORKS.....	13
2.1 Passive Star-Coupled WDM Optical Networks.....	13
2.2 MAC Protocols for Packet Transmission	18
2.2.1 Non Pre-Transmission Coordination Protocols	18
2.2.2 Pre-Transmission Coordination Protocols	19
2.3 MAC Protocols for Variable-Length Message Transmission.....	22
2.3.1 Basic MAC Protocols for Variable-Length Messages.....	22
2.3.2 Novel MAC Protocols for Variable-Length Messages.....	24
2.4 MAC Protocols with QoS Concerns	27
2.4.1 MAC Protocols for Real-Time Service	28

2.4.2 MAC Protocols for Multimedia Application	34
2.5 Summary	37
CHAPTER 3 . SYSTEM MODEL AND IMPRESSIVE PROTOCOLS.....	38
3.1 System Model	38
3.1.1 Network Architecture	39
3.1.2 Channel Structure and System Assumption.....	40
3.1.3 Transmission and Reception Procedure.....	41
3.2 Existing Protocols	42
3.2.1 Channel Assignment Algorithms.....	42
3.2.2 Message Sequencing Algorithms	45
3.3 Summary	49
CHAPTER 4 . PROPOSED SOLUTIONS TO IMPROVE NETWORK PERFORMANCE	52
4.1 Channel Assignment Algorithms.....	53
4.1.1 Latency Minimizing Scheduling (LMS).....	53
4.1.2 Continuous Channel Scheduling (CCS)	59
4.1.3 Continuous Channel-Minimum Scheduling Latency (CC-MSL)	65
4.1.4 Summary.....	71
4.2 Message Sequencing Algorithms.....	71
4.2.1 Minimum Flow Time Scheduling (MFTS).....	72
4.2.2 Scheduling Minimizing Idle Time (SMIT).....	88
4.2.3 Summary.....	100
4.3 Comparison.....	101
4.3.1 Channel Assignment Algorithms.....	101
4.3.2 Message Sequencing Algorithms	103
4.3.3 Combination of the Algorithms	106
4.4 Summary	108
CHAPTER 5 . PROPOSED SOLUTIONS TO SUPPORT QOS SERVICE	110
5.1 Differentiated Dropping Scheduling (DDS)	111
5.1.1 Scheduling Algorithm.....	113
5.1.2 Performance Analysis.....	124
5.1.3 Summary.....	140
5.2 Cost-Based Priority Scheduling (CBPS) Algorithm.....	140
5.2.1 Scheduling Algorithm.....	141
5.2.2 Performance Analysis.....	149
5.2.3 Summary.....	166
5.3 CBPS Based QoS Service Prediction	166
5.3.1 Introduction	166
5.3.2 Relative Class Mapping	169
5.3.3 Performance of Relative Classes Traffic	177
5.3.4 Performance of Network Traffic.....	198

5.3.5 Scheduling Parameters Searching.....	202
5.3.6 QoS Prediction Example.....	204
5.3.7 Summary.....	207
5.4 Summary.....	208
CHAPTER 6 . LINEAR PROGRAMMING BASED OFF-LINE SCHEDULING POLICY	210
6.1 Optimization by Linear Programming.....	210
6.2 Problem Formulation (Delay).....	212
6.2.1 Variable Definitions.....	213
6.2.2 System Constraints	214
6.2.3 Objective Function.....	220
6.3 Problem Formulation (Loss Rate).....	222
6.3.1 Variable Definitions.....	223
6.3.2 System Constraints	223
6.3.3 Objective Function.....	227
6.4 Examples.....	230
6.4.1 Delay.....	231
6.4.2 Loss Rate	234
6.5 Experimental Emulation and Results.....	237
6.5.1 Experimental Design	237
6.5.2 Experimental Results.....	241
6.6 Summary.....	244
CHAPTER 7 . CONCLUSION AND RECOMMENDATIONS	246
7.1 Conclusion	246
7.2 Recommendations for Future Research	251
7.2.1 Further Improvement on Current Results	251
7.2.2 Potential Research Directions.....	252
Author's Publications.....	257
Bibliography	259

List of Abbreviations

AF	Allocation Free
ARR-EATS	Adaptive Round-Robin and Earliest Available Time Scheduling
BSS	Binary Splitting Scheme
CAT	Channel Available Time
CBPS	Cost-Based Priority Scheduling
CBR	Constant Bit Rate
CC	Control Channel
CC-MSL	Continuous Channel-Minimum Scheduling Latency
CCS	Continuous Channel Scheduling
c.d.f	Cumulative Distribution Function
CRMA	Cyclic-Reservation Multiple-Access
CTMC	Continuous Time Homogeneous Markov Chain
DA	Destination Allocation
DAS	Dynamic Allocation Scheme
DDS	Differentiated Dropping
DQDB	Distributed Queue Dual Bus
DQMA	Distributed-Queue Multiple-Access
DT-WDMA	Dynamic Time-Wavelength Division Multiaccess
EATS	Earliest Available Time Scheduling
EPON	Ethernet Passive Optical Network
EDD	Earliest Due Date

FDQ	Fair Distributed Queue
FT-FR	Fixed Transmitter/Fixed Receiver
FT-TR	Fixed Transmitter/Tunable Receiver
HONET	Hybrid Optical Network
HRP	Hybrid Reservation Pre-Allocation
ILP	Integer Linear Programming
LAN	Local Area Network
LMS	Latency Minimizing Scheduling
LP	Linear Programming
LST	Laplace-Stieltjes Transform
M-WDMA	Multimedia Wavelength-Division Multiple-Access
MAC	Media Access Control
MAN	Metropolitan Area Networks
MCMS	Multi-Class Multi-Server
MDP	Markovian Decision Process
MFTS	Minimum Flow Time Scheduling
MILP	Mixed Integer Linear Programming
MLF	Minimum Laxity First
MLF-TTS	Minimum Laxity First with Time Tolerance Scheduling
Algorithm	
MSL	Minimum Scheduling Latency
OCRL	Optical Communications Research Laboratory
OLT	Optical Line Terminal
p.d.f	Probability Density Function
PDS	Priority-Differentiated Scheduling
PSC	Passive Star Coupler
QoS	Quality of Service

RAT	Receiver Available Time
RO-EATS	Receiver-Oriented Earliest Available Time Scheduling
SA	Source Allocation
SC	Source Channel
SJF	Shortest Job First
SMF	Shortest Message First
SMIT	Scheduling Minimizing Idle Time
SPA	Sampling Probe Algorithm
TD-TWDM	Time-Deterministic Time and Wavelength Division Multiple Access
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TSA	Time Slot Assignment
TT-FR	Tunable Transmitter/Fixed Receiver
VBR	Variable Bit Rate
VOD	Video-on-Demand
VOQ	Virtual Output Queueing
WDM	Wavelength Division Multiplexing
WDMA	Wavelength Division Multiple Access

List of Figures

Figure 2.1 A broadcast-and-select WDM network	14
Figure 3.1 Network architecture of a single-hop passive-star coupled WDM optical network (CC-FT/TT-FR/TR system)	39
Figure 3.2 Data and control channel configuration	40
Figure 3.3 Structure of the control channel	41
Figure 4.1 A simple example of relationships among the scheduling parameters.....	55
Figure 4.2 Impact of traffic load on the average delay of LMS	58
Figure 4.3 Impact of tuning time on the average delay of LMS.....	58
Figure 4.4 Impact of number of channels on average delay of LMS	59
Figure 4.5 Impact of traffic load on average delay of CCS	63
Figure 4.6 Impact of tuning time on average delay of CCS	64
Figure 4.7 Impact of number of channels on average delay of CCS	65
Figure 4.8 Impact of traffic load on average delay of CC-MSL.....	69
Figure 4.9 Impact of tuning time on average delay of CC-MSL	69
Figure 4.10 Impact of number of channels on average delay of CC-MSL.....	70
Figure 4.11 Sorting of adjacent messages.....	80
Figure 4.12 Impact of traffic load on average delay of MFTS	86
Figure 4.13 Impact of tuning time on average delay of MFTS	87
Figure 4.14 Impact of number of channels on average delay of MFTS	88
Figure 4.15 Timing relationships without the effect of tuning time and propagation delay	90

Figure 4.16. Timing relationships with the effect of tuning time and propagation delay.....	91
Figure 4.17 Information of messages in the example of SMIT	95
Figure 4.18 Results of message scheduling using SMF	96
Figure 4.19 Results of message scheduling using SMIT	97
Figure 4.20 Impact of traffic load on average delay of SMIT	99
Figure 4.21 Impact of tuning time on average delay of SMIT	99
Figure 4.22 Impact of number of channels on average delay of SMIT	100
Figure 4.23 Performance comparison among LMS, CCS and CC-MSL.....	103
Figure 4.24 Performance comparison between MFTS and SMIT	104
Figure 4.25 Performance comparison among combinational algorithms	106
Figure 5.1 Impact of traffic load on message loss rate of DDS	136
Figure 5.2 Impact of tuning time on message loss rate of DDS	136
Figure 5.3 Impact of number of channels on message loss rate of DDS	137
Figure 5.4 Impact of laxity on message loss rate of DDS	138
Figure 5.5 Impact of traffic load on throughput of DDS	139
Figure 5.6 Impact of traffic load on packet loss rate of DDS	140
Figure 5.7 Schedule using Moore and Hodgson's algorithm	147
Figure 5.8 Schedule using CBPS	148
Figure 5.9 Schedule using SMF	148
Figure 5.10 Interchange of adjacent messages	150
Figure 5.11 Impact of traffic load on mean delay of real-time messages of CBPS.....	155
Figure 5.12 Impact of traffic load on mean delay of non real-time messages of CBPS.....	156
Figure 5.13 Impact of traffic load on mean delay of CBPS.....	156
Figure 5.14 Impact of traffic load on loss rate of real-time messages of CBPS.....	157
Figure 5.15 Impact of traffic load on throughput of CBPS	157
Figure 5.16 Impact of number of channels on mean delay of CBPS.....	158
Figure 5.17 Impact of number of channels on loss rate of CBPS.....	159

Figure 5.18 Impact of tuning time on mean delay of CBPS	160
Figure 5.19 Impact of tuning time on loss rate of CBPS	160
Figure 5.20 Impact of α on mean delay	161
Figure 5.21 Impact of α on loss rate	162
Figure 5.22 Impact of β on mean delay	163
Figure 5.23 Impact of β on loss rate	164
Figure 5.24 Impact of γ on mean delay	165
Figure 5.25 Impact of γ on loss rate	165
Figure 5.26 Schematic description of QoS prediction process	168
Figure 5.27 Overview of QoS prediction structure	169
Figure 5.28 Distribution of the relative priority section	170
Figure 5.29 The relationship between static class and relative class	170
Figure 5.30 Structure of QoS service prediction	171
Figure 5.31 Transition diagram when $0 \leq r \leq n$	181
Figure 5.32 Transition diagram when $n < r < N$	181
Figure 5.33 Transition diagram when $r = N$	182
Figure 5.34 Transition diagram when $0 \leq r \leq n$	190
Figure 5.35 Transition diagram when $n < r < N$	190
Figure 5.36 Transition diagram when $r = N$	190
Figure 5.37 Average delay of real-time relative classes	197
Figure 5.38 Average delay of non-real-time relative classes	197
Figure 5.39 Loss rate of real-time relative classes	197
Figure 5.40 Average delay of real-time static classes	200
Figure 5.41 Loss rate of real-time static classes	201
Figure 5.42 Average delay of non real-time static classes	202
Figure 5.43 Flow chart for the searching process of the parameters	204
Figure 6.1 Message sequence of LPI	232

Figure 6.2 Message sequence of SMIT-CC-MSL algorithm.....	233
Figure 6.3 Message sequence of LPII.....	235
Figure 6.4 Message sequence of DDS-CC-MSL algorithm	236
Figure 6.5 MILP emulator	239
Figure 6.6 Virtual on-line emulation system	240
Figure 6.7 Experimental system for SMIT-CC-MSL and DDS-CC-MSL	240
Figure 6.8 Impact of traffic load on average delay of LPI.....	241
Figure 6.9 Impact of tuning time on average delay of LPI.....	242
Figure 6.10 Impact of traffic load on loss rate of LPII	243
Figure 6.11 Impact of tuning time on loss rate of LPII	244
Figure 7.1. Generic access network architecture	247

List of Tables

Table 2.1. Comparison of some protocols	27
Table 4.1 Current global information in the example of MFTS	76
Table 4.2 Information of messages in the example of MFTS.....	76
Table 4.3 Results of message scheduling using SMF-EATS	76
Table 4.4 Results of message scheduling using MFTS-EATS	77
Table 4.5 Results of message scheduling using MFTS-LMS.....	79
Table 4.6 Current global information in the example of SMIT	95
Table 5.1 Current global information in the example of DDS	120
Table 5.2 Information of messages in the example of DDS	120
Table 5.3 Results of message scheduling using EDD-EATS	120
Table 5.4 Results of message scheduling using Moore and Hodgson-EATS.....	121
Table 5.5 Results of message scheduling using DDS-EATS	122
Table 5.6 Current global information in the example of CBPS.....	145
Table 5.7 Information of the messages in the example of CBPS	146
Table 5.8 Region of relative priority.....	199
Table 5.9 Distribution of each static class	200
Table 5.10 Current QoS requirements	204
Table 5.11 New QoS requirements I.....	204
Table 5.12 Updated QoS requirements I.....	204
Table 5.13 New QoS requirements II	206
Table 5.14 Updated QoS requirements II	206

Table 5.15 QoS Prediction with current QoS	206
Table 5.16 QoS estimation I	206
Table 5.17 QoS estimation II	207
Table 5.18 Largest difference between the experimental and analytical values.....	207

Chapter 1 . Introduction

1.1 Motivation

With the enormous and still increasing bandwidth demand due to the incomparable popularity of the Internet, current local and wide area networks can barely cope with the huge demand for network bandwidth. As a result, there is a worldwide effort in upgrading current networks with high-bandwidth fiber-optic links that can potentially deliver terabits/s. One of the main problems in achieving this goal is the speed mismatch between slower electronic processing and faster optical transmission. Although a single-mode fiber's potential bandwidth is nearly 50 Tbps, the maximum rate at which an end-user, which can be a workstation or a gateway that interfaces with low-speed sub-networks, can access the network is limited by electronic processing speed, which is of the order of a few Gbps. This problem is known as the electronic bottleneck problem [1]. Wavelength Division Multiplexing (WDM) is an effective technology to avoid this mismatch by providing multiple channels on a single fiber, each operating at a maximum data rate set by electronic technology [2].

In WDM, the huge optical fiber bandwidth is split into a number of lower capacity wavelength bands, with each wavelength supporting a single communication channel operating at, e.g., peak electronic speed of a few tens of Gbit/s [3]. The resulting wavelength channels are then multiplexed onto the same fiber. Therefore, WDM exploits the vast

bandwidth by requiring that each end-user's equipment operates only at electronic transmission rates, but multiple traffic from different end-users may be multiplexed on the same fiber. Currently, typical specified channel bit rates for WDM networks according to the ITU are 2.5 Gbit/s (STM-16) and 10 Gbit/s (STM-64), but it is expected that channel transmission rates of up to 40 Gbit/s become applicable in the near future.

As the WDM technology continues its success story, its use in the field of Local Area Networks (LANs)/Metropolitan Area Networks (MANs) becomes more and more justified and affordable. Moreover, the emergence of new highly bandwidth-intensive applications makes effective bandwidth allocation in LANs/MANs an increasingly important issue. In general, WDM LANs/MANs can be built by using tunable optical transmitters or receivers. By tuning the transmitters/receivers to one or more wavelength channels, a node can transmit/receive on those channels. The tunability, including the tuning range and tuning time, of different transmitters and receivers that each node is equipped with can be potentially different. A passive star coupler (PSC), equipped with tunable transmitters and tunable receivers, can be used to construct a multi-access LAN/MAN using WDM channels.

One of the most relevant issues in building WDM LANs/MANs is the design of new Medium Access Control (MAC) protocols. MAC protocols arbitrate the access of the network nodes to the available bandwidth of the considered medium, i.e., the optical fiber, according to criteria such as network throughput, packet delays and access fairness. In WDM LANs/MANs, it is important to understand how system parameters such as the number of transmitters and receivers, number of wavelengths, transmitter and receiver tuning times, and propagation delays affect the network performance. It is also important to design MAC protocols which appropriately account for system limitations and utilize the network resources in an efficient manner.

Generally, the essential aspect of an efficient MAC protocol is to efficiently coordinate the transmissions between various nodes. For example, if each node is equipped with a tunable transmitter, *channel collisions* occur if more than one node transmits on the same

wavelength at the same time. Instead, when each node is equipped with tunable receivers, *destination collisions* occur when more than one messages arrives at the same destination at that same time. Therefore, a proper MAC protocol has to prevent such collisions. Additionally, tuning time of the transceivers is not negligible with respect to the transmission times and the tuning range is still limited. To reduce the negative impact of tuning times is also an important issue in the design of MAC protocols.

To date, many MAC protocols have been proposed for single-hop passive-star coupled WDM optical networks. The main goal of these protocols is to avoid channel collisions as well as receiver collisions by sharing some global information about the status of the channels and receivers. However, most of them have not taken the physical properties of the medium and the node equipments adequately into account. In WDM networks, there are three constraints on scheduling a message, which are receiver availability, channel availability and transceivers' tuning overhead. In order for one message to transmit, the channel, the receiver and the transmitter should be ready simultaneously. Moreover, the effect of the tuning time of the transceivers is unlikely to go away nowadays since the transceivers' capabilities are subject to both tuning range and tuning speed. Although many access schemes have been proposed, they either ignore the tuning overhead or rely on rapidly tunable transceivers to achieve efficiency. A careful access scheme design should mask the effect of tuning overhead to reduce the performance penalty as much as possible. Therefore, the design of new protocols, which are able to improve the network performance in terms of average delay by further elaborating the characteristic of the network, is still a valuable research direction in WDM optical networks.

Furthermore, the demand for QoS on the Internet is increasing due to its use by real-time application services such as video and voice streams. This coupled with the fact that Internet traffic will eventually be aggregated and carried over WDM optical networks makes it important to address the issue of QoS in WDM optical networks. Thus, one of the key issues in designing next generation photonic network is the additional support of real-time services

corresponding to time-sensitive applications. As a matter of fact, novel MAC protocols have to be developed which support real-time traffic with tight delay constraints in a flexible and efficient way. Though many MAC protocols have been proposed in the recent past, no such a scheme has considered the special characteristic of WDM networks adequately. Thus, to coordinate the transmission and reception of the messages efficiently while minimizing the fraction of messages violating their constraints, new protocols should take the physical characteristic of WDM networks and node equipments into account.

In addition, the current “best effort” model used for research in optical networks does not differentiate between different traffic types and cannot provide support for QoS. Thus, most of the MAC protocols proposed so far are not suitable for an integrated service environment since they have been designed with one generic traffic type in mind. They perform well for the traffic streams they have been designed for, but poorly for other traffic streams with different characteristics. As a result, some aspects of QoS for different types of traffic are still an open issue and subject to intensive research activities in single-hop passive-star coupled WDM optical networks. WDM technology provides wavelength partitioning as a simple way of sharing the bandwidth between different types of traffic. The problem of designing a QoS framework based on wavelength partitioning assumes significant importance in order to provide differentiated service in next generation optical networks.

As concerns designing a MAC protocol with QoS guarantee for optical networks, we also have to consider the issue of QoS prediction in optical networks. As we know, QoS requirements are different for different types of services. Even for the same types of service, different users may have different QoS requirements. Therefore, we should define several QoS levels based on the “satisfactory levels” and study the percentage of services that fit into each of these QoS levels. This is so-called QoS prediction. Before the QoS prediction structure is implemented, it is very important to define appropriate QoS measures that can capture the essence of the impairments. These QoS measures will be used as the criteria of the QoS prediction. Based on such criteria, the network can estimate how many applications

can be satisfied and to which level the QoS can be satisfied. Good predictions of the system can be run so that QoS degradation due to resource starvation will rarely happen. Furthermore, in the future generation network, one of the main challenges is providing flexible means for controlling network behavior as the surrounding environment changes. QoS service prediction is also an important topic involved in this issue, which is used to determine whether the connections can be accepted or not under certain individual user's specific QoS requirements. Therefore, the study of the new protocols, which are able to perform at real-time so as to provide QoS prediction when the network environment changes, has also become a valuable research direction in the WDM LANs/MANs.

Although WDM technique has the potential to significantly improve the performance of optical networks, in reality, it should be noted that the network size must be scalable, the number of devices of each computer used to connect to the WDM network may be limited, and the number of WDM channels that can be supported in a single fiber may also be insufficient compared with the number of connected computers. With such problems in practice, how to efficiently manage multiple information transmissions on the parallel data channels of a WDM network with given network resources constraints and some other constraints on the transmitted information becomes an important problem. This issue plays a key role in effectively utilizing the network resources and improving the network performance. So far, many scheduling algorithms that MAC protocols employ have been proposed. However, how to obtain an optimal scheduler to minimize the average delay or minimize the number of messages violating the time constraints remains an open issue. This is partly due to the difficulties of including system constraints in mathematical formulations that maintain sufficient structure to aid in the optimization. So, this is also a challenging direction in the design of scheduling algorithm for optical networks, of which we must take into account.

1.2 Objectives

The objective of this research is to design and develop various kinds of scheduling algorithms for MAC protocols on single-hop passive-star coupled WDM optical networks in order to efficiently exploit the latent huge bandwidth the optical fiber can supply.

The first objective of this work is to provide the service for non real-time applications. In order to evaluate the performance of non real-time scheduler, average delay is the most important metric. The purpose of our research is to develop efficient scheduling algorithms to manage variable-length message transmissions under the constraints of limited network resources. The new algorithms should take the physical characteristic of optical networks and node equipments into account so that the negative impact of the system constraints on the performance can be reduced or eliminated. Furthermore, these scheduling algorithms should be dynamic and distributed with the property that not only the network assignment aspect but also the message transmission sequencing aspect are perfectly considered.

The second objective of this work is to support QoS requirements for local and metro networks, which consists of three aspects as follows.

1) The first aspect is to provide the service for real-time applications. The most important aspect of the real-time applications is that a message must be received at the destination station within a given amount of time after its generation. This time is referred to as message *deadline*. If the delay of a message in the system exceeds its time constraint, the message is considered as lost. In a real-time communication system, the principle task for the scheduling policies is to maximize the number of messages that are delivered by their respective deadlines. So far many existing algorithms are available for scheduling real-time messages in simple systems. However, due to the special physical characteristic of WDM optical networks, more efficient protocols are still needed to provide the service for real-time applications on an optical switching system. The new algorithm should take the channel

constraints, destination constraints and tuning overhead of the transceivers adequately into account.

2) The second aspect is to provide differentiated service to the applications with different QoS requirements. Although many access protocols for single-hop passive-star coupled WDM optical networks have been proposed in the recent past, the integration of real-time services with certain QoS requirements and best-effort services remains open. As a matter of fact, it appears to be essential to develop novel MAC protocols which support the traffic with and without tight delay constraints in a flexible and efficient way directly in the optical transmission layer. In this research, we study the problem of scheduling and managing transmission when the specified network is applied with an integrated traffic so that the transmission of messages with and without time constraints could be benefited.

3) The third aspect is to develop a QoS prediction structure for various applications with various QoS requirements. The purpose of this QoS prediction structure is to estimate, given the QoS requirements of different applications, how many of them can be satisfied according to the predefined QoS level and how many of them have to be rejected or accepted under lower level. This structure should also be able to control the network behavior when the new application with certain QoS requirements requesting for the connection. The estimation that whether the new connection can be accepted or not should be given based on the current QoS requirements and the new QoS requirements.

The third objective of this work is to try to set up a local optimization framework, which is able to include system constraints, i.e., channel availability, receiver availability and tuning overhead, in mathematical formulations so that it is sufficient to obtain the optimal performance in terms of message delay and message loss rate. The new scheme should be able to prevent channel collisions as well as receiver collisions. Moreover, the negative effect of the tuning overhead should be incorporated into the new algorithm. The purpose of the new algorithm is to find a wavelength assignment that guarantees the delivery of the given traffic request, while minimizing the average delay or the loss rate.

1.3 Major Contribution of the Thesis

The main contributions in this thesis are as follows:

1. We take a general scheme, which addresses both channel assignment issue and message transmission sequencing issue, as a guideline for developing scheduling algorithms to schedule variable-length message transmission in the specified WDM optical networks. Based on this technique, we have put forward several channel assignment algorithms, which address the problem of selecting an appropriate channel and a time slot on that channel to transmit a message, and two message sequencing algorithms, which address the order in which messages are sent, to schedule messages without time constraints. The channel assignment algorithms are summarized as follows:

1) The Latency Minimizing Scheduling (LMS) algorithm selects the channels by considering the system constraints, including channel availability, receiver availability and tuning overhead, adequately in order to reduce the scheduling latency.

2) The Continuous Channel Scheduling (CCS) algorithm aims to reduce the impact of transceiver tuning overhead by avoiding unnecessary tuning of the transmitters. By using this algorithm, the message will be assigned to the channel that the source node has been tuned to whenever possible.

3) The Continuous Channel-Minimum Scheduling Latency (CC-MSL) algorithm combines the advantages of CCS and MSL algorithms [54]. This algorithm aims to reduce scheduling latency as well as the tuning overhead.

Similarly, the message sequencing algorithms are summarized as follows:

1) The Minimum Flow Time Scheduling (MFTS) algorithm determines the order of the message transmissions based on the flow time of the messages in order to decrease average message delay. The flow time of the message is composed of message waiting time and message transmission time. Moreover, due to the physical characteristic of WDM optical networks, the waiting time is dependent on channel availability and receiver availability.

Therefore, the time for a message to use the channel is not only its transmission time but also the time to wait for the channel and receiver to be ready. The salient feature of this algorithm is its ability to sequence the messages by taking the system constraints of WDM optical networks into account.

2) The Scheduling Minimizing Idle Time (SMIT) algorithm tries to reduce or eliminate the idle time durations along the data channels by scheduling the messages that will not introduce the idle time first. This algorithm is developed based on the observation that when a message selected for transmission is destined to a busy node, the message cannot start the transmission immediately even if the channel has been available for a while. In this case, the channel will be idle for some time to wait for the availability of the destination, which is defined as the channel idle time. This kind of time duration will impair the network performance in terms of message delay as well as channel utilization. SMIT algorithm is the first algorithm that has ever carefully considered the timing relationship of the message scheduling in WDM optical networks.

In this research, we have conducted extensive simulations to evaluate the performance of the proposed scheduling algorithms. Also, some of them are evaluated by theoretical analysis by comparing with other algorithms with similar functions. Furthermore, in order to show the superiority of these algorithms over each other, comparisons among the channel assignment algorithms, message sequencing algorithms and the algorithms combining these two kinds of algorithms are conducted respectively.

2. We have proposed several novel scheduling mechanisms to provide QoS support for single-hop passive-star coupled WDM LANs/MANs, which are summarized as follows.

1) We have proposed a Differentiated Dropping Scheduling (DDS) algorithm to handle real-time traffic. It is developed based on the observation that the message with tight constraints may possibly have the longer message length. Therefore, the sequencing should consider the time constraint as well as message length in order to avoid the blocking problem of longer messages so that more messages can meet their deadlines resulting in less message

loss rate. The significant feature of this new algorithm is its ability to sequence the messages by taking the special characteristic of WDM optical networks into account so that the negative impact on the performance caused by the receiver unavailability can be reduced. The performance of this algorithm is evaluated by experimental and theoretical analysis.

2) We have developed a novel scheduling algorithm, called Cost-Based Priority Scheduling (CBPS), to provide differentiated service to the messages with and without time constraints. The proposed algorithm considers the order of the message transmission based on the costs of messages incurred in the network. The message delay, message time constraints and message static priority are considered as the factors to message transmission costs and each of them is associated with one parameter. With the dynamic priority scheme, this algorithm is shown to be able to reduce not only the message loss rate but also the average message delay when an integrated traffic is applied to the network so that the transmission of both types of messages could be benefited. Also, it is shown that different combination of the cost parameters leads to different performance.

3) We have set up a CBPS based QoS prediction framework to provide QoS predictions to different kind of applications with various QoS requirements. The framework consists of four modules. *QoS Requirements Monitoring* detects the changes in the supplied QoS. These QoS requirements refer to the objective set. Based on the objective set, *QoS Prediction* performs at real-time to estimate whether one combination of parameters related to the cost factors in CBPS algorithm can be found so that the new QoS can be adapted to the current QoS. If the violation is detected by *QoS Violation*, a module of *QoS Estimation* is used to make the decision whether the hopeless new traffic is refused or accepted with acceptable QoS requirements. Therefore, this structure is able control network behavior when the surrounding environment changes.

3. We have set up a local optimization framework that tries to obtain the optimal performance in terms of average delay or loss rate. Two off-line scheduling algorithms are proposed based on two Mixed Integer Linear Programming (MILP) models. These two

MILP models have included system constraints into mathematical formulations so that the channel collision as well as receiver collision can be prevented. Furthermore, these two MILP models are able to reduce the negative impact of tuning overhead of the transceivers by selecting the channel that the source node or destination node has been tuned to. This way can avoid the retuning of the transceivers in some cases. And one off-line emulator is set up in order to evaluate the performance of the proposed algorithms.

1.4 Organization of the Thesis

This thesis is organized into seven chapters. Besides this introduction part (Chapter 1), a comprehensive overview on state-of-the-art MAC protocols for single-hop passive-star coupled WDM optical networks is given in Chapter 2. Depending on the characteristics, complexity, and capabilities of these MAC protocols, we have classified them as MAC protocols for packet transmission, MAC protocols for variable-length message transmission, and MAC protocols with QoS concerns. In Chapter 3, we present two channel assignment algorithms and three message sequencing algorithms based on a PSC-based broadcast-and-select topology, which can be viewed as representatives of reservation-based access protocols relying on a separate control channel. Meanwhile, the limitations of these algorithms are also presented in this chapter. By this survey, we will show that our research results are novel and important.

In Chapter 4, three novel channel assignment algorithms as well as two novel message sequencing algorithms which support network transmission service to variable-length messages are proposed. Detail algorithm descriptions as well as the mathematical analysis or simulation experiments of these two kinds of algorithms are presented in the first and second part of this chapter respectively. Furthermore, in order to show superiority of these algorithms over each other, some simple comparisons concerning three channel assignment algorithms, two message sequencing algorithms and the combinational algorithms are presented in the third part of this chapter.

In Chapter 5, three novel reservation-based algorithms to provide QoS for single-hop passive-star coupled WDM optical networks have been proposed and their performance is analyzed. First, the algorithm DDS, which aims to handle real-time traffic, is introduced. The results from mathematical analysis and simulation experiments are presented. Second, in order to provide differentiated service to the messages with different time constraints, one novel algorithm, called CBPS, is introduced. This algorithm is designed to handle messages with and without time constraints. Besides detail algorithm description, the theoretical analysis and experimental evaluation to show the performance of the specified WDM optical networks with our scheduling algorithm are presented. Last, based on CBPS algorithm, a novel QoS prediction structure used for the problem of providing QoS prediction to the various applications with various QoS requirements is set up. We have presented the detail description of the principle as well as the performance analysis for this prediction structure. Furthermore, some simple examples are given to show how the structure works given the QoS requirements of the applications.

In Chapter 6, a local optimization framework with the aim to obtain the optimal performance in terms of average message delay and message loss rate is set up. The framework consists of two MILP based off-line scheduling algorithms which are developed by including the system constraints in mathematical formulations. The detail descriptions for the construction of mathematical constraints and objective functions in the two MILP models are presented respectively. Meanwhile, the comparisons for these two off-line algorithms with other efficient algorithms with the similar functions are conducted respectively to show the superior performance of the new algorithms.

Finally, in Chapter 7, we conclude the thesis with a summary and a brief discussion of possible future research work.

Chapter 2 . Overview of MAC Protocols on Single-Hop Passive-Star Coupled WDM Optical Networks

WDM is an effective technique for utilizing the large bandwidth of an optical fiber. By allowing multiple messages to be simultaneously transmitted on a number of channels, WDM has the potential to significantly improve the performance of optical networks. A passive star coupler, equipped with tunable transmitters and tunable receivers, can be used to construct a multi-access LAN/MAN using WDM channels. It has the potential of sharing the enormous bandwidth of the optical medium among all the network users. In order to fully exploit the enormous available bandwidth of the optical fiber, efficient MAC protocols are needed to efficiently allocate and coordinate the system resources. Generally, the key requirements and features of access protocols for LANs/MANs comprise flexibility in terms of bandwidth allocation and configuration, low cost and compatibility with existing network architectures and protocols. This chapter will present an introduction to single-hop passive-star coupled WDM optical networks, followed by a comprehensive survey of state-the-art MAC protocols for WDM optical networks.

2.1 Passive Star-Coupled WDM Optical Networks

The passive star-coupled WDM optical network is the simplest and most popular topology for high-speed LANs/MANs. The star topology is attractive, first because of its logarithmic

splitting loss in the coupler (since the splitter portion of the coupler circuit is essentially a binary tree type structure), and second because of no tapping or insertion loss (as in linear bus). Moreover, the passive star network can typically support a larger number of users than, for example a linear bus topology because power loss and tapping loss in linear buses limit the number of users that can be attached without adding broadband optical amplifiers. In addition, the passive property of the optical star coupler is important for network reliability, since no power is needed to operate the coupler.

The passive star coupler is a broadcast device, in which all coming signals are combined onto a single fiber and then this is split to direct $1/n$ part of the combined signal back to each station. It is the key component to configure the broadcast-and-select network, which is set up by connecting computing nodes via two-way fibers to a passive star coupler, as shown in Figure 2.1. A fiber from each node is connected to a star coupler where the signals from all the nodes are mixed. Another set of fibers travels from the hub to each node. Communication between the source and destination nodes proceeds in one of the following two modes: single-hop, in which communication takes place directly between two nodes [4], or multi-hop, in which information from a source to a destination may be routed through the intermediate nodes of the network [5]. In our research, we will focus on the single-hop architecture.

Figure 2.1 A broadcast-and-select WDM network

In single-hop passive star-coupled WDM optical networks, a node sends its packets to the star coupler on one available wavelength by using a laser device which emits an optical

data stream. The data streams from multiple sources are optically combined at the star coupler and the signal power of each stream is evenly split and forwarded to all of the nodes. At the receiving node, the node's receiver, typically an optical WDM filter has to be properly tuned to one of the wavelengths to receive the respective data stream. Based on whether the nodal transceivers are tunable or not, WDM systems may be classified as: Fixed Transmitter/Fixed Receiver (FT-FR) systems, Tunable Transmitter/Fixed Receiver (TT-FR) systems, Fixed Transmitter/Tunable Receiver (FT-TR) systems, and Tunable Transmitter/Tunable Receiver (TT-TR) systems. Systems with pretransmission coordination, i.e., employing a Control Channel (CC), can be formally specified by using an additional index CC such as, for instance, CC-TT-TR.

So far, some experimental testbeds have recently been built in several research laboratories. The representatives of WDM LAN/MAN prototypes based on the passive-star topology are the LAMBDANET ([6], Bellcore), Rainbow ([7][8], IBM) and STARNET ([9], Stanford University) testbeds, which are discussed as follows.

(1) LAMBDANET

The maximum number of supported nodes (N) is 16 for LAMBDANET. The transmission speed of each node is 2 Gb/s and therefore the network throughput is 32Gb/s. The network range is 57.5km. In this system, the number of wavelength channels (C) accommodated has to be the same with the number of networks nodes (N), i.e., $N=C$. The LAMBDANET yields a FT-FR^N architecture, i.e., each node is equipped with one fixed-tuned transmitter and N fixed-tuned receivers. Each receiver is fixed to each available wavelength, a physical channel. The advantages of this prototype are its simplicity of design and architectural support of multicasting. No tunable components as well as no complicated protocols are required in this system. However, every node has to be equipped with N receivers so that the cost per node is proportional to the number of nodes. Moreover, this architecture is not scalable since C channels are needed. This design has a number of

potential applications including transmission of voice and data, and distribution of video services.

(2) Rainbow

The Rainbow LAN/MAN supports 32 nodes each operating at 300 Mb/S. Therefore, the network throughput is 9.6 Gb/s. Each node is equipped with one fixed transmitter which is fixed tuned to its own unique wavelength channel (home channel) and one tunable receiver which can be tuned to any available wavelength channel, i.e. yielding a FT-TR architecture. The tuning time of the receiver may be up to 25ms. When a receiver is idle, it scans all the incoming wavelengths in a round-robin fashion until it finds a channel with a setup request containing the receiver's address. After that, an acknowledgment is transmitted to the source node to make the actual connection setting up. This is so-called in-band receiver polling mechanism. By incorporating some higher-layer protocols, the Rainbow-II optical network [8] extends the Rainbow network in the aspect that the nodes are operating at 1 Gb/s over a distance of 10 to 20 km. The long setup-acknowledgment delay in this system makes it not suitable for packet-switched traffic. However, it works well for those circuit-switched applications, like telephony and video-conferencing.

(3) STARNET

STARNT is a passive star coupler based testbed constructed at the Optical Communications Research Laboratory (OCRL) of Stanford University. It offers all users two logical subnetworks: a high-speed reconfigurable packet-switched data subnetwork and a moderate-speed fix-tuned packet-switched control subnetwork. Thus STARNET supports traffic with a wide range of speed and continuity characteristics. Each node in this system contains one fixed transmitter and two tunable receivers, leading to a FT-TR² system. The transmitter is fixed-tuned to the node's home channel, while one receiver (main receiver) is dedicated to the high-speed data network and the second receiver (auxiliary receiver) is dedicated to the low-speed data network. The transmission speed is 2.5 GB/s for fast network and 125 MB/s FDDI compatible for slower control network. Using simulation and

few experimental results, it is shown that STARNET is highly suitable for low speed applications, like email and telephony, as well as high speed multimedia network applications, like video-conferencing and HDTV.

The testbeds addressed above represent proof-of-concept systems for investigating the practicality of WDM LAN/MAN and the deployment of different WDM devices. Note that, so far, only very simple access protocols have been used in these prototypes. However, the proposals for much more sophisticated and efficient MAC protocols are needed in future systems.

In single-hop networks, a significant amount of dynamic coordination between nodes is required. For a packet transmission to occur, one of the transmitters of the sending node and one of the receivers of the destination node must be tuned to the same wavelength for the duration of the packet's transmission. It is important that transmitters and receivers tune to the same channels quickly, so that packets can be transmitted and received in quick succession. However, in comparison to packet transmission times, the tuning time of these transceivers is relatively long. Moreover, the tunable range for these transceivers is small. So the key challenge in single-hop architecture is to develop appropriate MAC protocols for efficiently coordinating the data transmissions and efficiently exploiting the potential vast bandwidth of an optical fiber to meet the increasing information transmission demand under the constraints of the network resources and the constraints imposed on the transmitted information. Note that *channel collisions* will occur when two or more signals simultaneously arrive at the star coupler on the same wavelength and *receiver collisions* will occur when two or more signals are destined to the same destination. So a proper medium access protocol has to either prevent such collisions or efficiently resolve them when they occur.

According to the network service provided to the transmitted information, the MAC protocols of WDM networks can be roughly divided into three categories as follows: MAC

protocols for packet transmission, MAC protocols for variable-length message transmission, and MAC protocols with QoS concerns.

2.2 MAC Protocols for Packet Transmission

The MAC protocols dedicated for fixed-length packet transmission are so-called “legacy” protocols, which are often adopted from legacy shared medium networks. For a single-hop system to be efficient, the bandwidth allocation among the contending nodes must be dynamically managed. Such systems can be categorized into two categories: those employing pre-transmission coordination, and those not requiring any pre-transmission coordination.

2.2.1 Non Pre-Transmission Coordination Protocols

Protocols with non pre-transmission coordination do not have to reserve any channels. Arbitration of transmission rights is performed either in a pre-assigned fashion (fixed assignment protocols and partial fixed assignment protocols) or through contention-based data transmissions (random access protocols) on the regular data channels.

2.2.1.1 Fixed Assignment Protocols

A simple technique that allows single-hop communication is based on fixed assignment technique, Time Division Multiplexing (TDM) extended over a multi-channel environment [10]. Each node is equipped with one tunable transmitter and one tunable receiver; hence these systems are classified as TT-TR systems. It is pre-determined that a pair of nodes is allowed to communicate with each other in the specified time slot within a cycle on the specified channel. Several extensions to the above protocol have been proposed to improve the performance. One approach, named weighted TDM, assigns different number of time slots to different transmitting nodes according to the traffic load on each node [11]. Another approach proposed a versatile time-wavelength assignment algorithm [12]. In this protocol, node i is equipped with t^i transmitters and r^i receivers, all of which are tunable over all

available channels. This algorithm is designed such that, given a traffic demand matrix, it will minimize the tuning times in the schedule, while also minimizing the packet transmission duration. Based on [12], some new algorithms [13][14][15] study problems such as the performance of scheduling packet transmissions with an arbitrary traffic matrix and the effect of the tuning time on the performance.

2.2.1.2 Partial Fixed Assignment Protocols

The above fixed assignment protocols are too pessimistic because their main goal is to avoid both channel collisions and receiver collisions. However, alternative protocols can be defined in which the channel allocation procedures are less restrictive. A number of such protocols have also been studied in [10]. The first one is Destination allocation (DA) protocol. By using this protocol, the number of node pairs that can communicate over a slot is increased from the earlier value of N (the number of channels) to M (the number of nodes). The second one is Source Allocation (SA) protocol in which the control of access to the channels is further reduced. Similar to the SA protocol, Allocation Free (AF) protocol has been proposed, in which all source-destination pairs have full rights to transmit on any channel over any time slot duration.

2.2.1.3 Random Access Protocols

Two slotted ALOHA (SA) protocols were proposed in [16]. In the first protocol, time is slotted on all the channels, and these slots are synchronized across all channels. In the second protocol, each packet is considered to be of L minislots, and time across is synchronized across all channels over minislots. Another two similar protocols were proposed in [17].

2.2.2 Pre-Transmission Coordination Protocols

Pre-transmission coordination protocols allocate a channel as the control channel to transmit global information about the message to all the nodes in the system. These protocols can be

categorized according to the ways to access the control channels into the following subgroups.

2.2.2.1 Random Access Protocols

In [18], three random access protocols such as ALOHA, slotted-ALOHA, and CSMA are proposed to access the control channel. ALOHA, CSMA, and N-server switch scheme can be the sub-protocols for the data channels. In a typical ALOHA protocol, a node transmits a control packet over the control channel at a randomly selected time, after which it immediately transmits a data packet on a data channel, which is specified by the control packet. In [19], an improved protocol named slotted-ALOHA/delayed-ALOHA has been proposed. In this protocol, transmitting node will delay transmitting data on a data channel until it gets the knowledge of that its control packet has been successfully received by the destination node. This protocol can decrease the data channel collision and improve throughput comparing with the protocols in [18]. Similarly, in [20], one set of slotted-ALOHA protocols and one set of Reservation-ALOHA protocols have been proposed to improve the performance of the protocols in [18]. And in [21], a so-called Multi-Control-Channel protocol is proposed, which aims to improve Reservation-ALOHA-based protocols.

Both the protocols in [18] and [19] ignored “receiver collisions”, indicating that the probability of receiver collisions is small for large population systems and they would be taken care of by higher-level protocols. And [20] and [21] cannot prevent receiver collisions, too. A protocol, which is especially designed to avoid receiver collision, is presented in [23].

2.2.2.2 Reservation-Based Protocols

In [24], a Dynamic Time-Wavelength Division Multiaccess (DT-WDMA) protocol is proposed. In this protocol, a channel is reserved as control channel and fixed *time-division multi-access* (TDMA) is used within each slot on it. It requires two pairs of transmitters and receivers. One pair of the transceivers is fixed to the control channel and, another pair, with a fixed transmitter and a tunable receiver, is used for data channel, i.e., yielding a FT/FT-

FR/TR architecture. Although this protocol cannot avoid receiver collisions, it ensures that exactly one data can be successfully accepted when more than one data packets come to the same destination node simultaneously. One proposal [25] to improve DT-WDMA algorithm introduces the concept of resolving receiver collisions by incorporating a delay line receiver to buffer the potential collided packets. And a conflict-free protocol is proposed in [26].

In [27], a new algorithm, named Hybrid Reservation Preallocation/Time Slot Assignment (HRP/TSA), is proposed. It combines the concepts of receiver pre-allocation and reservation access. And nodes were allowed reservations on multiple channels in the same cycle. This work aims to reduce high time complexity and schedule lengths of the existing scheduling algorithms. It is shown that the slight increased computational overhead with scheduling is justified by reduced packet latency and higher utilization, especially for client-server traffic.

In [28], another two reservation-based protocols aiming at improving the DT-WDMA algorithm are outlined. The first one is called Dynamic Allocation Scheme (DAS), which dynamically assigns slots on a packet-by-packet basis. The second protocol is named Hybrid TDM, which combines the TDM and the DAS scheme and allows both pre-assigned and dynamic slot assignment. Time on the data channels is divided into frames consisting of several slots. In a certain period of time, one slot will be opened for a transmitting node to transmit data packets to any destination receiver.

A reservation-based multi-control-channel protocol can be found in [22]. By this protocol, x channels ($1 < x < N/2$) can be reserved as control channels to transmit control information, where N is the number of channels in the network. The objective to reserve multiple control channels in the network is to decrease the overhead of control information processing time as much as possible.

Based on the above survey, we find that, among non pre-transmission coordination protocols, those taking fixed-channel assignment approach can ensure that data is successfully transmitted and received, but they are sensitive to the dynamic bandwidth requirements of the network and they are difficult to scale in terms of the number of nodes.

And those taking contention-based channel assignment approach can adapt to the dynamic bandwidth requirements, but they introduce contention on data channels. As a result, either channel collision or receiver collision will occur.

On the other hand, among pre-transmission coordination protocols, the protocols with contention-based control channel assignment still have data channel collision and receiver collision because it introduces contention on control channel. However, by continuously testing the network states, some protocols proposed in [34][35] may have the capability to avoid both collisions. The reservation-based protocols, which take fixed control channel assignment approach, can only ensure data transmission without channel collisions. However, by introducing some information to make the network nodes intelligent, it may have potential to avoid receiver collisions as well. It also has potential to accommodate application traffic composed of variable-length messages.

2.3 MAC Protocols for Variable-Length Message Transmission

The “legacy” MAC protocols are designed to schedule fixed length packets. However, in the real world, traffic streams are often characterized as bursty, and consecutive arriving packets in a burst often have the same destination. Based on this observation, we can schedule all the fixed size packets as a whole rather than schedule them on a packet-by-packet basis. Using this kind of protocols has three advantages as follows: firstly, to an application, the performance metrics of its data units have more relevant performance measures than ones specified by individual packets; secondly, it perfectly fits the current trend of carrying IP traffic over WDM networks; and lastly, message fragmentation and reassembly are not needed.

2.3.1 Basic MAC Protocols for Variable-Length Messages

The first two MAC protocols in [20] proposed for variable-length message transmission are protocols with contention-based control channel assignment. Another two Reservation-

ALOHA-based protocols in [20] are presented in order to serve the long holding time traffic of variable-length messages. Data channel collisions can be avoided in the protocols presented in [20].

The protocol in [29][30][31] tries to improve the reservation-based DT-WDMA protocol in [24] in three aspects: the number of nodes is larger than the number of channels; the transmitted data is a variable-length message rather than a fixed length packet; and data transmission can start without delay.

The protocol introduced in [32], called FatMAC, is a hybrid approach that combines the advantage of receiver preallocation and reservation access strategies. It reserves access on preallocation-based channels through control packets. A reservation specifies the destination, the channel and the message length of next data transmission. This protocol is based on tunable transmitters and fixed receivers. LiteMAC in [33] is an extension to FatMAC. In LiteMAC protocol, each node is equipped with a tunable transmitter and a tunable receiver rather than a fixed receiver in FatMAC. LiteMAC has more flexibility than FatMAC because of its tunable receiver and special scheduling mechanism. Both these two algorithms can transmit variable-length packets without collisions. It has been proved that these two protocols have better performance than preallocation-based protocols while less transmission channels are used than reservation-based protocols.

An intelligent reservation-based protocol for scheduling variable-length message transmission has been proposed in [36]. It has the ability to avoid both channel collision and receiver collision by sharing some global information. This makes it a milestone in the development of MAC protocols for WDM optical networks. Each node is equipped with a fixed transmitter (called control transmitter) and a fixed receiver (called control receiver), both of which are tuned to the control channel. In addition, a tunable transmitter (called data transmitter) and a tunable receiver (called data receiver) are employed at each node to enable it to access the data channels. The number of channels may be much smaller than the number of nodes. A Time Division Multiple Access (TDMA) protocol is employed to access the

control channel so that the collision of control packets can be avoided. In [36], three data channel assignment algorithms have been proposed. The fundamental one is named Earliest Available Time Scheduling algorithm (EATS). This algorithm schedules the transmission of a message by selecting a data channel, which is available the soonest.

Some related protocols have been proposed to improve the performance of the network based on the same system architecture of [36]. In [37], two protocols for scheduling variable-length packet transmission have been proposed. The distinction between these protocols and other WDM protocols is that message transmissions are initiated by the receipt of a control message; other schemes schedule packet transmission for some fixed point in the future. This flexibility allows these protocols to avoid "head-of-line" blocking. In [38], the authors find that significant improvement in performance can be achieved using scheduling algorithms where message sequencing and channel assignment are simultaneously taken into consideration.

2.3.2 Novel MAC Protocols for Variable-Length Messages

As an example of the general scheduling scheme in [38], a new scheduling algorithm is proposed in [39]. The algorithm, named Receiver-Oriented Earliest Available Time Scheduling (RO-EATS), is designed based on the observation that two consecutive messages with the same destination may not fully use the available channels when EATS algorithm is employed. Therefore, this algorithm has the ability to decrease message transmission blocking caused by avoiding receiver collisions. By this protocol, the management of messages transmission and reception is the same as that by the protocol in [36]. The difference between the two protocols is on the scheduling algorithm for message transmission. In the RO-EATS, it first considers the earliest available receiver among all the nodes in the network and then selects a message, which is destined to this receiver from those messages. After that, a channel is selected and assigned to the selected message by the principle of EATS algorithm. The new algorithm enforces the idea of scheduling two

consecutive messages away from going to the same destination node. In this way, average message delay can be shown to be quite low and channel utilization can be shown to be high.

In [40], a novel signaling protocol called the Sampling Probe Algorithm (SPA) is proposed. This protocol is based on pre-transmission coordination, distinct from all the existing approaches in that it does not require a separated out-band signaling channel (control channel), thus alleviating the stringent requirement on the number of channels required in most control channel based protocols. The reservations occur on the same channels where data packets are transmitted, i.e., in-band signaling. This proposed scheme works well for systems under moderate or heavy traffic loading.

Recently, a persistent reservation protocol for variable-length messages in WDM-based local networks using a passive star topology is proposed in [41]. With this protocol, once a node reserves a data channel, the node persistently uses the channel until its message is completely transmitted. In this protocol, the control channel is shared by all nodes or on a contention basis using the slotted SLOHA protocol. And data channel and destination collision can be avoided with this protocol. The protocol is suitable for a network in which accommodation of variable-length messages (such as circuit-switched traffic or traffic with long holding times) is required. Moreover, the protocol enables any new node to join the network at anytime without network re-initialization.

A new channel reservation protocol using a counter for detection of a source conflict in a WDM single-hop network with non-equivalent propagation delay is introduced in [42]. In this protocol, a source conflict occurs when a source node has the right to transmit more than two messages to their destination nodes using different wavelengths in the same time slot. By investigating information about the final message which has succeeded in reservation, a source node can detect a source conflict before the assignment of wavelengths. With this protocol, the mean message delay can be dramatically reduced without degrading throughput performance as the offered load becomes large.

In [43], one novel reservation-based scheme is proposed to increase the utilization of the control slots and reduce the packet delays. Different from other reservation-based schemes, the proposed scheme can dynamically adjust the length of the control frames according to the traffic patterns of the nodes. In this way, the nodes can acquire and release the control slots dynamically so that the control slots on the control channel can be used efficiently and the packet delays can be reduced.

And in [44], one novel protocol for bursty traffic is proposed. In this paper, all the channels will be used to transmit messages. Each node is equipped with one tunable transmitter and one fixed receiver, i.e., yielding a TT/FR architecture. For each wavelength λ_i , each station maintains the information of the set of stations A_i which are estimated to be active for this wavelength. The stations in set A_i are granted permission to transmit on wavelength λ_i in a round-robin fashion. The stations which are granted permission to transmit at each time slot are selected by taking into account the network feedback information. This protocol is able to allocate the bandwidth of each wavelength to the stations according to their needs so that the number of idle slots is reduced, resulting in the increase of the network throughput.

So far, various architectures and protocols belonging to single-hop passive star coupled WDM optical network were reviewed. According to the survey, WDM systems can be divided into several types based on whether the nodal transceivers are tunable or not. In reality, tunable transceivers are more expensive than fixed transceivers. Table 2.1 provides a simple comparison among some protocols. As the table shows, in [19] and [23], only one pair of tunable transmitter and receiver is used, which is to avoid the costly network interface unit because of the additional fixed transmitter and fixed receiver to monitor control channel. However, the throughput achieved by these two protocols is not as high as that achieved by the protocols in [28] and [36]. In order to achieve better performance, more transmitters and receivers are used in [28] and [36]. For example, in [36], one pair of fixed transmitter and

receiver is used for control channel and one pair of tunable transmitter and receiver is dedicated for data channels. It becomes evident from this comparison that a tradeoff between equipment costs (related to the item architecture in Table 2.1) and the performance has to be considered when designing the MAC protocols.

protocol	architecture	control channel access protocol	processing complexity	throughput
Random Access [19]	CC-TT-TR	Slotted Aloha	low	low
DT-WDMA [24]	CC-FT/FT-FR/TR	TDMA	high	moderate
DAS, HTDM [28]	CC-FT/FT-FR/TR	TDMA	very high	high
TDMA-C [31]	CC-TT-FR/TR	TDMA	high	low
RCA[23]	CC-TT-TR	Slotted Aloha	moderate	moderate
EATS [36]	CC-FT/TT-FR/TR	TDMA	high	high

Table 2.1. Comparison of some protocols

Various contributions made on the research and development of MAC protocols for WDM-based local networks were discussed in this subsection. However, to provide real-time service to time-constrained application streams such as video or audio information become more and more important in the design of the high-speed computer networks such as WDM optical networks. It is explicit that MAC protocols are much needed to support QoS requirements. The protocols mentioned in this subsection can be a start point for research on this area.

2.4 MAC Protocols with QoS Concerns

One of the important issues of high-speed networks, such as WDM optical networks, is to support real-time message transmission with QoS demands and transmission service to multimedia applications with QoS requirements. The most important aspect of the former service is that a message generated at the source node must be received at the destination node within a given amount of time. This time is referred to as message deadline. Failure to

meet the deadlines of these tasks may lead to catastrophic consequences. The latter service needs a certain amount of bandwidth to deliver video/audio frames in time consistent with human perception.

2.4.1 MAC Protocols for Real-Time Service

A major challenge in the design of future generation high-speed networks is the provision of real-time service to time-constraint application streams such as video or audio information. If the delay of a message in the system exceeds its time constraint, the message is considered as late. The task of the scheduling algorithms for the protocols that provide real-time service to time-constrained messages is to schedule the transmission of the messages to meet the message time constraint as much as possible. Most of the MAC protocols that provide real-time service on passive star-coupled WDM optical networks are protocols with reservation-based pre-coordination. According to the type of the real-time service provided to the transmitted messages, the MAC protocols for real-time service can be simply classified into three types: protocols with best-effort service, protocols with deterministically guaranteed service and protocols for statistically guaranteed service.

2.4.1.1 MAC Protocols for Best-Effort Real-Time Service

A real-time protocol Time-Deterministic Time and Wavelength Division Multiple Access (TD-TWDMA), based on TDM (Time Division Multiplexing), for a fiber-optic star network is presented in [45]. Services for both guarantee-seeking messages and best-effort messages are supported by using this protocol. In this protocol, the access to each channel is divided into cycles of time-slots. Each node has a number of guaranteed slots to support guarantee-seeking messages. However, if there are no guarantee-seeking messages in a node the slots will be released for best effort messages from other nodes (or the same node) according to a predetermined scheme. Each node is equipped with one fixed-wavelength transmitter, which is always tuned to one specific wavelength channel, and tunable receivers, which can be tuned to an arbitrary wavelength channel. It is assumed that the number of wavelengths, C ,

equals the number of nodes, M . There are $2M$ queues in each transmitter, M for best-effort messages and M for guarantee-seeking messages. For each type of queues, one queue is for broadcast and $M-1$ queues for the single destination. Message transmission scheduling is based on the queue priority. Deadline guarantees are supported where the underlying deterministic bandwidth can be changed dynamically through slot reserving.

A reservation-based MAC protocol for best-effort real-time service can be found in [46]. This protocol is for the same network structure as that in [36]. Both hard real-time and soft real-time variable-length message transmissions have been considered. The scheduling algorithms of the protocol are based on the time related dynamic priority scheme, namely Minimum Laxity First (MLF) scheduling. The principle of this dynamic scheduling scheme is that the most stringent message will get the transmission service first. This work has confirmed that when real-time traffic involved in the networks, dynamic time-based priority assignment schemes as well as priority-based scheduling algorithms should be employed to improve the real-time performance of the networks as much as possible.

In [47], a novel reservation-based MAC protocol for real-time service has been proposed, which extends the function of the protocol in [46] to provide differentiated service to benefit both real-time and non real-time applications in one topology. The scheduling algorithm Minimum Laxity First with Time Tolerance Scheduling Algorithm (MLF-TTS) schedules real-time message transmission according to their time constraints. The basic MLF scheduling policy is adopted for scheduling real-time traffic. After the real-time messages have been scheduled to transmit on certain channels in certain time slots to their destination nodes, some of them could be blocked just because there may be more than two consecutive messages going to the same destination node in a very short time period. The MLF-TTS algorithm seeks and takes this waiting time period to schedule the transmission of non real-time messages under the condition that the transmission time of these messages should be less than the time that the blocked real-time messages are waiting for their destinations available. By the MLF-TTS algorithm, the average message delay for the messages without

time constraints could be expected to decrease while the message loss rate for hard real-time messages or message tardy rate for soft real-time messages is kept as low as those of the simple MLF algorithms. And the channel utilization could be expected to be high.

2.4.1.2 MAC Protocols with Deterministically Guaranteed Service

It is obvious that the QoS provided by a network service to real-time applications indicates the degree of how the real-time applications can meet their time constraints. However, best-effort real-time network service cannot ensure QoS because it cannot guarantee that real-time applications can meet their time constraints in certain degree when they are transmitted. Therefore, it is necessary to develop MAC protocols with deterministically guaranteed service concerns.

In [48], a preallocation-based Wavelength Division Multiple Access (WDMA) scheme is proposed to provide deterministic timing guarantees to support time constrained communication. A scheme called Binary Splitting Scheme (BSS) is proposed to assign each message stream sufficient and well-spaced slots to fulfill its timing requirement. Given a set of real-time message streams M specified by the maximum length of each stream C_i and the relative deadline of each stream D_i , this scheme can allocate time slots over as few channels as possible in such a way that at least C_i slots are assigned to M_i in any time window of size D_i slots so that the real-time constraints of the message streams can be guaranteed.

A modified pre-allocation based MAC protocol is proposed in [49] to guarantee reserved bandwidth and constant delay bound to the integrated traffic. The access to the transmission channels is controlled by the scheduler, which works based on the concept of computing maximal weighted matching, a generalization of maximal matching on unweighted graph. Based on this concept, several scheduling algorithms have been produced to provide scheduling. A Credit-Weighted Algorithm is proposed to serve guaranteed traffic. A Bucket-Credit Weighted Algorithm is designed to serve bursty traffic. And a Validated Queue Algorithm is a modification of the Bucket-Credit Weighted Algorithm to serve bursty traffic

and keep throughput guarantee at the same time. It has been proved that those scheduling algorithms can guarantee the bandwidth reservation to a certain percentages of the network capacity and ensure a small delay bound even when bursty traffic exists.

A reservation-based MAC protocol for deterministic guaranteed real-time service can be found in [50]. In [50], a systematic scheme comprised of admission control, traffic regulation, and message scheduling that provide guaranteed performance service for real-time application streams made up of variable-length messages. A traffic intensity oriented admission control policy is developed to manage flow level traffic. A g-regularity scheme based on the Max-plus algebra theory is employed to shape the traffic. An Adaptive Round-Robin and Earliest Available Time Scheduling (ARR-EATS) algorithm is proposed to schedule variable-length message transmission. All of those are integrated to ensure that the deterministic guaranteed real-time service can be achieved.

2.4.1.3 MAC Protocols for Statistically Guaranteed Service

The MAC protocols with deterministically guaranteed service can normally guarantee specific transmission delays to real-time applications. Or under certain time constraints imposed to the real-time applications, a specific percentage of real-time messages, which can meet the time constraints, can be predicted. However, MAC protocols for statistically guaranteed service cannot provide the deterministic guaranteed QoS service. Only an estimated percentage of real-time messages, which can meet their time constraints, can be evaluated statistically. Most of the MAC protocols in this category consider the issue of providing statistically guaranteed service to both real-time and non real-time applications, i.e., differentiated service. By these protocols, statistical QoS to real-time applications can be expected by sacrificing the transmission service to non real-time applications.

A novel reservation-based MAC protocol is proposed in [51] to support statistically guaranteed service in WDM networks by using a hierarchical scheduling framework. This work is developed from a similar network structure as that in [36]. The major advantage of

its protocol is that it divides the scheduling issue into flow scheduling or VC scheduling and transmission scheduling. The former is responsible for considering the order of traffic streams to be transmitted. The latter is to decide the order of the packets transmission. The packets involved in the transmission scheduling are those selected from the traffic streams by the flow schedule scheme. This protocol is expected to diminish the ratio of the packets which are over their deadlines. Another good point of this protocol is that a re-scheduling scheme is employed to compensate the failure scheduling result due to either output conflict or channel conflict. If the failure is from a real-time traffic, it certainly makes sense to re-schedule the very same packet as soon as possible. And if real-time traffic has more stringent QoS requirements, the re-scheduling scheme will ignore re-scheduling the failed non real-time packet to ensure the real-time traffic to meet its time constraints.

A MAC protocol, which is based on a multi-channel ring topology, has been presented in [52]. The transmitted information is in the form of fixed size packet. A collision free slotted MAC protocol is proposed, named as Synchronous Round Robin with Reservation (SR3), to support both QoS guarantee to real-time traffic and best-effort service to non real-time traffic. It combines a packet scheduling strategy (called SRR), a fairness control algorithm (called MMR), and a reservation mechanism. SRR achieves an efficient exploitation of the available bandwidth, MMR guarantees fair throughput access to each node, and SR3, by permitting slot reservations, leads to tighter control on access delays, and can thus effectively support traffic classes with different QoS requirements. This protocol has been proved to have the capability to provide packet-mode transport to multiple information flows with differentiated QoS requirements.

In [53], a scheduling scheme for Tbits/sec, input-queued, star-coupler WDM optical networks is presented. This work focuses on packets with fixed length. In this protocol, by using Virtual Output Queueing (VOQ), arriving cells are classified at a primal stage to a queue that corresponds to their designated destination. The scheduler determines which queue is served for transmission. Upon receiving a signal from the reservation scheduler,

each node performs wavelength reservation according to two primary guidelines: a) global switch resources status, i.e., available wavelengths at the reservation instances, and b) local considerations, i.e., the status and priorities of the node's internal queues. Using this protocol, class-differentiated low latency and extremely high throughput is achieved.

A novel scheduling scheme, namely Priority-Differentiated Scheduling (PDS), which is designed to handle real-time (high-priority) and non real-time (low-priority) packets in WDM star networks is introduced in [54]. This protocol is for the same network structure as that in [36]. PDS allows high-priority packets to preempt the prescheduled low-priority packets. By scheduling the high-priority packets first, and then having the preempted packets rescheduled, PDS guarantees that the high-priority packets can always achieve the earlier transmission than the others in order to meet the QoS requirements. In addition, it does not sacrifice the performance of low-priority packets. As a matter of fact, low-priority packets can also benefit from PDS algorithms.

In [55], a reservation-based MAC protocol enabling efficient integration of real-time traffic and data traffic is proposed. In this paper, the control channel is divided into contention-based (Slotted ALOHA) and contention-free (TDMA) fields within a control slot. Thereby, real-time transmission can be reserved by accessing one of the Slotted ALOHA minislots. These Slotted ALOHA minislots are located after each TDMA minislot in a cyclic manner. Data transmission is reserved via access to the TDMA minislots which are uniquely assigned to the respective network nodes. If the real-time control packet of one node fails to reserve the Slotted ALOHA slot, a new real-time control packet will be sent in the next TDMA minislot associated with itself, while the reservation for data traffic had to be deferred to the next control slot. So the reservation delay for real-time traffic is bounded by two round-trip delays.

2.4.2 MAC Protocols for Multimedia Application

There is a rapid growth in the number of multimedia applications recently. The transmission of the multimedia application is a kind of real-time and stream oriented communication. The quality of service required of a stream communication includes guaranteed bandwidth (throughput), delay and delay variation (jitter). Multimedia application integrates a variety of media, namely, audio, video, images, graphics, text, and data, each of which have different QoS requirements. The protocols that provide transmission service to multimedia application should support and ensure the variety of QoS requirements of different types of media. Many researchers have shown their interests on this issue. Some research results are generated from the existing protocols for real-time service. However, some protocols are completely novel or based on new network architectures dedicated to multimedia traffic.

In [56], the feasibility of several existing protocols based on the WDM bus LAN architecture to support multimedia application is studied. By the simulation study, the authors point out that several currently existing MAC protocols such as FairNet, WDMA, and nDQDB are not satisfactory for supporting multimedia traffic in the sense of that those protocols cannot guarantee that the total delay or jitter will not grow beyond the accepted value for different classes of multimedia applications.

In [57], a study on several MAC protocols to support multimedia traffic on WDM optical networks is carried out. These protocols including Distributed Queue Dual Bus (DQDB), Cyclic-Reservation Multiple-Access (CRMA), Distributed-Queue Multiple-Access (DQMA), Fair Distributed Queue (FDQ) are distributed reservation access schemes for WDM optical network based on slotted unidirectional bus structures. The performance of these four protocols is studied to simultaneously support synchronous traffic (for various real-time multimedia applications) and asynchronous traffic (for interactive terminal activities and data transfers). They have pointed out, by the extensive simulation results, that the reservation-based protocols are suitable for integrating real-time multimedia traffic with

bursty data traffic in the WDM optical network when delay constraint is somewhat relaxed. And the FDQ protocol stands out to support heterogeneous traffic.

In [58], a video-on-demand system over a passive star-coupler based WDM optical network is studied. The video-on-demand (VOD) traffic is a constant bit rate (CBR) traffic because the video/audio sources of the application are processed in advance, kept on the video server, and transmitted at a regular rate. The VOD traffic is desirable to be served by isochronous transmission service by the network. A centralized medium access control scheduler is employed to schedule the isochronous and the asynchronous traffic demands. A scheduling algorithm, named as KT-MTR, is employed for scheduling the asynchronous traffic only. And a scheduling algorithm, IATSA-MTR, is presented for scheduling both isochronous and asynchronous traffic coexisted in the network. Those scheduling algorithms are proved to be efficient for serving VOD applications in the WDM optical networks.

To efficiently support multimedia traffic streams, an efficient scheme Multimedia Wavelength-Division Multiple-Access (M-WDMA) is proposed in [59]. Each node has three tunable transmitters used to serve three different classes of traffic streams according to the corresponding sub-protocols. Three types of multimedia traffic streams, including a constant-bit-rate traffic (CBR), a variable-bit-rate traffic with large burstness (VBR1), and a variable-bit-rate traffic with longer inter-arrival times (VBR2), are considered by the proposed MAC protocol. The M-WDMA protocol consists of three sub-protocols. One is the TDM sub-protocol, which is an interleaved TDMA MAC protocol. The other one is a reservation-based sub-protocol, RSV, which controls the access to the data channels by using a multiple token method. The third one is a random access sub-protocol, CNT, which works in a way similar to that of the interleaved slotted ALOHA. The outstanding point of this protocol is that a dynamic bandwidth allocation scheme is incorporated into the protocol to dynamically adjust the portions of the bandwidth occupied by the three types of traffic streams according to their QoS demands. It has been proved that the performance of the M-WDMA is good enough for WDM optical networks in serving multimedia applications.

One bandwidth guaranteed multi-access protocol is proposed in [60], in which the control channel contains two types of minislots: reservation minislots and contention minislots. There are M access nodes and one control node in the network. Nodes requiring bandwidth guarantees, called guaranteed nodes, use reservation minislots that are assigned by the control node. The remaining nodes share contention minislots using a random access mechanism. The reservation minislots can guarantee a minimum bandwidth for the guaranteed nodes. The contention minislots enable on-demand services at the optical layer and achieve good fairness for the remaining bandwidth.

A novel MAC protocol for providing guaranteed QoS service to the MPEG compressed video/audio applications in the passive star-coupled WDM optical network is proposed in [61]. The QoS of transmission of the MPEG traffic is derived based on the frame size traces from the MPEG encoded real video sequences. A frame, which is considered as the basic element with variable-size of the MPEG traffic streams, is scheduled and transmitted at one time. A systematic scheme is proposed to guarantee the deterministic delay to the transmission of the MPEG traffic. This scheme includes an admission policy, a traffic characterization mechanism, and a scheduling algorithm as a whole to ensure the QoS of the transmission of MPEG traffic. Analytical evaluation of the guaranteed deterministic delay bound for the proposed system service schemes is based on the theory of max-plus algebra. The deterministic delay bound is verified by intensive trace-driven simulations and modeled MPEG traffic simulations. It is obvious that this protocol stands out as a state-of-the-art MAC protocol among seldom MAC protocols, which support QoS of the transmission of multimedia applications in WDM optical networks.

In [62], an interesting idea on the architecture of the WDM optical network is proposed to support varieties of traffic such as data, real-time traffic, and multicast/broadcast service. The proposed architecture, named as Hybrid Optical Network (HONET), tries to combine the single-hop and multihop WDM optical network architectures into a synergy architecture. The architecture of the HONET can be considered as a network, which consists of the

multihop network with an arbitrary virtual topology and a single-hop network based on a dynamically assigned T/WDMA MAC protocol. In this virtual network architecture, real-time traffic and other connection-oriented applications can be supported by single-hop network; while non real-time data traffic, which can tolerate relatively large delay is supported by multihop network. The advantage of this virtual architecture is that it is flexible to employ different topologies of the multihop network and different MAC protocols for the single-hop network to support varieties of traffic in the optical work according to the QoS of the traffic demands.

2.5 Summary

This chapter has summarized state-of-the-art medium access control protocols for the passive star-coupled WDM optical networks. Depending on the characteristics, complexity, and capabilities of these MAC protocols, we have classified them as data and message transmission MAC protocols, MAC protocols for real-time transmission service, and MAC protocols for multimedia applications. Most of these protocols focus local area environments. Architectural, qualitative and quantitative descriptions of various protocols within each category have been provided. Some important or milestone protocols have been given quite detailed explanations to present their underlined significances. This chapter can be served as a good stating point for researchers working on this area so as to give them an overview of the research efforts conducted for the past decade. The system model and some basic protocols proposed will be introduced in the next chapter.

Chapter 3 . System Model and Impressive Protocols

Single-hop passive-star coupled based WDM optical networks are very attractive mainly due to their low signal losses and their natural broadcast capability. Thus broadcast-and-select networks can be built where the individual wavelengths of the source nodes are broadcasted to all receiving nodes and the appropriate data channels may then be selected by the corresponding destination nodes. To date, many MAC protocols have been proposed for this kind of networks. Comprehensive surveys on these protocols are provided in Chapter 2. In this chapter, some existing basic reservation-based protocols for PSC-based single-hop networks are introduced. Meanwhile, the limitations of these algorithms are also presented.

3.1 System Model

Since channel as well as receiver collisions are most efficiently avoided by employing pre-transmission coordination, reservation-based access protocols are preferred over random access or static-assignment access schemes. Reservation-based access protocols for single-hop WDM networks using a separate control channel can be roughly divided into two groups according to underlying access strategy for the control channel, namely, contention and collision-free reservation protocols. In this thesis, we will focus our attention on collision-free reservation protocols.

3.1.1 Network Architecture

The network under study consists of N nodes, each connected to a passive star coupler via two-way fibers (see Figure 3.1). One direction of the fiber is used for transmission and the other for reception. Each direction of the fiber supports $C+1$ WDM channels ($\lambda_0, \lambda_1, \dots, \lambda_C$) with the same capacity. In general, $C \leq N$. The C channels ($\lambda_1, \dots, \lambda_C$), referred to as data channels, are used for message transmission. The remaining channel (λ_0), referred as control channel, is used to exchange global information among nodes about the messages to be sent. The control channel is the basic mechanism for implementing the reservation scheme. In the network, each node is equipped with two transmitters and two receivers. One transmitter and one receiver are fixed and tuned to the control channel (λ_0), and the other transmitter and receiver are tunable and can be tuned to any data channel to access messages on those channels, i.e., yielding a CC-FT/TT-FR/TR structure. In order for two nodes to communicate, a significant amount of dynamic coordination between nodes is required. For a packet transmission to occur, the transmitter of the sending node and the receiver of the destination node must be tuned to the same wavelength for the duration of the packet's transmission. It is important that transmitters and receivers tune to the same channels quickly so that packets can be transmitted and received in quick succession.

Figure 3.1 Network architecture of a single-hop passive-star coupled WDM optical network (CC-FT/TT-FR/TR system)

3.1.2 Channel Structure and System Assumption

Figure 3.2 illustrates some basic concepts of the channels in our model. As mentioned above, the system consists of C data channels and one control channel, giving $C+1$ channels in total. Time on data channels is divided into data slots, and network-wide synchronization of data slots over all data channels is assumed. The basic interval on the data channels is the transmission time of one fixed-length packet.

Figure 3.2 Data and control channel configuration

On the control channel, time is also divided into control frames. A control frame consists of N control slots, each of which can hold the transmission of one control packet. The detail description of the structure of the control channel can be found in Figure 3.3. The length of control packet is a system design parameter and depends on the number of messages l about which each node is allowed to broadcast and the amount of control information on each message (e.g., the destination node address, source node address, number of data packets in a message, generating time, indication of a message's real-time property, relative deadline, etc.) The larger the number of l of messages that are represented in a control packet, the more globally optimizing our scheduling algorithms will be. In our research, we only consider the case that $l=1$, which means that there are only one message that the node is allowed to broadcast. Access to the control channel is via Time Division Multiple Access

(TDMA) to avoid collision of control packets, i.e., node i can transmit its control packet only in the predetermined i th control slot in a control frame, where $i=1,2,\dots,N$. N control packets make up one control frame on the control channel. Length of data slot needs not equal to the length of control frame, and synchronization of data channels and control channels can be independent, thus enabling scalability of the network.

Figure 3.3 Structure of the control channel

In addition, we assume that round-trip propagation delay between a node and PSC is R_j time units for each node j . Also, transceiver's tuning time is assumed to be a constant of T time slots.

3.1.3 Transmission and Reception Procedure

The nodes can be divided into two non-disjoint sets of source nodes and destination nodes. A queue of messages to be transmitted is assumed to exist at each source node S_i which is shown in Figure 3.2. The nodes are assumed to generate messages with variable lengths which can be divided into several equal-sized packets. Like other reservation-based protocols, the procedure of message transmission and reception in this system model works as follows. Before a message can be sent, a node (say node i) transmits a control packet on the control channel on its assigned control slot before it sends a message. The control packet contains information about one (at the head of S_i ' queue) message it intends to send such as source-node identity (id), destination-node id, packet length, etc. After $R_j + F$ time slots, where R_j is the round-trip propagation delay between the PSC and destination of the message in node i , i.e., node j , and F is the duration of a control frame, node i will receive the control frame. At this point, an identical copy of the distributed scheduling algorithm is invoked by

node i (as well as by other nodes) to sequence the messages and determine the data channel and time duration over which messages will be transmitted. Once a message is scheduled, the sending node's transmitter will be tuned to the selected data channel before the scheduled transmission time and transmit the message at the scheduled transmission time. After another round-trip propagation delay, when the message arrives at its destination, the receiver at the destination should have been tuned to the same channel to receive the message. All the algorithms are executed independently at each node so that each node will reach the same unique schedule.

3.2 Existing Protocols

Generally, two important issues that need to be addressed when designing MAC protocols for WDM optical networks are message sequencing and channel assignment. Channel assignment addresses the problem of choosing an appropriate data channel via which a message is transmitted. This problem has been addressed extensively in [36]. On the other hand, message sequencing addresses the order in which messages are sent. In this section, some basic access protocols to address these two issues are introduced separately. The propagation delay is assumed to be identical (equals to R) for all the nodes in these algorithms.

3.2.1 Channel Assignment Algorithms

The scheduling algorithms which only address the channel assignment aspect schedule the messages individually and independently. Normally, they schedule the message instantly upon receiving the control information of it.

3.2.1.1 Earliest Available Time Scheduling (EATS)

The EATS protocol based on [36] is capable of accommodating variable-length messages and incorporates the transceiver tuning times as well as round-trip propagation delay of the network. The basic idea of EATS is to select the earliest available channel all the time. In

order to avoid channel and receiver collisions under this reservation-based protocol, global information is exchanged between the network nodes and control channel. The global information consists of two tables. An identical copy of this information is maintained and updated at each node. The two tables are Channel Available Time (*CAT*) and Receiver Available Time (*RAT*), which can be defined as follows:

- *CAT*: It is an array of C elements, one for each channel. $CAT[k]=m$, where $1 \leq k \leq C$, means that channel k will be available after m time slots. It is needed to avoid channel collisions on data channels.
- *RAT*: It is an array of N elements, one for each node. $RAT[j]=n$, where $1 \leq j \leq N$, means that node j 's receiver will become free after n time slots. It is needed to avoid the receiver collisions.

Using the above two tables and considering the transmitter and receiver tuning times T , the round-trip propagation delay R and the length of the message to be transmitted l , the following distributed algorithm running at each node is performed upon receiving a control packet.

- Choose a data channel k associated with the minimum *CAT*-value, such that $CAT[k] \leq CAT[n]$, $\forall n \neq k$, $k \geq 1$, $n \leq C$. If there are multiple channels with the same earliest available time, choose the one with the smallest channel number.
- Calculate the earliest time that channel k is ready for transmitting next message:

$$t_1 = \max(CAT[k], T).$$
- Calculate the earliest time that the destination node j is ready to receive next message:

$$r = RAT[j] + T.$$
- Calculate the time after which the source node could send a data message without receiver collision: $t_2 = \max(t_1 + R, r).$
- Schedule the time for the message to be transmitted: $t = t_2 - R.$
- Update *CAT* and *RAT*: $CTA[k] = t_2 - R + l$ and $RAT[j] = t_2 + l.$

As a result, this algorithm has the ability to reduce the effect of the propagation delay for the signaling message by allowing the length of messages to be variable. In this way, a long message can be scheduled with a single control packet transmission, which can significantly reduce the overhead of control packet transmissions. Moreover, this algorithm can avoid the channel collision and destination conflict based on the global information *CAT* and *RAT*. This salient feature makes it a milestone in the development of MAC protocols for WDM optical networks.

3.2.1.2 Minimum Scheduling Latency (MSL)

In [54], the authors find that EATS may cause a large scheduling latency along the selected channel due to the way it selects data channels. EATS always selects the earliest available channel independent of destination availability so that the scheduling latency will become quite large when the message is destined to a busy node, in which scheduling latency is defined as the time difference between when the channel starts transmission and when the channel becomes available. The proposed algorithm, MSL, is developed by considering the channel availability and destination availability simultaneously. In this way, the scheduling latency could be reduced.

Similar to EATS, in order to avoid channel and receiver collisions, global information, *CAT* and *RAT*, is exchanged between the networks nodes and the control channel. Suppose there is one message destined to node j to be scheduled. And the definition of t_1 , t_2 , t and r are the same with those used in EATS. The following two steps present how MSL selects the channels:

- Among all the channels, select one subset S , in which each channel has the earliest transmission time such that the scheduled transmission time $t(s) = \min\{t(c)\}$, where $s \in S$, and $c \in \{1, 2, \dots, C\}$. Here $t(c)$ stands for the scheduled transmission time obtained if channel c is selected.

- If $t(s) = \max(CAT[k], T)$, there is no channel scheduling latency by using these channels. Then select the channel with earliest available time. Otherwise, channel scheduling latency exists and varies with different channels in S . Among the channels in S , select one that has the latest available time.

This algorithm is shown to work better than EATS, which shows that smaller scheduling latency can lead to the reduction of the average delay.

3.2.2 Message Sequencing Algorithms

Different from the above protocols, the protocols that address channel assignment as well as message sequencing aspect schedule the messages upon receiving the whole control frame. The message sequencing algorithm will impose an order to the messages represented in the control frame according to some mechanism. And then the channel assignment algorithm is used to determine the data channel and time duration over which the message is transmitted. In this way, the performance of the network could be further improved since more existing global information of the network and the transmitted messages can be exploited. Also, one other good point of this method is that the scheduling algorithm is invoked when all the nodes in the network have received control information about one batch of messages, resulting in the reduction of the frequency in needing the scheduling algorithm to be invoked. This reduction leads to lower scheduling overheads and permits more times for transceivers' tuning.

3.2.2.1 Shortest Message First (SMF)

In [38], a general scheduling scheme, which combines the message sequencing techniques with the channel assignment algorithms, is proposed to schedule variable-length message transmission. Among the proposed sequencing algorithm introduced, Shortest-Job-First (SJF), which we will call Shortest Message First (SMF) hereafter, is shown to be effective to reduce the message delay. SMF is a priority scheme which imposes a priority on the order in which the shorter messages are scheduled before the longer messages.

The SMF-EATS algorithm, which combines the sequencing technique, SMF, and the channel assignment technique, EATS, schedules the messages after having collected the information of one batch of messages which represent in one control frame. Upon receiving the control frame by all the nodes, the identical copy of the scheduling algorithms will be invoked by each node, which sequences the messages according to SMF algorithm and decides the channel to transmit individual messages on the basis of the updated status tables according to EATS algorithm. Suppose there are n messages in one batch of the messages, the algorithmic description of SMF-EATS is as follows:

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Sequence the n messages according to SMF algorithm to obtain the sequence $E (M_1, M_2, \dots M_n)$ such that $l_k \leq l_{k+1}$, for $k=1, 2, \dots, n-1$, where l denotes message length;

Start from the first message in the sequence E , where $i=1$;

Select the earliest available data channel k for message i ;

Calculate $r=\text{RAT}[j]+T$, $t_1=\max(\text{CAT}[k], T)$ and $t_2=\max(t_1+R, r)$;

Schedule the transmission time of the message at $t=t_2-R$;

Update $\text{RAT}[j]=t_2+l_i$, $\text{CAT}[k]=t_2-R+l_i$, and $i=i+1$,

If $i \neq n+1$, return to **Select**;

Else, jump to **End**;

End

3.2.2.2 Receiver-Oriented Earliest Available Time Scheduling (RO-EATS)

Another novel message sequencing technique, RO-EATS, is proposed in [39]. This new algorithm decides the sequence of the message transmission by the information of the receiver's states. The motivation of the algorithm comes from the observation that two

consecutive message with the same destination may not fully use the available channels when the EATS algorithm is used as the channel assignment technique. This is because when there are two consecutive messages going to the same destination node, the first message will occupy one of the channels and the receiver will be tuned to that channel. Since the receiver can only receive one message at one period of time, the second message has to wait until the first message has been successfully transmitted and received even there may be other channels available. In this situation, the second message is blocked and some channels cannot be used. As a result, the performance in terms of the average delay and the channel utilization will be degraded. Based on this observation, the RO-EATS algorithm is designed to prevent two consecutive messages from going to the same destination node.

This algorithm always checks the table of RAT to see which node is the least used as a destination and chooses the message which is destined to this node to transmit. The channel assignment algorithm is also EATS. Suppose that there are n messages to be scheduled, the RO-EATS algorithm can be expressed in details as follows:

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Set $h=1$ and $i=1$ initially;

Sort $RAT[j]$ in non-decreasing order to form a new array $RAT'[l]$ such that

$$RAT'[l-1] \leq RAT'[l], \text{ where } 1 \leq l \leq N;$$

Check the messages not scheduled one by one according to the node number to see whether there is any destined to j , where $RAT[j] = RAT'[h]$;

If yes, jump to **Select**;

Else, $h=h+1$, and return to **Check**;

Select the earliest available data channel k ;

Calculate $r = \text{RAT}[j] + T$, $t_1 = \max(\text{CAT}[k], T)$, and $t_2 = \max(t_1 + R, r)$;

Schedule the transmission time of the message at $t = t_2 - R$;

Update $\text{RAT}[j] = t_2 + l_i$, $\text{CAT}[k] = t_2 - R + l_i$, where l_i denotes the message length, and $i = i + 1$;

If $i \neq n + 1$, return to **Sort**;

Else, jump to **End**;

End

By using this algorithm, average message delay can be shown to be quite low and channel utilization can be shown to be high if the difference of the message lengths is small.

3.2.2.3 Minimum Laxity First (MLF)

Both SMF and RO-EATS are designed for non real-time messages; however, one of the key issues in designing next generation photonic access networks is the additional support of real-time services corresponding to time-sensitive applications. As a matter of fact, it appears to be essential to develop novel medium access control protocols which support multimedia traffic with tight delay constraints in a flexible and efficient way directly. In [46], a reservation-based MAC protocol for best effort real-time service is proposed. The scheduling algorithm of this protocol is based on the time related dynamic priority scheme, namely Minimum Laxity First (MLF) scheduling. In this work, each message is associated with a laxity which directly represents the time constraint of the message. The laxity can be expressed as the difference between the deadline and the time the message stayed in the system already. If the waiting time of a message in the system exceeds its laxity, the message is considered as lost. By scheduling messages with minimum laxity first, MLF algorithm is expected to schedule and transmit tight time-constraint message first in order to reduce the message loss rate.

Using the above status tables, CAT and RAT , and considering the transceiver tuning time T , propagation delay R , the length of the messages to be transmitted l , and the laxity of the messages L , the following distributed algorithm running at each node is performed upon

detection of a control frame contained n messages. The channel assignment algorithm used here is also EATS.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Sequence the n messages according to MLF algorithm to obtain the sequence $E (M_1, M_2, \dots M_n)$ such that $L_k \leq L_{k+1}$, for $k=1, 2, \dots, n-1$;

Start from the first message in the sequence E , where $i=1$;

Select the earliest available data channel k for message i ;

Calculate $r=\text{RAT}[j]+T$, $t_1=\max(\text{CAT}[k], T)$ and $t_2=\max(t_1+R, r)$;

Test whether the message laxity can be exceeded;

If $((t_2-R)>L_i)$, drop the message, and $i=i+1$;

Otherwise schedule the transmission time of the message at $t=t_2-R$ and update

$\text{RAT}[j]=t_2+L_i$, $\text{CAT}[k]=t_2-R+L_i$, and $i=i+1$,

If $i \neq n+1$, return to **Select**;

Else, jump to **End**;

End

3.3 Summary

Two channel assignment algorithms and three message sequencing algorithms for single-hop passive star-coupled WDM optical networks have been introduced in this chapter. These algorithms based on a PSC-based broadcast-and-select topology and a CC-FT/TT-FR/TR architecture can be viewed as representatives of reservation-based access protocols relying on a separate control channel.

EATS and MSL address the channel selection issue and have the ability to avoid channel collision and receiver collision. EATS always selects the earliest available channel, which

will cause a large scheduling latency when the message is destined to a busy node. And MSL selects the channel by taking channel availability and receiver availability into account so that it is capable to reduce the scheduling latency of EATS. These two algorithms can support variable length messages. However, they just address channel assignment aspect of the scheduling problems and ignore that the way to choose the order of the message transmission may affect the performance of the network. Moreover, neither of these two algorithms has considered the tuning overhead of transceivers, which is one of major limitations on packet scheduling in photonic switching. So, we have the motivation to develop algorithms that has taken the effect of tuning overhead into account, which is presented in Chapter 4.

SMF-EATS and RO-EATS address the transmission sequence as well as the channel assignment aspect. These two algorithms are shown to provide better transmission service than those schedule messages transmitted individually since they not only take use of the global information about each message among receiving and transmitting nodes, but also consider multiple messages from different transmitting nodes simultaneously. The SMF-EATS algorithm decides the message sequence according to the message length, while the RO-EATS algorithm sequences the messages based on the state of the receivers. These two algorithms are shown to improve the network performance in terms of message delay. However, they just focus on characteristics of the messages, regardless of the negative effect on the network performance caused by the special physical characteristics of WDM optical networks. In fact, there are three constraints on scheduling a message, which are channel availability, receiver availability and tuning overhead. In order to make efficient use of the channel resources and improve the network performance, these three factors should be considered simultaneously. Based on this observation, we propose two novel message sequencing algorithms that aim to reduce the average delay by taking the three factors into account. These two algorithms are presented in Chapter 4.

Moreover, future communication networks have to accommodate a large variety of applications with different QoS requirements like bandwidth and delay guarantees. Therefore, one of the key issues in designing next generation photonic access networks is the additional support of real-time services corresponding to time-sensitive applications. In this chapter, one novel medium access control protocols, namely MLF, which supports real-time traffic with tight delay constraints (associated with message laxity), is introduced. However, this scheme just considers the laxity, regardless of the effect of length. Therefore, one scheme with the aim to achieve better performance in terms of loss rate is proposed in Chapter 5. On the other hand, MLF is not able to support different QoS classes, which is essential for the future generation of communication networks. So we develop a novel scheduling algorithm to provide differentiated service to messages with different time constraints, which is also presented in Chapter 5. Meanwhile, based on the proposed scheme which is to provide differentiated services, a novel QoS service prediction scheme is proposed in Chapter 5, which provides the prediction whether the new QoS can be adapted to the current QoS.

Although many schemes have been proposed to improve the network performance in terms of message delay or loss rate, however, no scheme so far has considered the optimization issue of the scheduling. Hence, in Chapter 6, one optimization scheme based on the Mixed Integer Linear Programming (MILP) is proposed, which aims to achieve local optimization of the networks.

Chapter 4 . Proposed Solutions to Improve Network Performance

To date, many MAC protocols have been proposed for single-hop passive-star coupled WDM optical networks. The main goal of these protocols is to avoid channel collisions as well as receiver collisions in a packet-switched optical network environment. Additionally, to reduce the negative impact of transmitter/receiver tuning times is also an important task for the MAC protocols. In order to evaluate the performance of non real-time schedulers, message delay is a normally used metric. Although there are some schemes proposed so far to reduce the message delay, most of them have not considered the characteristic of WDM optical networks completely. In WDM networks, there are three constraints on scheduling a message, which are receiver availability, channel availability and transceivers' tuning overhead. In order for one message to transmit, the channel, the receiver and the transmitter should be ready simultaneously. Therefore, even if the channel is available, the message has to be delayed for some time to wait for the receiver and transmitter to become available. In this chapter, several novel MAC protocols for single-hop passive-star coupled WDM optical networks are proposed, which try to further elaborate the characteristics of the specified WDM optical networks.

This chapter is organized as follows. In section 4.1, three novel reservation-based channel assignment techniques will be presented and analyzed respectively. And in Section

4.2, two messages sequencing techniques with aim to reduce message delay from different points of view are proposed. The corresponding performance analyses are also presented. Section 4.3 shows the comparisons for the proposed algorithms. Finally, section 4.4 gives the summary of this chapter.

4.1 Channel Assignment Algorithms

Traffic scheduling plays an essential role in photonic switching system. As mentioned in Chapter 3, there are two fundamental aspects that the scheduling algorithm should efficiently solve, namely, channel assignment and message sequencing. The channel assignment algorithms address the problem of selecting an appropriate channel and a time slot on that channel to transmit a message. These scheduling algorithms schedule the messages individually and independently. In this section, we will present three novel channel assignment algorithms.

4.1.1 Latency Minimizing Scheduling (LMS)

As described in Chapter 3, EATS is an algorithm which has received a lot of attention as an effective algorithm for channel assignment in single-hop passive-star coupled WDM optical networks. However, EATS may cause a large scheduling latency along a selected channel. For instance, if a channel is selected to transmit a message destined to a busy node, the message may be delayed for some time due to the unavailability of the destination even if the channel has been available by that time. In this case, the scheduling latency becomes quite large. Here we define scheduling latency as the difference between the time when the channel starts transmission and the time when the channel becomes available. Based on this algorithm, an algorithm, named MSL, is proposed to reduce the scheduling latency of EATS. In this subsection, we will introduce a channel assignment algorithm, namely Latency Minimizing Scheduling (LMS), which is the revised version of the MSL algorithm.

4.1.1.1 Algorithmic Description

Same as MSL, LMS aims to reduce the schedule latency of EATS algorithm. Some global information is required by this algorithm. One is RAT for each node and the other is CAT for each channel. $RAT[j]=m$, where $j = 1 \cdots N$, means that node j 's receiver will become free after m time slots. $CAT[k]=n$, where $k = 1 \cdots C$, means that channel k will become available after n time slots. By the use of ranging asymmetric distances may be accommodated in the passive-star configuration. Based on this observation, we assume that the round-trip propagation delay for each message is R_j time slots which depends on the distance of message i 's destination node j and the passive star coupler. The transceiver's tuning time is T time slots. In addition, we need the following definition of some parameters.

$$r = RAT[j] + T \quad (4.1)$$

where r is the time when node j can be ready to receive the next message. Then,

$$\begin{aligned} t_1 &= \max(CAT[k], T) \\ t_2 &= \max(t_1 + R_j, r) \end{aligned} \quad (4.2)$$

where t_1 is the earliest available time that transmitting node can transmit on data channel k , and t_2 is the time that destination node's receiver should be ready to receive on data channel k . So, the scheduled transmission time of the message, which is also the waiting time of the message, can be expressed by

$$t = W_i = t_2 - R_j \quad (4.3)$$

The flow time of message i , which can be defined as the time the message stays in the system, can be obtained from

$$F_i = W_i + l_i \quad (4.4)$$

where l_i is the message transmitting time.

In order to show the relationships of the above scheduling parameters, Figure 4.1 gives a simple example when a message i destined to a busy node j is selected to be transmitted. From the figure, it is easy to see that CAT of each channel is larger than tuning time T ,

therefore, according to (4.2), we can have that t_1 equals to the corresponding channel available time no matter which channel is selected. From the relationships shown in the figure and according to (4.1) and (4.2), it is clear that the scheduled transmission time will equal to $(r-R_j)$ no matter which channel is used to transmit the message. And scheduling latency exists in this case and varies with different channels. According to EATS, channel 1 will be selected to transmit the message since it has the earliest available time. However, this way of selection may cause the largest scheduling latency. Contrarily, if channel 2 is selected as the transmission channel, the scheduling latency can be reduced to the least. From the figure, it is easy to find that the larger the scheduling latency, the larger the wasteful time duration on the channel.

Figure 4.1 A simple example of relationships among the scheduling parameters

According to above analysis, we can easily find that, if a message destined to a busy node is selected to transmit, the message needs to wait longer than the channel available time due to the unavailability of destination receiver. Thus the scheduling latency may exist. In our system, there are three factors that are related to the scheduling latency, namely channel available time, destination available time and transmitter tuning time. The problem of EATS is that it selects the channel according to channel available time regardless of the other two factors. Thus if a message is destined to a busy node, or the channel available time is much

smaller than tuning time, the latency will become large. However, if we choose a channel by taking the destination available time and the transmitter tuning overhead into account instead always selecting the earliest available channel, the latency can be reduced. MSL is such an algorithm which reduces the latency of EATS, but it just considers destination and channel available time regardless of tuning time. Based on this algorithm, we introduce a new channel selection algorithm LMS, which selects the channels by taking the three factors into account.

The LMS algorithm, for each message i destined to node j , works as follows:

- If there is such a channel subset $\{CAT[k]\}$ in which all the channels satisfy $[\max(CAT[k], T) + R_j] < (RAT[j] + T)$, namely $t_1 + R_j < r$, it will produce schedule latency. Thus we should select the largest one in $\{CAT[k]\}$ so as to reduce the latency.
- If there is no such subset, we can find that W_i depends on $t_1 = \max(CAT[k], T)$. Then we should consider other two situations: one is that if there is $\{CAT[k]\} < T$, which means $t_1 = T$, there will be latency exists because the unavailability of transmitter. Similarly, selecting the largest one can reduce the latency. The other situation is that if there is no $\{CAT[k]\} < T$, we should select the earliest available channel.

This algorithm can be summarized as follows, in which i denotes the message and j denotes message i 's corresponding destination node.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control packet returns which indicates N messages to be sent;

Start:

For ($i=1$; $i \leq N$; $i++$)

{

If there is a channel subset $\{CAT[k]\}$ in which $[\max(CAT[k], T) + R_j] \leq (RAT[j] + T)$

{select the maximum $CAT[n] \in CAT[k]$ and $t_2 = r$. If there are multiple channels with the maximum available time, choose the one with the smallest channel number;}

Else

{ if there is a channel subset $\{CAT[k]\} \leq T$

{select the maximum $CAT[n] \in CAT[k]$ and $t_2 = T + R_j$;}

else

{select the minimum $CAT[n]$ among all the data channels and $t_2 = CAT[n] + R_j$;}

Schedule the message transmission at $t_2 - R_j$;

Update $RAT[j] = t_2 + l_i$, $CAT[n] = t_2 - R_j + l_i$, where l_i is the message length;

}

End.

4.1.1.2 Performance Evaluation

In this subsection, the performance of LMS algorithm is evaluated by simulations. We also compare its performance with that of the EATS algorithm. The performance is measured by the average message delay, which is defined as the duration from the time a message is scheduled to the time the message is finished with transmission.

The default values of system parameters can be set as follows. The number of nodes N is 50 and the number of channels C is set to 4. The round-trip propagation delay R is assumed to be identical for all the nodes here, which equals to 100 time slots. Message lengths vary according to an Exponential distribution with a mean value as 20 time slots. Message arrivals at each source node comply with an independent and identical Poisson processes. The destination nodes are selected according to a uniform probability distribution. In this set of experiments, no message sequencing will be imposed to the messages.

Figure 4.2 depicts the average delay versus traffic load. In this experiment, the tuning time T is fixed to 20 time slots. It is easy to see that the delay increases for both of the

algorithms when the traffic load increases. From the figure, we can see that, in a low traffic load, the difference between the two algorithms is not significant. However, as the traffic load increases, LMS outperforms EAST significantly. This is expected since LMS efficiently solves the problem of scheduling latency of EATS by considering channel availability, destination availability and tuning overhead simultaneously. When the traffic load is equal to 1, the delay with LMS is only 76% of that with EATS. This is the major achievement that we expect to see by using LMS.

Figure 4.2 Impact of traffic load on the average delay of LMS

Figure 4.3 Impact of tuning time on the average delay of LMS

Figure 4.3 illustrates the average delay versus tuning time. In this experiment, the number of channels is set to 4 and the traffic load is equal to 0.7. It is clearly that LMS

algorithm has achieved a significant improvement in the average delay, particularly when the tuning time is large. For example, when tuning time is 50, the average delay with LMS is only 77% of that with EATS. This is because EATS will cause large scheduling latency along the data channels due to the way it selects the channels. And LMS can better coordinate channel availability, destination availability and tuning overhead, and, therefore, reduce the scheduling latency. As a result, LMS achieves lower delay in the comparison.

Figure 4.4 shows the average delay versus number of channels. In this experiment, the tuning time is set to 20 time slots and the traffic load is equal to 0.7. It is clearly that the LMS consistently demonstrates its superior performance to EATS in this experiment. This is because that LMS has the ability to reduce the scheduling latency of EATS. From the figure, it is easy to see that the delay with both algorithms drops down as the number of wavelength channels increases, and, eventually, flattens out when the number of channels becomes large ($C > 7$). Further increase in the number of channels does not induce any more delay reduction, which means that the number of data channels is no longer a bottleneck.

Figure 4.4 Impact of number of channels on average delay of LMS

4.1.2 Continuous Channel Scheduling (CCS)

One of the fundamental limitations on packet scheduling in photonic switching is the signaling overhead involved in packet scheduling, which consists of propagation delay for the signaling message and optical devices' tuning time. The impact of the propagation delay

and the tuning time depends on the relative magnitude with respect to the packet transmission time, which varies from system to system. In order to reduce the effect of the propagation delay for the signaling message, a novel algorithm in [36] is proposed by allowing the length of messages to be variable. In this way, a long message can be scheduled with a single control packet transmission, which can significantly reduce the overhead of control packet transmissions. However, the effect of the tuning time of the transceivers is unlikely to go away nowadays since the transceivers' capabilities are subject to both tuning range and tuning speed. Although many access schemes have been proposed, they either ignore the tuning overhead or rely on rapidly tunable transceivers to achieve efficiency. A careful access scheme design should mask the effect of tuning overhead to reduce the performance penalty as much as possible. So far, we have seen only two schemes [36] which are designed to reduce the tuning overhead. However, the scheduling could be much simpler and more effective when some mechanisms are incorporated into the algorithms to reduce the impact of tuning overhead by avoiding unnecessary transmitter tuning.

In this subsection, one simple mechanism, namely CCS, is proposed to reduce the impact of transmitter tuning overhead on the message delay. Moreover, this scheme has an ability to prevent channel collision as well as destination conflict and support the transmission of variable-length messages.

4.1.2.1 Algorithmic Description

In the single-hop passive star-coupled WDM optical networks, the time required to tune the transceivers cannot be negligible with respect to the packet transmission time. In fact, with the current available optical transceivers, the tuning latency can be much larger than the packet transmission time. The objective of our proposed algorithm is to avoid or reduce the tuning overhead of the transmitter of source nodes whenever possible.

In order to achieve it, the algorithms have to rely on some global information, which can be grouped into three tables.

The first two tables are CAT and RAT. The other table is Source Channel (SC), which is an array of N elements, each of which is for one source node. $SC[i]=h$, where $0 \leq i \leq N$ and $0 \leq h \leq C$, means that, currently, the source node i is connected to channel h , namely connected-channel. After each time slot, all the three tables are decremented by one.

In the EATS, the transmitter's tuning overhead (T) is always incurred for every message, i.e., the minimum time period for one node to wait for starting transmission is at least T , even if the channel and the receiver are idle. The basic idea behind the CCS algorithm is to avoid the tuning overhead of the transmitters by choosing the connected-channel of the source node to transmit the messages.

Let $t_1 = \max(CAT[k], T)$, where k is the earliest available channel and t_1 is the earliest time that node i can transmit on data channel k under EATS. Also, let $t_c = CAT[SC[i]]$, where $SC[i]$ is the connected-channel of source node i and t_c is the earliest time that node i can transmit on data channel $SC[i]$. If we choose channel $SC[i]$ for message transmission, node i will start the transmission at t_c .

If $CAT[SC[i]] \leq T$, t_c will be smaller than t_1 as in EATS, which means that source node can start transmission earlier under CCS algorithm because no tuning is needed. Therefore in this case, it is beneficial to use data channel $SC[i]$ for message transmission. Then, node i 's transmitter tuning overhead can be avoided. If $CAT[SC[i]] > T$, CCS will schedule the message transmission on data channel k as under EATS since using continuous channel will not help to reduce the waiting time. In order to show the superior performance of CCS to EATS, only the case that $CAT[SC[i]] \leq T$ is needed to be considered since these two algorithms will achieve the same performance when $CAT[SC[i]] > T$.

Define $r = RAT[j] + T$, where r is the time when destination node j can be ready to receive the next message. If $(t_1 + R_j) \leq r$, the time to start transmission under EATS will equal to $(r - R_j)$, namely $t_{\text{EATS}} = (r - R_j)$. Since $CAT[SC[i]] \leq T$, we can get $CAT[SC[i]] + R_j \leq t_1 + R_j \leq r$. Hence the time to start transmission under CCS, namely t_{CCS} , equals to $(r - R_j)$, too. In this case, the time to start transmission of CCS is the same as that of EATS. However, since k is the

earliest available channel, we have $CAT[k] \leq CAT[SC[i]]$, by which the channel with larger available time can be used first by CCS. As a result, the waiting time for the following messages could be reduced resulting in total delay reduction finally.

If $(t_1 + R_j) > r$, the time to start transmission under EATS will equal to t_1 , namely $t_{\text{EATS}} = t_1$. There will be two cases under CCS.

- a. If $CAT[SC[i]] + R_j \leq r$, the time to start transmission under CCS will equal to $(r - R_j)$, namely $t_{\text{CCS}} = (r - R_j)$. Since $(t_1 + R_j) > r$, it is easy to get that $t_{\text{CCS}} = (r - R_j) < t_1 = t_{\text{EATS}}$.
- b. If $CAT[SC[i]] + R_j > r$, the time to start transmission under CCS will equal to $CAT[SC[i]]$.

Since $CAT[SC[i]] \leq T$, we can get $CAT[SC[i]] \leq t_1 = \max(CAT[k], T)$. Therefore, the time to start transmission of CCS is less than that of EATS.

It is proved from above derivation that CCS can achieve better performance in terms of average message delay than EATS.

This algorithm can be summarized as follows, in which i denotes the message and j denotes message i 's corresponding destination node.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control packet returns;

Start:

For ($i=1$; $i \leq N$; $i++$)

{ **if** ($CAT[SC[i]] \leq T$)

{select the connected-channel $SC[i]$;

calculate $r = RAT[j] + T$, $t_1 = CAT[SC[i]]$, $t_2 = \max(t_1 + R_j, r)$;

schedule the message transmission at $t_2 - R_j$;

update $RAT[j] = t_2 + l$, $CAT[SC[i]] = t_2 - R_j + l$, where l denotes the message length }

else

{select the earliest available data channel k ;

calculate $r = RAT[j] + T$, $t_1 = \max(CAT[k], T)$, $t_2 = \max(t_1 + R_j, r)$;


```

    schedule the message transmission at  $t_2 - R_j$ ;

    update  $RAT[j] = t_2 + l$ ,  $CAT[k] = t_2 - R_j + l$ ,  $SC[i] = k$ ;
}

```

End.

4.1.2.2 Performance Evaluation

In this subsection, we carry out the performance study of the proposed channel selection algorithm CCS. We also compare it with the EATS algorithm by extensive simulation experiments.

The parameters involved in the system design include the number of nodes (N), which is set to 50 and round-trip propagation delay (R), which is assumed to be identical for all the nodes and equals to 100 time slots. Message lengths vary according to an Exponential distribution with a mean value as 20 time slots. Message arrivals at each source node comply with independent and identical Poisson processes. The destination nodes are selected according to a uniform probability distribution. To evaluate the performance of the network, average message delay is adopted as one major metric.

Figure 4.5 Impact of traffic load on average delay of CCS

Figure 4.5 compares the average delay using two algorithms under varying traffic loads in a system. In this set of experiments, we assume that the number of channels C is set to 4 and the tuning time T is 20 time slots. Since the propagation delay R is always a part of the

message delay in all of the algorithms, it is not included in the message delay. Thus the delay incurred by queuing, tuning time overhead, and message transmission time will be our focus. From the figure, we can see that in the light traffic load the difference between the two algorithms is significant. This is because that CCS is designed to reduce the penalty of tuning overhead, which leads to the reduction of delay. When the traffic load is heavy, the node has enough time to tune its transmitter because the time to wait for starting transmission can be longer than the transmitter's tuning time. Therefore, CCS cannot achieve significantly better performance since no tuning overhead induced.

Figure 4.6 compares the characteristic of average delay under varying tuning time. In this set of experiments, the number of channels C is fixed to 4 and the traffic load is set to 0.7. It is easy to see that the delay increases with the tuning time for these two algorithms. This is because that when the tuning time increases, the source nodes take longer time to complete a message transmission. When tuning time equals to 0, the average delay of CCS equals to that of EATS since there is no tuning overhead occurred. However, when the tuning time increases, CCS consistently demonstrates its superior performance to EATS in terms of average message delay. This is because that CCS has the ability to reduce or avoid the tuning overhead.

Figure 4.6 Impact of tuning time on average delay of CCS

Figure 4.7 shows the effect of varying number of channels on the average delay for two algorithms. We assume that the tuning time T is fixed to 20 time slots and traffic load is set to 0.7. It is clearly that the delay decreases for both of the algorithms when the number of channels increases. Once again, we observe that CCS achieves better performance than EATS in terms of delay. CCS aims to reduce or avoid the tuning overhead, so it has the lower delay. When the number of channels becomes larger, the performance of all the algorithms will flatten out. And further increase in the number of channels will not induce any change, which means that the number of data channels is no longer a bottleneck.

Figure 4.7 Impact of number of channels on average delay of CCS

4.1.3 Continuous Channel-Minimum Scheduling Latency (CC-MSL)

As addressed above, scheduling latency and tuning overhead of the transceivers are two important issues concerning the channel assignment algorithms. The large scheduling latency in former algorithms is due to the way that they select the data channels. MSL and LSM are two algorithms with the aim to reduce the scheduling latency. The logic behind them is that they select the channels by considering the system constraints. On the other hand, in order to reduce the tuning overhead, CCS is proposed. The basic idea of CCS is to avoid the tuning of the transmitters at the source node by choosing the connected-channel of the source node for its transmitter to transmit the messages. Based on CCS and MSL

algorithm, we introduce a new channel selection algorithm, called CC-MSL, in this subsection. CC-MSL aims to reduce scheduling latency as well as the tuning overhead by combining the advantages of CCS and MSL algorithms.

4.1.3.1 Algorithmic Description

If a message is scheduled to a busy node, the message needs to wait longer than the channel available time due to the unavailability of the destination node. Thus the scheduling latency may exist. As mentioned above, in our network, there are three factors that may incur the scheduling latency, namely channel available time, destination available time and transceiver tuning time. The problem of EATS is that it selects the channel only according to channel available time regardless of other two factors. Hence, if a message is destined to a busy node or the channel available time is much less than the tuning time, the latency could be larger. However, if we choose a channel by taking the destination available time and the transmitter tuning overhead into account, the latency can be reduced. MSL is an algorithm to reduce the scheduling latency of EATS. But it just considers destination and channel available time regardless of the tuning overhead. Based on this algorithm, we introduce a new algorithm CC-MSL, which considers all these three factors. Moreover, it has the capability to reduce or avoid the tuning overhead of the transmitters as CCS does.

With the same global information as that used in CCS algorithm, the CC-MSL algorithm works as follows. For each message i with the destination available time $RAT[j]$:

- If there is such a subset of channels $\{CAT[n]\}$, in which all the channels satisfy $[\max(CAT[n], T) + R_j] \leq (RAT[j] + T)$, namely $t_1 + R_j \leq r$, it will produce scheduling latency. Thus we should select the largest one in $\{CAT[n]\}$ so as to reduce the latency. If there are multiple channels with the largest channel available time in $\{CAT[n]\}$, choose the one with the smallest channel number.
- If there is no such subset, we can find that the time to start transmission depends on $t_1 = \max(CAT[n], T)$. Then we should consider other two situations. One is that if there is a

subset of channels $\{CAT[n]\} \leq T$, which means $t_1=T$, scheduling latency may exist because the unavailability of transmitter. If $CAT[SC[i]] \leq T$, select the connected-channel $SC[i]$ to avoid the tuning overhead. Otherwise, select the largest one in $\{CAT[n]\}$ to reduce the scheduling latency. The other situation is that there is no such subset $\{CAT[n]\} \leq T$, we should select the earliest available channel k .

Algorithmic description of CC-MSL is as follows, in which i denotes the message and j denotes message i 's corresponding destination node.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control packet returns;

Start:

For ($i=1$; $i \leq N$; $i++$)

{

Calculate $r=RAT[j]+T$;

Select the channel according to CC-MSL:

if (there is a subset of channels $\{CAT[n]\}$ in which all the channels satisfy

$[\max(CAT[n], T)+R_j] \leq r$)

{select the maximum one in $\{CAT[n]\}$, namely $CAT[h]$, and the time to start transmission $t=r-R_j$;

update $RAT[j]=t+R_j+l$, $CAT[h]=t+l$, $SC[i]=h$;}
else

{**if** there is a subset of channels $\{CAT[n]\} \leq T$

{**if** ($CAT[SC[i]] \leq T$)

{select the connected-channel $SC[i]$ and the time to start transmission

$t=CAT[SC[i]]$;

update $RAT[j]=t+R_j+l$, $CAT[SC[i]]=t+l$;}
else

```

        {select the maximum one in  $\{CAT[n]\}$ , say  $CAT[h]$ , and the time to
        start transmission  $t=T$ ;
        update  $RAT[j]=t+R_j+l$ ,  $CAT[h]=t+l$ ,  $SC[i]=h$ ;}
    }
else
    {select the earliest available channel  $k$  among all the channels and the
    time to start transmission  $t=CAT[k]$ ;
    update  $RAT[j]=t+R_j+l$ ,  $CAT[k]=t+l$ ,  $SC[i]=k$ ;}
}
}

```

End.

4.1.3.2 Performance Evaluation

In this subsection, we study the performance of the proposed scheduling technique CC-MSL by comparing it with MSL and CCS algorithms by extensive simulation experiments. The following paragraphs present the design of these experiments and their results.

The parameters involved in the system design include the number of nodes N that is set to 50, the number of data channels C that is set to 4, and round-trip propagation delay R that is identical for all the nodes and equal to 100 time slots. Message lengths vary according to an Exponential distribution with a mean of 20 time slots. Message arrivals at each source node comply with an independent and identical Poisson processes. The destination nodes are selected according to a uniform probability distribution. And the performance metric is the average message delay.

Figure 4.8 depicts the average message delay using three algorithms under different traffic loads. In this experiment, the tuning time T is fixed to 20 time slots. It is easy to see that CC-MSL significantly outperforms MSL, especially, when the traffic load is light. This is expected since CC-MSL is designed to reduce the penalty of tuning overhead as well as

scheduling latency resulting in delay reduction. When the traffic load is heavy, the time to wait for starting transmission can be longer than the transmitter's tuning time, so the node will have enough time to tune its transmitter. As a result, CCS-MSL cannot achieve significantly better performance than MSL since no tuning overhead is induced. The figure also shows that smaller scheduling latency can lead to the reduction of the average message delay. It is demonstrated from the fact that CC-MSL works better than CCS under heavy traffic load.

Figure 4.8 Impact of traffic load on average delay of CC-MSL

Figure 4.9 Impact of tuning time on average delay of CC-MSL

Figure 4.9 illustrates the average delay versus the tuning time. In this experiment, the channel number C is fixed to 4 and the traffic load is set to 0.7. It is easy to see that the delay

increases with the tuning time for all the algorithms. This is because that when the tuning time is increased, the source nodes have to wait longer time for transmitters tuned to the right channels. When tuning time equals to 0, the average delay of CC-MSL equals to that of MSL since there is no tuning overhead occurred. However, when the tuning time increases, both CCS and CC-MSL consistently demonstrate their superior performance to MSL. This is because that CCS and CC-MSL algorithms have the capability to reduce or avoid the tuning overhead. Among the three algorithms, CC-MSL can achieve the best performance in terms of average message delay. For example, when the tuning time is 50, the delay with CC-MSL is only 70% of that with MSL. This is because that CC-MSL is designed to reduce the tuning overhead as well as the scheduling latency.

Figure 4.10 Impact of number of channels on average delay of CC-MSL

Figure 4.10 shows the average delay versus number of channels. In this experiment, we assume that the tuning time T is 20 time slots and the traffic load is 0.7. We clearly see that the delay decreases for all of the algorithms when the number of channels increases. Once again, we observe that CCS and CC-MSL achieve better performance than MSL algorithms. CCS and CC-MSL aim to reduce or avoid the tuning overhead, so they have the lower delay. When the number of channels becomes larger, the performance of all the algorithms will flatten out. And further increase in the number of channels will not induce any change, which means that the number of data channels is no longer a bottleneck. Once again, we

observe that the CC-MSL algorithm achieves the lowest delay due to its combination of the advantages of CCS and MSL.

4.1.4 Summary

In this section, we have proposed three novel channel selection schemes for single-hop passive-star coupled WDM optical networks, namely LMS, CCS and CC-MSL. LMS is introduced to minimize the scheduling latency by taking channel availability, destination availability and tuning overhead into consideration. The small scheduling latency leads to the reduction of the average message delay, which is shown by the numerical studies. On the other hand, with the additional information SC, the CCS and the CC-MSL algorithms can reduce the impact of transmitter tuning overhead by avoiding unnecessary tuning of the transmitters. The results of the numerical studies show that it is beneficial to schedule the transmissions on the same data channel as the one which the transmitter has been tuned to. Hereby it avoids transmitter tuning overhead. Among those algorithms in comparison, CC-MSL is introduced not only to reduce the transmitter tuning overhead but also to minimize the scheduling latency by taking the channel available time, the receiver available time and the tuning time into account. Hence it achieves better performance than CCS and MSL.

4.2 Message Sequencing Algorithms

The channel assignment algorithms have ignored that the way to choose the order of the message transmission may affect the performance of the network. However, the performance of the network could be further improved by the way of considering both channel assignment and message sequencing. In this section, we will present two novel message sequencing algorithms, which can be easily combined with one channel assignment algorithm to form the new algorithms.

4.2.1 Minimum Flow Time Scheduling (MFTS)

To the best of our knowledge, only a few papers [38][39] have addressed the issue of sequencing messages in the WDM networks. Based on the protocols in [38], the order of transmission is determined by the message length. And the protocol in [39] decides the sequence of the message transmission by the information of the receiver's states. These algorithms have been shown to improve the network performance of a WDM network. However, all these algorithms just focus on characteristics of the messages. They have not considered the whole flow time of the messages in a network, which can be affected by the network environment. Flow time is defined as a time period that a message exists in a network, which consists of message waiting time and message transmission time. Further, the message waiting time depends on tuning overhead of transceivers and the run-time availability of destination node and transmission channel. In this subsection, we propose a scheduling algorithm that determines the order of message transmissions based on the flow time of the messages, which is called Minimum Flow Time Scheduling (MFTS). This algorithm works by taking not only the message length but also the availability of the channels and the receivers into account. In this way, the algorithm can coordinate the transmission and reception of the messages efficiently while minimizing the average message delay of the network.

4.2.1.1 Algorithmic Description

MFTS is designed to reduce mean message flow time of the system. In the network, for a particular processing sequence, the mean flow time of message transmission can be

$$\begin{aligned}\bar{F} &= \frac{1}{n} \sum_{k=1}^n F_{ik} = \frac{1}{n} \sum_{k=1}^n (W_{ik} + l_{ik}) \\ &= \frac{1}{n} \sum_{k=1}^n W_{ik} + \frac{1}{n} \sum_{k=1}^n l_{ik}\end{aligned}$$

where i denotes the sequence and k denotes the position of the message in the sequence, and l_{ik} and W_{ik} denote the length and the waiting time of the message which is queued in the k

position of sequence i respectively. $\sum_{k=1}^n l_{ik}$, which does not change no matter how the messages are queued, denotes the sum length for all the messages in the queue. Hence in order to minimize $\bar{F}$, we must minimize $\sum_{k=1}^n W_{ik}$.

In order to minimize the mean flow time of the whole system, we have the following cases to consider:

$W_{i1}=0$ for all sequence since message M_{i1} , which is the message queued in the first place of sequence i , can be processed without waiting for other messages.

$W_{i2}=F_{i1}$ since M_{i2} must wait only for M_{i1} to be processed. Thus if we choose M_{i1} to have the shortest flow time $F_{i1}=W_{i1}+l_{i1}$ of all the messages in the list $\{M_1, M_2, \dots, M_n\}$, we shall minimize W_{i2} .

$W_{i3}=F_{i1}+F_{i2}$ since M_{i3} must wait only for M_{i1} and M_{i2} to be processed. In order to minimize W_{i3} , we should choose M_{i1} and M_{i2} to have the shortest and next shortest flow time from the list $\{M_1, M_2, \dots, M_n\}$. If we let M_{i1} still have the shortest, then we do not affect our minimization of W_{i2} . Therefore we can minimize W_{i2} and W_{i3} simultaneously.

Continuing in this way, we can build up a schedule by which the k th message has the shortest flow time among those remaining. Simultaneously, the total waiting time $\sum_{k=1}^n W_{ik}$ can be minimized. Thus minimize $\bar{F}$.

Based on the above consideration, we have an idea to generate a new message sequencing algorithm MFTS, which can reduce the mean flow time of the system significantly. In our proposed algorithm, a new priority assignment scheme is introduced, in which we consider not only the message transmission time but also the message waiting time. We have the following formula

$$F_i = W_i + l_i$$

to assign the priority to the messages that are going to be transmitted. By this priority scheme, the message with the least value of F_i will be considered as the message with the

highest priority. There is one point we need to address here is that the priority is not static. After we select one with the highest priority, the waiting time of the remainders will be calculated based on the updated global information, and then we select the new one with the least flow time again among the rest of the messages. In this way we can assure that the flow time of the system can be kept low all the time.

4.2.1.2 MFTS-LMS Algorithm

In this subsection, we combine the proposed message sequencing technique with the LMS algorithm to form the new algorithm. We describe it formally as the following.

We assume that there are N nodes and C data channels in the network. The propagation delay is R_j and the transceivers' tuning time is T . For simplicity, we will use D_j to stand for $RAT[j]$, which is for each node. $D_j = m$, where $j = 1 \cdots N$, means that node j 's receiver will become free after m time slots. And we will use C_k to stand for $CAT[k]$, which is for each channel. $C_k = n$, where $k = 1 \cdots C$, means that channel k will become available after n time slots. The messages can be transmitted from source node i to destination node j , where $i \neq j$, and $i, j \in N$.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Calculate the flow time of every message i destined to node j not scheduled:

Step 1) Select the channel according to the new channel selection algorithm:

- a. If there is a channel subset $\{C_k\}$ in which $[\max(C_k, T) + R_j] \leq (D_j + T)$, select the maximum $C_i \in \{C_k\}$.
- b. If there is no such $\{C_k\}$ that makes $[\max(C_k, T) + R_j] \leq (D_j + T)$, then
 - b.1 If there is $\{C_k\} \leq T$, select the maximum $C_i \in \{C_k\}$.

b.2 Else, select the minimum C_i among all the channels.

Step 2) Calculate $t_1 = \max(C_k, T)$, $t_2 = \max(t_1 + R_j, r)$, and obtain the time to start transmission $t = t_2 - R_j$. So the flow time of every message waiting to be scheduled is $F_i = t + l_i$, where l_i denotes the message length;

Choose the message with the least flow time F_i ;

Use the selected channel to transmit the selected message and schedule the transmission time at t ;

Update $D_j = t_2 + l_i$, $C_k = t_2 - R_j + l_i$;

Return to Start if any messages not scheduled;

End

The complexity of MFTS-LMS can be evaluated based on its operations. It has two sequencing procedures. One of the sequence procedures is to sort the messages according to their flow time. The other is to sort the C_k table. The number of the messages could be the number of nodes in the network in the worst case, and the number of C_k is the number of channels in the network. Let us assume that the number of nodes (N) is always larger than the number of channels (C). Hence we consider only the number of nodes when we estimate the complexity of the sorting algorithm. The complexity of a typical sorting algorithm is $O(N \log_2 N)$, where N can be mapped to the number of nodes in our case. Given that the worst case running time of the algorithm is two rounds of the sequence procedure, the complexity of the entire algorithm should be $O(N \log_2 N)$.

The bandwidth of the transmission link is defined as the number of bits transmitted per unit time. Since in our network, we assume that one packet can be transmitted in one time slot, the network bandwidth can be expressed as a function of the number of packets transmitted. The scheduling algorithm will not cost in terms of bandwidth as long as the scheduling procedure can be completed while messages in one control frame is being transmitted. The transmission time of all messages of one control frame can be

approximately by $m*N$, where N is the number of nodes in the network, and m is the mean message length. Therefore, the condition that our scheduling algorithm will not produce cost in terms of bandwidth can be formulated as $2*N \log_2 N \leq m*N$. This clearly shows that the mean message length can be a system design parameter. When it is large enough, our scheduling algorithm will not introduce any cost to the message transmission in the network.

4.2.1.3 Example

In order to illustrate the new algorithms effectively, we present an example in this subsection and compare its performance with that of SMF algorithm. The current global information is given in Table 4.1. And the information of the messages is given in Table 4.2, which contains five messages $M_1, M_2, \dots, M_5$, destined to nodes $j_1, j_2, \dots, j_5$ respectively. In this example, the number of data channel (C) is equal to 4, tuning time of the transceiver (T) is 2 time slots, and the propagation delay (R) is identical for all the nodes, which is equal to 4 time slots.

$C_1=5, C_2=13, C_3=15, C_4=0$
$D_{j1}=9, D_{j2}=20, D_{j3}=10, D_{j4}=10, D_{j5}=0$
$T=2, R=4$

Table 4.1 Current global information in the example of MFTS

Messages	M_1	M_2	M_3	M_4	M_5
Destination node j	j_1	j_2	j_3	j_4	j_5
Packet length L	7	9	10	11	13

Table 4.2 Information of messages in the example of MFTS

Messages	M_1	M_2	M_3	M_4	M_5
Destination node j	j_1	j_2	j_3	j_4	j_5
D_{jk}	9	20	10	10	0
Packet length L	7	9	10	11	13
Selected Channel	C_4	C_1	C_2	C_4	C_3
W_i	7	18	13	14	15
F_i	14	27	23	25	28
C_k (updated)	14	27	23	25	28
D_{jk} (updated)	18	31	27	29	32
$\bar{F}$	$(14+27+23+25+28)/5=23.4$				

Table 4.3 Results of message scheduling using SMF-EATS

We start our discussion by observing the behavior of SMF algorithm in this example. According to SMF algorithm, messages $M_1, M_2, \dots, M_5$ are sorted to M_1, M_2, M_3, M_4, M_5 , which is shown in Table 4.3. The channel selection algorithm used in this example is EATS. Firstly, running of the EATS algorithm will assign M_1 to channel 4. From Table 4.2, the destination available time for M_1 , namely D_{j1} , equals to 9. Comparing $(D_{j1}+T)$ with (C_4+R) , easily observed that the waiting time of the message M_1 is dependent on the destination available time and equal to 7 time slots according to (4.2) and (4.3). The flow time of the message M_1 is 14 time slots (7 time slots for waiting plus 7 time slots for transmission). After scheduling, the global information should be updated ($C_4=14$ and $D_{j1}=18$). At the current time, channel 1 becomes the earliest available channel. Message M_2 is then assigned to it. According to (4.2) and (4.3), the waiting time of message M_2 is also dependent on the destination available time and equal to 18 time slots. Thus the flow time of M_2 is 27 time slots (18 time slots for waiting plus 9 time slots for transmission). Similarly, the global information should be updated ($C_1=27$ and $D_{j2}=31$). The flow times of the rest of the messages can be calculated by following the same way. And the final result is shown in Table 4.3. The average flow times of these five messages can be calculated as: $(14+27+23+25+28)/5=23.4$.

Messages	M_1	M_3	M_4	M_2	M_5
Destination node j	j_1	j_3	j_4	j_2	j_5
D_{jk}	9	10	10	20	0
Packet length L	7	10	11	9	13
Selected Channel	C_4	C_1	C_2	C_4	C_3
W_i	7	8	13	18	15
F_i	14	18	24	27	28
C_k (updated)	14	18	24	27	28
D_{jk} (updated)	18	22	28	31	32
$\bar{F}$	$(14+18+24+27+28)/5=22.2$				

Table 4.4 Results of message scheduling using MFTS-EATS

Next we apply our new algorithm MFTS combining with EATS to our example. Firstly, we calculate the flow times of all these five messages based on the global information which is shown in Table 4.1. Currently, channel 4 has the earliest available time. According to

MFTS algorithm, we have to calculate the flow time of each message and then assign the highest priority to the one with the least flow time. For message M_1 , the destination available time (D_{j1}) is equal to 9. Comparing ($D_{j1}+T$) with (C_4+R), we can have that the waiting time for M_1 is equal to 7 time slots according to (4.2) and (4.3). Then M_1 's flow time is equal to 14 time slots (7 time slots for waiting plus 7 time slots for transmission). For M_2 , the destination available time (D_{j2}) is 20. So M_2 's waiting time is equal to 18 time slots. And the flow time of M_2 is 27 time slots (18 time slots for waiting plus 9 time slots for transmission). Following the same way, we can get the flow times of all the five messages at the first round of sorting, which are 14 (M_1), 27 (M_2), 18 (M_3), 19 (M_4) and 15 (M_5). Then M_1 will be the one with the least flow time and it will be scheduled first. After scheduling, the global information should be updated ($C_4=14$ and $D_{j1}=18$). Then, based on the new global information, we can calculate the flow times of the rest of the four messages. At this time every message will select channel 1 for transmission due to its earliest available time. Following the method as above, we can get the flow times of these four messages, which are 27 (M_2), 18 (M_3), 19 (M_4) and 18 (M_5). In this case, both M_3 and M_5 have the least flow time. We can select M_3 in the second round of sorting and assign the higher priority to it. By following this way, we finally sort the five messages to M_1, M_3, M_4, M_2, M_5 . The average flow time can be acquired from $(14+18+24+27+28)/5=22.2$. Until now we can find that MFTS can achieve better performance in terms of mean flow time than SMF. One feature of MFTS is that it doesn't assign priority at the beginning of scheduling. Instead it assigns the priority during the scheduling process.

At last, we discuss the performance result of MFTS-LMS algorithm with this example. In the first round of sorting, the respective flow time of every message will be calculated. For example, for M_1 , there are channel 1 and channel 4 that satisfy $[\max(C_k, T) + R] \leq (D_{j1} + T)$, so we will select channel 1 according to LMS. After that, according to (4.3) and (4.4), we will calculate the flow time of M_1 , which equals to 14

time slots. While for M_2 , all of the four channels satisfy $[\max(C_k, T) + R] \leq (D_{j_2} + T)$. Then in order to reduce the scheduling latency, channel 3 with the largest channel available time will be selected. And the flow time of the message is 27 time slots. By following the same way, we get the flow times of five messages that are 14 (M_1), 27 (M_2), 18 (M_3), 19 (M_4) and 15 (M_5) respectively in the first round of sorting. Thus the message M_1 with the least flow time will be assigned with the highest priority. After scheduling M_1 , we start the second round of calculation on the rest of the messages based on the updated global information. Similarly, the flow times of the remaining messages can be obtained, which are 27(M_2), 18 (M_3), 19 (M_4) and 15 (M_5). Therefore, at this round, the message M_5 with the least flow time will be selected. According to LMS, the channel 4 with the least channel available time will be assigned to this message. Finally, we can sort the five messages to M_1, M_5, M_3, M_4, M_2 . The final result is listed in Table 4.5. From the table, the average flow time can be easily calculated from $(14+15+23+25+27)/5=20.8$. Comparing this result with the result of MFTS-EATS above, we can find that LMS algorithm works better than EATS.

Messages	M_1	M_5	M_3	M_4	M_2
Destination node j	j_1	j_5	j_3	j_4	j_2
D_{jk}	9	0	10	10	20
Packet length L	7	13	10	11	9
Selected Channel	C_1	C_4	C_2	C_1	C_3
W_i	7	2	13	14	18
F_i (updated channel)	14	15	23	25	27
C_k (updated)	14	15	23	25	27
D_{jk} (updated)	18	19	27	29	31
$\bar{F}$	$(14+15+23+25+27)/5=20.8$				

Table 4.5 Results of message scheduling using MFTS-LMS

4.2.1.4 Performance Analysis

In this subsection, we will carry out the performance evaluation of MFTS algorithm by both mathematical analysis and experimental results.

4.2.1.4.1 Theoretical Analysis

In this subsection, we aim to compare the performance of MFTS and SMF in terms of mean flow time by adjacent pairwise interchange [64][65]. We consider an SMF sequence in which there must be a pair of adjacent messages M_1 and M_2 , with M_1 preceding M_2 due to $l_1 \leq l_2$, where l is the transmission time of the message, namely the message length. Then, we construct a new sequence, S' , in such a way that only messages M_1 and M_2 are sorted according to MFTS algorithm with other messages in S unaltered. We aim to prove that a strict improvement of performance can be achieved by the new sorting method. If we can show that the performance of the SMF sequence can be improved with respect to mean flow time by sorting an adjacent pair of messages in the way of MFTS algorithm, we can continue the sorting until eventually the MFTS sequence is constructed. The important part of our comparison is to prove the new sequence S' can achieve smaller mean flow time than original sequence S .

Figure 4.11 Sorting of adjacent messages

All the situations are depicted in Figure 4.11. We assume that in sequence S , M_1 precedes M_2 . And in this figure, we assume that after sorting according to MFTS, M_2 precedes M_1 . In Figure 4.11, A denotes the set of messages that precede messages M_1 and M_2 in both sequences, and B denotes the set of messages that follow M_1 and M_2 in both sequences. As we have mentioned above, messages of A and B will be left unchanged after the new sorting. C_k denotes the channel available time at which message M_1 begins to hold the channel in S

and at which message M_2 begins to hold the channel in S' . In the sequence S , $F_1(S)$ is the flow time of message M_1 . $F_1(S)$ also equals to the channel available time at which message M_2 begins to hold the channel in S , which is denoted by C'_k . The flow time of message M_2 in S is denoted by $F_2(S)$. Similarly, $F_1(S')$ is the flow time of message M_1 and $F_2(S')$ is the flow time of message M_2 in sequence S' .

We temporarily adopt the notation $F_k(S)$ to represent the flow time of message k under schedule S and $F_k(S')$ to represent the flow time of message k under schedule S' . Then

$$\sum_{k=1}^n F_k(S) = \sum_{k \in A} F_k(S) + F_1(S) + F_2(S) + \sum_{k \in B} F_k(S)$$

$$\sum_{k=1}^n F_k(S') = \sum_{k \in A} F_k(S') + F_2(S') + F_1(S') + \sum_{k \in B} F_k(S')$$

And because $\sum_{k \in A} F_k(S) = \sum_{k \in A} F_k(S')$ and $\sum_{k \in B} F_k(S) = \sum_{k \in B} F_k(S')$, we can get

$$\sum_{k=1}^N F_k(S) - \sum_{k=1}^N F_k(S') = F_1(S) + F_2(S) - F_2(S') - F_1(S')$$

Therefore we need to prove $F_1(S) + F_2(S) - F_2(S') - F_1(S') = F_S - F'_S \geq 0$.

Proof:

In sequence S , since $l_1 \leq l_2$, M_1 will be served before M_2 .

In sequence S' , the message with less flow time will be served first, so we need to compare the flow time of these two messages.

- i) If $F_1 \leq F_2$, M_1 will be served before M_2 , too. Therefore, MFTS can get the same sequence as SMF does. And the mean flow time of S' is the same as that of S .
- ii) If $F_1 > F_2$, according to MFTS, M_2 will precede M_1 . The situation is same with S' shown in Figure 4.11.

In order to prove that $F_S - F'_S \geq 0$ when $F_1 > F_2$, we make the following assumptions: the number of channels is 1; the receiver available times for M_1 and M_2 are D_1 and D_2

respectively; the tuning time T equals to 0; the propagation delay is assumed to be identical for the nodes, which is denoted by R . And we have the conditions that $l_1 \leq l_2$ and $F_1 > F_2$.

To obtain the flow time of these two messages, let's look at the relationship between $C_i + R$ and D_j , where C_i denotes the channel available time of the selected channel I and D_j denotes the receiver available time of destination node j . According to (4.2) and (4.3), we can easily get that if $C_i + R > D_j$, the flow time depends on channel available time C_i , otherwise, the flow time is determined by destination available time D_j . Based on the above assumptions, we have following properties that:

A. It is impossible for $C_k + R > D_1$ when $F_1 > F_2$.

The proof is as follows. In this case, we can have $W_1 = C_k$ and $F_1 = C_k + l_1$. Then we try to get the value of F_2 . For every message that is sorted at time C_k , the least waiting time equals C_k . So the waiting time for M_2 will be not less than C_k , namely $W_2 \geq C_k$. Then $F_2 = W_2 + l_2 \geq C_k + l_2$. And from $l_1 \leq l_2$, it is clearly that $(F_2 = W_2 + l_2) \geq (F_1 = C_k + l_1)$, which contradicts with $F_1 > F_2$. Therefore, $C_k + R > D_1$ cannot be true.

B. If $C_k + R \leq D_1$, the flow time of message M_1 depends on the receiver available time D_1 , and then the flow time of M_1 is $F_1 = D_1 - R + l_1$.

C. If $C_k + R \leq D_2$, the flow time of the message M_2 , namely F_2 , depends on the receiver available time D_2 , and $F_2 = D_2 - R + l_2$.

D. If $C_k + R > D_2$, the flow time of the message M_2 , namely F_2 , depends on channel available time C_k , and $F_2 = C_k + l_2$.

According to the above properties, there will be two cases to evaluate F_s and $F_{s'}$.

Case 1. $C_k + R \leq D_1$ and $C_k + R \leq D_2$

From $F_1 > F_2$ and property B and C, we have the relationship as follows

$$D_1 - R + l_1 > D_2 - R + l_2 \Rightarrow D_1 + l_1 > D_2 + l_2 \quad (4.5)$$

In S :

M_1 will precede M_2 . So the C'_k for M_2 equals to $F_1(S)$, as shown in Figure 4.11.

And from property B, we have $F_1(S)=D_1-R+l_1$, then

$$C'_k + R = F_1(S) + R = D_1 - R + l_1 + R = D_1 + l_1$$

From (4.5), we can get $C'_k + R > D_2 + l_2 > D_2$

So the flow time of message M_2 in sequence S is dependent on channel available time C'_k , and $F_2(S) = C'_k + l_2 = F_1(S) + l_2$.

Then the flow time of the two messages, M_1 and M_2 , is

$$F_S = F_1(S) + F_2(S) = 2(D_1 - R + l_1) + l_2$$

In S' :

In S' , M_2 precedes M_1 . From property C, we have $F_2(S') = D_2 - R + l_2$.

The C'_k for M_1 equals to $F_2(S')$, and

$$C'_k + R = F_2(S') + R = D_2 - R + l_2 + R = D_2 + l_2$$

Then in order to get the flow time of M_1 in sequence S' , we need to compare $C'_k + R$ with its destination available time D_1 .

1) if $D_2 + l_2 > D_1$,

Then $C'_k + R > D_1$. And $F_1(S') = C'_k + l_1 = F_2(S') + l_1 = D_2 - R + l_2 + l_1$

So $F'_S = F_2(S') + F_1(S') = 2(D_2 - R + l_2) + l_1$.

2) if $D_2 + l_2 < D_1$,

Then $C'_k + R < D_1$ and $F_1(S') = D_1 - R + l_1$

So $F'_S = F_2(S') + F_1(S') = D_2 - R + l_2 + D_1 - R + l_1$

Comparison:

1) $F_S - F'_S = 2(D_1 - R + l_1) + l_2 - 2(D_2 - R + l_2) - l_1 = 2(D_1 + l_1) - 2(D_2 + l_2) + l_2 - l_1$

From (4.5) and $l_2 \geq l_1$, we have $F_S - F'_S > 0$, namely,

$$F_1(S) + F_2(S) - F_2(S') - F_1(S') > 0.$$

2) $F_S - F'_S = 2(D_1 - R + l_1) + l_2 - (D_2 - R + l_2 + D_1 - R + l_1) = D_1 - D_2 + l_1$

From (4.5), we can easily get $F_S - F'_S > 0$, namely,

$$F_1(S) + F_2(S) - F_2(S') - F_1(S') > 0.$$

So in this case, sequence S' can get less flow time F than S .

Case 2. $C_k + R \leq D_1$ and $C_k + R > D_2$

From $F_1 > F_2$ and property B and D, we have

$$D_1 - R + l_1 > C_k + l_2 \quad (4.6)$$

In S :

M_1 will be served first. From property B, we get $F_1(S) = D_1 - R + l_1$.

Then the channel available time for M_2 will be C'_k , which equals to $F_1(S)$. And from

$C_k + R \leq D_1$ and $C_k + R > D_2$, we can get $D_1 \geq D_2$.

So, $C'_k + R = F_1(S) + R = D_1 - R + l_1 + R = D_1 + l_1 > D_2$

Thus, $F_2(S) = C'_k + l_2 = F_1(S) + l_2$.

Consequently, we obtain $F_S = F_1(S) + F_2(S) = 2(D_1 - R + l_1) + l_2$.

In S' :

For the S' , M_2 precedes M_1 . From property D, we can easily get $F_2(S') = C_k + l_2$.

As for M_1 , the channel available time for it, C'_k , equals to $F_2(S')$, then we need to compare $(C'_k + R)$ with D_1 , where $C'_k + R = F_2(S') + R = C_k + l_2 + R$.

1) if $C_k + l_2 + R > D_1$,

we can have $F_1(S') = C'_k + l_1 = C_k + l_2 + l_1$. Thus

$$F'_S = F_2(S') + F_1(S') = 2(C_k + l_2) + l_1$$

2) if $C_k + l_2 + R < D_1$

we have $F_1(S') = D_1 - R + l_1$. Then $F'_S = F_2(S') + F_1(S') = C_k + l_2 + D_1 - R + l_1$.

Comparison:

1) $F_S - F'_S = 2(D_1 - R + l_1) + l_2 - 2(C_k + l_2) - l_1$

From (4.6) and $l_1 \leq l_2$, we have $F_S - F'_S > 0$, namely,

$$F_1(S) + F_2(S) - F_2(S') - F_1(S') > 0.$$

$$2) F_S - F'_S = 2(D_1 - R + l_1) + l_2 - (C_k + l_2 + D_1 - R + l_1) = D_1 - R + l_1 - C_k$$

From $C_k + R \leq D_1$, we have $F_S - F'_S > 0$, namely,

$$F_1(S) + F_2(S) - F_2(S') - F_1(S') > 0.$$

So in this case, sequence S' can get less flow time F than S , too.

Therefore sequence S' can achieve less flow time than sequence S in both cases, which proves that SMF algorithm can be improved with respect to mean flow time by sorting an adjacent pair of messages in the way of MFTS algorithm. It follows that the overall MFTS sequence can achieve better mean flow time than SMF.

4.2.1.4.2 Experimental Evaluation

In this subsection, we carry out the performance study of the proposed scheduling algorithm MFTS. We also compare it with the SMF algorithm by extensive simulation experiments. The objectives of the simulation are twofold. First, we demonstrate that MFTS achieves better performance in terms of average delay than SMF. Second, we demonstrate that the new channel selection algorithm LMS works better than EATS. The following subsections provide a discussion of the design of these experiments and their results.

4.2.1.4.2.1. Experimental Design

The parameters involved in the system design include the number of nodes (N), which is set to 50, and round-trip propagation delay (R), which is identical for all the nodes and equals to 10 time slots. Message lengths vary according to an Exponential distribution with a mean value as 20 time slots. Packet arrivals at each source node comply with independent and identical Poisson processes. Traffic load across all nodes ranges from 0.4 to 1. The destination nodes are selected according to a uniform probability distribution. To evaluate the performance of computer networks, average message delay is normally adopted as one major metric. In our current research work, we follow the same way as other researchers to

take average message delay as the metric to evaluate the performance of the scheduling algorithms over the networks. The message delay is defined as the duration from the time a message is scheduled to the time the message finishes transmission.

4.2.1.4.2.2. Experimental Results

Figure 4.12 compares the average message delay using four algorithms under varying loads in a system. In this set of experiments, we assume that the number of channels C is 4 and the tuning time T is 10 time slots. Since the propagation delay (R) is always part of the message delay in all of the algorithms, it will not be included in the message delay. Thus the delay incurred by queuing, tuning time overhead, and message transmission time will be our focus. From the figure, we can see that in the low traffic load, the difference among these four algorithms is not significant. However, as the traffic load increases, MFTS-EATS and MFTS-LMS significantly outperform SMF-EATS and SMF-LMS respectively. For example, when traffic load is 1.0, the delay with MFTS-EATS is only 85% of that with SMF-EATS. This is because the MFTS algorithm sorts the messages according to the flow time, therefore resulting in delay reduction. Also, the figure shows the difference between EATS and LMS. It is easy to see that LMS can work better than EATS in terms of delay due to the minimized scheduling latency.

Figure 4.12 Impact of traffic load on average delay of MFTS

Figure 4.13 compares the characteristic of average message delay under varying tuning time. In this set of experiments, the number of channels C is fixed to 4 and the traffic load ρ is set to 0.9. It is easy to see that the delay increases with the tuning time for these four algorithms. This is because when the tuning time increases, the source nodes take longer time to complete a message transmission. Among these four algorithms, MFTS consistently demonstrates its superior performance to SMF in terms of delay. The difference in message delays becomes more significant when tuning time increases. This is expected since MFTS considers the flow time of the message rather than only the message length. The LMS is introduced mainly to minimize the scheduling latency. Therefore, MFTS-LMS and SMF-LMS can achieve lower delay than MFTS-EATS and SMF-EATS respectively.

Figure 4.13 Impact of tuning time on average delay of MFTS

Figure 4.14 shows the effect of varying number of channels on the average delay for four algorithms. We assume that the tuning time is fixed to 10 and the traffic load is set to 0.9. It is clear that the delay decreases for all of the algorithms when the number of channels increases. Once again, we observe that MFTS achieves better performance than SMF in terms of delay while using the same channel selection algorithm, especially when the number of channels is small. This is because, when the number of channel is small, the competition for channels becomes drastic and the effect of the message sequencing becomes important. MFTS considers message flow time rather than message length, so it has lower

delay. When the number of channels becomes large, the performance of all the algorithms will flatten out and further increase in the number of channels will not induce any change, which means that data channels are no longer a bottleneck. From the figure, we can see that combining with the same channel selection algorithm, MFTS can achieve better performance than SMF, while combining with the same message sequencing algorithm, LMS can achieve better performance than EATS.

Figure 4.14 Impact of number of channels on average delay of MFTS

4.2.2 Scheduling Minimizing Idle Time (SMIT)

MFTS schedules the messages based on the flow time by taking not only the message length but also the availability of the channels and the receivers into account. This algorithm is shown to be better than SMF. However, no algorithm so far has considered carefully the timing relationships of message transmissions in the scheduling process. Based on the study of the existent algorithms, we observe that some idle time slots along the data channels could be induced, which leads to the reduction of the channel utilization. Moreover, unnecessary delay will be introduced to the message transmission in the network. The idle time slots are mainly due to the unavailability of destinations. For example, when two consecutive messages are destined to the same destination, the first message will occupy one of the channels and the receiver will be tuned to that channel. Since the receiver can only receive

one message in one period of time, the second message has to wait until the first message has been successfully transmitted and received even there may be other channels available. During the second message's waiting time, the channel selected to transmit it will stay idle even the channel has already become available for a while. Moreover, other messages cannot use the selected channel even when they are ready to be sent. In this case, the bandwidth of the channel will be wasted. Contrarily, if we sequence the messages by avoiding two messages destined to the same destination to be transmitted continuously, the idle time slots caused by the unavailability of destinations can be avoided and the channels can start to transmit the messages immediately when they become available. Although MFTS has considered the effect of the destination availability, it cannot reduce or avoid the idle time slots significantly. In this subsection, we propose a novel algorithm, namely SMIT, which aims to reduce the idle time slots along the data channels so that the utilization of the data channels can be dramatically raised.

4.2.2.1 Scheduling Algorithm

For our proposed scheduling algorithm, we adopt EATS as the technique to assign data channels. The basic idea of EATS algorithm is to assign a message to a data channel that has the earliest available time among all the channels.

As for the message sequencing techniques, it is very important to order the messages by considering the timing relationships of the message transmission during the scheduling processes carefully. Besides CAT and RAT that has been defined before, there are other four items we need to define in order to show the timing relationships as follows.

- *Idle time*: The cause of the idle time is that even if the channel becomes available, the message has to wait due to the unavailability of the destination.
- *Transmission time*: The time to transmit the message.
- *Holding time*: The time duration for one message to hold the channel. During this period, other messages cannot use this channel.

- *Waiting time*: The time for one message to wait until it has been sent out.

The description of the relationships of these six time terms is given in Figure 4.15, where t_0 is the time to start the scheduling. For simplicity, we assume that the tuning time and propagation delay are equal to zero.

Figure 4.15 Timing relationships without the effect of tuning time and propagation delay

- (a) One message m destined to destination i is selected to transmit and $CAT < RAT[i]$.
 (b) One message n destined to destination j is selected to transmit and $CAT > RAT[j]$.

We present two cases to show the timing relationship of the message transmission. Consider case (a), in which one message destined to a busy destination i is selected to transmit and this message has to be delayed for a while even if a channel is available since it has to wait for its destination to become available. The delay duration is defined as the idle time. During this time period, the channel is held idle and no message can be transmitted on it. Obviously this period is wasted and will block other messages even if they are ready for transmission. Here we define the holding time as the sum of the idle time and the transmission time. It is easy to see that in case (a), the time for one message to hold the channel starts at the point that the channel has become available even if this message cannot be transmitted immediately. During the holding time period, no other messages can use this channel. Case (b) differs from (a) only in that the message selected isn't destined to a busy

node. The result is simple that the selected message can start transmission immediately without idle time induced. Therefore, the time for the selected message to hold the channel is only its transmission time. One could find that, among the six time parameters, the idle time is the major reason of the unnecessary delay induced to the system. So it should be eliminated from any schedule as much as possible. From the comparison between case (a) and case (b), it is sufficient to find that the idle time can be easily reduced by scheduling the message destined to an idle node first.

For further illustration, with consideration of the effects of the tuning time and propagation delay, then Figure 4.15 can be changed to Figure 4.16. It is assumed that the propagation delay is identical for all the nodes. Similarly, in the case (a), after the channel becomes available, it has to stay idle for some period, i.e., idle time, to wait for the availability of destination. It is because, in this case, one message destined to a busy destination is selected to transmit. On the other hand, in the case (b), a message that doesn't destine to a busy node is selected. As a result, no idle time along the data channel will be induced.

Figure 4.16. Timing relationships with the effect of tuning time and propagation delay

According to Figure 4.15 and Figure 4.16, we have the following relationship:

$$\text{idle time} = \begin{cases} \text{RAT}[j] + T - R_j - \max(\text{CAT}[k], T) & \text{if } \text{RAT}[j] + T > \max(\text{CAT}[k], T) + R_j \\ 0 & \text{otherwise} \end{cases} \quad (4.7)$$

$$\text{holding time} = \text{message transmission time} + \text{message idle time} \quad (4.8)$$

$$\text{waiting time} = \text{channel available time} + \text{idle time} \quad (4.9)$$

Combing the definition of (4.1) and (4.2), (4.7) can be revised to

$$\text{idle time} = \begin{cases} r - R_j - t_1 & \text{if } r > t_1 + R_j \\ 0 & \text{otherwise} \end{cases} \quad (4.10)$$

As described above, large idle time could exist in the network by previous scheduling algorithms. It is due to the way they select messages. These algorithms always select the messages based on only one specific characteristic of the messages such as the message length. They haven't taken into account the characteristics of the WDM optical network. In fact, as we can see from (4.7), there are two important factors on scheduling the message transmission. One is channel available time. The other is destination available time. If a message is selected regardless of the relationship between these two factors, its transmission time has to be scheduled far behind the data channel available time. That is, the idle time becomes quite large. Clearly, the idle time durations have not been used for transmission. Further, they will block the transmission of other messages. Hence, it is very important for a scheduling algorithm to consider the timing relationships of the message transmission.

Our proposed scheduling algorithm, SMIT, is designed to reduce the wasteful time durations on the transmission channels as much as possible. The logic behind SMIT is that, instead of always selecting the message according to one specific characteristic, SMIT generates a sequence of message transmission by taking the timing relationships and the message length into account.

The following describes how the SMIT algorithm is formulated.

Suppose that there are n messages to be scheduled. We need three steps to sequence these messages.

Step 1) Sorts the messages according to SMF to obtain the sequence $E (M_1, M_2, \dots M_n)$ such that $l_k \leq l_{k+1}$, for $k=1, 2, \dots, n-1$, where l denotes the message length.

Step 2) Find the first message that will not induce the idle time in sequence E , say M_l , which satisfies $\max(CAT[k], T) + R_j \geq RAT[j] + T$. If none, go to step 3. Otherwise, schedule the message found and take off this message from E . Then repeat Step 2.

Step 3) Schedule the first message in the current sequence E and take off this message from E . Then return to Step 2.

The following describes how the algorithm SMIT-EATS is formulated.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Sequence the n messages according to SMF algorithm to obtain the sequence $E (M_1, M_2, \dots M_n)$ such that $l_k \leq l_{k+1}$, for $k=1, 2, \dots, n-1$;

Select the earliest available data channel k such that $CAT[k] \leq CAT[n]$, $\forall n \neq k$, $k \geq 1$, $n \leq C$;

Start from the first message in the sequence E , where $i=1$;

Check whether the current message i destined to node j induces the idle time to the channel: calculate $r=RAT[j]+T$, $t_1=\max(CAT[k], T)$, $t_2=\max(t_1+R_j, r)$;

If $t_1+R_j > r$, schedule the transmission time of the message at $t=t_2-R_j$,

update $RAT[j]=t_2+l_i$, $CAT[k]=t_2-R_j+l_i$,

then remove the scheduled message from the sequence and form a new sequence

E , where $n=n-1$,

If $n \neq 0$, return to **Select**;

otherwise, jump to **End**;

Else $i=i+1$;

If $i \neq n+1$, return to **Check**;

otherwise, there is no such message that will not induce idle time found, jump to

Choose;

Choose the first message M_1 in the current E sequence,

calculate $r = RAT[j] + T$, $t_1 = \max(CAT[k], T)$, $t_2 = \max(t_1 + R_j, r)$,

schedule the transmission time of the message at $t = t_2 - R_j$,

update $RAT[j] = t_2 + l_1$, $CAT[k] = t_2 - R_j + l_1$,

then remove the scheduled message from the sequence and form a new sequence E ,

where $n = n - 1$,

If $n \neq 0$, return to **Select**;

otherwise, jump to **End**;

End.

4.2.2.2 Example

In order to illustrate the SMIT algorithm thoroughly, in this subsection, we give an example and compare its performance with SMF. The current global information is given in Table 4.6. In this example, we assume that the data channels are initially idle. Let $T=0$ and $R=0$ for all the nodes in this illustration. The information of the messages is shown in Figure 4.17. Figure 4.17 presents a simplified system model of Figure 3.2 and it shows a network of 5 nodes and 2 data channels. One control frame arrives and it requests the transmission of one batch of messages, which includes four messages, M_1 , M_2 , M_3 and M_4 . In this figure, every message is represented by one box and labeled by M_i . Each message has three attributes. One is the source node. In this figure, the name of the node that the message is attached to is considered as the message's first attribute. For example, M_1 is attached to n_1 , so M_1 's source node is n_1 . The second attribute is the length of the message and the third attribute is the destination node of this message. The number shown in the box represents the second and the third attribute of the messages respectively.

$CAT[1]=3, CAT[2]=5$
$RAT[1]=1, RAT[2]=5, RAT[3]=0, RAT[4]=8, RAT[5]=4$
$T=0, R=0$

Table 4.6 Current global information in the example of SMIT

Figure 4.17 Information of messages in the example of SMIT

Figure 4.17 also shows the control frame and its control packets which transmit global information about the messages. A control packet contains the information of the messages such as source node identity, message length and destination node identity. The number on the top of the control packets of each frame designates the corresponding source nodes. Similarly, the number inside the packets represents the message length and the destination respectively. Since every time each control packet just transmits information about the message at the head of the queue at its corresponding source node, the Frame 1 contains the information about messages M_1 , M_2 , M_3 and M_4 in control packets 1, 2, 3 and 4 respectively.

We start our discussion by observing the behavior of the SMF algorithm in this example. According to the SMF algorithm, messages M_1 , M_2 , M_3 , M_4 are sorted to M_1 , M_4 , M_3 , M_2 . Firstly, running of the EATS algorithm will assign M_1 to data channel C_1 . Comparing the available time of M_1 's destination ($RAT[1]$) with the channel available time ($CAT[1]$) and according to (4.10), it is easy to get that the idle time is equal to zero. So the scheduled

transmission time for M_1 is equal to $CAT[1]$, which is 3 time slots. The delay for the message M_1 is 12 time slots (3 time slots for waiting and 9 time slots for transmission). After scheduling, the global information should be updated ($CAT[1]$ (updated 1) =12 and $RAT[1]$ (updated 1) =12). After that, C_2 becomes the earliest available channel. Message M_4 is then assigned to it. Although M_4 starts to hold the channel at the time 5, it cannot be transmitted immediately after the channel becomes available since its destination has been updated to 12 ($RAT[1]$ (updated 1) =12). Therefore, it has to wait until M_1 has been transmitted. Message M_4 will be started at time 12 and the delay of it is 25 time slots (5 time slots for waiting the channel to become available, 7 time slots idle time and 13 time slots for transmission). Similarly, the global information should be updated ($CAT[2]$ (updated 1) =25 and $RAT[1]$ (updated 2) = 25). The delay of Message M_3 and M_2 can be scheduled following the same way. And the final result is shown in Figure 4.18. It is clear that one idle time duration, which is equal to 7 time slots, is introduced to the system. This period has blocked messages M_2 and M_3 even they are ready for transmission. In Figure 4.18, each horizontal axis represents the time of each channel. The average delay of the four messages of this example can be calculated as $(12+25+28+45)/4=27.5$.

Figure 4.18 Results of message scheduling using SMF

Figure 4.19 Results of message scheduling using SMIT

Our proposed algorithm uses the same control information as SMF, as shown in Table 4.6. In our example, the four messages are sorted according to message length first. So messages M_1, M_2, M_3, M_4 are sorted to M_1, M_4, M_3, M_2 . Currently channel C_1 is the first one with earliest available time. According to the three steps of the SMIT algorithm, we have to find the first messages in the sequence $\{M_1, M_4, M_3, M_2\}$, which will not induce idle time to the channel. At this time, message M_1 is found. And the delay for it is equal to 12 time slots. After scheduling, global information should be updated ($CAT[1]$ (updated 1) = 12 and $RAT[1]$ (updated 1) = 12). Then the searching of the message that will not induced to idle period starts for the second time. At this moment Channel C_2 is the earliest available channel and will be selected to transmit next message. The current sequence is $\{M_4, M_3, M_2\}$ and this time message M_3 is selected. So the delay for it is 21 time slots. Similarly, $CAT[2]$ and $RAT[2]$ are updated. The current sequence is then changed to $\{M_4, M_2\}$. At current time, channel C_1 will be the selected channel. And M_4 will be the message found which will not induce idle period. So M_4 is assigned to data channel C_1 . The delay of it is 25 time slots. Finally, M_2 is assigned the current earliest available channel C_2 . The delay for it is 41 time slots. The final result is shown in Figure 4.19. In comparison with the Figure 4.18, it is clear that the idle time induced by the SMF algorithm has been avoided by our new algorithm. And the average delay of the four messages can be calculated as $(12+21+25+41)/4=24.75$. It

is evident from the results of this example that the SMIT algorithm improves the average message delay in comparison with the SMF algorithm. This is attributed to the fact that SMIT has the ability to reduce the idle time induced to the channels.

4.2.2.3 Performance Evaluation

In this subsection, we study the performance of the proposed scheduling technique SMIT by comparing it with SMF algorithm by extensive simulation experiments. In these experiments, we adopt EATS as the channel assignment algorithm.

4.2.2.3.1 Experimental Design

The default values of system parameters can be set as follows. The parameters involved in the system design include the number of nodes (N), which is set to 50, and round-trip propagation delay (R), which is identical for all the nodes and equals to 100 time slots. Message lengths vary according to an Exponential distribution with a mean as 20 time slots. Message arrivals at each source node comply with independent and identical Poisson processes. Traffic load across all nodes ranges from 0.4 to 1. The destination nodes are selected according to a uniform probability distribution.

4.2.2.3.2. Experimental Results

In Figure 4.20, our study focuses on the average delay versus traffic load. In this set of experiments, we assume that the number of channels C is 4 and the tuning time T is 10 time slots. From the figure, we can see that the difference between the two algorithms is not significant when the traffic load is low. However, as the traffic load increases, SMIT algorithm significantly outperforms SMF algorithm. This is because SMIT has the capability to reduce the wasteful time durations on the channel, therefore resulting in delay reduction. For example, when $\rho = 1$, the delay with SMIT is only 72% of that with SMF. This is an affirmation of the efficiency of sequencing the message by considering the relationship between channel available time and destination available time.

Figure 4.20 Impact of traffic load on average delay of SMIT

Figure 4.21 Impact of tuning time on average delay of SMIT

Figure 4.21 shows the average delay of the two algorithms as the tuning time varies. In this set of experiments, the number of the channels C is fixed to 4 and the traffic load ρ is set to 0.9. We observe that the delay increases with the tuning time for these two algorithms. This is because that when the tuning time is increased, it is equivalent to increase the waiting time since it takes longer time to complete a message transmission. Between the two algorithms, SMIT always results in the lower average delay. The difference between the message delays becomes more significant when tuning time increases. This is expected since SMIT is designed to reduce the idle time induced by other scheduling methods, therefore

resulting in delay reduction. Also, we observe that the increase of delay with the SMIT algorithm is slower than that with the SMF algorithm, as the tuning time increases.

Figure 4.22 Impact of number of channels on average delay of SMIT

In this experiment, the tuning time T is set to 10 and the traffic load ρ is set to 0.9. Figure 4.22 shows the average delay of the two algorithms as the number of channels varies. It is easy to see that as the number of channels increases, the average delay of the two algorithms decreases. Once again, we observe that SMIT achieves better performance than SMF in terms of delay, especially when the number of channels is small. This is because, when the number of channel is small, the competition for channels becomes drastic and the effect of the message sequencing becomes important. SMIT can reduce the wasteful time durations so that the average delay of the network can also be reduced. When the number of channels becomes large, the performance of both algorithms will flatten out. Further increase in the number of channels will not induce any change, which means that data channels are no longer a bottleneck.

4.2.3 Summary

In this section, we have presented two novel message sequencing algorithms for the reservation-based MAC protocols in single-hop passive-star coupled WDM optical networks, namely MFTS and SMIT. These two algorithms aim to reduce the message delay

from different points of view. MFTS determines the order of the message transmissions based on the flow time of the messages, which consists of message waiting time and message transmission time. The experimental and theoretical studies have shown that MFTS can reduce the average message delay of the network dramatically. On the other hand, SMIT is designed to reduce the number of wasteful time durations along the data channels. Unlike many existing reservation-based techniques, the major feature of SMIT algorithm is that it has carefully considered the timing relationship of the message transmissions so that it has the capability to reduce the unused time durations on the data channels resulting in higher utilization of the data channels and lower average message delay. Much more improvement on the total performance of the network could be achieved by this algorithm, which is shown by numerical studies. The combination of simplicity and superiority in performance makes the MFTS and SMIT algorithm powerful and viable choices in MAC protocol design for passive star-coupled WDM optical networks.

4.3 Comparison

In this chapter, we have proposed three novel channel assignment algorithms and two novel message sequencing algorithms. In order to show the superiority over each other, we present a simple comparison for these two kinds of algorithms respectively.

4.3.1 Channel Assignment Algorithms

LMS, CCS and CC-MSL are three novel channel assignment algorithms proposed. LMS is the algorithm that aims to reduce the scheduling latency of EATS. It is developed based on the observation that EATS may cause large scheduling latency due to the way it selects the data channels. The major feature of LSM algorithm is that it selects the channels by considering the channel availability, destination availability and tuning overhead simultaneously. In this way, the scheduling latency can be reduced. CCS is the algorithm which is designed to reduce the penalty of the tuning time of the transceivers which is unlikely to go away nowadays since the transceivers' capabilities are subject to both tuning

range and tuning speed. In order to avoid the unnecessary transmitter tuning, CCS schedules the transmissions on the same data channel as the one which the transmitter has been tuned to. This method is shown to be effective, especially when the traffic load is light. CC-MSL is the algorithm which combines the advantage of CCS and MSL, with the capability to reduce scheduling latency as well as tuning overhead. In this subsection, a simple comparison among these three algorithms will be presented.

The default values of system parameters can be set as follows. The parameters involved in the system design include the number of nodes N which is set to 50, the number of data channels C which is set to 4, tuning time T which is set to 20 time slots and round-trip propagation delay R which is identical for all the nodes and equals to 100 time slots. Message lengths vary according to an Exponential distribution with a mean of 20 time slots. Message arrivals at each source node comply with an independent and identical Poisson processes. The traffic load across all nodes ranges from 0.4 to 1. And no message sequencing techniques will be imposed to the messages in this experiment.

Figure 4.23 depicts the average message delay using three algorithms under different traffic loads. When the traffic load is light, both CCS and CC-MSL work better than LMS. This is expected since CCS and CC-MSL are designed to reduce the penalty of tuning overhead resulting in delay reduction. When the traffic load is light, the main part of the message waiting time is the time to wait for the transmitter to be ready. In this case, it is beneficial to schedule the transmissions on the same data channel as the one which the transmitter has been tuned to. Hereby it avoids transmitter tuning overhead. When the traffic load is heavy, the time to wait for starting transmission can be longer than the transmitter's tuning time. So the node will have enough time to tune its transmitter. In this case, CCS cannot achieve better performance than LMS since less tuning overhead is induced. Contrarily, when the traffic load is heavy, LMS outperforms CCS, which shows that LMS has the ability to reduce the scheduling latency by considering destination availability, channel availability and tuning overhead simultaneously. It also arrives at a conclusion that small latency can

lead to the reduction of the message delay. CC-MSL is designed to reduce tuning overhead as well as scheduling latency. Therefore, it achieves the best performance among these three algorithms.

Figure 4.23 Performance comparison among LMS, CCS and CC-MSL

The conclusion we can draw from the comparison is that when the traffic load is light, it is beneficial to use CCS rather than LMS since the tuning overhead will be the major penalty of the tuning overhead in this case. On the other hand, when the traffic load is heavy, LMS will be a better choice since the tuning overhead will not be a major problem since the nodes have enough time to tune their transmitters. Contrarily, the scheduling latency may be a major part of the message delay in this case. In order to obtain the best performance, CC-MSL is the best choice due to its combination of advantages of CCS and MSL.

4.3.2 Message Sequencing Algorithms

MFTS and SMIT are two novel message sequencing algorithms that try to further elaborate the characteristics of the specified WDM optical networks. MFTS determines the order of the message transmissions based on the flow time of the messages, which consists of message waiting time and message transmission time. On the other hand, SMIT tries to reduce the number of idle time slots along the data channels resulting in its superior ability to reduce the unnecessary delay. These two algorithms aim to improve the network

performance in terms of message delay. And they solve the problems from different points of view. In this subsection, we will present a simple comparison between these two algorithms through simulation.

The simulation design is as follows. The parameters involved in the system design include the number of nodes (N), which is set to 50, and the number of channels (C), which is set to 4. Tuning time T is assumed to equal to 10 time slots. And round-trip propagation delay (R) equals to 100 time slots (identical for all the nodes). Message lengths vary according to an Exponential distribution with a mean value as 20 time slots. Message arrivals at each source node comply with independent and identical Poisson processes. Traffic load across all nodes ranges from 0.4 to 1. And the destination nodes are selected according to a uniform probability distribution. In this experiment, EATS algorithm will be adopted as the channel selection algorithm.

Figure 4.24 Performance comparison between MFTS and SMIT

Figure 4.24 demonstrates the message delays of these two algorithms under varying traffic load. It can be observed that SMIT offers better results than MFTS, especially when the traffic load is heavy. When the traffic load is light, the competition for the messages is not drastic and the effect of the sequencing is not significant. However, when the traffic load increases, different sequencing techniques will lead to different performance. Both of these

two algorithms are developed based on the observation that the destination unavailability may introduce delay to the messages since the message has to wait for the destination to be available even if the channel is available now. For example, when two consecutive messages are destined to the same destination, one of the messages has to wait until the other message has been successfully transmitted and received even there may be other channels available. In this situation, one message is blocked and some channels cannot be used to transmit messages even the channels have already become available for a while. Moreover, except the transmission delay and the queueing delay, another delay for waiting the destination to become available is introduced to the message. Based on the analysis above, it is easy to find that the delay caused by the destinations can be reduced or eliminated in most of the time if the sequence of the messages is decided carefully. MFTS has taken the effect of the destination availability into account by sequencing the messages according to the flow time. However, it cannot reduce the wasteful time slots caused by the way of sequencing to the maximum extent. On the other hand, SMIT is capable to consider the timing relationship of the scheduling carefully and reduce the unnecessary delay by scheduling the messages that will not be affected by the destination first. That is, the delay of the messages mainly focuses on their own transmission delay and the queueing delay. Therefore, SMIT will obtain the better performance in terms of message delay than MFTS.

From this comparison, we can arrive at the conclusion that in the scheduling problem of WDM optical networks, to reduce the wasteful time duration caused by the destination unavailability is the most efficient way to improve the performance in terms of message delay. This is the major achievement we expect to see by using SMIT. From this result, we can also find that the scheduling problem of WDM optical networks differs from other simple systems in the aspect that the messages cannot be transmitted in the same channel or destined to the same destination simultaneously. Therefore, except the message length, the coordination of channel availability and destination availability is also a very important factor which will affect the performance of the networks.

4.3.3 Combination of the Algorithms

Different combinations of channel assignment algorithms and message sequencing algorithms will achieve different performances. In this subsection, we will present one experimental comparison to show the performance of the combinations of those proposed algorithms. They are MFTS-LMS, MFTS-CCS, MFTS-CC-MLS, SMIT-LMS, SMIT-CCS and SMIT-CC-MLS.

The parameters involved in the system are designed as follows. The number of nodes (N) is set to 50 and the number of data channels (C) is equal to 4. Round-trip propagation delay (R) equals to 10 time slots (identical for all the nodes) and the tuning time (T) is set to 10 time slots. Message lengths vary according to an Exponential distribution with a mean value as 20 time slots. Packet arrivals at each source node comply with independent and identical Poisson processes. Traffic load across all nodes ranges from 0.4 to 1. And the destination nodes are selected according to a uniform probability distribution.

Figure 4.25 Performance comparison among combinational algorithms

Figure 4.25 presents the performance of different combinations of message sequencing algorithms and channel assignment algorithms. It is clear that the delay for all the algorithms increases with the traffic load. When the traffic load is light, the average delay of the two

message sequencing algorithms, namely MFTS and SMIT, get close to each other when combining with LMS, CCS and CC-MSL respectively. This means that the difference between MFTS and SMIT is not significant when the traffic load is light. Another result we can obtain from the figure is that, when the traffic load is light, CCS and CC-MSL work better than LMS when combining with MFTS and SMIT respectively. From this result, we can also arrive at the conclusion that when the traffic load is light, CCS and CC-MSL outperform LMS due to their high capability to reduce the penalty of tuning overhead. However, when the traffic load becomes heavy, combining with the same message sequencing algorithms, CCS works worse than other two channel assignment algorithms, which can be proved from the result that MFTS-LMS and MFTS-CC-MSL works better than MFTS-CCS. This means that the nodes have enough time to tuning the transmitters when the traffic load is heavy and the tuning overhead is not the major limitations of the packet scheduling in photonic switching anymore. Contrarily, the scheduling latency becomes the major obstacles of the scheduling in this case. Therefore, with the ability to reduce the scheduling latency, LMS and CC-MSL outperforms CCS in terms of message delay. Also, it is easy to find that MFTS-CC-MSL and SMIT-CC-MSL work better than MFTS-LMS and SMIT-LMS respectively. This is expected since CC-MSL works better than LMS because its combination advantage of CCS and MSL. The relationship among these three channel assignment algorithms is shown to be consistent with that of Figure 4.23, in which there are no message sequencing algorithms imposed to the messages. On the other hand, with the same message selection algorithms, SMIT always works significantly better than MFTS when the traffic load is heavy, which proves again that, with the ability to reduce the wasteful time durations along the data channels, SMIT is able to reduce the average message delay more efficiently. This result is also consistent with that of Figure 4.24, in which EATS algorithm is used as the channel assignment techniques. Among the six combinational algorithms, SMIT-CC-MSL achieves the best performance in terms of average message delay, which is shown in the figure. This is expected since SMIT is the one works better

between the two message sequencing algorithms and CC-MSL is the best one among the channel assignment techniques, and, therefore, these two form the best combination.

4.4 Summary

In this section, several novel reservation-based algorithms for single-hop passive-star coupled WDM optical networks have been proposed and their performances are analyzed. In principle, the network performance in terms of message average delay is significantly improved by considering the channel availability, destination availability and transceivers' tuning overhead simultaneously.

LMS, CCS and CC-MSL are three novel channel assignment algorithms proposed. LMS is introduced to reduce the scheduling latency. CCS is designed to reduce the penalty of tuning overhead. And CC-MSL combines the advantages of MSL and CCS. A simple comparison among these three algorithms is conducted. The result depicts that CCS works better than LMS when the traffic load is light. However, when the traffic load is heavy, CCS cannot outperform LMS since the time to wait for starting transmission can be longer than the transmitter's tuning time and no tuning overhead is induced. Contrarily, scheduling latency becomes the major obstacle of the scheduling in this case. With the ability to reduce the scheduling latency, LMS and CC-MSL outperforms CCS when the traffic load is heavy, which arrives at the conclusion that the small latency can lead to the reduction of the message delay. CC-MSL is an algorithm which is designed to reduce the tuning overhead as well as scheduling latency, so it obtains the best performance in the comparison.

MFTS and SMIT are two message sequencing techniques which aim to reduce the message delay from different points of view. MFTS algorithm sequences the messages based on the flow time of the messages, where flow time consists of message waiting time and message transmission time. And SMIT tries to reduce or eliminate the unused time durations along the data channels by scheduling the messages that will not introduce the idle time first. Both of these two algorithms have taken the effect of destination availability into account

when sequencing the messages. However, MFTS cannot reduce the wasteful time slots along the data channels to the maximum extent. SMIT, on the other hand, considers the characteristic of the WDM optical networks more adequately so that it can achieve better performance, which is shown in the comparison result between these two algorithms. The main conclusion can be also drawn from the obtained comparison result, which is that the most efficient way to reduce the average message delay is to reduce or eliminate the delay caused by the destination unavailability. This part of delay is mainly caused by the method of sequencing. With the ability to reduce the wasteful time durations caused by destination unavailability, SMIT is shown to perform more efficiently in single-hop passive-star coupled WDM networks.

In addition, a simple comparison among different combinational algorithms is conducted. SMIT-CC-MSL, which combines the message sequencing algorithm SMIT and the channel assignment algorithm CC-MSL, obtains the best performance in the comparison.

Chapter 5 . Proposed Solutions to Support QoS Service

One of the important issues involved in the future communication networks is to provide different quality of service (QoS) to meet the traffic requirements of a large variety of applications. Therefore, it appears to be essential to develop novel medium access control protocols which support real-time traffic with tight delay constraints (associated with message deadlines) in a flexible and efficient way directly in the optical transmission layer.

Although many access protocols for single-hop passive-star coupled WDM optical networks have been proposed in the recent past, there are just a few works in the literature related to the integration of real-time services with certain QoS requirements and best-effort services. Generally, since circuit-switched communication for real-time traffic leads to extremely inefficient resource allocations, packet switching seems to be a more promising approach for delay-sensitive traffic in terms of optical WDM lightwave networks.

In general, for guaranteeing QoS requirements, a control on packet delay or loss rate is required. As a first part in this chapter, one novel algorithm is proposed with the aim to reduce the message loss rate of real time messages as much as possible. Further, in order to support differentiated services, a novel scheme, named Cost-Based Priority Scheduling (CBPS), is proposed in the second part of this chapter to support the messages with different time constraints. Based on the CBPS algorithm, a QoS prediction structure is developed in

the third part of this chapter so as to provide a flexible mean for controlling network behavior as the QoS requirements changes. The final part provides a summary.

5.1 Differentiated Dropping Scheduling (DDS)

There has been increasing interest in providing real-time communication service for applications with time constraints in high-speed networks. The most important aspect of the time-constrained applications is that a message must be received at the destination station within a given amount of time after its generation. This time is referred to as message *deadline*. If the delay of a message in the system exceeds its time constraint, the message is considered as lost. In a real-time communication system, the principle task for the scheduling policies is to maximize the number of messages that are delivered by their respective deadlines. However, most of the former research work focuses on minimizing the average message delay, reducing call blocking probability, or increasing the network throughput. What is not fully addressed is how to support time-constrained communications for messages with delivery deadline. So far, only few work on multiple access protocols that provide real-time service to time-constrained messages on WDM optical networks have been found. One research in [63] proposes a reservation-based protocol for deadline critical services in a single-hop lightwave network with one control channel and multiple data channels. This scheme uses a token to pass the system information among all the nodes and to control the nodes to access the channels. However, the messages transmitted in that network are limited to fix-length packets. Another result in [48], proposes a pre-allocation-based protocol to provide deterministic timing guarantees for the messages with time constraints in single-hop optical networks. This paper considers a slot assignment issue so that enough slots can be assigned to each message streams before their delivery deadlines. However, in this paper, the transmitter/receiver tuning time is assumed to be incorporated into a slot time and hasn't accommodated the scheme with the tuning overhead. Also, in this scheme, every message has to be segmented into several time slots so that it can be

transmitted through separate time slots. As for the research result in [46], it proposes a reservation-based MAC protocol for best effort real-time service. The scheduling algorithm of this protocol is based on the time related dynamic priority scheme, namely Minimum Laxity First (MLF) scheduling. The principle of this dynamic scheduling scheme is that the most stringent message will get the transmission service first. However, the message with less laxity may be longer in length. It will cause the subsequent loss of other messages, which will lead to real-time performance degradation.

In this section, we propose a dynamic-priority-discipline-based protocol for scheduling real-time messages in single-hop passive star coupled WDM optical networks, which is called Differentiated Dropping Scheduling (DDS). This algorithm has the capability of preventing channel collision as well as destination conflict and supports real time variable length messages. It has the following characteristics to differentiate it from other algorithms mentioned above. First, the scheduling algorithm for this protocol is a complete reservation-based and distributed dynamic scheduling algorithm. Second, DDS schedules the messages after having collected the information of messages in one control frame. It makes our algorithm work better than existing approaches. Third, and more important, this algorithm takes the physical characteristics of the network into account. In this way, the algorithm can coordinate the transmission and reception of the messages efficiently while minimizing the fraction of messages violating their constraints.

In order to evaluate the performance of the new algorithm, extensive discrete-event simulations have been conducted by comparing it with the algorithms with the similar functions. The results demonstrate that significant improvement in terms of message loss rate can be obtained by using our new algorithm. We also estimate the performance of the algorithms in a mathematical way, which shows that DDS can achieve the best performance.

5.1.1 Scheduling Algorithm

5.1.1.1 Motivation

What is addressed in this section is how to reduce the fraction of messages violating their deadlines. So the task of the scheduling algorithms for the protocols is to schedule the transmission of messages to meet the time constraints as much as possible. A common way to do so nowadays is to incorporate a dynamic-priority-discipline-based sequencing algorithm into the protocol. The typical algorithm is Earliest Due Date (EDD), which sequences the messages according to the deadline. In this way, the most stringent message will get the transmission service first. However, the most stringent message may have larger transmission time, which will cause subsequent loss of other messages queuing behind it. So another one [66], named Moore and Hodgson's algorithm, is proposed to improve the performance of EDD. The basic idea of this algorithm is that when one message is found to violate the deadline, which we will call *potential tardy message* hereafter, the algorithm will discard the message with the maximum transmission time among the messages queueing before the potential tardy message in the message sequence rather than the potential tardy message itself. This algorithm is shown to be optimal in simple systems. However, optical systems are different in the aspect that the coordination of message transmissions should consider the channel availability and the receiver availability simultaneously. Therefore if we simply apply this algorithm to our network, the potential tardy message cannot be guaranteed to meet its deadline after the message with the maximum transmission time is discarded. For example, according to (4.1), (4.2) and (4.3), we can easily find that if $(t_1 + R_j) < (RAT[j] + T) = r$, the message needs to wait longer than the channel available time due to the unavailability of destination receiver. When the message with the largest transmission time is discarded according to Moore and Hodgson's algorithm, the channel available time may be reduced, but the destination available time may not change if the message discarded does not have the same destination with the potential tardy message. As a

result, it cannot be ensured that the potential tardy message will not violate its deadline after the longest message is discarded. In this case, Moore and Hodgson's algorithm cannot improve the performance; instead, it may produce worse performance. However, this algorithm gives us a good idea to minimize the number of tardy messages of our system. Following its idea, we propose a new algorithm which will take into account the physical characteristics of the optical networks, namely DDS.

5.1.1.2 DDS Algorithm Description

The basic sequencing algorithm in our new protocol is still EDD, in which the message with least deadline will be given the highest priority. In order to prevent the blocking problem of the longer messages in EDD as addressed above, the new algorithm will choose to sacrifice the longer messages, which we will call *blocking messages* hereafter, in order to schedule more messages to meet the time constraints.

To determine which message will be the one blocking others, we have to consider the timing relationship carefully, which is also addressed in Section 4.2.2. In our system, the time for the message to hold the channel, namely holding time, consists of two parts. One is message transmission time, and the other is message idle time. The cause of the message idle time is that even if the channel becomes available, the message has to wait due to the unavailability of the destination. So we can derive the relationships as follows:

- message idle time = the time to start the transmission of the message - the time the channel becomes available
- holding time = message transmission time + message idle time

Therefore, the message with the largest holding time may be the most possible one to block others rather than the one with the largest transmission time.

Moreover, in order to determine which message will be discarded when the potential tardy message found, we need to consider two causes of the potential tardy message separately. One is that the reason for the potential tardy message to be late is channel

unavailability. In this case, the message with the largest holding time will be discarded from the queue instead of the potential tardy message found. This way will be more efficient than discarding the potential tardy message if its holding time is not the largest. Moreover, after the message is discarded, the potential tardy message will no longer be late, which we will prove in the later performance comparison section. The other cause for the potential tardy messages to be late is destination unavailability. In this case, the blocking could be caused by the messages with the same destination as the potential tardy message. So it will be more efficient to discard the message with the largest transmission time among those that have the same destination as the potential tardy message. The reason to discard the message with the largest transmission time instead of largest holding time in this case can be expressed as follows. Here we use M_l to denote the potential tardy message. First we will define a set F ($M_{l(1)}, M_{l(2)}, \dots, M_{l(k)}$) from the messages queuing before M_l , in which every message has the same destination with M_l . Moreover, we assume that, in such a set, the waiting time of every message equals to the flow time of the message queuing right before it, where flow time is defined as the time that a message stays in the system. It means that every message in F cannot start the transmission until the one queuing before it completes the transmission. So we can find that the message idle time of each message in F is caused by the message right before it. Then from the fact that the waiting time of message M_l is equal to the flow time of message $M_{l(k)}$ and the waiting time of $M_{l(k)}$ is equal to the flow time of message $M_{l(k-1)}$, we can say that the flow time of message $M_{l(k-1)}$ has effect on the waiting time and the idle time of M_l . This also means that any change of the flow time $M_{l(k-1)}$ may also change the waiting time and the idle time of M_l . Following this idea, we can arrive at a conclusion that every message in F have effect on the waiting time and the idle time of all the messages queuing behind it. Therefore, if one message in F is discarded, the waiting time and the idle time for the messages queuing behind it will be reduced simultaneously. Hence in this case, it will save more time if the message with the largest transmission time is discarded. The detail proof will be addressed in the later section.

Following this idea, we propose a new sequencing scheme and it works as follows:

Step1. Sort the messages according to the EDD rule to obtain the sequence $E (M_1, M_2, \dots, M_n)$ such that $d_i \leq d_{(i+1)}$ for $i=1, 2, \dots, n-1$, where d denotes the message deadline.

Step 2. Find the first potential tardy message in sequence E , say M_i . If none, go to step 4, otherwise, go to step 3.

Step 3. Consider the two possible causes for the potential tardy message:

As defined in (4.1)~(4.4), t_2 depends on both the data channel's earliest available time t_1 and destination j 's earliest available time r .

Recall (4.3)

$$t_2 = \max(t_1 + R_j, r)$$

i) If $t_1 + R_j < r$, then $t_2 = r$. This means that the reason for the message to be late is receiver unavailability. In this case, determine a set of messages, say F as mentioned above, from $(M_1, M_2, \dots, M_i)$ first. And then find the message with the largest transmission time in it. Following that, compare the message found with M_i and discard the longer one from the current sequence E .

ii) If $t_1 + R_j \geq r$, then $t_2 = t_1 + R_j$. This means that the reason for the message to be late is channel unavailability. In this case, find the message in the sequence $(M_1, M_2, \dots, M_i)$ with the largest holding time and discard it from the sequence E .

Return to Step 2 with a new sequence E .

Step 4. The messages in the new sequence will be scheduled.

As mentioned above, the new algorithm will choose to sacrifice the blocking messages in order to schedule more messages to meet the time constraints. In our system, all the traffic streams are assumed to have the equal importance, so from loss rate's point of view, discarding one longer message will be identical to discarding one shorter message. Therefore, sacrificing the longer message will achieve better performance if the loss rate is defined as the fraction of messages violating their deadline. However, such sacrifice may be

possible to result in two problems. One is that it will lead to the performance degradation in terms of throughput. And the other is that it may increase loss rate if it is measured in the unit of packet. We will study these two problems later in the simulation section and the results will show that this kind of sequencing will not degrade the performance of throughput as well as the loss rate when the number of packets is used to measure them. This is because that discarding blocking message may save more time so that more messages can succeed in meeting their deadlines.

5.1.1.3 DDS-EATS Algorithm Description

DDS needs to specify a way of selecting an available data channel for packet transmission. In this section, we will adopt EATS as our basic channel assignment mechanism. The basic idea of EATS is to assign a message to a data channel that has the earliest available time among all the channels in the network. This algorithm effectively resolves destination conflicts. Its conflict-resolution characteristics are conveniently inherited by our algorithm, as well. The following describes how the DDS-EATS algorithm is formulated.

We assume that there are N nodes and C data channels in the network. The propagation delay is R_j . And the transceivers' tuning time is T . In this section, we will use $D[j]$ to stand for $RAT[j]$, for each node. $D[j]=a$, where $j=1\cdots N$, means that node j 's receiver will become free after a time slots. And we will use $C[k]$ to stand for $CAT[k]$, for each channel. $C[k]=b$, where $k=1\cdots C$, means that channel k will become available after b time slots. Moreover, the definitions of some parameters are the same as those of (4.1), (4.2), (4.3) and (4.4). The messages can be transmitted from source node s to destination node j , where $s \neq j$, and $s, j \in N$. We use l_i to denote the length and d_i to denote the deadline of message i . And we use a table $C[i][m]$ to store the available time of each channel m after scheduling the i th message, where $1 \leq i \leq N$ and $1 \leq m \leq C$, and a table $D[i][n]$ to store the available time of each destination node j after scheduling the i th message, where $1 \leq i \leq N$ and $1 \leq n \leq N$.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Save the current value of $C[k]$ and $D[j]$ to the other two tables $C'[k]$ and $D'[j]$ respectively;

Sequence the n messages represented in the control frame according to EDD algorithm to obtain the sequence $E(M_1, M_2, \dots, M_n)$ such that $d_i \leq d_{i+1}$;

Begin from the first message in the sequence E , where $i=1$;

Search the earliest available data channel k for message M_i such that $C[k] \leq C[n]$,
 $\forall n \neq k, k \geq 1, n \leq C$;

Check whether the deadline of current message M_i destined to node j can be exceeded:
 calculate $r=D[j]+T$, $t_1=\max(C[k],T)$, $t_2=\max(t_1+R_j,r)$ and obtain the message transmission time at $t=t_2-R_j$,

If $t+l_i+R_j > d_i$, it will be the potential tardy message found, then go to **Consider**;

Else, update $D[j] = t_2+l_i$, $C[k]=t_2-R_j+l_i$,

store $C[i][m]$ and $D[i][n]$:

for ($m=1; m \leq C; m++$)

$\{C[i][m]=C[m];\}$

for ($n=1; n \leq N; n++$)

$\{D[i][n]=D[n];\}$

and $i=i+1$;

if there are any messages not scheduled, return to **Search**,

otherwise, go to **End**;

Consider two causes of the tardy message,

i) If $t_1 + R_j < r$, determine a set of messages, say F , from $(M_1, M_2, \dots, M_i)$. Find the message with the largest transmission time from F and compare it with M_i . Then discard the longer one, say message M_h , from the current sequence E ;

ii) If $t_1 + R_j \geq r$, find the message in the sequence $(M_1, M_2, \dots, M_i)$ with the largest holding time, say message M_h , and discard it from the sequence E ;

Then $n=n-1$, recover the value of channel available time and destination available time as follows:

If $h \neq 1$, recover from the values stored after scheduling $(h-1)$ th message:

for $(m=1; m \leq C; m++)$

$\{C[m]=C[h-1][m];\}$

for $(n=1; n \leq N; n++)$

$\{D[n]=D[h-1][n];\}$

If $h=1$, recover the value of $C[k]$ and $D[j]$ from the tables $C'[k]$ and $D'[j]$;

return to **Search** with the new sequence and $i=h$;

End.

The complexity of DDS algorithm can be evaluated based on its operations. It has two sequence procedures and one searching procedure. One of the sequence procedures is to sort the messages according to its deadline, and the other is to sort the CAT table, which is denoted by $C[k]$ in this section. The number of the messages is the number of nodes in the network in the worst case and the number of CAT is the number of channels in the network. Let us assume that the number of nodes (N) is always larger than the number of channels (C). So we consider only the number of nodes when we estimate the complexity of the sorting algorithm. The complexity of a typical sorting algorithm is $O(N \log_2 N)$, where N can be mapped to the number of nodes in our case. The searching procedure is to search the largest message from the message sequence. The searching algorithm simply loops through the messages and its complexity is $O(N)$ in the worst case. Hence the complexity of the

algorithm to schedule one message will be $O(N)$ in the worst case. To schedule all the messages represented in one control frame will take $O(N^2)$.

5.1.1.4 Example

In this section, we discuss the details of DDS algorithm in the context of an example and compare its performance with Moore and Hodgson's algorithm and EDD algorithm. The channel assignment algorithm used in this example is EATS. We assume that the number of messages is 5 and the number of data channels C is 3. The initial $C[k]$ and $D[j]$ before scheduling are given in Table 5.1, and the information of the messages is given in Table 5.2, which contains five messages, $M_1, M_2, \dots, M_5$. The tuning time of the transceiver (T) is set to 0, and the propagation delay (R) is assumed to be identical for all the nodes and set to 4.

$C[1]=10, C[2]=2, C[3]=0$
$D[1]=7, D[2]=0, D[3]=5, D[4]=10, D[5]=0$
$T=0, R=4$

Table 5.1 Current global information in the example of DDS

Messages	M_1	M_2	M_3	M_4	M_5
Destination node j	3	1	2	2	2
$D[j]$	5	7	0	0	0
Message length l	14	14	13	7	20
Deadline d	27	22	19	22	35

Table 5.2 Information of messages in the example of DDS

Messages	M_3	M_2	M_4	M_1	M_5
Destination node j	2	1	2	3	2
$D[j]$	0	7	17	5	17
Message length l	13	14	7	14	20
Deadline d	19	22	22	27	35
Selected Channel	3	2	*	*	*
Flow time	17	21	*	*	*
$C[k]$ (updated)	13	17	*	*	*
$D[j]$ (updated)	17	21	*	*	*
Message loss or not	*	*	loss	loss	loss

Table 5.3 Results of message scheduling using EDD-EATS

We start our discussion by observing the behavior of the EDD-EATS algorithm in this example. According to EDD algorithm, messages $M_1, M_2, \dots, M_5$ are sorted to $M_3, M_2, M_4,$

M_1 , M_5 according to the deadline. Then we use the EATS technique to assign the data channels. At the current time, channel 3 is the earliest channel ($C[3]=0$). Message M_3 is then assigned to it at time 0. Since the destination of M_3 is currently idle ($D[2]=0$), the waiting time of M_3 is equal to 0 and the flow time is 17. And because the deadline for M_3 is 19, the message will not be late. Then $C[3]$ and $D[2]$ are updated. After updating, channel 2 will become the earliest available channel, so it is assigned to M_2 at time 2. Because the destination node of M_2 still has one more packet to receive ($D[1]=7$), the waiting time is dependent on the destination available time and equals to 3 ($D[1]-R=3$). Then we can get that the flow time of M_2 is 21. From the fact that the deadline of message M_2 is 22, the message will not be late, too. Similarly, $C[2]$ and $D[1]$ are updated. After that, channel 1 will be assigned to the message M_4 . And because the destination available time of M_4 has been updated to 17, the waiting time of it will be 13. Then we can get that the flow time of it is 24, which is larger than its deadline, so the message M_4 will violate its deadline. The final results of scheduling all the messages using EDD-EATS algorithm are shown in Table 5.3. As the table shows, the messages M_1 and M_5 are late similarly because the flow time exceeds their message deadlines. The total loss number is 3 in this case.

Messages	M_4	M_1	M_5	M_2	M_3
Destination node j	2	3	2	1	2
$D[j]$	0	5	11	7	31
Message length l	7	14	20	14	13
Deadline d	22	27	35	22	19
Selected Channel	3	2	3	*	*
Flow time	11	20	31	*	*
$C[k]$ (updated)	7	16	27	*	*
$D[j]$ (updated)	11	20	31	*	*
Message loss or not	*	*	*	loss	loss

Table 5.4 Results of message scheduling using Moore and Hodgson-EATS

Next we discuss the result of applying Moore and Hodgson-EATS to our example. Using this technique, the messages are first sorted to M_3 , M_2 , M_4 , M_1 , M_5 according to EDD. From Table 5.3, we can easily find that M_4 is the first potential tardy message found. According to Moore and Hodgson's algorithm, the message M_2 is the longest message among those

queuing before M_4 and it is also larger than M_4 , so it is discarded. Then the global information $C[k]$ and $D[j]$ will be recovered to the original value shown in Table 5.1 and then the new sequence without M_2 , namely M_3, M_4, M_1, M_5 , will be formed. At current time, the earliest available channel $C[3]$ is assigned to the message M_3 at time 0. Since the destination of M_3 is idle now, the waiting time for M_3 is 0. Then we can get that the flow time of M_3 is 17 and its deadline will not be violated. After that, $C[3]$ is updated to 13 and $D[2]$ is updated to 17. After scheduling, channel 2 will be the channel with the earliest available time and it is assigned to M_4 at time 2. Since the destination of M_4 , say $D[2]$, has been updated to 17, its waiting time equals to 13. Then its flow time equals to 24 and its deadline will be exceeded. Thus the message M_4 is found as the potential tardy message for the second time, which means that discarding M_2 does no help to the potential tardy message M_4 . This is because that the waiting time of M_4 depends on receiver available time rather than channel available time. In this case, message M_3 that has the largest transmission time will be discarded from the sequence. After M_2 and M_3 are discarded, the messages M_4, M_1, M_5 will not be late and the final results are shown in Table 5.4. The loss number is 2. It is easily to see from the Table 5.4 that in order to let the potential tardy message M_4 not to be late, two messages M_2 and M_3 before it are discarded. This is due to the fact that Moore and Hodgson's algorithm is not suitable for the optical networks.

Messages	M_2	M_4	M_1	M_5	M_3
Destination node j	1	2	3	2	2
D_{jk}	7	0	5	13	34
Message length l	14	7	14	20	13
Deadline d	22	22	27	35	19
Selected Channel	3	2	2	1	*
Flow time	21	13	27	34	*
$C[k]$ (updated)	17	9	23	30	*
$D[j]$ (updated)	21	13	27	34	*
Message loss or not	*	*	*	*	loss

Table 5.5 Results of message scheduling using DDS-EATS

Last we discuss the result of applying the DDS-EATS algorithm to our example. Using this technique, the messages are first sorted according to EDD. The result is the same as

Table 5.3. Currently, channel 3 has the earliest available time, so it will be assigned to transmit M_3 at time 0. As describe above, we can get the flow time of M_3 is 17 and it will not be late. Then $C[3]$ and $D[2]$ are updated and $C[1][m]$ and $D[1][n]$ are stored. After updating, channel 2 will be assigned to M_2 at time 2. Similarly, the message will not be late, too. So $C[2]$ and $D[1]$ are updated and $C[2][m]$ and $D[2][n]$ are stored. Then channel 1 becomes the channel with the earliest available time and is assigned to transmit message M_4 . Since the destination available time of it has been updated to 17, its waiting time is dependent on $D[2]$ and equals to 13. Then the flow time of it equals to 24, which is larger than its deadline. Thus M_4 is the first potential tardy message found. And from the fact that the waiting time of M_4 depends on $D[2]$, we select the messages that not only have the same destination with M_4 but have effect on the waiting time of M_4 among the messages queuing before M_4 , where we find M_3 . Since the transmission time of M_3 is larger than that of M_4 , M_3 is discarded. And because M_3 is the first message in the sequence, the value of all the $C[k]$ and $D[j]$ will be recovered to the initial value shown in Table 5.1 and the scheduling with the new sequence will be started. At current time, M_2 is the first message to be scheduled. It will select channel 3 with the earliest available time. Since the destination available time for it is 7, it is easy to get that the waiting time of M_2 equals to 3. So the flow time of M_2 is 21. Then $C[3]$ is updated to 17 and $D[1]$ is updated to 21. And $C[1][m]$ and $D[1][n]$ are stored, as well. After that, message M_4 will select channel 2 with the earliest available time and the waiting time for it is 2. Therefore the flow time of it equals to 13 and the message will not be late. $C[2]$ and $D[2]$ are updated ($C[2]=9$, $D[2]=13$). Similarly, $C[2][m]$ and $D[2][n]$ are stored. At this time, channel 2 is still the earliest available channel and will be assigned to message M_1 . Since the destination available time for M_1 is equal to 5, the waiting time for it is dependent on the channel available time, which equals to 9. So the flow time of it is equal to 27 and the deadline of it won't be violated. Similarly, $C[2]$ and $D[3]$ are updated and $C[3][m]$ and $D[3][n]$ are stored. Following the same way, all the messages are scheduled and the final results are shown in Table 5.5. The loss number is 1. From the results, we can find that after

M_3 is discarded, the remainder messages will meet their deadlines finally, which means that discarding M_3 will be more efficient.

It is evident from the results that the DDS algorithm improves the network performance in terms of loss rate when comparing with Moore and Hodgson's algorithm and EDD algorithm.

5.1.2 Performance Analysis

In this subsection, we estimate the performance of DDS, EDD and Moore and Hodgson's algorithm by theoretical analysis and simulations.

5.1.2.1 Theoretical Analysis

The objectives of the theoretical analysis are twofold. First, we prove that DDS will achieve better performance than other two algorithms. Second, we demonstrate that the proposed algorithm DDS can avoid the blocking of longer messages while guaranteeing the potential tardy messages to meet their deadlines after blocking messages are discarded

5.1.2.1.1 Comparison between DDS and EDD

We consider an EDD sequence S in which there are two tardy messages M_{α_1} and M_{α_2} , due to $d_{\alpha_1} < f_{\alpha_1}$ and $d_{\alpha_2} < f_{\alpha_2}$, where d is the deadline and f is the flow time. Then, we construct a new sequence, S' , in such a way that the messages before M_{α_2} are sorted according to DDS algorithm with other messages in S unaltered. If we can show that the performance of the EDD sequence can be improved with respect to loss rate by sorting the messages in the way of DDS algorithm, we can continue the sorting until eventually the DDS sequence is constructed. So the important part of our comparison is to prove that the new sequence S' can achieve smaller number of loss messages, namely N_T , than original sequence S .

Here we use d_i to denote the message deadline, t_i to denote the message transmission time and δ_i to denote the message idle time of each message M_i . So the holding time for the message to hold a channel, which is denoted by h_i , is equal to $t_i + \delta_i$. And the flow time of

each message is denoted by f_i and the waiting time is denoted by W_i . Without loss of generality, we may assume that the tuning time T is 0 and the propagation delay R is identical for all the nodes, which equals to 0, too.

In our system, the waiting time of the messages consists of two parts. One is the time to wait for the channel to become available, i.e., channel available time. And the other is the time to wait for the availability of the destination even if the channel has become available, i.e., message idle time. Therefore, for any message M_i , we can have $W_i = C[k]_i + \delta_i$, where $C[k]_i$ is the channel available time before the scheduling of message i . And from the relationships $f_i = W_i + t_i$, we can have $f_i = C[k]_i + t_i + \delta_i$. After scheduling message i , the channel available time $C[k]_i$ should be updated according to $C[k]_{i+1} = f_i$. So for the message queuing after the message M_i , say $M_{(i+1)}$, there should have $W_{i+1} = C[k]_{i+1} + \delta_{i+1} = f_i + \delta_{i+1}$ and $f_{i+1} = C[k]_{i+1} + t_{i+1} + \delta_{i+1} = f_i + t_{i+1} + \delta_{i+1}$. It is assumed that the channel is idle initially, so we have $f_1 = t_1 + \delta_1$. Thus for any message M_i , the channel available time before the scheduling is equal to the sum of the holding time of all the messages queuing before it, i.e., $C[k]_i = \sum_{j=1}^{i-1} (t_j + \delta_j)$ and the flow time is $f_i = \sum_{j=1}^i (t_j + \delta_j)$.

In order to compare the performance of EDD and DDS, we have to consider the causes of the tardy message M_{α_1} separately.

Case 1. The cause of the tardy message is channel unavailability.

In S :

EDD will simply drop the message M_{α_1} , so the channel available time before the scheduling of the message M_{α_1+1} will be

$$C[k]_{\alpha_1+1}(EDD) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\alpha_1} + \delta_{\alpha_1})$$

As for the second tardy message M_{α_2} , it is late because

$$d_{\alpha_2} < f_{\alpha_2}(EDD)$$

and

$$f_{\alpha_2}(EDD) = C[k]_{\alpha_1+1}(EDD) + \sum_{j=\alpha_1+1}^{\alpha_2} (t_j + \delta_j)$$

In S' :

When the first tardy message M_{α_1} found, DDS will discard the message, say M_{β_1} , with the largest holding time according to

$$(t_{\beta_1} + \delta_{\beta_1}) = \max_{i=1,2,\dots,\alpha_1} \{t_i + \delta_i\} \quad (5.1)$$

So the channel available time for the message M_{α_1+1} will be

$$C[k]_{\alpha_1+1}(DDS) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\beta_1} + \delta_{\beta_1})$$

From (5.1), we can easily find that $C[k]_{\alpha_1+1}(DDS) \leq C[k]_{\alpha_1+1}(EDD)$.

As for second tardy message M_{α_2} found, the flow time for M_{α_2} will be

$$f_{\alpha_2}(DDS) = C[k]_{\alpha_1+1}(DDS) + \sum_{j=\alpha_1+1}^{\alpha_2} (t_j + \delta_j)$$

From $C[k]_{\alpha_1+1}(DDS) \leq C[k]_{\alpha_1+1}(EDD)$, we can find that $f_{\alpha_2}(DDS) \leq f_{\alpha_2}(EDD)$. So there will be two cases:

- a. If $d_{\alpha_2} < f_{\alpha_2}(DDS)$, M_{α_2} will be late. So in this case, S' will get the same N_T as S .
- b. If $d_{\alpha_2} \geq f_{\alpha_2}(DDS)$, M_{α_2} won't be late. So in this case, S' will get smaller N_T than S .

So in this case, S' is possible to get smaller N_T than S . Also we can get the conclusion that $N_T(EDD) \geq N_T(DDS)$.

Case 2. The cause of the tardy message is destination unavailability.

In S :

EDD will simply drop the message M_{α_1} , so the channel available time before the scheduling of the message M_{α_1+1} will be

$$C[k]_{\alpha_1+1}(EDD) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\alpha_1} + \delta_{\alpha_1})$$

As for the second tardy message M_{α_2} , it is late because

$$d_{\alpha_2} < f_{\alpha_2}(EDD)$$

and

$$f_{\alpha_2}(EDD) = C[k]_{\alpha_1+1}(EDD) + \sum_{j=\alpha_1+1}^{\alpha_2} (t_j + \delta_j)$$

In S' :

When the first tardy message M_{α_1} found, DDS will determine a set of messages F ($M_{l(1)}, M_{l(2)}, \dots, M_{l(k)}$), in which the messages not only have the same destination as the message M_{α_1} but also have effect on the waiting time of M_{α_1} . Then the longest message in F will be selected and compared with M_{α_1} . After that, the longer message, say M_{d_1} , will be discarded. So

$$t_{d_1} = \max \{t_i : M_i \in F \text{ or } M_i = M_{\alpha_1}\} \quad (5.2)$$

Assume that message $M_{l(m)}$ is the message with the largest transmission time which accords with (5.2). So F will change to $F' (M_{l(1)}, M_{l(2)}, \dots, M_{l(m-1)}, M_{l(m+1)}, \dots, M_{l(k)})$ after $M_{l(m)}$ is discarded. In F' , we can find that every message after $M_{l(m)}$ will go forward one position when comparing with F . For example, $M_{l(m+1)}$ will take the place of $M_{l(m)}$, $M_{l(m+2)}$ will take the place of $M_{l(m+1)}$ and so on.

For every message, $M_{l(i)}$ in F , it cannot transmit only if the message before it, say $M_{l(i-1)}$, has finished the transmission, so we can arrive at the conclusion that the waiting time of $M_{l(i)}$ is equal to the flow time of $M_{l(i-1)}$. So the waiting time for $M_{l(i)}$ is got from $W_{l(i)} = f_{l(i-1)}$, where $f_{l(i-1)}$ is the flow time of message $M_{l(i-1)}$. And from the relationship $W_{l(i)} = C[k]_{l(i)} + \delta_{l(i)}$, where $C[k]_{l(i)}$ is the current channel available time before the scheduling of message $M_{l(i)}$, we can derive $W_{l(i)} = f_{l(i-1)} = C[k]_{l(i)} + \delta_{l(i)} \Rightarrow \delta_{l(i)} = f_{l(i-1)} - C[k]_{l(i)}$. It is easy to see that the reason of the message idle time of $M_{l(i)}$ is that $f_{l(i-1)}$ is larger than the channel available time. Therefore, when $f_{l(i-1)}$ increases, $\delta_{l(i)}$ will increase, too. Moreover, the flow time of $M_{l(i)}$

equals the flow time of $M_{l(i-1)}$ plus the transmission time of $M_{l(i)}$, namely $f_{l(i)} = W_{l(i)} + t_{l(i)} = f_{l(i-1)} + t_{l(i)}$.

As for $M_{l(m+1)}$ in F' , its waiting time will change to be dependent on $M_{l(m-1)}$ because it will take the place of $M_{l(m)}$. Hence the waiting time of $M_{l(m+1)}$ equals to $f'_{l(m-1)}$, where $f'_{l(m-1)}$ denotes the flow time of $M_{l(m-1)}$ in F' . And because the position of the messages before $M_{l(m)}$ do not change in F' , we can get $f'_{l(m-1)} = f_{l(m-1)}$. Then the message idle time of $M_{l(m+1)}$ in F' will be the same with that of $M_{l(m)}$ in F . Here we use $\delta'_{l(m+1)}$ to denote the message idle time for $M_{l(m+1)}$ in F' , then $\delta'_{l(m+1)} = \delta_{l(m)}$. And from (5.2), we have $t_{l(m)} \geq t_{l(m+1)}$, from which we can derive $f_{l(m)} = f_{l(m-1)} + t_{l(m)} \geq f'_{l(m+1)} = f'_{l(m-1)} + t_{l(m+1)}$. Also, message $M_{l(m+2)}$ in F' will take the place of $M_{l(m+1)}$ when comparing with F . From $f_{l(m)} \geq f'_{l(m+1)}$, we can get that $\delta_{l(m+1)} \geq \delta'_{l(m+2)}$. Considering in the same way, it is easy to prove that for any message $M_{l(i)}$ behind $M_{l(m)}$ we have $\delta_{l(i)} \geq \delta'_{l(i+1)}$. Similarly, since M_{α_1} will take the place of $M_{l(k)}$, we can have $\delta_{l(k)} \geq \delta'_{\alpha_1}$.

As described before, in EDD, the channel available time before the scheduling of the message M_{α_1+1} will be

$$C[k]_{\alpha_1+1}(EDD) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\alpha_1} + \delta_{\alpha_1}) = \sum_{j=1}^{\alpha_1-1} (t_j + \delta_j)$$

It can also be expressed as

$$\begin{aligned} C[k]_{\alpha_1+1}(EDD) &= \sum_{M_j \notin F} (t_j + \delta_j) + \sum_{M_j \in F} (t_j + \delta_j) \\ &= \sum_{M_j \notin F} (t_j + \delta_j) + \sum_{\substack{M_j \in F \\ l(j) < l(m)}} (t_j + \delta_j) + t_{l(m)} + \delta_{l(m)} + t_{l(m+1)} + \delta_{l(m+1)} + \cdots + t_{l(k)} + \delta_{l(k)} \end{aligned}$$

In DDS, message $M_{l(m)}$ will be discarded instead of M_{α_1} , so the channel available time before the scheduling of message M_{α_1+1} will be

$$\begin{aligned}
 C[k]_{\alpha_1+1}(DDS) &= \sum_{M_j \notin F} (t_j + \delta_j) + \sum_{M_j \in F} (t_j + \delta_j) \\
 &= \sum_{M_j \notin F} (t_j + \delta_j) + \sum_{\substack{M_j \in F \& \\ l(j) < l(m)}} (t_j + \delta_j) + t_{l(m+1)} + \delta'_{l(m+1)} + \dots + t_{l(k)} + \delta'_{l(k)} + t_{\alpha_1} + \delta'_{\alpha_1}
 \end{aligned}$$

Since for every message M_i , we have $\delta_{l(i)} \geq \delta'_{l(i+1)}$ and from $t_{l(m)} \geq t_{\alpha_1}$, we can get

$$t_{l(m)} + \delta_{l(m)} + t_{l(m+1)} + \delta_{l(m+1)} + \dots + t_{l(k)} + \delta_{l(k)} \geq t_{l(m+1)} + \delta'_{l(m+1)} + \dots + t_{l(k)} + \delta'_{l(k)} + t_{\alpha_1} + \delta'_{\alpha_1}$$

Therefore, we have $C[k]_{\alpha_1+1}(DDS) \leq C[k]_{\alpha_1+1}(EDD)$.

All in all, since every message in F has effect on the waiting time of every message queuing behind it, no matter which message is dropped, the idle time of the messages queuing behind it will be reduced simultaneously. DDS can reduce more time than EDD in this case simply because it has discarded the message with maximum transmission time in F .

As for the second tardy message M_{α_2} , the flow time for it is

$$f_{\alpha_2}(DDS) = C[k]_{\alpha_1+1}(DDS) + \sum_{j=\alpha_1+1}^{\alpha_2} (t_j + \delta_j)$$

From $C[k]_{\alpha_1+1}(DDS) \leq C[k]_{\alpha_1+1}(EDD)$, we can have $f_{\alpha_2}(DDS) \leq f_{\alpha_2}(EDD)$. So there will be two cases:

- a. If $d_{\alpha_2} < f_{\alpha_2}(DDS)$, M_{α_2} will be late. In this case, S' will get the same N_T as S .
- b. If $d_{\alpha_2} \geq f_{\alpha_2}(DDS)$, M_{α_2} won't be late. In this case, S' will get smaller N_T than S .

So in this case, S' is possible to get smaller N_T than S . Also we can arrive at the conclusion that $N_T(EDD) \geq N_T(DDS)$.

5.1.2.1.2 Comparison between DDS and Moore and Hodgson's Algorithm

Firstly, we consider an EDD sequence in which all the messages are sorted according to the deadline. Then, we construct two new sequences, S' and S'' , in which the messages are ordered according to Moore and Hodgson's algorithm and DDS algorithm respectively. If we can show that S'' can achieve better performance with respect to loss rate than S' , we can prove that DDS performs better than Moore and Hodgson's algorithm. We assume that M_{α_1}

is the first tardy message found for both algorithms due to $d_{\alpha_1} < f_{\alpha_1}$. And we use M_{α_2} to denote the second tardy message found in Moore and Hodgson's algorithm. If we can show that M_{α_2} possibly will not be late in DDS algorithm, we can prove that DDS can achieve better performance than Moore and Hodgson's algorithm. Similarly, there are two cases.

Case 1. The cause of the tardy message is channel unavailability.

In S' :

When the first tardy message M_{α_1} is found, Moore and Hodgson's algorithm will discard the message with the largest transmission time, namely M_{φ_1} . So the channel available time before the scheduling of message M_{α_1+1} will be

$$C[k]_{\alpha_1+1}(Moore) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\varphi_1} + \delta_{\varphi_1})$$

As for the second tardy message M_{α_2} , it is late because

$$d_{\alpha_2} < f_{\alpha_2}(Moore)$$

and

$$f_{\alpha_2}(Moore) = C[k]_{\alpha_1+1}(Moore) + \sum_{j=\alpha_1+1}^{\alpha_2} (t_j + \delta_j)$$

In S'' :

DDS will discard the message with the largest holding time, say M_{β_1} , which accords with (5.1). Then the for the message M_{α_1+1} , the channel available time will be

$$C[k]_{\alpha_1+1}(DDS) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\beta_1} + \delta_{\beta_1})$$

So we can derive the fact that

$$d_{\alpha_1} \geq d_{\alpha_1-1} \geq f_{\alpha_1-1} = \sum_{j=1}^{\alpha_1-1} (t_j + \delta_j) = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\alpha_1} + \delta_{\alpha_1}) \geq \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\beta_1} + \delta_{\beta_1})$$

where d_{α_1-1} is the deadline of M_{α_1-1} and M_{α_1-1} is the message right before M_{α_1} . After discarding M_{β_1} , the flow time of M_{α_1} will be $f_{\alpha_1} = \sum_{j=1}^{\alpha_1} (t_j + \delta_j) - (t_{\beta_1} + \delta_{\beta_1})$. Hence we can prove that the potential tardy message M_{α_1} will meet the deadline after M_{β_1} is discarded.

From (5.1), we can easily get that $t_{\beta_1} + \delta_{\beta_1} \geq t_{\varphi_1} + \delta_{\varphi_1}$, so

$$\sum_{j=1}^{\alpha_l} (t_j + \delta_j) - (t_{\beta_1} + \delta_{\beta_1}) \leq \sum_{j=1}^{\alpha_l} (t_j + \delta_j) - (t_{\varphi_1} + \delta_{\varphi_1})$$

i.e., $C[k]_{\alpha_1+1}(DDS) \leq C[k]_{\alpha_1+1}(Moore)$.

As for the second tardy message M_{α_2} , the flow time can be obtained from

$$f_{\alpha_2}(DDS) = C[k]_{\alpha_1+1}(DDS) + \sum_{j=\alpha_1+1}^{\alpha_2} (t_j + \delta_j)$$

From $C[k]_{\alpha_1+1}(DDS) \leq C[k]_{\alpha_1+1}(Moore)$, we can get that $f_{\alpha_2}(DDS) \leq f_{\alpha_2}(Moore)$. So

there will be two cases for M_{α_2} :

- a. If $d_{\alpha_2} < f_{\alpha_2}(DDS)$, M_{α_2} will be late. So in this case, S'' will get the same N_T as S' .
- b. If $d_{\alpha_2} \geq f_{\alpha_2}(DDS)$, M_{α_2} won't be late. So in this case, S'' will get smaller N_T than S' .

Therefore, in this case, S'' is possible to get smaller N_T than S' . Then we can draw the conclusion that $N_T(Moore \text{ and } Hodgson's \text{ algorithm}) \geq N_T(DDS)$.

Case 2. The cause of the tardy message is destination unavailability.

In S' :

Moore and Hodgson's algorithm will discard the message with the largest transmission time regardless of the cause of the tardy message.

The basic idea of Moore and Hodgson's algorithm is that once a message, say M_{α_1} , violates its deadline, the longest message before it will be discarded. Let M_{δ_1} denotes the message discarded according to Moore and Hodgson's algorithm. So there will be two cases:

- a. Discarding message M_{δ_1} did not change the destination available time for M_{α_1} or the destination available time reduced is not enough so that the message M_{α_1} is still late. In this case, the message M_{δ_1} with the largest transmission time will be

discarded from the sequence. Then the new sequence is formed and the scheduling restarts. Since M_{α_1} is still be late, it will be the potential tardy message found for the second time and then another longest message will be discarded. If the sacrifice of this longest message, say M_{δ_2} , still cannot ensure M_{α_1} will not be late, M_{α_1} will be the tardy message found for the third time. The worst case is that in order to avoid the blocking for the tardy message M_{α_1} , several messages queuing before it will be discarded. In this case, $N_T(\text{Moore and Hodgson's algorithm}) > 1$ when the potential tardy message M_{α_1} is confirmed not to be late.

b. If discarding the message M_{δ_1} changes the destination available time for M_{α_1} , the message M_{α_1} is not late anymore. In this case, M_{δ_1} should be in the set F in which all the messages not only have the same destination of M_{α_1} but also have effect on the waiting time of M_{α_1} . And because M_{δ_1} is the message with the largest transmission time, it is also the largest among the messages in F and accords with (5.2).

In S'' :

By DDS, it will select a set of messages $F (M_{l(1)}, M_{l(2)}, \dots M_{l(k)})$ as mentioned above. Then message M_{α_1} is selected according to (5.2). Because the reason for the tardy message is destination available time, the waiting time for the message M_{α_1} is $W_{\alpha_1} = f_{l(k)}$, where $f_{l(k)}$ is the flow time of the last message in the F . And it is easy to have $f_{\alpha_1} = f_{l(k)} + t_{\alpha_1}$. As addressed above, for any message i in F , we have $f_{l(i)} = W_{l(i)} + t_{l(i)} = f_{l(i-1)} + t_{l(i)}$. Therefore, we have $f_{\alpha_1} = f_{l(1)} + t_{l(1)} + \dots + t_{l(k)} + t_{\alpha_1}$.

In the following we will describe the performance comparison between S'' and S' according to the two cases of S' above one by one.

a. For case a , M_{α_1} is late because

$$d_{\alpha_1} < f_{l(k)} + t_{\alpha_1}.$$

After discarding the message M_{d_1} , the flow time of M_{α_1} will be $f_{l(k)} - t_{d_1} + t_{\alpha_1}$.

Since the deadline of M_{α_1} is larger than $M_{l(k)}$, we have

$$d_{\alpha_1} \geq d_{l(k)} \geq f_{l(k)}$$

From (5.2), we can get

$$d_{\alpha_1} \geq f_{l(k)} = f_{l(k)} + t_{\alpha_1} - t_{\alpha_1} \geq f_{l(k)} + t_{\alpha_1} - t_{d_1}$$

So we can easily prove that the message M_{α_1} will meet the deadline.

Therefore in our algorithm, we can assure that after the message M_{d_1} is discarded, the potential tardy message will not violate the deadline. From this point of view, we can prove that DDS can achieve better performance than Moore and Hodgson's algorithm.

b. Because the message discarded accords with (5.2), the message discarded by DDS will be the same as Moore and Hodgson's algorithm. Therefore, in this case DDS and Moore and Hodgson's algorithm will achieve the same performance.

All in all, in this case, S'' is possible to get smaller N_T than S' . Also we can get the conclusion that $N_T(\text{Moore and Hodgson's algorithm}) \geq N_T(\text{DDS})$.

5.1.2.2 Simulation Results

We are interested in the system performance measures under various traffic conditions and the impact on the system performance from a wide range of system parameters (in particular the number of channels, the tuning overhead of the transceivers and the message time constraints). In this section, we carry out the performance studies of the proposed scheduling technique DDS and also compare it with Moore and Hodgson's algorithm and EDD algorithm by extensive simulation experiments. The following paragraphs provide a discussion of the design of these experiments and their results.

5.1.2.2.1 Experimental Design

The following summarizes the main assumptions used in the simulation:

- The number of nodes in the system is set to 50;
- The fiber is divided into five channels, one is control channel and four are data channels;
- The system is synchronized with the slot time equals to a data packet transmission time;
- The length of the messages is given in number of slots, which is a random variable following an Exponential distribution with a mean value 20;
- The propagation delay is assumed to be identical for all the nodes and given in number of slots;
- Messages are generated at each node following an independent and identical Poisson process;
- Traffic load ρ is defined as the ratio of the arrival rate over the service rate, which ranges from 0.4 to 1;
- The destination nodes are selected according to a uniform probability distribution;
- For the real-time messages, the message time constraint is expressed as message deadline. In order to avoid the case that the deadline of one message is smaller than its length or the propagation delay, which means that the message violates its deadline when it is generated, we will define the message laxity first. Message laxity can be expressed through the following relationship:

$$\text{message deadline} = \text{message laxity} + \text{message length} + \text{propagation delay}$$

In this section, we assume that message laxity is a random variable following an Exponential distribution with a mean value 50. So, the deadline of each message can be obtained by adding the laxity, the length and the propagation delay. Moreover, we assume that the deadline of each batch of the messages is calculated from the time that this batch of the messages starts to be scheduled.

In order to assume the propagation delay, we need to get the ratio of propagation delay to the packet transmission time. From the fact that the size of Ethernet frames ranges from 64 to 1518 bytes, we approximately assume that the length of one data packet in our system is 1000 bits. And the four data channels are of the same speed that is set to 10 Gb/s. So it will take one packet $0.1 \mu\text{s}$ to transmit. For a typical local area network with a diameter of two kilometers, the propagation delay is about $10 \mu\text{s}$. Then the ratio of propagation delay to packet transmission time is 100. According to the above, we assume that the slot time equals to one data packet transmission time, therefore, the propagation delay is 100 time slots.

Metrics of performance in the experiments are the loss rate and throughput. The message loss rate is measured in two ways. One is defined as the ratio of the number of messages dropped over total number of messages entering into the network, which we will call message loss rate hereafter. And the other is defined as the ratio of the number of packets dropped over the total number of packets entering into the network, which is called packet loss rate. The throughput is defined as the number of packets transmitted per time slot.

5.1.2.2.2 Experimental Results

In the following section, we will present the experimental results to show the real-time properties of the network using DDS algorithm when it handles real-time messages.

Figure 5.1 presents message loss rate versus traffic load ρ . In this set of experiments, we assume that the number of channels is 4 and the tuning time is fixed to 0. As shown in the figure, the DDS algorithm outperforms other two algorithms. The reason why the DDS algorithm can work better than Moore and Hodgson's algorithm is that DDS algorithm is developed depending on the characteristic of single-hop star networks and considers not only the channel available time but also the receiver available time. So it has the ability to efficiently discard the message which will block others and consequently more messages can be successfully scheduled to meet their deadlines. Moreover, it can ensure that potential tardy message will not be late after the blocking message is discarded. Among the three

algorithms, EDD performs worst in terms of loss rate. This is because EDD just simply drops the tardy message, regardless of the message length.

Figure 5.1 Impact of traffic load on message loss rate of DDS

Figure 5.2 Impact of tuning time on message loss rate of DDS

Figure 5.2 compares the characteristic of mean message loss rate under varying tuning time. The number of channels is fixed to 4 and the traffic load ρ is set to 0.9. It is easy to see that the loss rate increases with the tuning time for these three algorithms. This is because that when the tuning time is increased, it is equivalent to increase the waiting time since it takes longer time to complete a message transmission. However, among these three algorithms, DDS consistently demonstrates its superior performance to the other two algorithms across the whole tuning time spectrum. This is expected since DDS has the

ability to avoid the blocking message to block other messages and, therefore, improve the performance of the network.

Figure 5.3 shows the effect of varying number of channels on the message loss rate for three algorithms. Here we assume that the tuning time is fixed to 0 and the traffic load ρ is set to 0.9. We clearly see that the loss rate decreases for all of the algorithms when the number of channels increases. Once again, we observe that DDS achieves better performance than the other two algorithms in terms of message loss rate, especially when the number of channels is small. This is because, when the number of channel is small, the competition for channels becomes drastic and the effect of the message sequencing becomes important. DDS has the ability to efficiently discard the message which will block other messages, so it can achieve better performance than other two algorithms in terms of loss rate. When the number of channels becomes large, the performance of all the algorithms will flatten out and further increase in the number of channels will not induce any change, which means that data channels are no longer a bottleneck.

Figure 5.3 Impact of number of channels on message loss rate of DDS

Figure 5.4 shows the relationship between the message loss rate and message laxity. Here we assume that the number of channels is 4, the tuning time is 0 and the traffic load ρ is 0.9. We can clearly see that the DDS algorithms have achieved a significant improvement in terms of loss rate. This is because that DDS takes not only the channel available time but

also the destination available time into account and it has the capability to efficiently discard the longer messages so that more messages queuing behind can meet the deadline. As shown in the figure, the loss rate decreases as the laxity increases for these three algorithms. For the DDS algorithm, when the message deadline increases from 30 to 100, the message loss rate gets nearly 95% of its original value decreased. However, for the EDD algorithm, the message loss rate gets 90% of its original value decreased. This implies that DDS algorithm have better real-time performance.

Figure 5.4 Impact of laxity on message loss rate of DDS

The former experiments mainly focus on the performance measure of the message loss rate, which is defined as the fraction of messages violating the deadlines. The following experiments will take the actual packet size into account to study whether the sacrifice of the blocking messages will degrade the throughput or increase the loss rate in terms of the fraction of packets violating the deadline.

Figure 5.5 presents the throughput versus traffic load ρ using different algorithms. As mentioned above, the throughput is defined as the number of packets transmitted per time slot. And the message loss rate is defined as the ratio of the number of messages dropped over total number of messages entering into the network, in which one message consists of several packets. So the algorithm with the smaller message loss rate may not achieve the higher throughput. From the figure, we can see that among the three algorithms, DDS still

achieves best performance. This result shows that although DDS drops “longer” messages when the potential tardy message found, the total number of packets dropped by it is still smaller. This is mainly because that the sacrifice of selected messages can resolve blockings so that several shorter potential tardy messages following them will meet their deadlines successfully. Moreover, the “longer” message is not in terms of message length. But in DDS algorithm, we consider that a message is long due to its holding time instead of just only transmission time. Recall that holding time = message transmission time + message idle time, the message with the largest holding time may not the one with the largest transmission time. While Moore and Hodgson’s algorithm always drops the messages with the largest transmission time, so its throughput will be smaller than DDS. Therefore, even if the message loss rate of the Moore and Hodgson’s algorithm is smaller than EDD, the total number of packets dropped by it is larger than EDD.

Figure 5.5 Impact of traffic load on throughput of DDS

Figure 5.6 depicts the loss rate when the packet size is taken into account, namely packet loss rate, versus the traffic load ρ . The objective of this study is to show whether the discarding of the blocking messages will increase the loss rate when considering the actual packet size. As the figure shows, DDS still performs the best among these three algorithms. This is mainly because that several shorter potential tardy messages will meet their deadlines after one selected blocking message is discarded. And the total length of the shorter messages will be larger than the length of one blocking message. Therefore, the performance

of packet loss rate will not be degraded. However, in this experiment, Moore and Hodgson's algorithm cannot work better than EDD. This is expected since Moore and Hodgson's algorithm always discard the messages with the largest transmission time and especially it cannot ensure this sacrifice will be efficient.

Figure 5.6 Impact of traffic load on packet loss rate of DDS

5.1.3 Summary

In this section, we propose a novel scheduling scheme, namely Differentiated Dropping Scheduling (DDS), which is designed to handle real-time traffic in passive star coupled WDM optical networks. The scheduling algorithm is designed and developed based on the destination and channel availability of a single-hop passive-star coupled WDM optical network. This algorithm can efficiently avoid blocking of longer messages in order to make more messages meet their deadlines resulting in less message loss rate. The results from extensive simulation experiments and mathematical estimations have shown that our proposed algorithm works better than the algorithms with the similar functions in terms of message loss rate, packet loss rate and throughput.

5.2 Cost-Based Priority Scheduling (CBPS) Algorithm

There are a large amount of research results on scheduling algorithms to provide real-time service to the messages with time constraints in various network environments. Some reports on medium access control protocols, which provide real-time service to time-constrained

messages on WDM optical networks, have been found such as the ones in [46][51][54]. Some other reports try to provide real-time service to integrated traffic by designing combined and/or dynamic priority schemes [67][68][69][70][71][72]. However, there are few reports on providing differentiated services in WDM optical networks. In this section, we develop a novel scheduling algorithm, named as Cost-Based Priority Scheduling (CBPS) algorithm, to provide differentiated service to messages with different time constraints. The new algorithm differs from the others in the following aspects. First, it has the ability of preventing channel collisions and destination conflicts. Second, it uses a combined dynamic priority scheme considering not only the laxity but also the message length. Third, CBPS has the advantage of reducing the loss rate of real-time messages without sacrificing the performance of non real-time messages.

We evaluate our algorithm by comparing its performance with Shortest Message First (SMF) and Moore and Hodgson's algorithm [66] using extensive discrete-event simulations. The results demonstrate that the algorithm can reduce not only the message loss rate but also the average message delay when an integrated traffic is applied to the network so that the transmission of both messages with and without time constraints could be benefited.

5.2.1 Scheduling Algorithm

5.2.1.1 Algorithm Description

For our proposed scheduling algorithm, we adopt EATS algorithm as the technique to assign data channels and transmission time slots to selected messages.

To determine the message transmission sequence, priority schemes are adopted widely. Messages can be sequenced for transmission after the entire control frame has been received by all the nodes. As part of the distributed scheduling algorithm at each node, we can apply a priority mechanism to sequence the messages represented in the control frame. This sequencing will impose an order on the messages that are assigned to the channel. Recently there are various policies to assign priorities to the transmitted messages. However, some of

these priority schemes schedule message transmission without considering the time constraints of the messages resulting in high message loss rate. Some other priority schemes schedule message transmission just based on the stringency of the messages resulting in the sacrifice of the performance of the messages without time constraints. Here we develop a combined dynamic priority scheme as well as a new scheduling algorithm, name as Cost-Based Priority Scheduling (CBPS) algorithm, to provide differentiated services in order to benefit both types of the messages.

When the cost-based priority assignment scheme is considered, we admit that if a message stays in the network longer, it will occupy more of the network resources. Or it costs more to be transmitted by the network. We consider the message length, the relative laxity of a message, and the importance (or static priority) of a message to be the factors to the cost of message transmission. We have the following formula to assign the priority to the messages, which are going to be transmitted.

$$P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$$

where P_i is the dynamic priority assigned to message i , SP_i is the static priority of message i which shows the different importance of different kinds of messages (The message with the least value SP_i has the highest priority.), T_{ni} is the normalized relative laxity of message i which presents the stringency of the message and it should be less or equal to 1, L_{ni} is the normalized length of message i which should be less or equal to 1, α is the normalized parameter for the static priority, β is the normalized parameter for the relative laxity and γ is the normalized parameter for the message length. The priority assigned to a message is a dynamic one in the sense that it changes with time and network situation. It is also an integrated scheme in the meaning that it is a combined priority assignment scheme. By this priority scheme, the message with the least value of P_i will be considered as the message with the highest priority, which takes the least cost of network resources.

In order to get the normalized value of message length and message laxity, we need to find the maximum value of them. The message length and laxity are random variables following different Exponential distributions with different mean values. For simplicity, we consider the 95% data of the probability density function. For a random variable following an Exponential distribution with mean message length as $1/\lambda$, the maximum value of that variable can be approximated as:

$$\int_0^x \lambda e^{-\lambda x} dx = 0.95 \Rightarrow x_{\max} = \frac{3}{\lambda} \quad (5.3)$$

If $1/\lambda=20$ for message length, we have $x_{\max}=L_{\max}=60$. If $1/\lambda=25$ for message laxity, we have $x_{\max}=T_{\max}=75$. Then the normalized value of the message length equals to $\text{length}/L_{\max}$. And the normalized value of the relative laxity is $\text{laxity}/T_{\max}$.

We combine the proposed priority scheme as well as the message sequencing technique with the channel assignment algorithm EATS to form the new algorithm, CBPS. We describe it formally as follows.

We assume that there are N nodes and C channels in the network. The propagation delay is R_j and the transceivers' tuning time is T . Messages have variable lengths that follow an Exponential distribution. And the length of message i is denoted by L_i . Message laxities also follow an Exponential distribution. And the laxity of message i is denoted by T_i . In addition, messages are indicated with different importance. Similar to MFTS algorithm in Chapter 4, we will use D_j to stand for $\text{RAT}[j]$ and C_k to stand for $\text{CAT}[k]$. $D_j=m$, where $j=1 \cdots N$, means that node j 's receiver will become free after m time slots. $C_k = n$, where $k=1 \cdots C$, means that channel k will become available after n time slots.

Begin:

Transmit a control packet on the control channel for every node;

Wait until the control frame returns;

Start:

Assign priority to messages based on the formula;

Choose the message with the highest priority, say M_i ;

Search the earliest available data channel k such that $C_k \leq C_n$, $\forall n \neq k$, $k \geq 1$, $n \leq C$;

Calculate $r=D_j+T$, $t_1=\max(C_k,T)$, $t_2=\max(t_1+R_j,r)$;

Schedule the transmission time at $t=t_2-R_j$;

If the message is a hard real-time message,

Test whether the message laxity can be exceeded;

If transmission time $t >$ message laxity T_i ,

Drop the message;

Otherwise update $D_j=t_2+L_i$, $C_k=t_2-R_j+L_i$;

Else, update $D_j=t_2+L_i$, $C_k=t_2-R_j+L_i$;

Return to **Start** if any messages not scheduled;

Otherwise to **End**;

End.

The complexity of the CBPS algorithm can be evaluated according to its operation. It can be found that the algorithm has two searching procedures to schedule each message. The first is to search the message with the least cost among those presented in one control frame. The other procedure is to search for a channel with the earliest available time among all the channels in the network. Since we have assumed the maximum number of message requests per source node is 1, the maximum number of messages requests per control frame is equal to the number of nodes (N). If the number of nodes is larger than the number of channels, the complexity of the algorithm to schedule one message will be $O(N)$ in the worst case, where N is the number of nodes in the network. To schedule all the messages represented in one control frame will take $O(N \log_2 N)$.

5.2.1.2 Example

In this section, we discuss the details of three algorithms, i.e., Moore and Hodgson's algorithm, SMF and CBPS, in the context of an example. The current C_k and D_j are given in

Table 5.6, and the information of the messages is given in Table 5.7, which contains twenty messages $M_1, M_2, \dots, M_{20}$, destined to node $j_1, j_2, \dots, j_{20}$ respectively. The number of data channels (C) is equal to 4, tuning time of the transceiver (T) is 0, and the propagation delay (R) is set to 4 for all the nodes. $M_1, M_4, M_{10}, M_{13}, M_{14}$ and M_{16} are hard real-time messages with the highest static priority. M_5, M_7, M_8 , and M_{11} are soft real-time messages. And $M_2, M_3, M_6, M_9, M_{12}, M_{15}, M_{17}, M_{18}, M_{19}$, and M_{20} are non real-time messages with the lowest priority. In this example, we assume that $\alpha = 0.1$, $\beta = 1$ and $\gamma = 1$. Since the laxity of the non real-time messages is infinity, we assume that the laxity of them is equal to $2 * T_{max}$, where T_{max} equals to 75 as described above. With these assumptions, for a non real-time message i , we have $P_i = 0.1 * SP_i + 2 * T_{max} / T_{max} + L_{ni} = 0.1 * SP_i + 2 + L_{ni}$. And for a real-time message i , we have $P_i = 0.1 * SP_i + T_{ni} + L_{ni}$. Because L_{ni} and T_{ni} are normalized value, and both of them are less than 1, the value P of real-time messages will be less than $0.1 * SP_i + 2$. Together with the fact that the value SP of real-time messages is smaller than that of non real-time message, it is easy to get that the value P of non real-time messages is larger than that of real-time messages all the time. That is, the non real-time messages will not be served first than real-time messages. From the priority formula, we can get the P_i of $M_1, M_2, \dots, M_{20}$, which is shown in Table 5.7.

$C_1=35, C_2=11, C_3=24, C_4=0$
$D_j=0$ for all the j and $0 \leq j \leq N$
$T=0, R=4$

Table 5.6 Current global information in the example of CBPS

Messages	Destination node j	Packet length L	Laxity T	Static Priority SP	Dynamic Priority P
M_1	j_1	60	14	0	1.18
M_2	j_2	22	150	2	2.56
M_3	j_3	4	150	2	2.26
M_4	j_4	9	13	0	0.32
M_5	j_5	43	53	1	1.52
M_6	j_6	10	150	2	2.36
M_7	j_7	16	13	1	0.54
M_8	j_8	7	9	1	0.33
M_9	j_9	39	150	2	2.85
M_{10}	j_{10}	15	66	0	1.13
M_{11}	j_{11}	4	55	1	0.9
M_{12}	j_{12}	15	150	2	2.45
M_{13}	j_{13}	15	4	0	0.3

M_{14}	j_{14}	16	33	0	0.7
M_{15}	j_{15}	3	150	2	2.25
M_{16}	j_{16}	10	12	0	0.32
M_{17}	j_{17}	66	150	2	3.3
M_{18}	j_{18}	6	150	2	2.3
M_{19}	j_{19}	20	150	2	2.53
M_{20}	j_{20}	32	150	2	2.73

Table 5.7 Information of the messages in the example of CBPS

We start our discussion by observing the behavior of the Moore and Hodgson's algorithm in this example. According to Moore and Hodgson's algorithm, messages $M_1, M_2, \dots, M_{20}$ are first sorted according to minimum laxity first (MLF) to $M_{13}, M_8, M_{16}, M_4, M_7, M_1, M_{14}, M_5, M_{11}, M_{10}, M_2, M_3, M_6, M_9, M_{12}, M_{15}, M_{17}, M_{18}, M_{19}, M_{20}$. Then the EATS technique is used to assign messages to data channels. At this time, channel 4 has the earliest time 0, so it will be assigned to transmit M_{13} at time 0. According to EATS, the transmission time for M_{13} is 0. So M_{13} will not be late and the channel available time as well as destination available time will be updated after scheduling. And then M_8 is assigned to channel 2 with the earliest available time. M_8 is a soft real-time message and it will not be late, too. Following this way, we can find the first message to be late, namely M_{16} . According to Moore and Hodgson's algorithm, M_{13} is the message with the largest length among the messages before M_{16} and it will be removed. After that, the C_i and D_j will be recovered to the original value as in Table 5.6 and the messages will be rescheduled. At this time, M_8 will be the first message to be scheduled. It will select channel 4 with the earliest available time. According to EATS, the transmission time for it will be 0. So M_8 will not be late and the channel available time as well as destination available time will be updated after scheduling. Following the same method, we can find that at this time the first message to be late is M_1 . Comparing M_1 with the messages queuing before M_1 , M_1 is the message with the largest length. So it will be removed from the queue. After that, the global information will be recovered and all the messages will be rescheduled. The final sequence of the messages is $M_8, M_{16}, M_4, M_7, M_{14}, M_5, M_{11}, M_{10}, M_2, M_3, M_6, M_9, M_{12}, M_{15}, M_{17}, M_{18}, M_{19}, M_{20}, M_{13}, M_1$, where M_1 and M_{13} are lost messages. The final result of scheduling all the messages using

Moore and Hodgson's algorithm is shown in Figure 5.7. The average delay of 18 messages, excluding the loss message M_1 and M_{13} , using Moore and Hodgson's algorithm is calculated as: $(7+17+20+33+36+67+37+50+58+41+51+89+66+61+127+72+87+104)/18=56.8$. The loss rate can be expressed as: $2/20=10\%$.

Figure 5.7 Schedule using Moore and Hodgson's algorithm
Loss messages: M_1 and M_{13}

Next we discuss the result of applying the CBPS to our example. Using this technique, the message queuing at the source nodes are first sorted according to dynamic priority scheme, which includes the static priority, message length and message laxity. According to the dynamic priority shown in Table 5.7, messages $M_1, M_2, \dots, M_{20}$ are sorted to $M_{13}, M_4, M_{16}, M_8, M_7, M_{14}, M_{11}, M_{10}, M_1, M_5, M_{15}, M_3, M_{18}, M_6, M_{12}, M_{19}, M_2, M_{20}, M_9, M_{17}$. At the beginning, the message M_{13} will select channel 4 with the earliest available time. And the transmission time for it is 0. So it will not be late and C_4 will be updated to 15. After that, M_4 will select channel 2 and the transmission time for it equals to 11, which is less than its laxity. Therefore, M_4 will not be late, too. As for M_{16} , it will select channel 4 with the earliest available time and the transmission time is 15, which is larger than its laxity 12. Because it is a hard real-time message, it will be dropped. Following the same way, all the messages will be scheduled and the final result is shown in Figure 5.8. The loss messages are M_1 and M_{16} . The average delay of the 18 messages, excluding the loss messages, is calculated as $(15+20+22+36+38+28+43+78+39+42+45+52+58+65+74+90+104+140)/18=54.9$. The loss rate can be obtained from: $2/20=10\%$.

Figure 5.8 Schedule using CBPS

Loss messages: M_1 and M_{16}

Last we discuss the result of applying the SMF to our example. SMF first sequences the messages according to their message length and then EATS is called to assign the messages to the data channels. Therefore messages $M_1, M_2, \dots, M_{20}$ can be sequenced to $M_{15}, M_3, M_{11}, M_{18}, M_8, M_4, M_6, M_{16}, M_{10}, M_{12}, M_{13}, M_7, M_{14}, M_{19}, M_2, M_{20}, M_9, M_5, M_1, M_{17}$, in increasing order of their lengths. They are then assigned to the channels. M_{15} is the first message to be scheduled. At this point of time, channel 4 will be assigned to it and the transmission time for it is 0. Then it will not be late and C_4 will be updated to 3. After that, channel 4 is still the one with the earliest available time and it will be assigned to M_3 . So the transmission time for M_3 is 3 and it will not be late, too. In the same way, all the messages will be scheduled and the final result is shown in Figure 5.9. The loss messages are M_1, M_4, M_{13} and M_6 . The average delay of 16 messages, excluding the loss messages, using SMF can be calculated as: $(3+7+11+17+18+27+33+39+43+49+55+61+75+88+98+127)/16=46.9$. The loss rate is: $4/20=20\%$.

Figure 5.9 Schedule using SMF

Loss messages: M_1, M_4, M_{13} and M_{16}

It is evident from the results of the example that CBPS improves the loss rate when compared to the SMF and the average delay when compared to Moore and Hodgson's

algorithm. This is attributed to the fact that CBPS considers not only message length but also message laxity.

5.2.2 Performance Analysis

In this subsection, we will carry out the performance evaluation of CBPS algorithm by both mathematical analysis and experimental results.

5.2.2.1 Theoretical Comparison

5.2.2.1.1 Comparison on average delay

Since the probability that two messages are destined to the same destination is small, it is assumed to be zero in this subsection for simplicity. Therefore, destination constraints are removed so that our system is changed to be close to a simple system. According to [64], SMF can reduce the mean flow time, namely average delay, in the simple systems. That is, SMF can achieve the smaller delay than other two algorithms. Therefore, only the performance comparison between Moore and Hodgson's algorithm and CBPS is presented as follows.

The way we use to compare the performance is adjacent pairwise interchange [64]. We consider a Moore and Hodgson's sequence in which there must be a pair of adjacent messages M_i and M_j , with M_i preceding M_j due to $T_i \leq T_j$, where T is the laxity of the messages. Then, we construct a new sequence, S' , in such a way that only messages M_i and M_j are sorted according to CBPS algorithm with other messages in S unaltered. If we can show that the performance of the Moore and Hodgson's sequence can be improved with respect to mean flow time by sorting an adjacent pair of messages in the way of CBPS algorithm, we can continue the sorting until eventually the CBPS sequence is constructed. So the important part of our comparison is to prove that the new sequence S' can achieve smaller mean flow time than original sequence S .

We assume that in sequence S , M_i precedes M_j . After sequence S is constructed, there are two situations for these two messages. One situation is that these two messages are

interchanged so that M_j precedes M_i . The other situation is that these two messages are not interchanged so that the sequence S' is the same with S . Figure 5.10 depicts the former situation, in which the two messages are interchanged. As figure shows, A denotes the set of messages preceding messages M_i and M_j in both sequences. B denotes the set of messages following M_i and M_j in both sequences. As we have mentioned above, messages of A and B will leave unchanged after the new sorting. t_A is the point in time at which message i begins in S and at which message j begins in S' . We temporarily adopt the notation $F_k(S)$ to represent the flow time of message k under schedule S and $F_k(S')$ to represent the flow time of message k under schedule S' . In addition, we use L_k to denote message length.

Figure 5.10 Interchange of adjacent messages

From $\bar{F} = \frac{\sum_{k=1}^N F_k(S)}{N}$, N is constant for all the algorithms, thus what we need to compare is

$\sum_{k=1}^N F_k(S)$ for the two algorithms. For the easy comparison, it is assumed that $\alpha = 0$,

$\beta = 1$ and $\gamma = 1$, so the formula to assign priority in CBPS is $P_k = T_k + L_k$. In addition, the number of data channels is assumed to be 1. The receiver available times for M_i and M_j are assumed to be 0. And the tuning time T and the propagation delay R (identical for all the nodes) are assumed to be 0.

1. If $T_i + L_i \leq T_j + L_j$, according to $P_k = T_k + L_k$, M_i still precedes M_j in S' . So S' will get the same sequence as S . Thus $F_i(S) + F_j(S) = F_i(S') + F_j(S')$.

From

$$\begin{aligned}\sum_{k=1}^N F_k(S) &= \sum_{k \in A} F_k(S) + (t_A + L_i) + (t_A + L_i + L_j) + \sum_{k \in B} F_k(S) \\ \sum_{k=1}^N F_k(S') &= \sum_{k \in A} F_k(S') + (t_A + L_i) + (t_A + L_i + L_j) + \sum_{k \in B} F_k(S)\end{aligned}$$

We obtain $\sum_{k=1}^N F_k(S) = \sum_{k=1}^N F_k(S') .$

2. If $T_i + L_i > T_j + L_j$, from the fact that $T_i \leq T_j$, we have

$$L_i > L_j. \quad (5.4)$$

In S

According to Moore and Hodgson's algorithm, M_i precedes M_j . Thus according to Figure 5.10, it is easy to obtain

$$F_i(S) = t_A + L_i$$

$$F_j(S) = F_i(S) + L_j = t_A + L_i + L_j$$

In S'

According to CBPS, M_j precedes M_i . Thus

$$F_j(S') = t_A + L_j$$

$$F_i(S') = t_A + L_j + L_i$$

So

$$\begin{aligned}\sum_{k=1}^N F_k(S) &= \sum_{k \in A} F_k(S) + (t_A + L_i) + (t_A + L_i + L_j) + \sum_{k \in B} F_k(S) \\ \sum_{k=1}^N F_k(S') &= \sum_{k \in A} F_k(S') + (t_A + L_j) + (t_A + L_j + L_i) + \sum_{k \in B} F_k(S)\end{aligned}$$

Then

$$\sum_{k=1}^N F_k(S) - \sum_{k=1}^N F_k(S') = L_i - L_j$$

And from (5.4), we have

$$\sum_{k=1}^N F_k(S) - \sum_{k=1}^N F_k(S') > 0$$

Thus S' may have smaller mean flow time than S and Moore and Hodgson's algorithm can be improved by the sorting of messages M_i and M_j according to CBPS. If the adjacent messages are sorted according to CBPS iteratively, we can get the sequence of CBPS. Also we can prove that $\bar{F}_{CBPS} < \bar{F}_{Moore}$. So, for these three algorithms, $\bar{F}_{SMF} < \bar{F}_{CBPS} < \bar{F}_{Moore}$.

5.2.2.1.2 Comparison on Loss Rate

In the simple systems, Moore and Hodgson's algorithm works best among the three algorithms concerning N_T [64], which is the number of loss messages. Hence, only the comparison between SMF and CBPS is presented as follows.

From $G = \frac{N_T}{N}$, where G is loss rate, it is easy to find that what we need to compare between SMF and CBPS is N_T . The method we use is the same with above. Similarly, we assume that $\alpha = 0$, $\beta = 1$ and $\gamma = 1$, so the formula to assign the priority in CBPS becomes $P_k = T_k + L_k$. In addition, the number of data channels is assumed to be 1. The receiver available times for M_i and M_j are assumed to be 0. And tuning time T and propagation delay R (assumed to be identical for all the nodes) are assumed to be 0.

Consider a sequence S which is sorted according to CBPS sequence. That is, for any two pairs of adjacent messages, M_i and M_j , with M_j following M_i , we have $T_i + L_i \leq T_j + L_j$. Now construct a new sequence, S' , in which message M_i and M_j are sequenced according to SMF with other messages in S unaltered. Now we need to prove that the new sequence S' will get larger loss rate than the original sequence S .

Similarly, we temporarily adopt the notation $F_k(S)$ to represent the flow time of message k under schedule S and $F_k(S')$ to represent the flow time of message k under schedule S' . In addition, we adopt $N_T(S)$ to represent the number of loss messages under S and $N_T(S')$ to represent the number of loss messages under S' .

According to the assumption above, by CBPS, M_j follows M_i , so

$$F_i(S) = t_A + L_i$$

$$F_j(S) = F_i(S) + L_j = t_A + L_i + L_j$$

Since M_i and M_j will not be late in CBPS, then

$$T_i \geq t_A \tag{5.5}$$

$$T_j \geq t_A + L_i$$

In order to evaluate the performance of the sequence S' , there will be two cases.

1. If $L_i \leq L_j$, S' will get the same sequence as S , so $N_T(S) = N_T(S')$.
2. If $L_i > L_j$, the situation will be the same as that in Figure 5.10.

In S' , M_j precedes M_i , then

$$F_j(S') = t_A + L_j$$

$$F_i(S') = t_A + L_j + L_i$$

For message M_j , from (5.5), we have $T_j \geq t_A$.

So the laxity of M_j will not be exceeded in S' , i.e., M_j will not be late.

On the other hand, for M_i , there will be two cases:

- a. If $T_i \geq t_A + L_j$, M_i will not be late, too. So in this case, S' will get the same loss number as S , namely $N_T(S) = N_T(S')$.
- b. If $T_i < t_A + L_j$, M_i will be late and it will be dropped. So in this case, S' will get larger loss number than S , namely $N_T(S) < N_T(S')$.

All in all, the sorting of the adjacent messages in CBPS according to SMF will be possible to increase the loss rate. If the sorting of the adjacent messages according to SMF can continue iteratively, we can get the sequence of SMF. Also we can prove that $N_T(\text{CBPS}) < N_T(\text{SMF})$. So for these three algorithms, we have $N_T(\text{Moore}) < N_T(\text{CBPS}) < N_T(\text{SMF})$.

5.2.2.2 Experimental Evaluation

In this section, we discuss the results of a set of experiments that evaluate the performance of the proposed scheduling technique CBPS. We also compare it with the Moore and

Hodgson's algorithm and the SMF algorithm. The objectives of the simulation are twofold. First, we demonstrate that CBPS can benefit the transmission of both types of messages with and without time constraints. Second, we demonstrate that CBPS will have different performance when changing the parameters in the priority formula. The following subsections provide a discussion of the design of these experiments and their results.

5.2.2.2.1 Experimental Design

The following parameters have been involved in the experiment design. The number of nodes (N) is set to 50. The Round-trip propagation delay (R) is assumed to be identical for all the nodes and equals to 10 time slots. Message length varies according to an Exponential distribution with a mean value as 20 time slots. Message arrivals at each source node comply with an independent Poisson process. Traffic load across the entire network ranges from 0.4 to 1. The destination nodes are selected according to a uniform probability distribution. And message laxity is a random variable following an Exponential distribution with mean laxity as 25 time slots. In our system, we apply two types of traffic to the network. One type is the messages with time constraints. The other type is the messages without time constraints. Each type of the traffic occupies 50% of the total population. For real-time messages, one half of them are hard real-time messages. Another half of them are soft real-time messages. The behavior of the candidate algorithms is observed over a simulation period of 1,000,000 time slots. Metrics of performance in the experiments are the message loss rate, message delay and throughput. The message loss rate is defined as the ratio of the number of messages dropped to total number of messages entering into the network. The message delay is defined as the duration from the time a message is scheduled to the time the message finishes transmission. And the throughput is defined as the number of messages that are transmitted per time slot.

5.2.2.2.2 Experimental Results

a. Message Delay/Loss Rate/Throughput vs. Traffic Load

In this set of experiments, we assume that the number of channels C is 4, the tuning time is 0, α is 0.1, β is 1, and γ is 1.

Figure 5.11 compares the mean delay of real-time messages using three algorithms under varying loads in a system. From the figure, we can see that CBPS significantly outperforms other two algorithms. This is because, according to the formula $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$, CBPS can arrange the real-time messages to be served earlier than non real-time messages. As such, it is capable to achieve better performance than SMF, which sequences the messages without the consideration of time constraints. On the other hand, although Moore and Hodgson's algorithm can let the real-time messages to be served first, it doesn't consider message length as one item that has an effect on scheduling. So CBPS can achieve better performance in terms of average delay than Moore and Hodgson's algorithm.

Figure 5.11 Impact of traffic load on mean delay of real-time messages of CBPS

Figure 5.12 shows the mean delay of non real-time messages versus traffic load. It is easy to find that SMF outperforms the other two algorithms. However, it is of interest to note that CBPS algorithm has improved the network performance in the sense that the mean message delay of it is smaller than that of Moore and Hodgson's algorithm. The reason of this improvement is that when the message transmission is being scheduled, the CBPS has considered not only the time constraints of the messages but also the message length, which

can also incur cost of the network resources. The messages with shorter message length, which incur less cost, will be transmitted at earlier time.

Figure 5.12 Impact of traffic load on mean delay of non real-time messages of CBPS

Figure 5.13 illustrates the overall mean delay of both types of the messages versus traffic load. From the figure, it is obvious that when the traffic load increases, SMF performs best among these three algorithms. However, CBPS has improved the network performance in the sense that the mean delay of CBPS algorithm is about 85% of that of Moore and Hodgson's algorithm when the traffic load is 1.

Figure 5.13 Impact of traffic load on mean delay of CBPS

Figure 5.14 shows the real-time performance of the system using three different scheduling algorithms. The performance of Moore and Hodgson's algorithm is the best

among these three algorithms. However, CBPS has much improved the real-time performance in terms of that the message loss rate of CBPS is only 50% of that of SMF when the traffic load is high. The reason of the improvement is that when the order of the message transmission is considered by CBPS, the message constraint has been counted to be a very important factor. The message with less laxity will be transmitted at earlier time.

Figure 5.14 Impact of traffic load on loss rate of real-time messages of CBPS

Figure 5.15 Impact of traffic load on throughput of CBPS

Figure 5.15 presents the throughput versus traffic load using different algorithms. The throughput is defined as the number of messages transmitted per time slot. Thus, the larger the loss rate, the smaller the throughput. As observed in the previous study, Moore and

Hodgson's algorithm has the smallest loss rate, so it achieves the largest throughput. Similarly, SMF algorithm has the largest loss rate, so it achieves the worst performance in terms of throughput. From the figure, we can see that CBPS can get similar performance as Moore and Hodgson's algorithm due to its consideration of time constraints.

b. Message Delay/Loss Rate vs. Number of Channels

We assume that the tuning time is 0, α is 0.1, β and γ are 1, and the traffic load in this set of experiments is set to 0.9.

Figure 5.16 shows the effect of varying the number of channels on the mean delay for three algorithms. We clearly see that the delay decreases for all of the algorithms when the number of channels increases. Once again, we observe that CBPS achieves better performance than Moore and Hodgson's algorithm in terms of delay, especially when the number of channels is small, due to its consideration of not only message laxity but also message length. When the number of channels becomes large, the performance of all the algorithms will flatten out and further increase in the number of channels will not induce any change, which means that data channels are no longer a bottleneck.

Figure 5.16 Impact of number of channels on mean delay of CBPS

Figure 5.17 depicts the loss rate versus the number of channels. It is clearly that the loss rate of all the algorithms drops down as the number of channels increases. Among these three algorithms, Moore and Hodgson's algorithm has achieved best performance. However,

the CBPS algorithm has achieved an improvement in the loss rate than SMF, especially when the number of channels is small. For example, when $C=4$, the loss rate of CBPS is only 40% of that of SMF algorithm. This is because, when the number of channel is small, the competition for channels becomes drastic and the effect of differentiating traffic becomes important. When the number of channel becomes large, the performance of CBPS and Moore and Hodgson's algorithm narrows down and finally flattens out, which means that channel number is not the bottleneck anymore. In our system, hard real-time message occupies 25% of the total population, so in one batch, the maximum number of hard real-time message will be $50 \times 25\% = 13$. From the figure, we can see that when the number of channel exceeds 15, further increase in the number of channels will not induce any more loss rate. As for the SMF, all the messages has the same chance to go first and real-time messages may not go first than non real-time messages, so it will not flatten out when the channel number is increased to 15.

Figure 5.17 Impact of number of channels on loss rate of CBPS

c. Message Delay/Loss Rate vs. Tuning Time

In this set of experiments, we will evaluate the performance of these three algorithms against the tuning time. The number of channels is fixed to 4 and the traffic load is set to 0.9. We also assume that α is 0.1, and β and γ are 1.

Figure 5.18 Impact of tuning time on mean delay of CBPS

Figure 5.18 compares the characteristic of mean delay under varying tuning time. It is easy to see that the delay increases with the tuning time for these three algorithms. This is because that when the tuning time is increased, the source nodes take longer time to complete a message transmission. Among these three algorithms, CBPS consistently demonstrates its superior performance to Moore and Hodgson's algorithm in terms of delay.

Figure 5.19 Impact of tuning time on loss rate of CBPS

Figure 5.19 plots the loss rate versus tuning time. This figure clearly displays that CBPS has improved the network performance in the sense that the loss rate is smaller than SMF, especially when the tuning time is small. For example, when the tuning time is 0, the loss rate of CBPS is only 46% of that of SMF. When the tuning time becomes large, the loss rate

of all the algorithms gets close to 25%. It means that almost all the hard real-time message will be dropped for these three algorithms. This is because that increasing tuning time will increase the waiting time for all the messages and cause the waiting time of hard real-time messages to exceed their laxities.

d. Message Delay/Loss Rate with effect of α

In this set of experiments, the number of channels is set to 4, tuning time is fixed at 0, and β and γ is 1.

Figure 5.20 presents mean delay with effect of α . It is clearly that the mean delay increases as α increases. This is because, according to the formula $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$, when α increases, the weigh of SP_i becomes larger and message length L_{ni} will have less effect to the sorting scheme. From Figure 5.20, we can see that when α becomes large, the delay gets close to each other, which means that α has dominated over other two parameters in priority formula and further increases of α will not cause any delay change. In this case, all the messages will be queued according to the SP_i .

Figure 5.20 Impact of α on mean delay

Figure 5.21 shows the loss rate with effect of α . From the figure, we can see that the loss rate decreases as α increases, and, eventually, unaltered when α becomes large. It is clearly

that when α is no less than 0.5, CBPS can achieve better performance than Moore and Hodgson and Hodgson's algorithm in terms of loss rate. The reason of the improvement is, when α increases, SP_i will have more important effect to the sorting scheme. Meanwhile, hard real-time messages have smaller SP_i than soft real-time messages, thus hard real-time messages will have more chance to go first than soft real-time messages and achieve smaller loss rate. When α becomes larger, almost all the hard real-time messages will go first than other messages, and the hard real-time messages will be queued at the head of the line. In this case, the loss of the messages will be caused only by the queuing delay among the hard real-time messages and other kinds of messages will have no effect to it, then the loss rate is unaltered eventually.

Figure 5.21 Impact of α on loss rate

e. Message Delay/Loss Rate with effect of β

In this set of experiments, the number of channels is set to 4, tuning time is fixed at 0, α is 0.1 and γ is 1. We aim to study the performance of CBPS by comparing it with Minimum Laxity First algorithm (MLF), which sorts the messages according to the message laxity. The

reason the MLF is used to compare with the CBPS in this group of experiments is that with changes of β , the CBPS will approach to the MLF algorithm.

Figure 5.22 presents mean delay with effect of β . It is clearly that the mean delay increases as β increase. This is because, according to the formula $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$, when β increases, the weigh of T_{ni} becomes larger and message length L_{ni} will have less effect on the sorting scheme. Thus the delay increases with β , and unaltered eventually. From the Figure 5.22, we can find that even if β equals to 61, which means that the weigh of laxity exceeds the other two parameters enormously and the messages will be sorted according to the message laxity, the delay of CBPS is still smaller than MLF. The reason of the improvement is that when the laxities of two messages are not equal, these two messages will be sorted according to MLF. However, when the laxities of two messages are equal to each other, the message with less $\alpha SP + \gamma L_{ni}$ will have the higher priority. This means that further sorting will happen to the messages with the same laxity. This sorting will improve the CBPS in terms of delay and makes it better than MLF. If α and γ are set to 0, the CBPS algorithm will get the same result as that of the MLF algorithm.

Figure 5.22 Impact of β on mean delay

Figure 5.23 presents the loss rate with effect of β . From the figure, we can see that CBPS can achieve better performance than MLF in terms of loss rate. This is because MLF just

considers laxity and the message that has less laxity may have the larger length and cause the messages following it to be late. When β increases, the loss rate of CBPS will get close to MLF. This is because, when β is large enough, the messages will be sorted according to a way similar to MLF. Similarly, CBPS will be lightly better than MLF even if β is increased to 61 because of the reason as mentioned above.

Figure 5.23 Impact of β on loss rate

f. Message Delay/Loss Rate with effect of γ

In this set of experiments, we assume the number of channels is 4, tuning time is 0, α is 0.1, and β is 1.

Figure 5.24 presents mean delay with effect of γ . It is clearly that the mean delay decreases as γ increases. This is because, according to the formula $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$, when γ increases, the weigh of L_{ni} becomes larger and message laxity T_{ni} will have less effect to the sorting scheme. This means that the messages will be sorted according to the message length. From Figure 5.24, we can see that when γ increases, the delay of CBPS will be close to SMF. If we set α and β to 0, we can get the same results as SMF, which is shown in the figure.

Figure 5.24 Impact of γ on mean delay

Figure 5.25 Impact of γ on loss rate

Figure 5.25 shows the loss rate with effect of γ , and we can see that even if γ equals to 51, the loss rate of CBPS is still smaller than SMF. This is because when the length of two messages is not equal, they will be queued according to SMF, while the length of these two messages is equal to each other, the message with less $\alpha SP + \beta T_{ni}$ will have higher priority. This means that sequence happens in the messages with the same length. This kind of change

makes CBPS achieve better performance than SMF in the sense of loss rate even if γ is very large.

5.2.3 Summary

In this section, we have proposed a new reservation-based algorithm for scheduling variable-length messages with different time constraints in single-hop passive-star coupled WDM optical networks. The proposed algorithm, which is designed to handle messages with and without time constraints, considers the order of the message transmission based on the costs of messages incurred in the network. And it assigns the priority of messages dynamically. CBPS comes with data channel selection algorithm EATS, which can help to avoid channel collisions and receiver collisions. The studies have shown that the new algorithm can not only reduce the messages loss rate but also reduce the average message delay when an integrated traffic is applied to the network so that the transmission of both types of messages could be benefited. Moreover, we also conduct a set of experiments to study the adaptability of CBPS algorithm. Results show that the changes of the parameters that associate with the costs of messages, i.e., α , β , and γ , will lead to different performance in terms of delay and loss rate.

5.3 CBPS Based QoS Service Prediction

5.3.1 Introduction

CBPS is an algorithm to provide differentiated service so that real time messages as well as non real-time messages can be benefited. The supported differentiation service depends on the priority of each message. And it is a dynamic priority defined by $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$. As shown in the last section, CBPS algorithm can not only reduce the message loss rate but also reduce the average message delay when an integrated traffic is applied to the network. However, the specific QoS requirements of individual traffic are not considered by this algorithm. In the future generation network, one of the main challenges is to deliver a

seamless QoS service based on the individual user's needs. In order to provide acceptable QoS, the network must provide different levels of service to different categories of customers. In the issue of QoS guarantee in WDM optical networks, QoS service prediction is one of the important open topics, which is used to determine whether the connections can be accepted or not under certain QoS requirements. In this section, a CBPS based QoS prediction scheme for single-hop passive-star coupled WDM optical networks is proposed. To guarantee QoS service of a set of traffic with given QoS requirements, proposed scheme is able to detect future changes in the network and provide the prediction based on these changes. This CBPS based QoS prediction is presented to enhance the performance of WDM optical network management so that service providers can offer better services to their clients.

Figure 5.26 shows the framework of the proposed QoS prediction scheme, which includes four modules. *QoS Requirements Monitoring* observes the current QoS requirements and detects changes in the QoS. *QoS Prediction* is to predict whether the QoS of individual traffic can be guaranteed under QoS requirements. If *QoS Violation* detects that violation happens, *QoS Estimation* will decide to refuse or give out a recommendation to the hopeless requests. Among these four modules, *QoS Prediction* performs the most important function, which is also the main part to be introduced in this section. This module is developed based on the CBPS algorithm, i.e., the priority of each message in the network will be defined according to $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$, where P_i is the dynamic priority assigned to message i , SP_i is the static priority of message i which shows the different importance of different kinds of messages (The message with the least value SP_i has the highest priority.), T_{ni} is the normalized relative laxity of message i which presents the stringency of the message, and L_{ni} is the normalized length of message i . Based on the priority, different traffic streams will achieve different QoS service. The detail description of QoS prediction process will be presented as follows.

Figure 5.26 Schematic description of QoS prediction process

As the results of CBPS show, different combination of scheduling parameters α , β , and γ will lead to different performance in terms of delay and loss rate. In other word, given the traffic load and one set of performance requirements, it is possible to predict whether QoS of one traffic stream can be met with one combination of these three parameters. Figure 5.27 shows an overview of QoS prediction structure with six classes of traffic. And the class of the traffic is defined as the static class with index $c \in \{1, \dots, 6\}$, where messages in static class $c-1$ stream have higher static priority over messages in static class c stream. According to the CBPS algorithm, the priority for each message is dynamic depending on its static priority, message length and message laxity. This also means that the priority value P for each message of each static class is not constant, so the messages of higher static priority may not always go first than those of lower static priority. Based on this observation, we get the motivation to map the dynamic priority of each message to one kind of relative priority so that the messages with higher relative priority definitely go first than those with lower relative priority. It is assumed that the message with one relative priority belongs to one corresponding relative class. With such kind of mapping, it will be quite easy to get the performance in terms of average delay/loss rate of each relative class. Therefore, the main problem of the design of QoS prediction structure becomes how to map the dynamic priority of the messages to the relative priority. If the mapping method is found, the relationship between static class and relative class can be obtained. With the relationship between static

class and relative class given one set of three parameters and the performance of each relative class, the performance of each static class can be obtained easily.

Figure 5.27 Overview of QoS prediction structure

5.3.2 Relative Class Mapping

5.3.2.1 Mapping Process

This subsection devotes entirely to introduce the relative class mapping. First we define six static classes of traffic, of which three are real time messages and another three are non real time messages, which are shown in Figure 5.29. Class 1, Class 2 and Class 3 are real time classes, and Class 4, Class 5 and Class 6 are non real time classes. The method to map dynamic priority to relative priority can be explained by Figure 5.28, in which, the domain of the value for dynamic priority P of all the messages is divided into sections $\{I_1, I_2, \dots, I_{10}\}$ where I_k starts at $p(k)$ and is δ_k wide. Each section is corresponding to one relative class, i.e., if the value P of one message drops on section I_k , its relative priority will be k . The message with less relative priority will have higher priority. $p(1) \sim p(6)$ are for real-time messages, in which $p(1)$ equals to the minimum value of P among all the real-time messages and $p(6)$ equals to the maximum value of P among all the real-time messages. $p(6) \sim p(11)$ are for non real-time messages, in which $p(6)$ equals to the minimum value of P among all the non real-time messages and $p(11)$ equals to the maximum value of P among all the non real-time messages. The reason to divide the section of real-time messages and non real-time messages respectively is that the value P of non real-time messages will not be less than that of real-

time messages because the laxity of non real-time messages is infinity. Normally $P(\text{rt})_{\max} \neq P(\text{nrt})_{\min}$, i.e, $p(6)$ can only stand for real time messages or non real time messages. For simplicity, we put them together because there will be no value of P between $P(\text{rt})_{\max}$ and $P(\text{nrt})_{\min}$. Therefore, $p(6)$ stands for the maximum value of P among all the real-time messages as well as the minimum value of P among all the non real-time messages.

Figure 5.28 Distribution of the relative priority section

Figure 5.29 The relationship between static class and relative class

Once each section of relative class is fixed, the relationship between static class and relative class can be obtained in Figure 5.29. Let n_k^c denotes the proportion of static classes distributing to the relative classes, where c denotes the static class and k denotes the relative class. As the figure shows, among all the messages of Class 1, n_1^1 of them belong to section I_1 , n_2^1 of them belong to section I_2 and so on, where $n_1^1 + n_2^1 + n_3^1 + n_4^1 + n_5^1 = 1$. Since $I_6 \sim I_{10}$ are for non real-time messages, no messages of Class 1 will belong to these sections. Following the same way, the proportion of the messages from one static priority distributing to one relative priority can be obtained, which is shown in Figure 5.29. The proportion of these

static classes distributing to the relative classes depends on the parameters, α , β and γ , i.e., the changes of the three parameters will lead to the changes of n_k^c .

If the relationship between the static class and the relative class is found, the structure of QoS prediction can be changed to the one shown in Figure 5.30. As addressed above, it is much easy to predict the performance of relative class since the message with higher relative priority will definitely go first than that with lower relative priority. Therefore, if we can predict such performance, it is also easy to get the performance of static class given the parameters, α , β and γ . The connection between static class and relative class is the proportion of each static class c distributing to the relative classes k , namely n_k^c . Therefore, the main problems to be solved are how to obtain the section of relative class and how to obtain the relationship between static class c with λ_c and the relative class k with λ'_k .

Figure 5.30 Structure of QoS service prediction

5.3.2.2 Relative Priority Division

The criterion for the section division of P is that traffic density of each section should be equal to each other for real-time messages and non real-time messages. That means it has to satisfy $\lambda'_k = \lambda'_l$, where k and l are any relative class. The way of such division is to balance

the traffic among the relative classes so that the event that the traffic falls on only several sections can be avoided. Moreover, in this way, once the total traffic load of static classes is fixed, the traffic load of each relative class can be fixed, too. We have $\lambda'_k = n_k^1 \lambda_1 + n_k^2 \lambda_2 + n_k^3 \lambda_3$ if k is a real time relative class, where n_k^c denotes the proportion of static class c distributing to relative class k . Similarly, $\lambda'_k = n_k^4 \lambda_4 + n_k^5 \lambda_5 + n_k^6 \lambda_6$ if k is non real-time relative class. Based on the criterion mentioned, if the total traffic load of the six static classes of traffic is ρ , the respective traffic load of each relative class should be $\rho/10$. It also indicates that the traffic load of each relative class has no connection with the three parameters. That is, no matter how the combination of three parameters changes, as long as the total traffic load ρ is fixed, the performance of the relative class can be predicted. Let $\text{delay}(k)$ and $\text{loss}(k)$ denote the delay and loss rate of each relative class k we get respectively, it is easy to get the delay of each static class. Take static class 1 as an example, the delay and loss rate of it equals to $n_1^1 * \text{delay}(1) + n_1^2 * \text{delay}(2) + n_1^3 * \text{delay}(3) + n_1^4 * \text{delay}(4) + n_1^5 * \text{delay}(5)$ and $n_1^1 * \text{loss}(1) + n_1^2 * \text{loss}(2) + n_1^3 * \text{loss}(3) + n_1^4 * \text{loss}(4) + n_1^5 * \text{loss}(5)$ respectively. As addressed above, the proportion of each static class distributing to each relative class depends on the three parameters. Therefore, it can be found what QoS prediction structure actually is to change the proportion of each static class distributing to each relative class by changing the three parameters until one combination that satisfies all the QoS requirements found.

To comply with the criterion for the section division of P , the division of the sections should associate with the analysis of probability density function (p.d.f) of the dynamic priority P so that the total probability that the traffic of the static classes falling on each section is equal to each other. Moreover, once the section of P is fixed, the proportion of each static class to each relative class can be obtained by the analysis of p.d.f..

As defined in CBPS algorithm, T_{ni} is the normalized relative laxity of message i , L_{ni} is the normalized length of message i . Both T_{ni} and L_{ni} should be less or equal to 1. And the

message length and laxity are random variables following different Exponential distributions with different mean values. The normalized value of the message length equals to $\text{length}/L_{\max}$. And the normalized value of the relative laxity is $\text{laxity}/T_{\max}$.

Suppose the p.d.f of the laxity is $f_x(x) = \eta e^{-\eta x}$, so the p.d.f of T_{ni} is $f_x(x) = \eta T_{\max} e^{-\eta T_{\max} x}$. And the p.d.f of βT_{ni} is $f_x(x) = \eta T_{\max} / \beta e^{-(\eta T_{\max} / \beta)x}$. Similarly, suppose the p.d.f of the length is $f_y(y) = \mu e^{-\mu y}$, so the p.d.f of L_{ni} is $f_y(y) = \mu L_{\max} e^{-\mu L_{\max} y}$. And the p.d.f of γL_{ni} is $f_y(y) = \mu L_{\max} / \gamma e^{-(\mu L_{\max} / \gamma)y}$. Then the p.d.f $f_z(z)$ of P_i of real-time static classes, where $P_i = \alpha SP_i + \beta T_{ni} + \gamma L_{ni}$, can be as follows according to [73]

$$f_z(z) = \int_0^z \mu * L_{\max} / \gamma e^{-(\mu * L_{\max} / \gamma)(z-x-\alpha)} * \eta * T_{\max} / \beta e^{-(\eta * T_{\max} / \beta)x} dx$$

Following we will give a simple example to show that how to obtain section $\{I_1, I_2, \dots, I_{10}\}$. For real-time messages, we define that $\alpha=0.5$, $\beta=1$, $\gamma=1$, $L_{\max}=60$, $T_{\max}=75$, $1/\mu=20$, and $1/\eta=25$. Moreover, SP equals to 0, 1 and 2 for static class 1, 2 and 3 respectively. Take a message i in static class 2 for example, it is easy to get that $P_i = 0.5 + T_{ni} + L_{ni}$. Let $f_{z_c}(z_c)$ to denote the p.d.f function of the variables P of class c . Then the p.d.f $f_{z_2}(z_2)$ can be got from

$$\begin{aligned} f_{z_2}(z_2) &= \int_0^{z_2} \mu * L_{\max} / \gamma e^{-(\mu * L_{\max} / \gamma)(z_2-x-\alpha)} * \eta * T_{\max} / \beta e^{-(\eta * T_{\max} / \beta)x} dx \\ &= \int_0^{z_2} \frac{1}{20} * 60 e^{-\frac{1}{20} * 60(z_2-x-0.5)} * \frac{1}{25} * 75 e^{-\frac{1}{25} * 75x} dx \\ &= 9 \int_0^{z_2} e^{-3(z_2-0.5)} dx \\ &= 9(z_2 - 0.5) e^{-3(z_2-0.5)} \end{aligned}$$

Similarly, the p.d.f of P of Class 1 ($SP=0$) and Class 3 ($SP=2$) are $f_{z_1}(z_1) = 9z_1 e^{-3z_1}$ and $f_{z_3}(z_3) = 9(z_3 - 1) e^{-3(z_3-1)}$ respectively. Let $P_{\min}^c$ and $P_{\max}^c$ denotes the minimum value and maximum value of P for static class c . Then from the formula $P_i = 0.5 * SP_i + T_{ni} + L_{ni}$ and the fact that $T_{ni} \in [0,1]$ and $L_{ni} \in [0,1]$, we have $P_{\min}^1 = 0$, $P_{\max}^1 = 2$, $P_{\min}^2 = 0.5$, $P_{\max}^2 = 2.5$, $P_{\min}^3 = 1$ and $P_{\max}^3 = 3$.

Let $F_{Z_c}(z_c)$ denotes the cumulative distribution function (c.d.f) of dynamic priority P of each static class c , where $F_{Z_c}(z_c) = P(P_{\min}^c \leq Z_c \leq z_c) = \int_{P_{\min}^c}^{z_c} f_{z_c}(z_c) dz_c$. Therefore, from the fact that $P_{\min}^1 = 0$, $P_{\min}^2 = 0.5$ and $P_{\min}^3 = 1$, the c.d.f of these three class are $F_{Z_1}(z_1) = -3z_1e^{-3z_1} - e^{-3z_1} + 1$, $F_{Z_2}(z_2) = -3z_2e^{-3z_2+1.5} + 0.5e^{-3z_2+1.5} + 1$ and $F_{Z_3}(z_3) = -3z_3e^{-3z_3+3} + 2e^{-3z_3+3} + 1$ respectively. Then, we change each $F_{Z_c}(z_c)$ according to the following criterion

$$F_{Z_c}(z_c) = 0, \quad z_c < P_{\min}^c \text{ or } z_c > P_{\max}^c$$

Therefore, we have

$$\begin{aligned} F_{Z_1}(z_1) &= -3z_1e^{-3z_1} - e^{-3z_1} + 1 & P_{\min}^1 \leq z_1 \leq P_{\max}^1 \\ F_{Z_1}(z_1) &= 0 & z_1 < P_{\min}^1 \text{ or } z_1 > P_{\max}^1 \end{aligned} \quad (5.7.1)$$

$$\begin{aligned} F_{Z_2}(z_2) &= -3z_2e^{-3z_2+1.5} + 0.5e^{-3z_2+1.5} + 1 & P_{\min}^2 \leq z_2 \leq P_{\max}^2 \\ F_{Z_2}(z_2) &= 0 & z_2 < P_{\min}^2 \text{ or } z_2 > P_{\max}^2 \end{aligned} \quad (5.7.2)$$

$$\begin{aligned} F_{Z_3}(z_3) &= -3z_3e^{-3z_3+3} + 2e^{-3z_3+3} + 1 & P_{\min}^3 \leq z_3 \leq P_{\max}^3 \\ F_{Z_3}(z_3) &= 0 & z_3 < P_{\min}^3 \text{ or } z_3 > P_{\max}^3 \end{aligned} \quad (5.7.3)$$

In addition, let D_c to denote the probability that the value of z_c , falls on the domain $[P_{\min}^c, P_{\max}^c]$, i.e., $D_c = F_{Z_c}(P_{\max}^c) - F_{Z_c}(P_{\min}^c)$. In this way, we have $D_1 = F_{Z_1}(P_{\max}^1) - F_{Z_1}(P_{\min}^1)$, $D_2 = F_{Z_2}(P_{\max}^2) - F_{Z_2}(P_{\min}^2)$ and $D_3 = F_{Z_3}(P_{\max}^3) - F_{Z_3}(P_{\min}^3)$. Seen from Figure 5.28, the domain for the real-time relative class is $[p(1), p(6)]$, in which $p(1) = P_{\min}^1$ and $p(6) = P_{\max}^3$. Therefore, the total probability that the random variable P falls on the domain $[P_{\min}^1, P_{\max}^3]$ is equal to $D_1 + D_2 + D_3$. In order to achieve that the traffic intensity of each relative class is equal to each other, it is necessary to get that the total probability that the dynamic priority P falls on each section of relative priority section I_k with domain $[p(k), p(k+1)]$ equal to $\frac{D_1 + D_2 + D_3}{5}$.

So in order to get the division of each section, we have the following constraint

$$F_{z_1}(p(k)) - F_{z_1}(p(k-1)) + F_{z_2}(p(k)) - F_{z_2}(p(k-1)) + F_{z_3}(p(k)) - F_{z_3}(p(k-1)) = \frac{D_1 + D_2 + D_3}{5} \quad (5.8)$$

It is easy to find that the value $p(k)$ can be acquired from (5.8) recursively. As addressed above, $p(1) = P_{\min}^1$. Therefore, substituting $p(1)$ into (5.8) and according to (5.7.1), (5.7.2) and (5.7.3), $p(2)$ can be easily obtained. Once $p(2)$ is obtained, $p(3)$ can be got from (5.8), too. Following this method, $p(2) \sim p(5)$ can be obtained recursively. In addition, $p(6)$ equals to the $P_{\max}^3$, i.e., $p(6) = P_{\max}^3 = 3$. As such, the sections $\{I_1, I_2, \dots, I_5\}$ are obtained, of which the density the messages of each section is equal to $\frac{D_1 + D_2 + D_3}{5}$. Once the sections are fixed, the proportion of each real time static class can be obtained. Take static class 1 as an example, $n_1^1 = \int_{p(1)}^{p(2)} f_{z_1}(z_1) dz_1$, where $f_{z_1}(z_1)$ is the p.d.f of P of static class 1. Following the same way, all the connection between the real-time static class and the relative class can be obtained.

The way to divide the section $\{I_6, I_7, \dots, I_{10}\}$ is similar to above. However, since the laxity of the non real-time messages is infinity, we assume that the laxity of them is fixed to one constant so that the value P of non real-time messages will not be smaller than that of real-time messages. In this example, we assume that the normalized laxity of the non real-time messages is fixed to 2 since $\gamma L_{ni} \leq 1$ and $\beta T_{ni} \leq 1$ for real-time messages. That is, the priority formula for non real-time messages becomes $P_i = 0.5 * SP_i + 2 + L_{ni}$. And except the assumption of common parameters above, it is assumed that SP equals to 3, 4, and 5 for static class 4, 5 and 6 respectively.

Similarly, the p.d.f of the length is $f_y(y) = \mu e^{-\mu y}$, so the p.d.f of L_{ni} is $f_y(y) = \mu L_{\max} e^{-\mu L_{\max} y}$. And the p.d.f of γL_{ni} is $f_y(y) = \mu L_{\max} / \gamma e^{-(\mu L_{\max} / \gamma) y}$. Then the p.d.f $f_z(z)$ of dynamic priority P_i for non real-time static class, where $P_i = \alpha SP_i + 2 + \gamma L_{ni}$, can be obtained according to

(5.9)

$$f_z(z) = f_y(z - \alpha SP_i - 2)$$

Taking static class 4 as an example, for each message i , we have $P_i = 3.5 + L_{ni}$. So the p.d.f of P of this class is $f_{z_4}(z_4) = 3e^{-3(z_4-3.5)}$. Then from the formula $P_i = 0.5 * SP_i + 2 + L_{ni}$ and the fact that $L_{ni} \in [0,1]$, we have $P_{\min}^4 = 3.5$ and $P_{\max}^4 = 4.5$. Hence the c.d.f of P of static class 4 is $F_{Z_4}(z_4) = -e^{-3z_4+21/2} + 1$. Similarly, from the fact that $P_{\min}^5 = 4$, $P_{\max}^5 = 5$, $P_{\min}^6 = 4.5$ and $P_{\max}^6 = 5.5$, the c.d.f of P of static class 5 and static class 6 are $F_{Z_5}(z_5) = -e^{-3z_5+12} + 1$ and $F_{Z_6}(z_6) = -e^{-3z_6+27/2} + 1$ respectively. And then we change them to

$$\begin{aligned} F_{Z_4}(z_4) &= -e^{-3z_4+21/2} + 1 & P_{\min}^4 \leq z_4 \leq P_{\max}^4 \\ F_{Z_4}(z_4) &= 0 & z_4 < P_{\min}^4 \text{ or } z_4 > P_{\max}^4 \end{aligned} \quad (5.10.1)$$

$$\begin{aligned} F_{Z_5}(z_5) &= -e^{-3z_5+12} + 1 & P_{\min}^5 \leq z_5 \leq P_{\max}^5 \\ F_{Z_5}(z_5) &= 0 & z_5 < P_{\min}^5 \text{ or } z_5 > P_{\max}^5 \end{aligned} \quad (5.10.2)$$

$$\begin{aligned} F_{Z_6}(z_6) &= -e^{-3z_6+27/2} + 1 & P_{\min}^6 \leq z_6 \leq P_{\max}^6 \\ F_{Z_6}(z_6) &= 0 & z_6 < P_{\min}^6 \text{ or } z_6 > P_{\max}^6 \end{aligned} \quad (5.10.3)$$

$$\text{Let } D_4 = F_{Z_4}(P_{\max}^4) - F_{Z_4}(P_{\min}^4), \quad D_5 = F_{Z_5}(P_{\max}^5) - F_{Z_5}(P_{\min}^5) \text{ and } D_6 = F_{Z_6}(P_{\max}^6) - F_{Z_6}(P_{\min}^6).$$

Therefore, the total probability that the random variable P falls on the domain $[p(6), p(11)]$, i.e., $[P_{\min}^4, P_{\max}^6]$, is equal to $D_4 + D_5 + D_6$. In order to achieve that the traffic intensity of each relative class is equal to each other, it is necessary to get that the total probability that the dynamic priority P falls on each relative priority section k with the domain $[p(k), p(k+1)]$ equal to $\frac{D_4 + D_5 + D_6}{5}$. Therefore, we have

$$F_{Z_4}(p(k)) - F_{Z_4}(p(k-1)) + F_{Z_5}(p(k)) - F_{Z_5}(p(k-1)) + F_{Z_6}(p(k)) - F_{Z_6}(p(k-1)) = \frac{D_4 + D_5 + D_6}{5} \quad (5.11)$$

From (5.11), it is easy to find that the value $p(k)$ can be acquired recursively. As addressed above, $p(6) = P_{\min}^4$. Therefore, substituting $p(6)$ into (5.11) and according to (5.10.1), (5.10.2) and (5.10.3), $p(7)$ can be easily obtained. Once $p(7)$ is obtained, $p(8)$ can be got from (5.11),

too. Following this method, $p(7) \sim p(10)$ can be obtained recursively. In addition, $p(11)$ equals to $P_{\max}^6$, i.e., $p(11) = P_{\max}^6 = 5.5$. As such the sections $\{I_6, I_7, \dots, I_{10}\}$ are obtained, of which the density the messages is equal to $\frac{D_4 + D_5 + D_6}{5}$. Once the sections of relative class is found, the proportion of each class to the relative classes is fixed, too.

Until now, it can be found that by changing the combination α , β and γ , the proportion of each static class to each relative class can be changed so that the performance of each static class can be changed simultaneously. Hence, what we need to do next is to analyze the performance of relative classes.

With the introduction of relative class, our model can be modeled to the one with ten relative classes, of which five are real-time classes and another five are non real-time classes. The arrival rate of each class is the same with each other. Moreover, the real-time messages have strict laxities before the beginning of their service. Therefore, it can be defined as a finite-capacity multi-class, multi-server (MCMS) model with non-preemptive priorities, which are a generalization of the well-known $M/M/k$ queue. This is a finite queue with state-dependent Poisson arrival process, exponential service times, multiple servers, and exponential customer impatience for real-time classes. The detail analysis of such a model will be presented in the next subsections.

5.3.3 Performance of Relative Classes Traffic

MCMS models arise when clients with different service characteristics ask for the same capacity and performance characteristics are needed for each class separately. There is quite some literature on single server priority queueing systems. However, multi-server priority queueing systems have been received less attention. Such models have been studied in [74][75][76][77][78][79][80][81][82][83]. The model we set up is much close to the one in [81], but we have considered the time constraints of the real-time messages as that addressed in [84]. In the context of queueing theory, customers with limited waiting time are usually

referred to as “impatient customers”. An impatient customer has a laxity before which it is available for service and after which it must leave the system.

5.3.3.1 Definitions and Notation

It is assumed that the population is finite, and a message can be generated only from an idle node. Therefore, our model can be considered as a multi-class system with n servers and $N-n$ waiting room, where N is the number of nodes in the network. There are I classes in the system. The class with index $k \in \{1, \dots, I/2\}$ is real time class and that with index $k \in \{I/2+1, \dots, I\}$ is non real time class. Customers are processed according to a non-preemptive priority rule, where class $k-1$ customers have higher priority over class k customers. If all the servers are busy at the arrival instant of a customer, the customer joins the queue behind all customers of the same class. If there are less than n customers in the system, an arriving customer selects a free server at random.

Let

- The total arrival rate of the arrival stream of I classes is assumed to be independent Poisson processes with rate $\lambda \neq 0$. And $\lambda = \sum_{k=1}^I \lambda_k$.
- The service times of the customers are identical, exponentially distributed random variables with mean $1/\mu$.
- The laxity of each real time class is exponentially distributed random variable with mean $1/\eta$.

Consider now a customer with class $k' \in \{1, \dots, I\}$ entering the queue. To determine the delay of this customer we have to know the number of messages with higher or equal priority and the number of messages with lower priority in the waiting line at the arrival instant. For real-time messages, we have to know 1) the number of real-time messages with higher or equal priority; 2) the number of real-time messages with lower priority; and 3) the number of non real-time messages. On the other hand, for non real-time messages, we have to know 1) the number of real-time messages; 2) the number of non real-time messages with

higher or equal priority; and 3) the number of non real-time messages with lower priority. It is easily to find that the scenarios considered for real-time and non real-time messages are different. Therefore, in order to analyze the performance of real-time and non real-time messages in the same system, we define different models for them respectively. And the two models are as follows.

5.3.3.2 Performance for Real-Time Message Scenario

5.3.3.2.1 The Steady State Probability

As addressed above, for any arriving real time class $k' \in \{1, \dots, I/2\}$, we have to know 1) the number of real-time messages with higher or equal priority, denoted by $l(k', 3)(t)$; 2) the number of real-time messages with lower priority, denoted by $l(k', 2)(t)$; and 3) the number of non real-time messages, denoted by $l(k', 1)(t)$, in the waiting line at the arrival instant t .

Then we define $l(k', 3)(t) = \sum_{k=1}^{k'} l_k(t)$, $l(k', 2)(t) = \sum_{k=k'+1}^{I/2} l_k(t)$ and $l(k', 1)(t) = \sum_{k=I/2+1}^I l_k(t)$,

where $0 \leq l_k(t) \leq (N - n)$ is the number of waiting customers of class k at time t .

We use $(r(t), l(k', 3)(t), l(k', 2)(t), l(k', 1)(t))$ to denote the state of the system at time t , where $r(t) = l(k', 3)(t) + l(k', 2)(t) + l(k', 1)(t) + n$ is the total number of messages in the system, including those in the queue and those in the servers, and $0 \leq r(t) \leq N$. So the steady state distribution of the continuous time homogeneous Markov Chain (CTMC) is denoted by

$$\pi_{(r, l(k', 3), l(k', 2), l(k', 1))}^{k'} = \lim_{t \rightarrow \infty} \text{Prob}\{Z(t) = (r(t), l(k', 3)(t), l(k', 2)(t), l(k', 1)(t))\}$$

Let $\lambda_{k', 3}$ denote the total arrival rate of the real-time messages with higher or equal priority, $\lambda_{k', 2}$ denote the total arrival rate of the real-time messages with lower priority and $\lambda_{k', 1}$ denote the total arrival rate of the non real-time messages. Then we have $\lambda_{k', 3} = \sum_{k=1}^{k'} \lambda_k$, $\lambda_{k', 2} = \sum_{k=k'+1}^{I/2} \lambda_k$ and $\lambda_{k', 1} = \sum_{k=I/2+1}^I \lambda_k$. Moreover, as addressed in [36], the messages destined to the same destination cannot be processed at the same time in order to avoid the destination collision. But in EATS, the m channels are assigned to m messages

if $m < n$ or n channels are assigned to m messages if $m \geq n$, independent of whether they can be transmitted immediately or not. Hence, under EATS, given that there are m messages occupying the channels, i.e., servers, only i out of them which are destined to different nodes can be in service. That is, the number of the servers in service is dependent on the messages currently occupying them. So the service rate can be obtained from following:

- When $m \leq n$, where n is the number of servers, we have

$$\mu_m = \mu \sum_{i=1}^m i \alpha_i(m) \quad (5.12.1)$$

where $\alpha_i(m)$ is the probability that, out of m messages, there are i destined to different

nodes. We have $\alpha_i(m) = \frac{\binom{N}{i} f(i, m)}{N^m}$. $f(i, m)$ is the total number of ways that m

messages can be destined to the i selected destinations and $f(i, m) = i^m - \sum_{j=1}^{i-1} \binom{i}{j} f(j, m)$.

$\alpha_i(m)$ can also be expressed as $\alpha_i(m) = \frac{\binom{N}{i} i! S(i, m)}{N^m}$, where $S(i, m) = f(i, m) / i!$ is the sterling number.

- When $m > n$, all servers are assigned to n messages, so

$$\mu_m = \mu_n = \mu \sum_{i=1}^n i \alpha_i(n) \quad (5.12.2)$$

The system can be modeled as a stochastic process whose states are $(r, l(k', 3), l(k', 2), l(k', 1))$ (refer to Figure 5.31, Figure 5.32 and Figure 5.33). As an example, for steady state $(r, l(k', 3), l(k', 2), l(k', 1))$ in Figure 5.32, where $n < r < N$, there are twelve transitions into and out of that state. As assumed before, the message with higher priority definitely goes first than that with lower priority. According to [84], given that the laxity follows exponential distribution with mean value $1/\eta$ and there are h messages in the system, the probability that a message misses its laxity is $h\eta$. Therefore, the transition rate

from $(r, l(k', 3), l(k', 2), l(k', 1))$ to $(r, l(k', 3) - 1, l(k', 2), l(k', 1))$ is $\mu_n + l(k', 3)\eta$, which means that a message in $l(k', 3)$ leaves the system with probability μ_n because of the end of the service or with probability $l(k', 3)\eta$ because of the violation of the time constraints. The message is considered as lost in the latter case. On the other hand, the transition rate from $(r, l(k', 3), l(k', 2), l(k', 1))$ to $(r, l(k', 3), l(k', 2) - 1, l(k', 1))$ is $(\mu_n + l(k', 2)\eta) * 1_{l(k', 3)=0} + l(k', 2)\eta * 1_{l(k', 3) \neq 0}$, where $1_{condition} = 1$ when condition is true, otherwise, $1_{condition} = 0$. This means that if $l(k', 3) = 0$, the message in $l(k', 2)$ leaves the system with rate μ_n because of the end of the service or with rate $l(k', 2)\eta$ because of the violation of the time constraints, otherwise, the message in $l(k', 2)$ cannot be serviced and its leave is only because of the violation of the time constraint. Following the same way, the steady state distribution can be expressed as follows.

Figure 5.31 Transition diagram when $0 \leq r \leq n$

Figure 5.32 Transition diagram when $n < r < N$

Figure 5.33 Transition diagram when $r=N$

For $r=0$,

$$0 = \pi_{(0,0,0,0)}^{k'}(-\lambda) + \pi_{(1,0,0,0)}^{k'}\mu_1$$

for $0 < r < n$

$$0 = \pi_{(r,0,0,0)}^{k'}(-\lambda - \mu_r) + \pi_{(r-1,0,0,0)}^{k'}\lambda + \pi_{(r+1,0,0,0)}^{k'}\mu_{r+1}$$

for $r=n$

$$0 = \pi_{(n,0,0,0)}^{k'}(-\lambda - \mu_n) + \pi_{(n-1,0,0,0)}^{k'}\lambda + \pi_{(n+1,1,0,0)}^{k'}(\mu_n + \eta) + \pi_{(n+1,0,1,0)}^{k'}(\mu_n + \eta) + \pi_{(n+1,0,0,1)}^{k'}\mu_n$$

for $n < r < N$ and $l(k',3) \neq 0$

$$0 = \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'}(-\mu_n - \lambda - (l(k',3) + l(k',2))\eta) + \sum_{k=1}^3 \pi_{(r-1,\dots,l(k',k)-(1-\delta_{l(k',k),0}),\dots)}^{k'}\lambda_{k',k}(1-\delta_{l(k',k),0}) \\ + \pi_{(r+1,l(k',3)+1,l(k',2),l(k',1))}^{k'}(\mu_n + (l(k',3) + 1)\eta) + \pi_{(r+1,l(k',3),l(k',2)+1,l(k',1))}^{k'}(l(k',2) + 1)\eta$$

where $\delta_{l(k',k),0}$ denotes the Kronecker-delta, $\delta_{l(k',k),0} = 1$ when $l(k',k) = 0$, and vice versa.

for $n < r < N$, $l(k',3) = 0$ and $l(k',2) \neq 0$

$$0 = \pi_{(r,0,l(k',2),l(k',1))}^{k'}(-\mu_n - \lambda - l(k',2)\eta) + \sum_{k=1}^2 \pi_{(r-1,\dots,l(k',k)-(1-\delta_{l(k',k),0}),\dots)}^{k'}\lambda_{k',k}(1-\delta_{l(k',k),0}) \\ + \pi_{(r+1,1,l(k',2),l(k',1))}^{k'}(\mu_n + \eta) + \pi_{(r+1,0,l(k',2)+1,l(k',1))}^{k'}(\mu_n + (l(k',2) + 1)\eta)$$

for $n < r < N$, $l(k',3) = 0$, $l(k',2) = 0$ and $l(k',1) \neq 0$

$$0 = \pi_{(r,0,0,l(k',1))}^{k'}(-\mu_n - \lambda) + \pi_{(r-1,0,0,l(k',1)-1)}^{k'}\lambda_{k',1} \\ + \pi_{(r+1,1,0,l(k',1))}^{k'}(\mu_n + \eta) + \pi_{(r+1,0,1,l(k',1))}^{k'}(\mu_n + \eta) + \pi_{(r+1,0,0,l(k',1)+1)}^{k'}\mu_n$$

for $r=N$

$$0 = \pi_{(N,l(k',3),l(k',2),l(k',1))}^{k'}(-\mu_n - (l(k',3) + l(k',2))\eta) + \sum_{k=1}^3 \pi_{(N-1,\dots,l(k',k)-(1-\delta_{l(k',k),0}),\dots)}^{k'}\lambda_{k',k}(1-\delta_{l(k',k),0})$$

Let $r_{x \rightarrow x'}$ be the transition rate from the current state $x = (r(t), l(k', 3)(t), l(k', 2)(t), l(k', 1)(t))$ to the destination legal state $x' = (r'(t), l'(k', 3)(t), l'(k', 2)(t), l'(k', 1)(t))$. The transition rate is given in the following equation.

$$r_{x \rightarrow x'} = \begin{cases} \lambda & r' = r + 1, l'_{(k', k)} = l_{(k', k)} = 0, r < n, k = 1, 2, 3 \\ \mu_r & r' = r - 1, l'_{(k', k)} = l_{(k', k)} = 0, r < n, k = 1, 2, 3 \\ \lambda_{k', k} & r' = r + 1, l'_{(k', k)} = l_{(k', k)} + 1, n \leq r < N, k = 1, 2, 3 \\ (\mu_n + l_{(k', 3)}\eta) & r' = r - 1, l'_{(k', 3)} = l_{(k', 3)} - 1, n < r \leq N \\ (\mu_n + l_{(k', 2)}\eta) * 1_{l_{(k', 3)}=0} + l_{(k', 2)}\eta * 1_{l_{(k', 3)} \neq 0} & r' = r - 1, l'_{(k', 2)} = l_{(k', 2)} - 1, n < r \leq N \\ \mu_n * 1_{l_{(k', 3)} + l_{(k', 2)} = 0} & r' = r - 1, l'_{(k', 1)} = l_{(k', 1)} - 1, n < r \leq N \end{cases}$$

From above, the transition rate matrix, $\mathbf{Q}$, governing the evolution of the multi-dimensional Markov chain, can be easily obtained as follows.

$$\begin{bmatrix} -\lambda & \lambda & 0 & \cdots & 0 & 0 & 0 \\ \mu_1 & -\lambda - \mu_1 & \lambda & \cdots & 0 & 0 & 0 \\ 0 & \mu_2 & -\lambda - \mu_2 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & \ddots & \ddots & \ddots & -\mu_n - (N - n)\eta & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 & -\mu_n - (N - n)\eta \end{bmatrix}$$

The following probabilities are said to constitute the steady-state probability distribution of the CTMC.

$$\boldsymbol{\pi} = [\pi_{(0,0,0,0)}^{k'}, \pi_{(1,0,0,0)}^{k'}, \cdots, \pi_{(r, l(k', 3), l(k', 2), l(k', 1))}^{k'}, \cdots]$$

According to [85], $\boldsymbol{\pi}$ can be obtained by the solution of

$$\begin{cases} \boldsymbol{\pi} \mathbf{Q} = 0 \\ \sum_{\forall (r, l(k', 3), l(k', 2), l(k', 1))} \pi_{(r, l(k', 3), l(k', 2), l(k', 1))}^{k'} = 1 \\ \pi_{(r, l(k', 3), l(k', 2), l(k', 1))}^{k'} \geq 0 \end{cases}$$

5.3.2.2.2 Delay and Loss Rate

In order to analyze the delay of real-time messages, we analyze the waiting time first. Suppose that an arriving customer of class k' starts his waiting time in state $(r, l_1, \dots, l_I)$, where l_k denotes the state of each class in the system. Assume that there are $l_{k'} - 1$ customers

of the same priority class and $\sum_{k=1}^{k'-1} l_k$ customers of classes $k \in \{1, \dots, k'-1\}$ are in front of it in the waiting line. Therefore, for one message to be served, it has to wait for the customers with higher or equal priority to leave the system. Moreover, it has to wait for the messages with higher priority which arrives during its waiting time. We define by $W_{w', B_1, B_2, F}(x)$ the conditional probability that the CTMC $Z(t)$, starting in state $(r(0), l_1(0), \dots, l_I(0)) = (r, l_1, \dots, l_I)$ with $B_1 = \sum_{k=1}^{k'} l_k$ waiting customers of the priority classes $k \in \{1, \dots, k'\}$, $B_2 = \sum_{k=k'+1}^{I/2} l_k$ waiting customers of the priority classes $k \in \{k'+1, \dots, I/2\}$ and $F = N - \sum_{k=1}^I l_k$ free waiting places at time $t=0$, reaches a state $(r(t), l_1(t), \dots, l_I(t)) = (r', 0, \dots, 0, l'_k, \dots, l'_I)$ for the first time no later than at time x . The Laplace-Stieltjes Transform (LST) is given by

$$\tilde{W}_{k', B_1, B_2, F}(s) = \int_0^\infty e^{-sx} dW_{k', B_1, B_2, F}(x)$$

To calculate the LST $\tilde{W}_{k', l_1 + \dots + l_{k'}, l_{k'+1} + \dots + l_{I/2}, N - (l_1 + \dots + l_I)}(s)$ we use the homogeneity of the CTMC $Z(t)$ and take into account the effects of arrival and departure events to the variables B_1 , B_2 and F . For the real-time message with class k' , let $\hat{\lambda}_{k', 3}$ denote the total arrival rate of the real-time messages with higher priority, $\hat{\lambda}_{k', 2}$ denote the total arrival rate of the real-time messages with equal or lower priority, and $\hat{\lambda}_{k', 1}$ denotes the total arrival rate of the non real-time messages. Then setting $\hat{\lambda}_{k', 1} = \sum_{k=I/2+1}^I \lambda_k$, $\hat{\lambda}_{k', 2} = \sum_{k=k'}^{I/2} \lambda_k$ and $\hat{\lambda}_{k', 3} = \sum_{k=1}^{k'-1} \lambda_k$, we obtain the following relationships.

For $B_1 \neq 1$, $B_2 \neq 0$ and $F \neq 0$:

$$\begin{aligned} \tilde{W}_{k', B_1, B_2, F}(s) &= [s + \mu_n + \lambda + (B_1 + B_2)\eta]^{-1} (\mu_n + B_1\eta) \tilde{W}_{k', B_1-1, B_2, F+1}(s) \\ &+ [s + \mu_n + \lambda + (B_1 + B_2)\eta]^{-1} \hat{\lambda}_{k', 3} \tilde{W}_{k', B_1+1, B_2, F-1}(s) \\ &+ [s + \mu_n + \lambda + (B_1 + B_2)\eta]^{-1} \hat{\lambda}_{k', 2} \tilde{W}_{k', B_1, B_2+1, F-1}(s) \\ &+ [s + \mu_n + \lambda + (B_1 + B_2)\eta]^{-1} \hat{\lambda}_{k', 1} \tilde{W}_{k', B_1, B_2, F-1}(s) \\ &+ [s + \mu_n + \lambda + (B_1 + B_2)\eta]^{-1} B_2 \eta \tilde{W}_{k', B_1, B_2-1, F+1}(s) \end{aligned}$$

For $F = 0$:

$$\begin{aligned}\tilde{W}_{k',B_1,B_2,0}(s) &= [s + \mu_n + (B_1 + B_2)\eta]^{-1}(\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,1}(s) \\ &+ [s + \mu_n + (B_1 + B_2)\eta]^{-1}B_2\eta\tilde{W}_{k',B_1,B_2-1,1}(s)\end{aligned}$$

Thus it follows for $B_1 \neq 1$, $B_2 \neq 0$ and $F \neq 0$:

$$\begin{aligned}&[s + \mu_n + \lambda + (B_1 + B_2)\eta]\tilde{W}_{k',B_1,B_2,F}(s) - (\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,F+1}(s) - \hat{\lambda}_{k',3}\tilde{W}_{k',B_1+1,B_2,F-1}(s) \\ &- \hat{\lambda}_{k',2}\tilde{W}_{k',B_1,B_2+1,F-1}(s) - B_2\eta\tilde{W}_{k',B_1,B_2-1,F+1}(s) \\ &= \hat{\lambda}_{k',1}\tilde{W}_{k',B_1,B_2,F-1}(s)\end{aligned}\quad (5.13)$$

and for $F = 0$:

$$[s + \mu_n + (B_1 + B_2)\eta]\tilde{W}_{k',B_1,B_2,0}(s) - (\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,1}(s) - B_2\eta\tilde{W}_{k',B_1,B_2-1,1}(s) = 0 \quad (5.14)$$

Then the LST of $\tilde{W}_{k',B_1,B_2,F}(s)$ can be obtained. Let $B_1 + B_2 + F = M$. Then on the left hand sides of above two equations it holds that $B_1 + B_2 + F = M$ and on the right hand sides $B_1 + B_2 + F = M - 1$. Ordering $\tilde{W}_{k',B_1,B_2,F}(s)$ and setting

$$\begin{aligned}\tilde{W}_{k',0,M}(s) &= (\tilde{W}_{k',1,0,M-1}(s), \tilde{W}_{k',2,0,M-2}(s), \dots, \tilde{W}_{k',M,0,0}(s))^t \\ \tilde{W}_{k',1,M}(s) &= (\tilde{W}_{k',1,1,M-2}(s), \tilde{W}_{k',2,1,M-3}(s), \dots, \tilde{W}_{k',M-1,1,0}(s))^t \\ &\vdots \\ \tilde{W}_{k',M-1,1}(s) &= (\tilde{W}_{k',1,M-1,0}(s))^t\end{aligned}$$

The equations (5.13) and (5.14) yield an inhomogeneous system of linear equations

$$A_{k',M}(s) \begin{pmatrix} \tilde{W}_{k',0,M}(s) \\ \tilde{W}_{k',1,M}(s) \\ \vdots \\ \tilde{W}_{k',M-2,M}(s) \\ \tilde{W}_{k',M-1,M}(s) \end{pmatrix} = -\hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}(s) \\ 0 \\ \tilde{W}_{k',1,M-1}(s) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-2,M-1}(s) \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \mu_n + \eta \\ 0_{C-1} \\ \mu_n + \eta \\ 0_{C-2} \\ \vdots \\ \mu_n + \eta \\ 0 \\ \mu_n + \eta \end{pmatrix} \quad (5.15)$$

where $0_{M-1} \in \mathfrak{R}^{M-1 \times 1}$, and $\mathfrak{R}^{i \times j}$ denotes a matrix with the dimension $i \times j$. The matrix

$A_{k',M}(s) \in \mathfrak{R}^{M(M+1)/2 \times M(M+1)/2}$ is given by

$$A_{k',M}(s) = \begin{pmatrix} -s - \mu_n - \lambda - \eta & \hat{\lambda}_{k',3} & 0 & \cdots & \cdots & \cdots & 0 \\ \mu_n + 2\eta & -s - \mu_n - \lambda - 2\eta & \hat{\lambda}_{k',3} & \cdots & \cdots & \cdots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \eta & -s - \mu_n - \lambda - M\eta & 0 \\ 0 & \ddots & \ddots & \ddots & (M-1)\eta & 0 & -s - \mu_n - M\eta \end{pmatrix}$$

We obtain the LST of the actual waiting time of class $1 \leq k' \leq I/2$ customers by means of the distribution $\pi_{(r,l(k',3),l(k',2),l(k',1))}$. Setting $B_1 = \sum_{k=1}^{k'} l_k + 1 = l(k',3) + 1$,

$B_2 = \sum_{k=k'}^{I/2} l_k = l(k',2)$ and $F = N - (l(k',3) + l(k',2) + l(k',1) + 1)$, for $k' > 1$, we have

$$W_{k'}(s) = \sum_{\substack{\forall (r,l(k',3),l(k',2),l(k',1)) \\ n \leq r < N}} \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'} \tilde{W}_{k',l(k',3)+1,l(k',2),(N-(l(k',3)+l(k',2)+l(k',1)+1))}(s) + \sum_{r=0}^{n-1} \pi_{(r,0,0,0)}^{k'} \\ + \sum_{\substack{\forall (r,l(k',3),l(k',2),l(k',1)) \\ r=N}} \pi_{(N,l(k',3),l(k',2),l(k',1))}^{k'}$$

And for $k' = 1$,

$$W_{k'}(s) = \sum_{\substack{\forall (r,l(1,3),l(1,2),l(1,1)) \\ n \leq r < N}} \pi_{(r,l(1,3),l(1,2),l(1,1))}^1 ((n\mu + (l(1,3) + 1)\eta) / (s + (n\mu + (l(1,3) + 1)\eta)))^{l(1,3)+1} \\ + \sum_{r=0}^{n-1} \pi_{(r,0,0,0)}^1 + \sum_{\substack{\forall (r,l(1,3),l(1,2),l(1,1)) \\ r=N}} \pi_{(N,l(1,3),l(1,2),l(1,1))}^1$$

The LST of the conditional waiting time $W_{k',cond}(S)$ the can be obtained by the following relationship

$$W_{k',cond}(S) = W_{k'}(S) / (1 - \sum_{\substack{\forall (r,l(k',3),l(k',2),l(k',1)) \\ r=N}} \pi_{(N,l(k',3),l(k',2),l(k',1))}^{k'})$$

Therefore, the waiting time of class k' traffic can be obtained from

$$W_{mean} = -dW_{k',cond}(S) / ds \big|_{s=0}. \quad (5.16)$$

In order to solve the equation (5.16), we can change the form of (5.15) by differential.

$$A_{k',M}'(0) \begin{pmatrix} \tilde{W}_{k',0,M}(0) \\ \tilde{W}_{k',1,M}(0) \\ \vdots \\ \tilde{W}_{k',M-2,M}(0) \\ \tilde{W}_{k',M-1,M}(0) \end{pmatrix} + A_{k',M}(0) \begin{pmatrix} \tilde{W}_{k',0,M}'(0) \\ \tilde{W}_{k',1,M}'(0) \\ \vdots \\ \tilde{W}_{k',M-2,M}'(0) \\ \tilde{W}_{k',M-1,M}'(0) \end{pmatrix} = -\hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}'(0) \\ 0 \\ \tilde{W}_{k',1,M-1}'(0) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-2,M-1}'(0) \\ 0 \\ 0 \end{pmatrix} \quad (5.17)$$

Also, from (5.15) and (5.17), we can have

$$A_{k',M}^2(0) \begin{pmatrix} \tilde{W}_{k',0,M}'(0) \\ \tilde{W}_{k',1,M}'(0) \\ \vdots \\ \tilde{W}_{k',M-2,M}'(0) \\ \tilde{W}_{k',M-1,M}'(0) \end{pmatrix} = -A_{k',M}'(0) \begin{pmatrix} -\hat{\lambda}_{k',1} \\ \begin{pmatrix} \tilde{W}_{k',0,M-1}(0) \\ 0 \\ \tilde{W}_{k',1,M-1}(0) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-2,M-1}(0) \\ 0 \\ 0 \end{pmatrix} \\ \begin{pmatrix} \mu_n + \eta \\ 0_{M-1} \\ \mu_n + \eta \\ 0_{M-2} \\ \vdots \\ \mu_n + \eta \\ 0 \\ \mu_n + \eta \end{pmatrix} \end{pmatrix} - A_{k',M}(0) \hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}'(0) \\ 0 \\ \tilde{W}_{k',1,M-1}'(0) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-2,M-1}'(0) \\ 0 \\ 0 \end{pmatrix}$$

For $M=1$ it holds that $A_{k',1}(s) = -s - n\mu - \eta$, so the solution is given by $\tilde{W}_{k',1,0,0}(0) = 1$ and

$\tilde{W}_{k',1,0,0}'(0) = -1/(n\mu + \eta)$. The unique solution $\tilde{W}_{k',B_1,B_2,F}'(0)$ can be calculated for all

$M = B_1 + B_2 + F$ by increasing values of M starting with $M=1$.

Let $D_{k',cond}(s)$ denotes the LST of the p.d.f of conditional delay d for the real-time message with class k' and $L_{k'}(s)$ denotes the LST of the p.d.f of transmission time l for the real-time message with class k' . And from the fact that $d=l+w$, where w is the waiting time, we have $D_{k',cond}(s) = L_{k'}(s) * W_{k',cond}(s)$. Because message length follows exponential distribution, we have $L_{k'}(s) = \mu/(s + \mu)$. Then the average delay can be obtained from

$$\begin{aligned} D_{mean} &= -dD_{k',cond}(s)/ds|_{s=0} \\ &= -\left(\frac{\mu}{s + \mu} \times dW_{k',cond}(s)\right)|_{s=0} - \left(W_{k',cond}(s) \times \frac{-\mu}{(s + \mu)^2}\right)|_{s=0} \\ &= -dW_{k',cond}(s)|_{s=0} + \frac{1}{\mu} W_{k',cond}(0) \\ &= -dW_{k',cond}(s)|_{s=0} + \frac{1}{\mu} \end{aligned} \quad (5.18)$$

Therefore, the average delay of the real-time class k' can be obtained from (5.18) once $W_{k',cond}(S)$ is obtained.

On the other hand, the loss rate, which is defined as the probability that the message is lost due to the violation of laxity, can be obtained from

$$\begin{aligned}\Pr(T < w) &= \int_0^\infty \Pr(T < t \mid w = t) f_w(t) dt \\ &= \int_0^\infty (1 - e^{-\eta t}) f_w(t) dt \\ &= 1 - \int_0^\infty e^{-\eta t} f_w(t) dt \\ &= 1 - W_{k',cond}(\eta)\end{aligned}$$

where T denotes laxity, w denotes the waiting time and $f_w(t)$ denotes the probability density function of waiting time. Then from (5.15), we have

$$A_{k',M}(\eta) \begin{pmatrix} \tilde{W}_{k',0,M}(\eta) \\ \tilde{W}_{k',1,M}(\eta) \\ \vdots \\ \tilde{W}_{k',M-2,M}(\eta) \\ \tilde{W}_{k',M-1,M}(\eta) \end{pmatrix} = -\hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}(\eta) \\ 0 \\ \tilde{W}_{k',1,M-1}(\eta) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-2,M-1}(\eta) \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \mu_n + \eta \\ 0_{M-1} \\ \mu_n + \eta \\ 0_{M-2} \\ \vdots \\ \mu_n + \eta \\ 0 \\ \mu_n + \eta \end{pmatrix}$$

For $M=1$ it holds that $A_{k',1}(s) = -s - \mu_n - \eta$, so the solution is given by

$\tilde{W}_{k',1,0,0}(\eta) = (\mu_n + \eta) / (\mu_n + 2\eta)$. The unique solution $\tilde{W}_{k',B_1,B_2,F}'(\eta)$ can be calculated for all $M = B_1 + B_2 + F$ by increasing values of M starting with $M=1$. As such, the loss rate of the real-time class k' is obtained.

5.3.2.3 Performance for Non Real-Time Message Scenario

5.3.2.3.1 The Steady State Distribution

For any arriving non real time class $k' \in \{I/2+1, \dots, I\}$, we have to know 1) the number of real-time messages, denoted by $l(k',3)(t)$; 2) the number of non real-time messages with higher or equal priority, denoted by $l(k',2)(t)$; and 3) the number of non real-time messages

with lower priority, denoted by $l(k',1)(t)$, in the waiting line at the arrival instant t . Then we define $l(k',3)(t) = \sum_{k=1}^{I/2} l_k(t)$, $l(k',2)(t) = \sum_{k=I/2+1}^{k'} l_k(t)$ and $l(k',1)(t) = \sum_{k=k'+1}^I l_k(t)$.

Similar to the analysis of real-time messages, we use $(r(t), l(k',3)(t), l(k',2)(t), l(k',1)(t))$ to denote the state of the system at time t , where $r(t) = l(k',3)(t) + l(k',2)(t) + l(k',1)(t) + n$ is the total number of messages in the system, including those in the queue and those in the servers, and $0 \leq r(t) \leq N$.

Let $\lambda_{k',3}$ denotes the total arrival rate of the real-time messages, $\lambda_{k',2}$ denote the total arrival rate of the real-time messages with higher or equal priority and $\lambda_{k',1}$ denote the total arrival rate of the non real-time messages with lower priority. We define $\lambda_{k',3} = \sum_{k=1}^{I/2} \lambda_k$, $\lambda_{k',2} = \sum_{k=I/2+1}^{k'} \lambda_k$ and $\lambda_{k',1} = \sum_{k=k'+1}^I \lambda_k$. Similar to the model to analyze the performance of real-time messages, the service rate is dependent on the messages currently occupying the servers according to (5.12.1) and (5.12.2).

Then the steady state distribution is shown in Figure 5.34, Figure 5.35 and Figure 5.36. As an example, for steady state $(r, l(k',3), l(k',2), l(k',1))$ in Figure 5.35, where $n < r < N$, there are twelve transitions into and out of that state. Based on the criterion that the message with higher priority definitely goes first than that with lower priority. Therefore, the transition rate from $(r, l(k',3), l(k',2), l(k',1))$ to $(r, l(k',3)-1, l(k',2), l(k',1))$ is $\mu_n + l(k',3)\eta$, which means that a message in $l(k',3)$ leaves the system with probability of μ_n because of the end of the service or with probability of $l(k',3)\eta$ because of the violation of the time constraints. The message is considered as lost in the latter case. On the other hand, the transition rate from $(r, l(k',3), l(k',2), l(k',1))$ to $(r, l(k',3), l(k',2)-1, l(k',1))$ is $\mu_n * 1_{l(k',3)=0}$, which is different from the case in Figure 5.32. This is because that, in this model for non real-time messages, $l(k',2)$ denotes the non real-time messages with higher or lower priority and no time constraints are associated with them. Moreover, it can only be served under the

condition that $l(k',3)=0$. Following the same way, the steady state distribution can be expressed as follows.

Figure 5.34 Transition diagram when $0 \leq r \leq n$

Figure 5.35 Transition diagram when $n < r < N$

Figure 5.36 Transition diagram when $r=N$

For $r=0$;

$$0 = \pi_{(0,0,0,0)}^{k'}(-\lambda) + \pi_{(1,0,0,0)}^{k'}\mu_1$$

for $0 < r < n$

$$0 = \pi_{(r,0,0,0)}^{k'}(-\lambda - \mu_r) + \pi_{(r-1,0,0,0)}^{k'}\lambda + \pi_{(r+1,0,0,0)}^{k'}\mu_{r+1}$$

for $r=n$

$$0 = \pi_{(n,0,0,0)}^{k'}(-\lambda - \mu_n) + \pi_{(n-1,0,0,0)}^{k'}\lambda + \pi_{(n+1,1,0,0)}^{k'}(\mu_n + \eta) + \pi_{(n+1,0,1,0)}^{k'}\mu_n + \pi_{(n+1,0,0,1)}^{k'}\mu_n$$

for $n < r < N$ and $l(k', 3) \neq 0$

$$0 = \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'}(-\mu_n - \lambda - l(k', 3)\eta) + \sum_{k=1}^3 \pi_{(r-1,\dots,l(k',k)-(1-\delta_{l(k',k),0}),\dots)}^{k'}\lambda_{k',k}(1-\delta_{l(k',k),0}) \\ + \pi_{(r+1,l(k',3)+1,l(k',2),l(k',1))}^{k'}(\mu_n + (l(k', 3) + 1)\eta)$$

for $n < r < N$, $l(k', 3) = 0$ and $l(k', 2) \neq 0$

$$0 = \pi_{(r,0,l(k',2),l(k',1))}^{k'}(-\mu_n - \lambda) + \sum_{k=1}^2 \pi_{(r-1,\dots,l(k',k)-(1-\delta_{l(k',k),0}),\dots)}^{k'}\lambda_{k',k}(1-\delta_{l(k',k),0}) \\ + \pi_{(r+1,1,l(k',2),l(k',1))}^{k'}(\mu_n + \eta) + \pi_{(r+1,0,l(k',2)+1,l(k',1))}^{k'}\mu_n$$

for $n < r < N$, $l(k', 3) = 0$, $l(k', 2) = 0$ and $l(k', 1) \neq 0$

$$0 = \pi_{(r,0,0,l(k',1))}^{k'}(-\mu_n - \lambda) + \pi_{(r-1,0,0,l(k',1)-1)}^{k'}\lambda_{k',1} \\ + \pi_{(r+1,1,0,l(k',1))}^{k'}(\mu_n + \eta) + \pi_{(r+1,0,1,l(k',1))}^{k'}\mu_n + \pi_{(r+1,0,0,l(k',1)+1)}^{k'}\mu_n$$

for $r = N$

$$0 = \pi_{(N,l(k',3),l(k',2),l(k',1))}^{k'}(-\mu_n - l(k', 3)\eta) + \sum_{k=1}^3 \pi_{(N-1,\dots,l(k',k)-(1-\delta_{l(k',k),0}),\dots)}^{k'}\lambda_{k',k}(1-\delta_{l(k',k),0})$$

Let $r_{x \rightarrow x'}$ be the transition rate from the current state $x = (r(t), l(k', 3)(t), l(k', 2)(t), l(k', 1)(t))$ to the destination legal state $x' = (r'(t), l'(k', 3)(t), l'(k', 2)(t), l'(k', 1)(t))$. The transition rate is given in the following equation.

$$r_{x \rightarrow x'} = \begin{cases} \lambda & r' = r + 1, l'_{(k',k)} = l_{(k',k)} = 0, r < n, k = 1, 2, 3 \\ \mu_r & r' = r - 1, l'_{(k',k)} = l_{(k',k)} = 0, r < n, k = 1, 2, 3 \\ \lambda_{k',k} & r' = r + 1, l'_{(k',k)} = l_{(k',k)} + 1, n \leq r < N, k = 1, 2, 3 \\ (\mu_n + l_{(k',3)}\eta) & r' = r - 1, l'_{(k',3)} = l_{(k',3)} - 1, n < r \leq N \\ \mu_n * 1_{l_{(k',3)}=0} & r' = r - 1, l'_{(k',2)} = l_{(k',2)} - 1, n < r \leq N \\ \mu_n * 1_{l_{(k',3)}+l_{(k',2)}=0} & r' = r - 1, l'_{(k',1)} = l_{(k',1)} - 1, n < r \leq N \end{cases}$$

From above, the transition rate matrix, $\mathbf{Q}$, can be easily obtained as follows.

$$\begin{bmatrix} -\lambda & \lambda & 0 & \dots & 0 & 0 & 0 \\ \mu_1 & -\lambda - \mu_1 & \lambda & \dots & 0 & 0 & 0 \\ 0 & \mu_2 & -\lambda - \mu_2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ 0 & 0 & \ddots & \ddots & \ddots & -\mu_n - (N - n - 1)\eta & 0 \\ 0 & 0 & \ddots & \ddots & 0 & 0 & -\mu_n - (N - n)\eta \end{bmatrix}$$

The probabilities

$$\pi = [\pi_{(0,0,0,0)}^{k'}, \pi_{(1,0,0,0)}^{k'}, \dots, \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'}, \dots]$$

are said to constitute the steady-state probability distribution of the CTMC.

Then π can be obtained by the solution of

$$\begin{cases} \pi Q = 0 \\ \sum_{\forall (r,l(k',3),l(k',2),l(k',1))} \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'} = 1 \\ \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'} \geq 0 \end{cases}$$

5.3.2.3.2 Delay

In order to analyze the delay of real-time messages, we analyze the waiting time first.

Suppose that an arriving customer of class k' starts his waiting time in state $(r, l_1, \dots, l_I)$.

Assume that there are $l_{k'} - 1$ customers of the same priority class and $\sum_{k=1}^{k'-1} l_k$ customers of

classes $k \in \{1, \dots, k'-1\}$ are in front of it in the waiting line. For one message to be served, it

has to wait for the customers with higher or equal priority, i.e., $\sum_{k=1}^{k'} l_k$, to leave the system.

Moreover, it has to wait for the messages with higher priority which arrives during its waiting time. We define by $W_{w', B_1, B_2, F}(x)$ the conditional probability that the CTMC $Z(t)$,

starting in state $(r(0), l_1(0), \dots, l_I(0)) = (r, l_1, \dots, l_I)$ with $B_1 = \sum_{k=1}^{I/2} l_k$ waiting customers of real

time messages in the priority classes $k \in \{1, \dots, I/2\}$, $B_2 = \sum_{k=I/2+1}^{k'} l_k$ waiting customers of

the priority classes $k \in \{I/2+1, \dots, k'\}$ and $F = N - \sum_{k=1}^I l_k$ free waiting places at time $t=0$,

reaches a state $(r(t), l_1(t), \dots, l_I(t)) = (r', 0, \dots, 0, l'_1, \dots, l'_I)$ for the first time no later than at time

x . The Laplace Transform is given by

$$\tilde{W}_{k', B_1, B_2, F}(s) = \int_0^\infty e^{-sx} dW_{k', B_1, B_2, F}(x)$$

To calculate the LST $\tilde{W}_{k', l_1 + \dots + l_{I/2+1}, l_{I/2} + \dots + l_{k'}, N - (l_1 + \dots + l_I)}(s)$ we use the homogeneity of the CTMC

$Z(t)$ and take into account the effects of arrival and departure events to the variables B_1 , B_2

and F .

For the non real-time message with class k' , let $\hat{\lambda}_{k',3}$ denote the total arrival rate of the real-time messages, $\hat{\lambda}_{k',2}$ denote the total arrival rate of the non real-time messages with higher priority, and $\hat{\lambda}_{k',1}$ denotes the total arrival rate of the non real-time messages with equal or lower priority. Setting $\hat{\lambda}_{k',1} = \sum_{k=k'}^I \lambda_k$, $\hat{\lambda}_{k',2} = \sum_{k=I/2+1}^{k'-1} \lambda_k$ and $\hat{\lambda}_{k',3} = \sum_{k=1}^{I/2} \lambda_k$, we obtain the following relationships.

For $B_1 \neq 1$, $B_2 \neq 0$ and $F \neq 0$:

$$\begin{aligned} \tilde{W}_{k',B_1,B_2,F}(s) = & [s + \mu_n + \lambda + B_1\eta]^{-1}(\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,F+1}(s) \\ & + [s + \mu_n + \lambda + B_1\eta]^{-1}\hat{\lambda}_{k',3}\tilde{W}_{k',B_1+1,B_2,F-1}(s) \\ & + [s + \mu_n + \lambda + B_1\eta]^{-1}\hat{\lambda}_{k',2}\tilde{W}_{k',B_1,B_2+1,F-1}(s) \\ & + [s + \mu_n + \lambda + B_1\eta]^{-1}\hat{\lambda}_{k',1}\tilde{W}_{k',B_1,B_2,F-1}(s) \end{aligned}$$

For $F = 0$:
$$\tilde{W}_{k',B_1,B_2,0}(s) = [s + \mu_n + B_1\eta]^{-1}(\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,1}(s)$$

Thus it follows for $B_1 \neq 1$, $B_2 \neq 0$ and $F \neq 0$:

$$\begin{aligned} & [s + \mu_n + \lambda + B_1\eta]\tilde{W}_{k',B_1,B_2,F}(s) - \hat{\lambda}_{k',3}\tilde{W}_{k',B_1+1,B_2,F-1}(s) \\ & - (\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,F+1}(s) - \hat{\lambda}_{k',2}\tilde{W}_{k',B_1,B_2+1,F-1}(s) \\ & = \hat{\lambda}_{k',1}\tilde{W}_{k',B_1,B_2,F-1}(s) \end{aligned}$$

and for $F = 0$:
$$[s + \mu_n + B_1\eta]\tilde{W}_{k',B_1,B_2,0}(s) - (\mu_n + B_1\eta)\tilde{W}_{k',B_1-1,B_2,1}(s) = 0$$

Now the LST of $\tilde{W}_{k',B_1,B_2,F}(s)$ can be calculated. Let $B_1 + B_2 + F = M$. Then on the left hand sides of both equations it holds that $B_1 + B_2 + F = M$ and on the right hand sides $B_1 + B_2 + F = M - 1$. Ordering $\tilde{W}_{k',B_1,B_2,F}(s)$ and setting

$$\begin{aligned} \tilde{W}_{k',0,M}(s) &= (\tilde{W}_{k',1,0,M-1}(s), \tilde{W}_{k',2,0,M-2}(s), \dots, \tilde{W}_{k',M,0,0}(s))^t \\ \tilde{W}_{k',1,M}(s) &= (\tilde{W}_{k',0,1,M-2}(s), \tilde{W}_{k',1,1,M-2}(s), \tilde{W}_{k',2,1,M-3}(s), \dots, \tilde{W}_{k',M-1,1,0}(s))^t \\ &\vdots \\ \tilde{W}_{k',M,0}(s) &= (\tilde{W}_{k',0,M,0}(s))^t \end{aligned}$$

The above two equations yield an inhomogeneous system of linear equations

$$A_{k',M}(s) \begin{pmatrix} \tilde{W}_{k',0,M}(s) \\ \tilde{W}_{k',1,M}(s) \\ \vdots \\ \tilde{W}_{k',M-1,M}(s) \\ \tilde{W}_{k',M,M}(s) \end{pmatrix} = -\hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}(s) \\ 0 \\ \tilde{W}_{k',1,M-1}(s) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-1,M-1}(s) \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \mu_n + \eta \\ 0_{M-1} \\ \mu_n \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} \quad (5.19)$$

$0_{M-1} \in \Re^{M-1 \times 1}$. The matrix $A_{k',M}(s) \in \Re^{M(M+1)/2 \times M(M+1)/2}$ is given by

$$A_{k',M}(s) = \begin{pmatrix} -s - \mu_n - \lambda - \eta & \hat{\lambda}_{k',3} & 0 & \cdots & \cdots & \cdots & 0 \\ \mu_n + 2\eta & -s - \mu_n - \lambda - 2\eta & \hat{\lambda}_{k',3} & \cdots & \cdots & \cdots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \mu_n + \eta & -s - \mu_n - \eta & 0 \\ 0 & \ddots & \ddots & \ddots & \mu_n & 0 & -s - \mu_n \end{pmatrix}$$

For $M=1$ it holds that $A_{k',1}(s) = -s - \mu_n - \eta$, so the solution is given by

$\tilde{W}_{k',1,0,0}(s) = (s + \mu_n + \eta)^{-1}(\mu_n + \eta)$ and $\tilde{W}_{k',0,1,0}(s) = (s + \mu_n)^{-1}\mu_n$. The unique solution $\tilde{W}_{k',B_1,B_2,F}(s)$ can be calculated for all $M = B_1 + B_2 + F$ by increasing values of M starting with $M=1$. This will be used in the following section.

We obtain the LST of the actual waiting time of class $I/2+1 \leq k' \leq I$ customers by means of the distribution $\pi_{(r,l(k',3),l(k',2),l(k',1))}$. Setting $B_1 = \sum_{k=1}^{I/2} l_k + 1 = l(k',3) + 1$,

$B_2 = \sum_{k=I/2+1}^{k'-1} l_k = l(k',2)$ and $F = N - (l(k',3) + l(k',2) + l(k',1) + 1)$, we have

$$W_{k'}(s) = \sum_{\substack{\forall (r,l(k',3),l(k',2),l(k',1)) \\ n \leq r < N}} \pi_{(r,l(k',3),l(k',2),l(k',1))}^{k'} \tilde{W}_{k',l(k',3)+1,l(k',2),(m-(l(k',3)+l(k',2)+l(k',1)+1))}(s) + \sum_{r=0}^{n-1} \pi_{(r,0,0,0)}^{k'} \\ + \sum_{\substack{\forall (r,l(k',3),l(k',2),l(k',1)) \\ r=N}} \pi_{(N,l(k',3),l(k',2),l(k',1))}^{k'}$$

If we denote $W_{k',cond}(S)$ as the LST of the waiting time distribution under the condition that an arriving customer can enter the system, then we have the following relationship as

$$W_{k',cond}(S) = W_{k'}(S) / (1 - \sum_{r=N}^{\infty} \pi_{(N,l(k',3),l(k',2),l(k',1))}^{k'})$$

The waiting time of class k traffic can be obtained from $W_{mean} = -dW_{k',cond}(S)/ds|_{s=0}$.

In order to obtain the average waiting time W_{mean} , we can change the form of (5.19) by differential.

$$A_{k',M}'(0) \begin{pmatrix} \tilde{W}_{k',0,M}(0) \\ \tilde{W}_{k',1,M}(0) \\ \vdots \\ \tilde{W}_{k',M-1,M}(0) \\ \tilde{W}_{k',M,M}(0) \end{pmatrix} + A_{k',C}(0) \begin{pmatrix} \tilde{W}_{k',0,M}'(0) \\ \tilde{W}_{k',1,M}'(0) \\ \vdots \\ \tilde{W}_{k',M-1,M}'(0) \\ \tilde{W}_{k',M,M}'(0) \end{pmatrix} = -\hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}'(0) \\ 0 \\ \tilde{W}_{k',1,M-1}'(0) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-2,M-1}'(0) \\ 0 \\ 0 \end{pmatrix} \quad (5.20)$$

Also, from (5.19) and (5.20), we can have

$$A_{k',M}^2(0) \begin{pmatrix} \tilde{W}_{k',0,M}'(0) \\ \tilde{W}_{k',1,M}'(0) \\ \vdots \\ \tilde{W}_{k',M-1,M}'(0) \\ \tilde{W}_{k',M,M}'(0) \end{pmatrix} = -A_{k',M}'(0) \begin{pmatrix} -\hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}(0) \\ 0 \\ \tilde{W}_{k',1,M-1}(0) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-1,M-1}(0) \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \mu_n + \eta \\ 0_{M-1} \\ \mu_n \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} - A_{k',M}(0) \hat{\lambda}_{k',1} \begin{pmatrix} \tilde{W}_{k',0,M-1}'(0) \\ 0 \\ \tilde{W}_{k',1,M-1}'(0) \\ 0 \\ \vdots \\ \tilde{W}_{k',M-1,M-1}'(0) \\ 0 \\ 0 \end{pmatrix} \end{pmatrix}$$

For $M=1$ it holds that $A_{k',1}(s) = -s - \mu_n - \eta$, so the solution is given by $\tilde{W}_{k',1,0,0}(0) = 1$,

$\tilde{W}_{k',1,0,0}'(0) = -1/(\mu_n + \eta)$, $\tilde{W}_{k',0,1,0}(0) = 1$ and $\tilde{W}_{k',0,1,0}'(0) = -1/\mu_n$. The unique solution

$\tilde{W}_{k',B_1,B_2,F}'(0)$ can be calculated for all $M = B_1 + B_2 + F$ by increasing values of M starting with $M=1$.

From the fact that $D_{mean} = -dW_{k',cond}(s)|_{s=0} + \frac{1}{\mu}$, the average delay of the non real-time

message with class k' can be obtained.

5.3.2.4 Numerical Results

Due to the computation complexity, we reduce our former system model ($N=50$, $n=4$) to current model ($N=25$, $n=2$) pro rata, where N denotes the number of nodes in the network, n denotes the number of channels. In addition, we assume that message length varies according to an Exponential distribution with a mean value as 20 time slots. The message laxity is a random variable following an Exponential distribution with mean laxity as 25 time slots. Message arrivals at each source node comply with an independent Poisson process. The destination nodes are selected according to a uniform probability distribution. And the traffic intensity of each relative class is the same.

We have calculated the average delay for each relative class and the loss rate of each relative class. The results are presented in Figure 5.37, Figure 5.38 and Figure 5.39. Figure 5.37 depicts the average message delay for the real-time relative classes under different traffic loads. It is easy to see that the class with higher relative priority outperforms that with lower relative priority, especially the traffic load is high. This is expected since we have defined that the messages with higher relative priority definitely go first than those with lower relative priority. Similarly, in Figure 5.38, which illustrates the average delay of non real-time relative class with traffic load, the class with higher relative priority outperforms that with lower relative priority. Moreover, when the traffic load increases, delay of the class with lower relative priority increases faster than that of the class with higher relative priority. This is because that the message with lower relative priority cannot be served until all the messages with higher relative priority before it has been served. Therefore, when the traffic load becomes large, the number of messages with higher relative priority increases, which cause the waiting time for the messages with lower relative priority to increase, too. Figure 5.39 depicts the loss rate of real-time relative class versus traffic load. It is easy to see that the loss rates for these five real-time relative classes increase when the traffic load increases. However, the relative class with higher priority outperforms that with lower priority, especially when the traffic load is high. This is because that the messages in the relative class

with higher priority can be served first than those in the relative class with lower priority, resulting in the lower loss rate.

Figure 5.37 Average delay of real-time relative classes

Figure 5.38 Average delay of non-real-time relative classes

Figure 5.39 Loss rate of real-time relative classes

5.3.4 Performance of Network Traffic

The process to obtain the performance of the network traffic can be summarized as follows.

1) Firstly, according to the scheduling parameters α , β , γ , the value of dynamic priority P is divided into ten sections $\{I_1, I_2, \dots, I_{10}\}$ based on the criterion that the traffic intensity of each section is equal to each other. Each section is referred to the relative priority section. To comply with this criterion, the division of the sections should associate with the analysis of p.d.f and c.d.f of the dynamic priority P . Based on the p.d.f of message length and message laxity, the p.d.f of the dynamic priority P for real-time static class and non real-time static class can be obtained according to (5.6) and (5.9) respectively. After that, c.d.f of the dynamic priority P for each static class can be obtained. According to the c.d.f and the equations (5.8) and (5.11), the section of P can be acquired so that the probability that the value of dynamic priority P of static classes drops on each specific relative priority section is equal to each other. Once the section is obtained, the proportion of each static class c distributing to each static class k , namely n_k^c , can be obtained from

$\int_{p(k)}^{p(k+1)} f_{z_c}(z_c) dz_c$, where $f_{z_c}(z_c)$ is the p.d.f of the dynamic priority P of static class c .

2) Secondly, after relative class mapping is finished, our system can be modeled to the one with ten relative classes, which can be defined as a MCMS model with non-preemptive priorities. In order to analyze the performance of real-time and non real-time relative class traffic, different scenarios for these two kinds of traffic have been defined, which is shown in Section 5.3.3. And then the LST of the waiting time for each type of traffic is obtained. Based on the LST of waiting time, average delay for two kinds of traffic and loss rate for real-time traffic can be acquired.

3) Last, based on the performance of the relative class traffic and the proportion of each static class distributing to each relative class, the performance of static class can be obtained. Let $\text{delay}(i)$ denote the delay of real-time or non real-time static class i and $\text{loss}(i)$ denote

the loss rate of real-time relative class. Then the average delay of each real-time static class c can be acquired according to $n_1^c * \text{delay}(1) + n_2^c * \text{delay}(2) + n_3^c * \text{delay}(3) + n_4^c * \text{delay}(4) + n_5^c * \text{delay}(5)$, and the loss rate of each real-time static class c can be acquired according to $n_1^c * \text{loss}(1) + n_2^c * \text{loss}(2) + n_3^c * \text{loss}(3) + n_4^c * \text{loss}(4) + n_5^c * \text{loss}(5)$. And the average delay of each non real-time static class c can be obtained according to $n_6^c * \text{delay}(6) + n_7^c * \text{delay}(7) + n_8^c * \text{delay}(8) + n_9^c * \text{delay}(9) + n_{10}^c * \text{delay}(10)$.

Following we will present the result of a simple example when $\alpha = 0.5$, $\beta = 1$ and $\gamma = 1$ to show the process to obtain the performance of the network traffic. Firstly, based on the parameters, the p.d.f of dynamic priority P of real-time static class and non real-time static class can be obtained according to (5.6) and (5.9) respectively. And then, the c.d.f of each real-time static class can be obtained, which is shown in (5.7.1), (5.7.2) and (5.7.3). Similarly, the c.d.f of each non real-time static class can be acquired, which is shown in (5.10.1), (5.10.2) and (5.10.3). To comply with the criterion that the traffic intensity of each section is equal to each other, the section of P can be obtained according to the c.d.f of the static classes and the equations (5.8) and (5.11). The final result of the division of P is shown in Table 5.8.

Sections of P	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}
Region	0~ 0.6098	0.6098~ 0.9572	0.9572~ 1.2745	1.2745~ 1.6203	1.6203~ 3	3.5~ 3.7814	3.7814~ 4.1175	4.1175~ 4.4802	4.4802~ 4.7008	4.7008~ 5.5

Table 5.8 Region of relative priority

Therefore, the proportion of each static class c distributing to each relative class k , namely n_k^c , is obtained, as shown in Table 5.9. It is easy to see that, for static class 1, 55.55% of it drop on the region of relative class 1, 23.91% of it drop on the region of relative class 2 and so on. And the sum of the proportion for static class 1 distributing to each relative class is $55.55\% + 23.91\% + 11.58\% + 6.11\% + 2.85\% = 1$. In Table 5.9, the sum of the proportion for one specific static class distributing to each relative class is presented.

Static class Region of relative class	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
I_1	55.55%	4.45%	0	0	0	0
I_2	23.91%	36.09%	0	0	0	0
I_3	11.58%	28.11%	20.31%	0	0	0
I_4	6.11%	17.72%	36.17%	0	0	0
I_5	2.85%	13.63%	43.52%	0	0	0
I_6	0	0	0	60 %	0	0
I_7	0	0	0	28.73%	31.27%	0
I_8	0	0	0	10.95%	49.05%	0
I_9	0	0	0	0.032%	12.06%	47.62%
I_{10}	0	0	0	0	7.62%	52.38%
Sum	1	1	1	1	1	1

Table 5.9 Distribution of each static class

From Table 5.9 and the results of relative classes which are shown in Figure 5.37, Figure 5.38 and Figure 5.39, it is easy to obtain the performance results of each static class. For example, for static class 1, the average delay of it can be obtained from $n_1^1 * \text{delay}(1) + n_2^1 * \text{delay}(2) + n_3^1 * \text{delay}(3) + n_4^1 * \text{delay}(4) + n_5^1 * \text{delay}(5)$, where $\text{delay}(i)$ denotes the average delay of each relative class i , and loss rate of it can be obtained from $n_1^1 * \text{loss}(1) + n_2^1 * \text{loss}(2) + n_3^1 * \text{loss}(3) + n_4^1 * \text{loss}(4) + n_5^1 * \text{loss}(5)$, where $\text{loss}(i)$ denotes the loss rate of each relative class i . The final results are shown in Figure 5.40, Figure 5.41 and Figure 5.42.

Figure 5.40 Average delay of real-time static classes

In Figure 5.40, our study focuses on the average delay of real-time static classes versus the traffic load. From the figure, we can see that, in a low traffic load, the difference among

the three classes is not significant. However, as the traffic load increases, the class with higher static priority outperforms that with lower priority. This is because that, as Table 5.9 shows, the proportion of the real-time static class with higher priority distributing to the real-time relative class with higher priority will be larger than the proportion of the real-time static class with lower priority. For example, for static class 1, 55.55% out of its messages drop on the relative class 1, while, for static class 2, only 4.45% out of its messages drop on the relative class 1. Therefore, most of the messages in static class 1 will be served first than the messages in static class 2. That is, the larger the proportion of one real-time static class distributing to the real-time relative classes with higher priority, the better the performance this static class can achieve. Because of the same reason, the loss rate of real-time static class with higher priority is smaller than that of real-time static class with lower priority, which is shown in Figure 5.41. From these two results, we can arrive at a conclusion that in order to achieve good performance for one real-time static class, the efficient way is to adjust the proportion distribution of it so that the majority of its traffic drops on the relative real-time class with higher priority.

Figure 5.41 Loss rate of real-time static classes

Figure 5.42 illustrates the average delay of non real-time static classes versus traffic load. Similarly, the average delay of static class with higher priority works better than that of static class with lower priority. This is expected because, as Table 5.9 shows, the proportion of the

non real-time static class with higher priority distributing to the non real-time relative class with higher priority will be larger than the proportion of non real-time static class with lower priority. From this result, we can arrive at a conclusion that different proportion distribution of each non real-time static class distributing to the non real-time relative classes will lead to different performance. In order to achieve better performance for one non real-time static class, the efficient way is to make the proportion of it distributing the non real-time relative class with higher priority as large as possible.

Figure 5.42 Average delay of non real-time static classes

Until now, we have obtained the delay/loss rate of each static class. So by changing α , β and γ , the proportion of each static class distributing to the relative classes can be changed so that the different performance of each static class can be achieved. Based on this observation, a QoS service prediction structure can be set up to predict whether a set of α , β and γ can be found to satisfy the QoS requirement of the traffic coming in.

5.3.5 Scheduling Parameters Searching

As addressed in CBPS, different combinations of α , β and γ will lead to different performance. And, as Figure 5.20~Figure 5.25 shows, when two of the parameters are fixed, the performance will not change when the other parameter has been increased to some value. From these results, we observe that what actually affects the performance is the proportion

among these three parameters, i.e., the set of the parameters $\alpha = 1$, $\beta = 1$, $\gamma = 1$ will achieve the same performance as the set of the parameters $\alpha = 10$, $\beta = 10$, $\gamma = 10$ since $\alpha : \beta : \gamma = 1 : 1 : 1$ for both cases. Hence, it is necessary to define a scheduling domain in order to make the searching more efficient. From the results in Figure 5.20~Figure 5.25, it is easy to find that when β and γ is increased to 50, further change will hardly affect the performance. Therefore, it is possible for us to define a *searching region*, in which we have $\alpha \in [1, 50]$, $\beta \in [1, 50]$, $\gamma \in [1, 50]$ and $\alpha + \beta + \gamma = 50$. As such, the parameter generation can be processed by increasing the value of α first, and then increasing the value of β . The value of γ can be obtained from $50 - \alpha - \beta$. The parameter generation procedure from scheduling domain can be summarized as follows.

```

for (  $\alpha = 1$ ;  $\alpha \leq 50$ ;  $\alpha++$  )
{ for (  $\beta = 1$ ;  $\beta \leq 50$ ;  $\beta++$  )
{  $\gamma = 50 - \alpha - \beta$ ;
...
}
}

```

The detail description for the searching process of the parameters is presented in Figure 5.43. As the figure shows, there are three stages for the searching process. The first is to generate one set of the parameters in the scheduling region. And then based on the parameters, the division for the region of relative class as well as the proportion of each static class distributing to each relative class can be obtained according to the way addressed before. Last, with the results of region division of relative classes and proportion distribution of static classes with the results on the performance of relative classes, the performance of each static class based on the given set of parameters can be obtained. If the QoS requirements of all the traffic can be fulfilled, the searching will be ended with the given set

of parameters. Otherwise, the new searching process will start and another set of parameters in the scheduling region will be generated.

Figure 5.43 Flow chart for the searching process of the parameters

5.3.6 QoS Prediction Example

In order to show how the proposed QoS prediction scheme (see Figure 5.26) works when the traffic with specified QoS requirements comes into the network, one example is presented in this subsection. Table 5.10 shows the objective set of six classes of traffic coming in with different traffic loads. The class expresses its requirements in terms of delay or loss rate. These parameters are in turn used by the *QoS Prediction* as its objective set. After prediction process, a combination of α , β and γ is found and the performance obtained is also shown in Table 5.10. It is clearly that the QoS of these six classes of traffic can be fulfilled. And the combination of parameters of CBPS is $\alpha=1$, $\beta=1$, $\gamma=1$. Meanwhile, in order to test the accuracy of the developed model, its results are compared to those obtained from simulations. In Table 5.10, a good agreement between analysis and simulation can be observed.

Class	Traffic Load ρ	QoS Requirement	QoS Requirement	Search Result	Performance Predicted	Performance Predicted	Simulation Results	Simulation Results
-------	---------------------	-----------------	-----------------	---------------	-----------------------	-----------------------	--------------------	--------------------

		(delay)	(loss rate)		(delay)	(loss rate)	(delay)	(loss rate)
1	0.075	/	0.15	$\alpha=1,$ $\beta=1,$ $\gamma=1$	22.9481	0.0912	23.2144	0.0971
2	0.075	/	0.17		23.2340	0.0974	23.3808	0.1030
3	0.15	/	0.18		23.6284	0.1054	23.7551	0.1117
4	0.075	31	/		26.1890	/	27.3966	/
5	0.15	38	/		28.4574	/	29.3094	/
6	0.075	55	/		32.7322	/	33.0737	/

Table 5.10 Current QoS requirements

In the next stage, the case that *QoS Requirements Monitoring* detects two new traffic with different QoS requirements coming in is considered. The objective set of new traffic is shown in Table 5.11. So the traffic requirements are updated to Table 5.12. Following the prediction, a combination of α , β and γ is found by *QoS Prediction*, which is able to adapt the new QoS to the current QoS. The results are presented in Table 5.12. Similarly, the simulation results are presented, which agrees well with the results of analysis.

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)
2	0.05	/	0.165
6	0.05	55	/

Table 5.11 New QoS requirements I

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)	Search Result	Performance Predicted (delay)	Performance Predicted (loss rate)	Simulation Results (delay)	Simulation Results (loss rate)
1	0.075	/	0.15	$\alpha=1,$ $\beta=1,$ $\gamma=1$	23.6875	0.1142	23.8649	0.1180
2	0.125	/	0.165		24.1509	0.1241	24.3824	0.1313
3	0.15	/	0.18		24.7737	0.1365	24.6395	0.1440
4	0.075	31	/		28.4733	/	29.6725	/
5	0.15	38	/		32.0108	/	32.0438	/
6	0.125	55	/		39.8445	/	39.3336	/

Table 5.12 Updated QoS requirements I

In the following stage, *QoS Requirements Monitoring* detects another two new traffic so that the total traffic load is increased to 0.8. The objectivity set of the new traffic is shown in Table 5.13. According to the updated QoS requirements in Table 5.14, QoS prediction find that no such a combination of α , β and γ found, which comes to the *QoS Violation*. *QoS Estimation* will be invoked to determine whether the hopeless new traffic will be refused or not. Firstly, a prediction based on the current QoS is given in Table 5.15, which shows that the QoS does not violate. (It is worth mentioning that if the current QoS cannot be satisfied, the new traffic have to be refused.) Therefore, two estimations have been given out, which is

shown in Table 5.16 and Table 5.17 respectively. The results in Table 5.16 are obtained with the aim to give the least loss rate for the new traffic of class 1, regardless the requirement of new traffic of class 5. Meanwhile, the QoS of current QoS requirements has to be fulfilled. On the other hand, the results in Table 5.17 gives a recommendation to give the least delay for the new traffic of class 5, regardless of the requirement of new traffic of class 1. Similarly, the current QoS cannot be violated. The network can control the activities based on these two estimations and the network policies. The corresponding simulation results for each combination of parameters are also shown in the tables.

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)
1	0.05	/	0.12
5	0.05	36	/

Table 5.13 New QoS requirements II

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)	Search Result
1	0.125	/	0.12	QoS Violation
2	0.125	/	0.165	
3	0.15	/	0.18	
4	0.075	31	/	
5	0.2	36	/	
6	0.125	55	/	

Table 5.14 Updated QoS requirements II

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)	Search Result	Performance Predicted (delay)	Performance Predicted (loss rate)	Simulation Results (delay)	Simulation Results (loss rate)
1	0.125	/	0.15	$\alpha=19,$ $\beta=30,$ $\gamma=1$	24.5216	0.14	25.1661	0.1468
2	0.125	/	0.165		25.1957	0.1538	25.5421	0.1612
3	0.15	/	0.18		26.0864	0.171	28.3188	0.1783
4	0.075	31	/		29.4813	/	30.8716	/
5	0.2	38	/		37.5129	/	37.2348	/
6	0.125	55	/		53.8248	/	54.2929	/

Table 5.15 QoS Prediction with current QoS

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)	Search Result	Performance Predicted (delay)	Performance Predicted (loss rate)	Simulation Results (delay)	Simulation Results (loss rate)
1	0.125	/	0.15	$\alpha=24,$ $\beta=25,$ $\gamma=1$	24.1515	0.1374	25.0459	0.1418
2	0.125	/	0.165		25.1369	0.1534	25.3861	0.1590
3	0.15	/	0.18		26.1128	0.1715	25.8575	0.1798
4	0.075	31	/		29.4813	/	31.7710	/
5	0.2	38	/		37.5129	/	37.0983	/
6	0.125	55	/		53.8248	/	53.8855	/

Table 5.16 QoS estimation I

Class	Traffic Load ρ	QoS Requirement (delay)	QoS Requirement (loss rate)	Search Result	Performance Predicted (delay)	Performance Predicted (loss rate)	Simulation Results (delay)	Simulation Results (loss rate)
1	0.125	/	0.15	$\alpha=42,$ $\beta=3,$ $\gamma=5$	24.3836	0.1388	24.8687	0.1419
2	0.125	/	0.165		25.1904	0.1538	25.2119	0.1597
3	0.15	/	0.18		26.1128	0.1715	25.8646	0.1791
4	0.075	31	/		29.4813	/	30.8658	/
5	0.2	38	/		37.5129	/	38.7501	/
6	0.125	55	/		53.8248	/	53.5243	/

Table 5.17 QoS estimation II

As mentioned above, in order to check the accuracy of the analytical models, experimental simulations are conducted for each prediction (see Table 5.10, Table 5.12, Table 5.15, Table 5.16 and Table 5.17). Table 5.18 depicts the largest difference between the experimental and analytical values for each prediction table, in which the difference is computed as

$$\text{Difference} = |(\text{Result}_{\text{experimental}} - \text{Result}_{\text{analytical}})| / \text{Result}_{\text{experimental}}$$

From the table, we observe that the largest difference of these five predictions is 7.88%, which proves the accuracy of the analytical model.

	Largest difference (delay)	Largest difference (loss rate)
Table 5.10	4.4%	6.07%
Table 5.12	4.04%	5.48%
Table 5.15	7.88%	4.6%
Table 5.16	7.2%	4.6%
Table 5.17	4.48%	4.24%

Table 5.18 Largest difference between the experimental and analytical values

5.3.7 Summary

In this section, we have presented a novel framework for QoS service prediction in WDM optical networks. The architecture is composed of four modules that cooperate to dynamically adapt network behavior in order to maintain the level of the delivered QoS to end users. *QoS Requirements Monitoring* detects the changes in the supplied QoS. These changes are then reported to a *QoS Prediction* which dynamically adapt the network policies to meet the user's requirements. A module of *QoS Estimation* is used to make the decision whether the hopeless new traffic is refused or accepted with acceptable QoS

requirements when *QoS Violation* detects that the violation happens. The novelty of the proposed algorithm lies in that given sets of objectives, QoS prediction scheme performs at run-time. It has been shown that adapting policies at run-time provides flexible means for controlling network behavior as the surrounding environment changes.

The main part of this prediction structure is *QoS Prediction*, which aims to predict whether the current QoS or the new QoS can be satisfied or not. In order to set up the relationship between the static priority with the α , β , and γ , the concept of relative priority is introduced so that our system can be modeled as a MCMS model. In order to obtain the solution of the MCMS model, we have derived a recursive algorithm to compute the steady-state distribution of the number of customers in the system. From the results of MCMS model, it is easily to obtain the performance of each static class. Therefore, by changing α , β , and γ , it is easy to predict whether the QoS of the traffic can be met or not. In order to show the accuracy of the developed model, the simulations are conducted. A good agreement between analysis and simulation can be observed.

5.4 Summary

In this chapter, two novel reservation-based algorithms to provide Quality of Service (QoS) for single-hop passive-star coupled WDM optical networks have been proposed and their performance analyzed.

DDS algorithm is developed to handle real-time traffic. With the ability to consider the destination and channel availability of a single-hop passive-star coupled WDM optical network, it can efficiently avoid blocking of longer messages in order to make more messages meet their deadlines resulting in less message loss rate. Moreover, the sacrifice of the longer messages does not worsen the performance in terms of loss rate and throughput when the actual packet size is taken into account. Instead, such performance is improved by this technique.

In order to provide differentiated service to the messages with different time constraints, one novel algorithm, called CBPS, is introduced in this chapter. This algorithm is designed to handle messages with and without time constraints. And it considers the order of the message transmission based on the costs of messages incurred in the network. The message length, the relative laxity of a message, and the importance (or static priority) of a message are considered to be the factors to the cost of message transmission. This algorithm is proved to be able to reduce the messages loss rate as well as the average message delay when an integrated traffic is applied to the network so that the transmission of both types of messages could be benefited. Moreover, one set of experiments is conducted to evaluate the adaptability of CBPS algorithm, which shows that different combination of the parameters related to the cost factors will lead to different performance.

Based on the CBPS algorithm, one QoS service prediction structure is developed. This structure is designed based on the observation that by changing the parameters of CBPS, different results on performance can be obtained. Since it is hard to set up a direct relationship between the parameters of CBPS and the performance of each static priority directly, the relative priority is introduced so that our system can be modeled as a MCMS model. With the introduction of relative priority, QoS prediction structure can easily predict whether one suitable set of the parameters can be found to fulfill the QoS requirements of the connected traffic. Moreover, one novel framework for QoS service prediction in WDM optical networks is developed, which consists of four modules. *QoS Requirements Monitoring* detects the changes in the supplied QoS. And *QoS Prediction* predicts whether the new QoS can be adapted to the current QoS. If the violation is detected by *QoS Violation*, a module of *QoS Estimation* is used to make the decision whether the hopeless new traffic is refused or accepted with acceptable QoS requirements. Therefore, given a set of new QoS requirements, QoS prediction scheme has the ability to perform at run-time, which provides a flexible mean to control network behavior as the surrounding environment changes.

Chapter 6 . Linear Programming Based Off-Line Scheduling Policy

In the previous two chapters, several reservation-based access protocols for passive-star coupled WDM systems relying on CC-FT/TT-FR/TR architecture were proposed. These protocols aim to reduce either average delay or loss rate. And their superior performance has been shown. However, no attempt has been made to obtain the optimal performance. This is partly due to the difficulties of including both channel and destination constraints in a mathematical formulation that maintains sufficient structure to aid in the optimization. This chapter is devoted entirely to discuss a novel scheduling policy that is developed based on a Linear Programming approach to obtain the optimal scheduler.

6.1 Optimization by Linear Programming

Linear Programming [86][87] deals with the problem of minimizing or maximizing a linear function in the presence of linear inequalities. The high efficiency of the solving method helps to popularize the subject of linear programming and makes the industrial and military application practical. The population of linear programming can be attributed to many factors including its ability to model large and complex problems, and the ability to solve large problems in a reasonable amount of time.

Generally, linear programming is a mathematical approach to maximize or minimize a linear function of n variables $x_1, \dots, x_n$ of the form

$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

The maximization or minimization of function z is subject to m linear inequality constraints of the form

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i, \quad i = 1, \dots, m$$

where a_{ij} and b_i are constants.

A set of values $x_1, \dots, x_n$ that satisfies the above set of constraints is called a feasible point or a feasible vector. The set of all such points constitutes the feasible region or the feasible space. The function is to be maximized or minimized is called the objective function. The feasible vector that maximizes or minimizes the objective function is called the optimal feasible vector. However, an optimal feasible vector can fail to exist for two major reasons: (i) there are no feasible vectors due to incompatible constraints, or (ii) there is no maximum or minimum giving an unbounded value for the objective function.

Based on the LP approach, the problem of finding an optimal transmission schedule for a batch of messages in our system can be formulated in the following way. Given a traffic request matrix D , whose elements represent the length or the laxity of the messages that must be transmitted from any source node s to any destination node d , FIND a message sequence that guarantees the delivery of the requested traffic, while MINIMIZING the average delay or the loss rate for all the transmissions, SUBJECT TO technological constraints. Due to the difficulty to predict the traffic pattern in a long run, the optimization is done based on batch-by-batch basis, i.e., this is so-called local optimization.

Similar work can be found in [11][12][14][88][89]. However, our new scheme has the following salient features to differ it from other works. Firstly, the network with $N \geq C$ is considered and each node is assumed to have only one tunable transmitter and one tunable receiver to access the data channels. Secondly, to reduce the ratio of propagation delay to

message transmission time, we allow the length of the messages to be variable and only one variable-length message is allowed to transmit from one source node at one time. Thirdly, tuning time is not negligible with respect to the transmission times and will be incorporated in the scheduling algorithm. Moreover, the new scheme has the ability to avoid the tuning overhead of transmitters and receivers whenever possible. Fourthly, the algorithms are capable to avoid channel collisions as well as receiver collisions. Lastly, one of the algorithms supports the real-time traffic with time constraints. Therefore, the total number of the messages dropped due to the violation of the time constraints can be minimized.

We have developed two models. One is to minimize the average delay, which is presented in Section 6.2. The other is to minimize the loss rate, which is presented in Section 6.3.

6.2 Problem Formulation (Delay)

In this section, we develop a model to minimize the average delay, which is called LPI. We consider a network with N nodes, which are interconnected through an optical broadcast medium that can support C channels. It is assumed that $N \geq C$. Each node of network is provided with an FT/TT-FR/TR architecture, in which one pair of transceivers is fixed and tuned to the control channel, and another pair of transceivers is tunable with significant latencies and can be tuned to any data channel. Besides CAT and RAT, there are other two tables needed in the new algorithm. One is Source Channel (SC), which is an array of N elements, each of which is for one source node. $SC[i]=h$, where $0 \leq i \leq N$ and $0 \leq h \leq C$, means that, currently, the transmitter of source node i is connected to channel h , namely source channel. The other table is Destination Channel (DC), which is an array of N elements, each of which is for one destination node. $DC[j]=k$, where $0 \leq j \leq N$ and $0 \leq k \leq C$, means that, currently, the receiver of destination node j is connected to channel k , namely destination channel. The traffic demand is characterized by a $N \times N$ matrix, denoted by D_1 . Entry d_{1ij} of D_1 represents the demand in terms of message length for interconnections

from node i to node j . The problem is to decide the optimal sequencing of the messages to minimize the average delay, which can be formulated as an Integer Linear Programming (ILP) problem, which includes only integer variables.

6.2.1 Variable Definitions

Listed below are notations of variables and parameters used in this problem formulation.

Parameters:

N	= Number of nodes in the network
C	= Number of channels in the network
M	= Number of non-zero entries in the traffic demand matrix D_1 , i.e., number of messages waiting for transmission
T	= Tuning time of transceivers
R_j	= Propagation delay from the passive star coupler to the node j
$CAT[m]$	= Channel available time of channel m , $m = 1, \dots, C$
$RAT[r]$	= Receiver available time of destination r 's receiver, $r = 1, \dots, N$
$SC[i]$	= Source channel of source node s , $s = 1, \dots, N$
$DC[i]$	= Destination channel of destination node d , $d = 1, \dots, N$
t_i	= Transmission time of message i in the network, $i = 1, \dots, M$
H, X	= Arbitrary big number
A	= Arbitrary small number and $0 < A < 1$

Variables:

S_{im}	= Starting time of message i on channel m , $i = 1, \dots, M$, $m = 1, \dots, C$
S_i	= Actual starting time of message i
a_{im}	= The status of message processing on channel m (binary, i.e., 0 or 1)
$p_{i,j}$	= Status of operation sequence between message i and j , where message i and j are destined to the same destination (binary), $i = 1, \dots, M$, $j = 1, \dots, M$

$y_{i,j,m}$ = The status of operation sequence between message i and j in channel m ,
where message i and j are destined to the different destinations (binary)

6.2.2 System Constraints

The delay of each message is composed of two elements: 1) the message transmission time and 2) the overhead incurred due to the tunability of the system's transmitters and receivers. Since tuning times can be quite large with respect to transmission times, two transmissions from the same source at different wavelengths must be separated by at least T empty slots. This retuning generally yield very limited throughput. As a consequence, schedules must include the case that does not require retuning of transmitters or receivers, hence at the same channels, namely source channel or destination channel.

Additionally, there are two important issues involved in the design of the scheduling algorithms in WDM optical networks with FT/TT-FR/TR structure, which are to avoid channel collision and receiver collision. If more than one message is scheduled on the same channels at the same time duration, channel collision occurs. And if more than one message are transmitted to the same destination simultaneously, receiver collision occurs. These two constraints should be incorporated in the scheduling algorithm so that no time durations will be wasted due to the retransmission.

1) *Channel Constraints I*: This constraint is used to avoid the case that more than one wavelength is selected to transmit the message, i.e., one message can be processed on one channel only. Hence, a *zero-one* variable a_{im} is used to specify the status of message processing, i.e.,

$$a_{im} = 0, \text{ if message } i \text{ is not processed on channel } m$$

$$a_{im} = 1, \text{ if message } i \text{ is processed on channel } m$$

and

$$\sum_{m=1}^C a_{im} = 1 \quad (6.1)$$

In order to limit the value of S_{im} , which is the starting time of message i on channel m , it is necessary to define the constraints as follows. If m is not the source channel of the source node of message i , then

$$\begin{aligned} S_{im} &= T, \text{ if message } i \text{ is not processed on channel } m \\ S_{im} &\geq T+1, \text{ if message } i \text{ is processed on channel } m \end{aligned} \quad (6.2.1)$$

and the constraint between a_{im} and S_{im} is

$$\frac{S_{im} - T - A}{X} \leq a_{im} \leq S_{im} - T \quad (6.3.1)$$

where A is a small number and $0 < A < 1$.

With (6.2.1) and (6.3.1), it is easy to find that if message i is not processed on channel m , $a_{im}=0$. Then (6.3.1) becomes $\frac{S_{im} - T - A}{X} \leq 0 \leq S_{im} - T$, which forced S_{im} to be T . On the other hand, if message i is processed on channel m , $a_{im}=1$. Then (6.3.1) becomes $\frac{S_{im} - T - A}{X} \leq 1 \leq S_{im} - T$, which force S_{im} to be no less than $T+1$. From (6.1), it is easy to find that one and only value in $\{S_{im}\}$ is allowed to be no less than $T+1$. That is, one and only channel is assigned to the message.

And if m is the source channel of the source node of message i , i.e., no tuning overhead of transmitter will be induced, then

$$\begin{aligned} S_{im} &= 0, \text{ if message } i \text{ is not processed on channel } m \\ S_{im} &\geq 1, \text{ if message } i \text{ is processed on channel } m \end{aligned} \quad (6.2.2)$$

and the constraint between a_{im} and S_{im} is

$$\frac{S_{im} - A}{X} \leq a_{im} \leq S_{im} \quad (6.3.2)$$

Here constraints (6.2.1) and (6.2.2) compose the constraint (6.2), and (6.3.1) and (6.3.2) compose the constraint (6.3). Constraints (6.1)~(6.3) enforces the restrictive assumption that no more than one channels are used to transmit one message as well as the restrictive

assumption that message must start the transmission on one channel. Also, these three constraints incorporate the case that messages select the source channel to avoid transmitter retuning whenever possible.

With the constraints (6.2.1) and (6.2.2), it is easy to observe that if message chooses its source channel to transmit the message,

$$\sum_{m=1}^C S_{im} = S_{im'} + (C-1)*T + 1$$

where m' denotes the source channel of message i . And the actual starting time of message, which is denoted by S_i , should be

$$S_i = S_{im'} = \sum_{m=1}^C S_{im} - (C-1)*T - 1$$

Otherwise,

$$\sum_{m=1}^C S_{im} = S_{ik} + (C-2)*T + 1$$

where k denotes one arbitrary channel rather than the source channel, and the actual waiting is

$$S_i = S_{ik} = \sum_{m=1}^C S_{im} - (C-2)*T - 1$$

It is clear that the ways to obtain the actual starting time are different when the selected channel is different.

2) *Channel Constraint II*: This constraint is developed to avoid the channel collisions, i.e., two messages cannot be processed on the same channel simultaneously. For example, if message i and message j are not destined to the same destination and message i precedes message j on channel m , then

$$S_{jm} - S_{im} \geq a_{im} t_i \quad i = 1, \dots, M, \quad m = 1, \dots, C$$

On the other hand, if message j precedes message i , then

$$S_{im} - S_{jm} \geq a_{jm} t_j$$

This information relies on a *zero-one* variable y_{ijm} to specify operation sequence, i.e.,

$$y_{ijm} = 1, \text{ if message } i \text{ precedes } j \text{ on machine } m \\ = 0, \text{ otherwise}$$

According to (6.2.1) and (6.2.2), the starting times of each message are different when the channel considered is the source channel or not. Therefore, it is also necessary to distinguish the channels when defining the constraints. If message i selects its source channel m to transmit the message, and m is not message j 's source channel, we have

$$S_{im} \geq 1$$

If message j doesn't select channel m , its starting time is

$$S_{jm} = T$$

In this case, it is equivalent to that message j precedes message i on channel m , it should be

$$S_{im} - S_{jm} \geq 0$$

However, it will add one unnecessary constraint to S_{im} , which is $S_{im} \geq T$. Therefore, it is necessary to change the formula to

$$S_{im} + T - S_{jm} \geq t_j$$

With the similar considerations, the constraints can be defined as follows. If channel m is not the source channel of both message i and message j or if channel m is the source channel of message i 's as well as message j 's

$$S_{jm} - S_{im} + H(1 - y_{ijm}) \geq a_{im}t_i \quad (6.4.1)$$

$$S_{im} - S_{jm} + Hy_{ijm} \geq a_{jm}t_j \quad (6.5.1)$$

where H represents an arbitrary positive number such that (6.4.1) and (6.5.1) can be satisfied.

Otherwise, if channel m is the source channel of message i and not message j 's

$$S_{jm} - (S_{im} + (1 - a_{jm}) * T) + H(1 - y_{ijm}) \geq a_{im}t_i \quad (6.4.2)$$

$$S_{im} + (1 - a_{jm}) * T - S_{jm} + Hy_{ijm} \geq a_{jm}t_j \quad (6.5.2)$$

And if channel m is the source channel of message j and not message i 's

$$S_{jm} + (1 - a_{im}) * T - S_{im} + H(1 - y_{ijm}) \geq a_{im} t_i \quad (6.4.3)$$

$$S_{im} - (S_{jm} + (1 - a_{im}) * T) + Hy_{ijm} \geq a_{jm} t_j \quad (6.5.3)$$

The constraints (6.4) and (6.5) enforce the constraint that no more one message is transmitted on the same channel at the same time and are used to avoid the channel collisions.

3) *Destination Constraint*: Two messages cannot be destined to the same destination simultaneously. For example, if message i and message j are destined to the same destination and message i precedes message j , i.e., message j starts after the completion of message i , then

$$S_i - S_j \geq t_j$$

On the other hand, if message j precedes message i , then it is also necessary that

$$S_j - S_i \geq t_i$$

This set of *disjunctive constraints* enforces the restrictive assumption that each destination does not receive more than one message at any time. The formulation relies on a *zero-one* variable p_{ij} to specify operation sequence, i.e.,

$$\begin{aligned} p_{ij} &= 1, & \text{if message } i &\text{ precedes } j \\ &= 0, & \text{otherwise} \end{aligned}$$

As addressed above, if message chooses its source channel to transmit the message, the actual starting time, which is denoted by S_i , of message should be

$$S_i = S_{im'} = \sum_{m=1}^C S_{im} - (C-1) * T - 1$$

where m' denotes the source channel of message i .

Otherwise, the actual waiting is

$$S_i = S_{ik} = \sum_{m=1}^C S_{im} - (C-2) * T - 1$$

where k denotes an arbitrary channel rather than source channel.

Therefore, for message i , its actual starting time is

$$S_i = \sum_{m=1}^C S_{im} - ((C-1) - (1 - a_{im'})) * T - 1$$

Then constraints become

$$\sum_{m=1}^C S_{jm} - (C - 2 + a_{jm''}) * T - (\sum_{m=1}^C S_{im} - (C - 2 + a_{im'}) * T) + H(1 - p_{ij}) \geq t_i \quad (6.6)$$

$$\sum_{m=1}^C S_{im} - (C - 2 + a_{im'}) * T - (\sum_{m=1}^C S_{jm} - (C - 2 + a_{jm''}) * T) + Hp_{ij} \geq t_j \quad (6.7)$$

where m' denotes the source channel of message i and m'' denotes the source channel of message j .

The constraints (6.6) and (6.7) enforce that no destination collisions will happen.

4) *Channel Availability Constraint*: No any message can start the transmission on channel m before channel m becomes available. The inequalities representing this constraint in order for a set of S_{im} to be feasible are

$$S_{im} - 1 + H(1 - a_{im}) \geq CAT[m] \quad (6.8)$$

There are two cases for this constraint. One is that if message i selects channel m for transmission, we have $a_{im} = 1$. So (6.8) becomes $S_{im} - 1 \geq CAT[m]$, where the actual waiting time satisfies $S_i = S_{im} - 1$. And the other case is that if message i doesn't select channel m for transmission, we have $a_{im} = 0$. So (6.8) becomes $S_{im} - 1 + H \geq CAT[m]$. H is a very big constant, so $S_{im} - 1 + H \geq CAT[m]$ can be satisfied all the time.

5) *Destination Availability Constraint*: No any message can arrive at destination j before its receiver becomes available. If message i selects its source channel, say m' , to transmit the message, then

$$S_i = \sum_{m=1}^C S_{im} - (C - 1) * T - 1$$

Otherwise
$$S_i = \sum_{m=1}^C S_{im} - (C - 2) * T - 1$$

where k is an arbitrary channel rather than the source channel.

Moreover, since the tuning times can be quite large with respect to transmission times, schedules that require retuning of receivers for each packet also impact the performance in

terms of average message delay. As a consequence, schedules must include the case that does not require retuning of receivers, hence at the same channel with that receivers has been tuned to, namely destination channel.

If the channel selected is not the destination channel, then the constraint should be

$$S_i \geq RAT[r] + T - R_r$$

Otherwise,

$$S_i \geq RAT[r] - R_r$$

R_r is the propagation delay from passive star coupler to the destination node r .

Therefore, the corresponding constraints becomes

$$\sum_{m=1}^C S_{im} - [(C-2) + a_{jm'}] * T - 1 \geq RAT[r] + (1 - a_{jm''}) * T - R_r \quad (6.9)$$

where r denotes the destination of message i and m'' is the destination channel that the receiver has been tuned to.

6) *Basic Constraints*: Following are the constraints for some variables:

$$a_{im} = 0 \text{ or } 1$$

$$p_{ij} = 0 \text{ or } 1$$

$$y_{ijm} = 0 \text{ or } 1$$

(6.10)

6.2.3 Objective Function

The criterion for the optimization of the schedule is the minimization of the average delay of the messages. Let $\bar{D}$ denotes the average delay. Then

$$\bar{D} = \sum_{i=1}^M (S_i + t_i) / M$$

Hence, the objective function can be formulated with the objectivity to minimize the sum of actual starting times of all the messages, i.e., $\sum_{i=1}^M S_i$.

For any message i , if it selects its source channel m' as the transmission channel,

$$S_i = S_{im'} = \sum_{m=1}^C S_{im} - (C-1) * T - 1$$

Otherwise,

$$S_i = S_{ik} = \sum_{m=1}^C S_{im} - (C-2)*T - 1$$

where k is not the source channel.

To combine these two cases, we have

$$S_i = \sum_{m=1}^C S_{im} - (C-2 + a_{im'})*T - 1$$

Therefore, the objective function becomes

MINIMIZE

$$\sum_{i=1}^M (\sum_{m=1}^C S_{im} - (C-2 + a_{im'})*T) \quad (6.11)$$

Therefore, the determination of an optimal schedule can be formulated as an ILP model, with a set of precedence constraints and restrictive assumptions as follows

SUBJECT TO

$$\frac{S_{im} - T - A}{X} \leq a_{im} \leq S_{im} - T \quad 1 \leq i \leq M, 1 \leq m \leq C \text{ and } SC[i] \neq m$$

$$\frac{S_{im} - A}{X} \leq a_{im} \leq S_{im} \quad 1 \leq i \leq M, 1 \leq m \leq C \text{ and } SC[i] = m$$

$$\sum_{m=1}^C a_{im} = 1 \quad 1 \leq i \leq M, 1 \leq m \leq C$$

$$\left. \begin{aligned} \sum_{m=1}^C S_{jm} - (C-2 + a_{jm'})*T - (\sum_{m=1}^C S_{im} - (C-2 + a_{im'})*T) + H(1 - p_{ij}) &\geq t_i \\ \sum_{m=1}^C S_{im} - (C-2 + a_{im'})*T - (\sum_{m=1}^C S_{jm} - (C-2 + a_{jm'})*T) + Hp_{ij} &\geq t_j \end{aligned} \right\}$$

if i and j are destined to the same destination, m' is the source channel of the source node of message i and m'' is the source channel of the source node of message j

$$\left. \begin{array}{l} S_{jm} - S_{im} + H(1 - y_{ijm}) \geq t_i \\ S_{im} - S_{jm} + Hy_{ijm} \geq t_j \end{array} \right\} \quad \text{if } i \text{ and } j \text{ are not destined to the same destination and } m \text{ is}$$

the source channel of message i and message j or m is not the source channel of message i and message j

$$\left. \begin{array}{l} S_{jm} - (S_{im} + (1 - a_{jm}) * T) + H(1 - y_{ijm}) \geq a_{im} t_i \\ S_{im} + (1 - a_{jm}) * T - S_{jm} + Hy_{ijm} \geq a_{jm} t_j \end{array} \right\} \quad \text{if } i \text{ and } j \text{ are not destined to the same}$$

destination and m is the source channel of message i and not message j 's

$$\left. \begin{array}{l} S_{jm} + (1 - a_{im}) * T - S_{im} + H(1 - y_{ijm}) \geq a_{im} t_i \\ S_{im} - (S_{jm} + (1 - a_{im}) * T) + Hy_{ijm} \geq a_{jm} t_j \end{array} \right\} \quad \text{if } i \text{ and } j \text{ are not destined to the same}$$

destination and m is the source channel of message j and not message i 's

$$S_{im} - 1 + H(1 - a_{im}) \geq CAT[m] \quad 1 \leq i \leq M, 1 \leq m \leq C$$

$$\sum_{m=1}^C S_{im} - [(C - 2) + a_{jm'}] * T - 1 \geq RAT[r] + (1 - a_{jm''}) * T - R_r \quad 1 \leq i \leq M, 1 \leq m \leq C,$$

$1 \leq r \leq N$, and m' is the source channel of the source node of message i

and m'' is the destination channel of the destination node of message i

$$S_{im} \geq T \quad \text{if } m \text{ is not the source channel of the source node of message } i$$

$$S_{im} \geq 0 \quad \text{if } m \text{ is the source channel of the source node of message } i$$

$$a_{im} = 0 \text{ or } 1$$

$$p_{ij} = 0 \text{ or } 1, \text{ and}$$

$$y_{ijm} = 0 \text{ or } 1$$

This minimization requires the solution of an ILP problem.

6.3 Problem Formulation (Loss Rate)

In this section, we develop a model to minimize the loss rate, which is called LPII. The network model considered in this section is the same with that in the last section. Except the

traffic demand D_1 addressed in LPI, another traffic demand, characterized by a $N \times N$ matrix, is denoted by D_2 . Entry d_{2ij} of D_2 represents the demand in terms of message deadline for interconnections from node i to node j . Similarly, in our approach, the determination of an optimal schedule can be formulated as a cost function to minimize the loss rate, subject to a set of linear constraints. The loss rate is defined as the ratio of number of messages dropped due to the violation of the time constraints to the total number of messages. In this chapter, the time constraint of the messages is expressed as *deadline*, which is the time constraint until the end of the service. If the delay of a message in the system exceeds its time constraint, the message is considered as lost and will be dropped. The problem is to decide the optimal sequencing of the messages to minimize the loss rate, which can be formulated as a Mixed Integer Linear Programming (MILP) problem, which includes both integer and real variables.

6.3.1 Variable Definitions

All the variables and parameters defined in the Subsection 6.2.1 are also the variables and parameters in this subsection. Besides, there are two new variables and one parameters listed below.

Parameters:

D_i = Deadline of message i in the network

Variables:

Δ_i = Difference between the laxity and starting time of message i

k_i = Status of message loss or not

6.3.2 System Constraints

In this subsection, the objective for the scheduling policies is to maximize the number of messages that are delivered by their respective deadlines. Similar to the former model LPI, the new model, namely LP II, has to take channel constraints, destination constraints and

other constraints into account. LPII and LPI are defined with the same architecture, so most of the constraints in LPI are also applicable in LPII except some special cases.

1) *Channel Constraints I*: One message can be processed on one channel only. Similar to the LPI, a *zero-one* variable a_{im} is used to specify the status of message processing, *i.e.*,

$$a_{im} = 0, \text{ if message } i \text{ is not processed on channel } m$$

$$a_{im} = 1, \text{ if message } i \text{ is processed on channel } m$$

Different from LPI, if the message violates its time constraint, it will be dropped before its transmission. That is, it will not be processed in any channel. Therefore constraint (6.1) becomes

$$\sum_{m=1}^C a_{im} \leq 1 \quad (6.12)$$

With (6.12), it is clear that if the message is not lost, $\sum_{m=1}^C a_{im} = 1$, which means that one channel will be assigned to the message. Otherwise, we have $\sum_{m=1}^C a_{im} = 0$, which means that the message will not start transmission at any channel.

The constraints to limit the value of S_{im} is also the same with those in LPI. Therefore, constraints (6.1)~(6.3) are applicable here.

Similarly, with the constraints (6.2.1) and (6.2.2), it is easy to observe that if message chooses its source channel to transmit the message, the actual starting time of message should be

$$S_i = S_{im'} = \sum_{m=1}^C S_{im} - (C-1)*T - 1$$

where m' denotes the source channel of message i .

And if message doesn't choose its source channel to transmit the message,

$$S_i = S_{ik} = \sum_{m=1}^C S_{im} - (C-2)*T - 1$$

where k denotes one arbitrary channel rather than the source channel.

Additionally, there is a special case in LPII, which is that the message will not start transmission if it is lost. In this case, no channel will be selected. So

$$S_i = 0$$

In order to combine these three cases, the actual starting time should be

$$S_i = \sum_{\forall m \& m \neq m'} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} \quad (6.13)$$

2) *Channel Constraint II*: Two messages cannot be processed on the same channel simultaneously. The constraints (6.4) and (6.5) in LPI are also applicable here.

3) *Destination Constraint*: Two messages cannot be destined to the same destination simultaneously. It is also necessary to consider the special case that the message is lost. In this case, the message will not be processed on any channel. That is, $\sum_{m=1}^C a_{im} = 0$.

Therefore, with the result in (6.13), the constraints (6.6) and (6.7) become

$$\begin{aligned} & \sum_{\forall m \& m \neq m''} (S_{jm} - (1 - a_{jm}) * T) + S_{jm'} - \sum_{\forall m} a_{jm} - \left(\sum_{\forall m \& m \neq m''} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} \right) + H(1 - p_{ij}) \\ & \geq \sum_{\forall m} a_{im} * t_i \end{aligned} \quad (6.14)$$

$$\begin{aligned} & \sum_{\forall m \& m \neq m''} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} - \left(\sum_{\forall m \& m \neq m''} (S_{jm} - (1 - a_{jm}) * T) + S_{jm'} - \sum_{\forall m} a_{jm} \right) + Hp_{ij} \\ & \geq \sum_{\forall m} a_{jm} * t_j \end{aligned} \quad (6.15)$$

where m'' denotes the source channel of message j and m' denotes the source channel of message i .

4) *Channel Availability Constraint*: No any message can start the transmission on channel m before it becomes available. The corresponding constraint (6.8) in LPI is also applicable here.

5) *Destination Availability Constraint*: No any message can arrive at destination j before its receiver becomes available. Constraint (6.9) is applicable to the messages that do not violate their time constraints. However, it is necessary to include the constraints for those lost messages in the new model. Therefore, with result (6.13), constraint (6.9) becomes

$$\sum_{\forall m \& m \neq m'} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} \geq \sum_{\forall m} a_{im} * (RAT[r] - R_r) + \left(\sum_{\forall m} a_{im} - a_{jm''} \right) * T \quad (6.16)$$

where r denotes the destination of message i , m' is the source channel that the transmitter has been tuned to and m'' is the destination channel that the receiver of the destination has been tuned to.

6) *Basic Constraints*: Constraint (6.10) is also applicable in LPII.

7) *Message Loss*: Δ_i is the value for judging whether a message is loss. There are three cases to consider. First, if the channel selected is the source channel of the message i 's source node, denoted by m' , we have

$$\Delta_i = D_i - S_i = D_i - (S_{im'} - 1 + t_i)$$

Second, if the channel selected is not the source channel of the message i 's source node,

$$\Delta_i = D_i - S_i = D_i - \left(\sum_{\forall m \& m \neq m'} (S_{im} - (1 - a_{im}) * T) - 1 + t_i \right)$$

Last, if the message is lost, no channel will be selected, then

$$\Delta_i = 0$$

Therefore, the constraint that includes these three cases is

$$\Delta_i = \sum_{\forall m} a_{im} * D_i - \left(\sum_{\forall m \& m \neq m'} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} + \sum_{\forall m} a_{im} * t_i \right)$$

With the above constraint, if the message is lost, $\Delta_i = 0$. However, it will be hard to differ it from the case that when the message deadline equals to the message delay, namely $\Delta_i = 0$, the message does not violate its time constraint. Therefore, the constraint should be changed to

$$\Delta_i = \sum_{\forall m} a_{im} * D_i - \left(\sum_{\substack{\forall m \& \\ m \neq m'}} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} + \sum_{\forall m} a_{im} * t_i \right) + \sum_{\forall m} a_{im} - 0.5 \quad (6.17)$$

Then, if the message deadline equals to the message delay, $\Delta_i = 0.5$. On the other hand, if the message violates its time constraint, $\Delta_i = -0.5$.

That is

if $\Delta_i > 0$, message i does not violate the time constraint

if $\Delta_i < 0$, message i violates the time constraint

This information relies on $\sum_{\forall m} a_{im}$, and

$$\begin{aligned} \text{if } \Delta_i > 0, \sum_{\forall m} a_{im} &= 1 \\ \text{if } \Delta_i < 0, \sum_{\forall m} a_{im} &= 0 \end{aligned}$$

In order to represent the relationship between Δ_i and $\sum_{\forall m} a_{im}$, we have the constraint that

$$X^* (\sum_{\forall m} a_{im} - 1) \leq \Delta_i \leq X^* \sum_{\forall m} a_{im} \quad (6.18)$$

where W is a big number. Also, to indicate the status of message loss, k_i is used, then

$$\begin{aligned} \text{if } \Delta_i > 0, k_i &= 1 \\ \text{if } \Delta_i < 0, k_i &= 0 \end{aligned}$$

And the constraint is

$$\sum_{\forall m} a_{im} = k_i \quad (6.19)$$

The constraints (6.17)~(6.19) enforce that if the message violating the time constraint, $k_i=0$ and otherwise, $k_i=1$.

6.3.3 Objective Function

The objectivity of the MILP model is to minimize the loss rate. This problem is also equivalent to the one that aims to maximize the number of the messages meeting the time constraint.

Hence, the MILP model to maximize k_i of all messages can be formulated as follows

MAXIMIZE

$$\sum_{i=1}^M k_i \quad (6.20)$$

and subject to a set of constraints below:

SUBJECT TO

$$\frac{S_{im} - T - A}{X} \leq a_{im} \leq S_{im} - T \quad 1 \leq i \leq M, 1 \leq m \leq C \text{ and } SC[i] \neq m$$

$$\frac{S_{im} - A}{X} \leq a_{im} \leq S_{im} \quad 1 \leq i \leq M, 1 \leq m \leq C \text{ and } SC[i] = m$$

$$\sum_{m=1}^C a_{im} \leq 1 \quad 1 \leq i \leq M, 1 \leq m \leq C$$

$$\left. \begin{aligned} & \sum_{\substack{\forall m \& \\ m \neq m'}} (S_{jm} - (1 - a_{jm}) * T) + S_{jm'} - \sum_{\forall m} a_{jm} - \left(\sum_{\substack{\forall m \& \\ m \neq m''}} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} \right) + H(1 - p_{ij}) \\ & \geq \sum_{\forall m} a_{im} * t_i \\ & \sum_{\substack{\forall m \& \\ m \neq m''}} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} - \left(\sum_{\substack{\forall m \& \\ m \neq m'}} (S_{jm} - (1 - a_{jm}) * T) + S_{jm''} - \sum_{\forall m} a_{jm} \right) + H p_{ij} \\ & \geq \sum_{\forall m} a_{jm} * t_j \end{aligned} \right\}$$

if i and j are destined to the same destination, m' is the source channel of the source node of message i and m'' is the source channel of the source node of message j

$$\left. \begin{aligned} S_{jm} - S_{im} + H(1 - y_{ijm}) &\geq a_{im} t_i \\ S_{im} - S_{jm} + H y_{ijm} &\geq a_{jm} t_j \end{aligned} \right\} \text{ if } i \text{ and } j \text{ are not destined to the same destination and } m$$

is the source channel of message i and message j or m is not the source channel of message i and message j

$$\left. \begin{aligned} S_{jm} - (S_{im} + (1 - a_{jm}) * T) + H(1 - y_{ijm}) &\geq a_{im} t_i \\ S_{im} + (1 - a_{jm}) * T - S_{jm} + H y_{ijm} &\geq a_{jm} t_j \end{aligned} \right\} \text{ if } i \text{ and } j \text{ are not destined to the same}$$

destination and m is the source channel of message i and not message j 's

$$\left. \begin{aligned} S_{jm} + (1 - a_{im}) * T - S_{im} + H(1 - y_{ijm}) &\geq a_{im} t_i \\ S_{im} - (S_{jm} + (1 - a_{im}) * T) + H y_{ijm} &\geq a_{jm} t_j \end{aligned} \right\} \text{ if } i \text{ and } j \text{ are not destined to the same}$$

destination and m is the source channel of message j and not message i 's

$$S_{im} - 1 + H(1 - a_{im}) \geq CAT[m] \quad 1 \leq i \leq M, 1 \leq m \leq C$$

$$\sum_{\substack{\forall m \& \\ m \neq m'}} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} \geq \sum_{\forall m} a_{im} * (RAT[r] - R_r) + (\sum_{\forall m} a_{im} - a_{jm''}) * T$$

$1 \leq i \leq M, 1 \leq m \leq C, 1 \leq r \leq N$, and m' is the source channel of the source node of message i and m'' is the destination channel of the destination node of message i

$$\Delta_i = \sum_{\forall m} a_{im} * D_i - (\sum_{\substack{\forall m \& \\ m \neq m'}} (S_{im} - (1 - a_{im}) * T) + S_{im'} - \sum_{\forall m} a_{im} + \sum_{\forall m} a_{im} * t_i) + \sum_{\forall m} a_{im} - 0.5$$

$1 \leq i \leq M, 1 \leq m \leq C$, and m' is the source channel of the source node of message i

$$X * (\sum_{\forall m} a_{im} - 1) \leq \Delta_i \leq X * \sum_{\forall m} a_{im} \quad 1 \leq i \leq M, 1 \leq m \leq C$$

$$\sum_{\forall m} a_{im} = k_i$$

$$S_{im} \geq T \quad \text{if } m \text{ is not the source channel of the source node of message } i$$

$$S_{im} \geq 0 \quad \text{if } m \text{ is the source channel of the source node of message } i$$

$$a_{im} = 0 \text{ or } 1$$

$$p_{ij} = 0 \text{ or } 1$$

$$y_{ijm} = 0 \text{ or } 1, \text{ and}$$

$$k_i = 0 \text{ or } 1$$

This minimization requires the solution of MILP problem.

To solve the ILP/MILP problems, generally speaking, there are two types of methods. One is due to Gomory [91] and the other is due to Benders [92]. The two methods are called cutting-plane and branch-and-bound respectively. The cutting-plane algorithm obtains the optimal solution by adding constraint to an ILP that does not exclude integer feasible points. A linear constraint that does not exclude any integer feasible points is called a cutting plane.

One such constraint is added at a time until the solution to the ILP relaxation is integer. Due to the fact that no integer feasible points are excluded, the final solution to the relaxed ILP with added constraints will solve the original ILP. The branch-and-bound method is based on the idea of intelligently enumerating all the feasible points of a combinatorial optimization problem. That is, this method tries to construct a proof that a solution is optimal, based on successive partitioning of the solution space. The branch in branch-and-bound refers to this partitioning process; the bound refers to lower bounds that are used to construct a proof of optimality without exhaustive search. In this thesis, the ILP/MILP problems will be solved by using the software CPLEX, which is a sophisticated and computationally efficient solver that can handle ILP/MILP problems by using one of these two methods. CPLEX will solve the problems by calling one of these two functions automatically. Since ILP can also be regarded as a special case of MILP, both of the models we developed above will be denoted by MILP hereafter.

6.4 Examples

Consider a small WDM star network with $N=10$ and $C=2$ and the following traffic are demand matrix D_1 in terms of message length and demand matrix D_2 in terms of message deadline. Each entry d_{1ij} in D_1 represents the length of the message from source node i to destination j , and, similarly, each entry d_{2ij} in D_2 represents the deadline of the message from source node i to destination j , where $1 \leq i \leq 10$, $1 \leq j \leq 10$ and $i \neq j$.

$$D_1 = \begin{bmatrix} 0 & 0 & 33 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 59 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 24 & 0 & 0 & 0 \\ 24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 28 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 41 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 \\ 0 & 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 73 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 74 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 104 & 0 & 0 & 0 \\ 44 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 98 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 31 & 0 & 0 \\ 0 & 0 & 91 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 103 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 80 \\ 0 & 62 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

It is clearly that there are 10 messages requesting message transmissions.

The matrix SC and DC are as follows:

$$SC = [1 \ 1 \ 2 \ 1 \ 2 \ 2 \ 1 \ 1 \ 1 \ 2]$$

$$DC = [1 \ 2 \ 2 \ 1 \ 1 \ 2 \ 1 \ 1 \ 2 \ 2]$$

Each entry in SC represents the channel that the transmitter of each node is currently tuned to and each entry in DC represents the channel that the receiver of each node is currently tuned to.

And CAT for each channel and RAT for each node is

$$CAT = [0 \ 3]$$

$$RAT = [2 \ 35 \ 4 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$$

The tuning time T is assumed to be 10 time slots. And propagation delay R is assumed to be identical for all the nodes, which is also assumed to be 10 time slots.

6.4.1 Delay

Firstly, the MILP model that aims to obtain the optimal delay, say LPI, is solved. The time constraints of the messages will not be considered in this model. So no messages will be dropped. The optimal solution is as follows.

The status of message processing on channel m , say a_{im} , can be presented as the following matrix, in which each entry a_{im} denotes the status of message i processing on channel m , $1 \leq i \leq 10$ and $1 \leq m \leq 2$.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

And the starting of each message on each channel is represented in the following matrix, in which each entry S_{im} denotes the starting time of message i on channel m .

$$\begin{bmatrix} 63 & 0 & 10 & 0 & 10 & 10 & 96 & 1 & 19 & 29 \\ 10 & 79 & 5 & 39 & 51 & 4 & 10 & 10 & 10 & 0 \end{bmatrix}$$

The actual starting time of each message is

$$[62 \quad 78 \quad 4 \quad 38 \quad 50 \quad 3 \quad 95 \quad 0 \quad 18 \quad 28]$$

And the sequencing of the messages can be shown in Figure 6.1.

Figure 6.1 Message sequence of LPI

In order to show the performance of the proposed algorithm, the scheduling result of SMIT-CC-MSL algorithm, which is shown to work best among all the combinational algorithms in Chapter 4, is presented in Figure 6.2. And the starting time of the messages using SMIT-CC-MSL algorithm is

$$[30 \quad 92 \quad 18 \quad 63 \quad 64 \quad 3 \quad 87 \quad 0 \quad 10 \quad 42]$$

Figure 6.2 Message sequence of SMIT-CC-MSL algorithm

Initially, the CAT of each channel is equal to 0 and 3 respectively. According to SMIT-CC-MSL, messages are ordered according to the length first, and the results are M_6 , M_8 , M_9 , M_{10} , M_3 , M_4 , M_5 , M_1 , M_7 , and M_2 . M_6 is the message with the least length. According to the CC-MSL algorithm which is proved to work best among the channel selection algorithms in Chapter 4, the channel 2 instead of earliest available channel, namely channel 1, will be selected since the transmitter of the source node of M_6 is currently connected to channel 2. In this way, the retuning of the transmitter can be eliminated and the impact of the delay for M_6 by the tuning overhead will be avoided. Seen from Figure 6.1, with the constraints of MILP model, the idea of CC-MSL is also included in the new algorithm, which is proved from the fact that M_6 is scheduled on channel 2. Similarly, M_8 is scheduled on channel 1 to avoid the retuning of the transmitter. As for M_9 , according to CC-MSL algorithm, channel 2 is selected. However, in this case, one idle time duration on channel 2 is induced. Although SMIT is designed to reduce the idle time of the scheduling, it mainly aims at the idle time caused by the receiver unavailability regardless of that caused by the tuning overhead. Therefore, this idle time duration cannot be avoided by SMIT algorithm. On the other hand, with LPII, M_9 will be scheduled on channel 1 so that no idle time duration is introduced. It shows that the new algorithm has considered the constraints of the system more adequately than SMIT-CC-MSL. Moreover, in SMIT-CC-MSL sequence, without the ability to avoid the receiver retuning, M_{10} has to be scheduled after M_3 and M_1 because it is destined

to a busy node ($RAT[2]=35$). However, in the LPII sequence, with the capability to avoid the retuning of the receiver, M_{10} can be scheduled right after M_3 with the starting time of 28. It shows again that the new algorithm has considered the system constraints more adequately. Further, in the sequence of LPII, the ability to reduce the transmitter retuning is also presented in the case when scheduling M_3 and M_4 . These two messages are with the same length, but the transmitters corresponding to their respective source node is not tuned to the same channel. Therefore, in order to avoid the overhead of the transmitter tuning, M_3 is first scheduled on channel 1 and no tuning overhead is induced. After that M_4 is scheduled. In our system, the delay is equal to the starting time, message length and propagation delay. Therefore, with the final results, the average delay of the messages with the LPII algorithm is $(62+33+10+78+59+10+4+24+10+38+24+10+50+28+10+3+1+10+95+41+10+0+18+10+18+20+10+28+22+10)/10=74.6$. And the average delay with SMIT-CC-MSL is $(30+33+10+92+59+10+18+24+10+63+24+10+64+28+10+3+1+10+87+41+10+0+18+10+10+20+10+42+22+10)/10=77.9$. It is proved from the fact that the new algorithm proposed achieves better performance than SMIT-CC-MSL algorithm.

6.4.2 Loss Rate

Next, the MILP model that aims to obtain the optimal loss rate, say LPII, is solved. In this case, the time constraints of the messages will be considered. So the messages will be dropped if their time constraints are violated. The major results are as follows.

The status of message processing on channel m can be presented as the following matrix, in which each entry a_{im} denotes the status of message i processing on channel m .

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

It is clear that M_2 and M_7 are lost due to the violation of the time constraints. Therefore,

$$\sum_{\forall m} a_{2m} = 0 \text{ and } \sum_{\forall m} a_{7m} = 0.$$

The starting of each message on each channel is represented in the following matrix, in which each entry S_{im} denotes the starting time of message i on channel m .

$$\begin{bmatrix} 25 & 0 & 10 & 1 & 58 & 10 & 0 & 0 & 0 & 10 \\ 10 & 10 & 5 & 10 & 0 & 4 & 10 & 71 & 51 & 29 \end{bmatrix}$$

And the actual starting time of each message in the LP11 is

$$[24 \ 0 \ 4 \ 0 \ 57 \ 3 \ 0 \ 70 \ 50 \ 28]$$

The sequencing of the messages can be shown in Figure 6.3. And the lost messages are presented as the rectangles with the slash.

Figure 6.3 Message sequence of LP11

In order to show the performance of LP11, the scheduling result of DDS-CC-MSL algorithm, which combines DDS algorithm in Chapter 5 and CC-MSL algorithm in Chapter 4, is presented in Figure 6.4. The starting time of the messages using DDS-CC-MSL algorithm is

$$[4 \ 0 \ 62 \ 0 \ 37 \ 3 \ 0 \ 44 \ 24 \ 0]$$

Similarly, the lost messages are presented as the rectangles with the slash.

Figure 6.4 Message sequence of DDS-CC-MSL algorithm

Initially, the CAT of each channel is equal to 0 and 3 respectively. As Figure 6.3 shows, although RAT of the destination of M_{10} is equal to 35, it still can be scheduled on channel 2 with the starting time 28. It is because that the corresponding receiver is currently connected to channel 2 and no retuning of receiver is needed. Moreover, M_4 , M_6 , and M_3 can start the transmission with the starting time smaller than the tuning time, which shows that these messages are assigned to the source channels of the corresponding source nodes. From this case, it is reinforced that the new algorithm has the ability to reduce the impairment caused by the retuning of transmitters or receivers. Moreover, no idle time durations caused by receiver unavailability or tuning overhead along the data channels are induced. The delay matrix of LPII is

$$[67 \ 0 \ 38 \ 34 \ 95 \ 14 \ 0 \ 98 \ 80 \ 60]$$

As figure shows, M_2 and M_7 are lost, so the status of message loss, say k_i , is

$$[1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1]$$

Hence, the loss rate is $2/10=0.2$.

On the other hand, by using DDS-CC-MSL algorithm, the messages will be ordered according to EDD first so that the message with stringent time constraint will be ordered first. That is, the initial message sequence is M_6 , M_4 , M_{10} , M_1 , M_2 , M_9 , M_7 , M_5 , M_8 , and M_3 . According to CC-MSL algorithm, the channel 2 will be selected to transmit M_6 since the

transmitter of the source node of M_6 is currently tuned to channel 2. In this way, the retuning of the transmitter can be avoided. Similarly, M_4 is scheduled on channel 1 and no tuning overhead of the transmitter will be induced. However, as for M_{10} , it is destined to a busy node ($RAT[2]=35$). Therefore, its starting time will be equal to the receiver available time. And its delay is $35+22+10=67$. It is clearly that its delay exceeds its time constraint. As such, M_{10} will be dropped. This is because that CC-MSL algorithm cannot avoid the negative effect of the receiver as LP11 does. With the ability to avoid the retuning of the receiver, message M_{10} is successful scheduled by using LP11, which is shown in Figure 6.3. The final result of DDS-CC-MSL algorithm is shown in Figure 6.4. The delay matrix of DDS-CC-MSL is

$$[47 \ 0 \ 96 \ 34 \ 75 \ 14 \ 0 \ 72 \ 54 \ 0]$$

And as the figure shows, except M_2 and M_7 , M_{10} is lost, too. So the status of message loss, say k_i , is

$$[1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$$

Hence, the loss rate is $3/10=0.3$.

It is evident from the result that CC-MSL algorithm just considers avoiding the retuning of the transmitters, regardless of the negative impact of the receivers. On the other hand, LP11 has considered the system constraints adequately so that the negative impact caused by the retuning of transmitters or receivers can be avoided whenever possible. Therefore, LP11 achieves better performance.

6.5 Experimental Emulation and Results

6.5.1 Experimental Design

The MILP problem (i.e., with 0-1 variables) and combinatorial optimization problems are challenging from a computational standpoint: they are NP-complete [90]. And it is widely believed that no Polynomial algorithm exists for solving them. As a consequence, their

solution requires considerable time and expertise in both the application domain (modeling) and algorithm design (solving). Therefore, it is very difficult for our algorithm to be implemented on-line with the capability of today's computer, especially when the number of nodes in the network is very large. Therefore, it can only be used on off-line scheduling. In order to emulate the activity of the network, we have designed a virtual on-line experiment system to evaluate the performance of the proposed scheme.

Firstly, we design a MILP emulator, which is shown in Figure 6.5. *Traffic Generator* generates the traffic according to some distribution. In the real network, the scheduling algorithms will be invoked to schedule the messages upon receiving the whole control frame which contains a batch of the messages. And the time for one control frame to be received by all the nodes is its transfer time. Therefore, the time interval for the scheduling algorithm to be invoked is also equal to the transfer time of one control frame. In order to emulate the real network, the traffic generated from *Traffic Generator* during the transfer time of a control frame will be stored in *Database I* as a batch. That is, the messages stored in *Database I* will be on batch-by-batch basis. Meanwhile, the properties associated with each message, i.e., message destination node, message length and message deadline will be also stored. After that, each batch of the messages generated is imported to *MILP* as the traffic demand. The required input for the *MILP* includes the number of messages and the properties of each message. Besides this, global information, including CAT and RAT, is also a part of the input data. The scheduler LPI in subsection 6.2 or the scheduler LPII in subsection 6.3 is then called to sequence of the messages based on the objective of the system. There are two stages for LPI/LPII to obtain the optimal sequence of the messages. One is to formulate the MILP models provided with data input and the global information, and the other is to solve the MILP problems by calling CPLEX optimizer. CPLEX is a sophisticated and computationally efficient solver that can handle linear programming problems with hundreds of thousands of variables and mixed integer linear programming problems with tens of thousands of variables. With these two stages, the optimal sequence of the input messages

can be obtained by the scheduler LPI/LPII. Once the solution is found, results are transferred to *Database II*, in which every optimal result is corresponding to every batch of the messages in *Database I*. Meanwhile, Global information is updated based on the results.

Figure 6.5 MILP emulator

The purpose of the emulator is to use MILP processor to get an optimal schedule result off-line. Then the off-line results can be used directly in the virtual on-line emulation system. Based on the emulator, one virtual on-line emulation system can be set up to emulate the activity of the network, which is shown in Figure 6.6. As addressed above, *Database I* stores the traffic generated from *Traffic Generator*. And *Database II* stores the optimal results from LPI/LPII emulator. Then these two databases can be used as the input of the network and the scheduler. The traffic is generated from the *Database I*, and the data will be retrieved on batch-by-batch basis every transfer time of control frame. As addressed above, each result stored in *Database II* is corresponding to each batch of the messages in *Database I*, and, therefore, *Scheduler* just simply retrieves the sequence results from *Database II* based on the input traffic and then imports the results to the *Performance Analysis*. This virtual on-line system can be used to emulate the activity when LPI/LPII emulator performs as an on-line scheduler.

Figure 6.6 Virtual on-line emulation system

Additionally, in order to show the performance of the schedulers LPI and LPII, the comparisons between them and SMIT-CC-MSL and DDS-CC-MSL are conducted respectively by extensive simulation experiments. And the experimental structure for SMIT-CC-MSL and DDS-CC-MSL algorithm is shown in Figure 6.7. In order to obtain the same traffic with the new algorithms, *Database I* remains the traffic generator. After every transfer time of a control frame, one batch of the messages stored in *Database I* will be retrieved as the input of the network. Different from Figure 6.6, *Scheduler* schedules the input messages on-line by using SMIT-CC-MSL or DDS-CC-MSL algorithm respectively and then the scheduling results are imported to the *Performance Analysis*.

Figure 6.7 Experimental system for SMIT-CC-MSL and DDS-CC-MSL

The parameters of the experiments are defined as follows. In this set of experiments, the number of nodes (N) is set to 25 and the number of channels (C) is set to 2. Round-trip propagation delay R is assumed to be identical for all the nodes, which equals to 10 time slots. And tuning time (T) is also assumed to be 10 time slots. Message lengths vary according to an Exponential distribution with a mean value as 20 time slots. Message arrivals at each source node comply with an independent Poisson process. And the destination nodes are selected according to a uniform probability distribution.

For the real-time messages, the message time constraint is expressed as message deadline. In order to avoid the case that the deadline of one message is smaller than its length or the propagation delay, which means that the message violates its deadline when it is generated, we will define the message laxity first. Message laxity can be expressed through the following relationship:

$$\text{message deadline} = \text{message laxity} + \text{message length} + \text{propagation delay}$$

In this section, we assume that message laxity is a random variable following an Exponential distribution with a mean value 80 time slots. So, the deadline of each message can be obtained by adding the laxity, the length and the propagation delay. Moreover, we assume that the deadline of each batch of the messages is calculated from the time that this batch of the messages starts to be scheduled.

6.5.2 Experimental Results

Figure 6.8 Impact of traffic load on average delay of LPI

The performances of the LPI and SMIT-CC-MSL are plotted in Figure 6.8 as curves of the average message delay versus the traffic load ρ . It is clearly that the delay increases for the two algorithms when the traffic load increases. In a low traffic load, the difference between the two algorithms is not significant. However, as the traffic load increases, the algorithm

developed based on MILP algorithm outperforms SMIT-CC-MSL. This is expected because LPI is able to include the system constraints in the mathematical formulations, and the optimal performance in terms of message delay can be obtained under such constraints and an objective function. Furthermore, LPI has considered reducing the idle time caused by the tuning overhead as well as avoiding the retuning of the receivers whenever possible, therefore resulting in better performance. From this comparison, we can also arrive at a conclusion that the consideration of special characteristics of optical network and node equipments in SMIT-CC-MSL is still not adequately enough thought it works best among the algorithms addressed in Chapter 4.

Figure 6.9 Impact of tuning time on average delay of LPI

Figure 6.9 shows the effect of varying tuning time on message delay for LPI and SMIT-CC-MSL. The delays of both algorithms increase linearly with the tuning time. This is because when the tuning time is increased, it is equivalent to increasing the offered load since it takes longer time to complete a message transmission. From the figure, we observe that LPI always results in the lower message delay when the tuning time is larger than 0. And the difference between the message delays becomes more significant when tuning time increases. This is expected since LPI is able to obtain the optimal sequence of the messages

so that the average delay of the messages can be minimized. Moreover, the salient feature of the new algorithms is that they have the ability to reduce the idle time caused by tuning time as well as the ability to reduce or eliminate the tuning overhead cause by the retuning of the transceivers. As a consequence, LPII achieves better performance than SMIT-CC-MSL, especially when the tuning time becomes large.

Figure 6.10 Impact of traffic load on loss rate of LPII

In Figure 6.10, our study focuses on the loss rate versus the traffic load ρ . In order to show the performance of LPII, it is compared with DDS algorithm in Chapter 5 which is mainly developed to reduce the loss rate of real-time traffic. In addition, DDS will be combined with CC-MSL algorithm in Chapter 4, which is shown to work best among the channel assignment algorithms proposed. We see that the LPII algorithm works better in the loss rate. This is expected since LPII is able to obtain an optimal result by solving MILP model with mathematical constraints and an objective function. Furthermore, LPII has included all the constraints of the optical networks to the mathematical formulations so that the negative impact caused by destination and tuning time is reduced to the maximum extent. On the other hand, although DDS has ordered the messages by taking the message length and laxity into account, it cannot reduce the idle time caused by the destination or tuning time. Moreover, although CC-MSL is able to reduce the impairment of the performance

caused by the retuning of the transmitter, it is incapable of decreasing the negative impact of the retuning of the receivers. Therefore, DDS-CC-MSL algorithm works worse in terms of loss rate.

Figure 6.11 Impact of tuning time on loss rate of LPII

Figure 6.11 depicts the loss rate versus the tuning time. It is clearly that the loss rate increases with the tuning time for both algorithms. However, between the two algorithms, LPII consistently demonstrates its superior performance to DDS-CC-MSL across the whole tuning time spectrum. This is because LPII has the ability to include the system constraints of WDM optical networks to mathematical formulations so that the optimal loss rate can be obtained. On the other hand, DDS-CC-MSL cannot avoid the idle time caused by receiver unavailability or tuning time unavailability, and, moreover, it cannot reduce the tuning overhead caused by the retuning of the receivers. As a result, DDS-CC-MSL works worse than LPII.

6.6 Summary

In this chapter, two MILP based off-line scheduling algorithms to schedule variable-length message transmission in the specified WDM optical networks are proposed. The purposes of

these two off-line scheduling algorithms are to obtain the optimal performance in terms of average message delay or loss rate. We have defined the two scheduling problems as MILP models that have included the system constraints of single-hop passive-star coupled WDM optical networks, i.e., channel constraints, receiver constraints and tuning overhead of the transceivers, in mathematical formulations. With the constraints and objective function in the form of linear formulations, these two models can work out optimal solutions. To evaluate the performance of the new algorithms, one virtual on-line system model has been set up and extensive experiments have been conducted by comparing the new algorithms with those with similar functions. The results show that with the ability to avoid the retuning of the transmitter and receivers as well as the ability to avoid channel collisions and receiver collisions, the new algorithms achieve better performance in terms of average message delay or loss rate.

Chapter 7 . Conclusion and Recommendations

7.1 Conclusion

Increasingly bandwidth-demand applications and services justify the need for high-speed photonic LAN/MAN. In order to make sure that WDM local/metro networks can be considered as viable promising alternatives to current LAN architectures such as Gigabit Ethernet and FDDI, several requirements have to be fulfilled. First, they have to be cost-effective. In this context, low-cost tunable transmitters and receivers are of paramount concern. Second, they should provide scalability with respect to the number of channels and nodes. Third, they have to support variable QoS requirements. So far, several experimental prototypes have been built based on passive-star topology in several research laboratories [6][7][8][9]. Among these prototypes, only simple access protocols have been used. However, in future systems, in order to achieve much better performance, more sophisticated and efficient protocols may be required. The schemes proposed in this dissertation could be viable candidates for industry in the future.

Among the existent prototypes, STARNET [9] is a broadband backbone WDM network, which is built by Prof. kazovsky of Stanford University. It provides two logical subnetworks to serve these applications in the campus area: a high-speed reconfigurable packet-switched data subnetwork and a moderate-speed fix-tuned packet-switched control subnetwork. While what we study in this dissertation is a WDM LAN, which can also be configured as an

access network. In order to achieve better performance for end users, our WDM LAN can be connected to STARNET by using the internetworking devices, such as gateways. Figure 7.1 shows the generic access network architecture. As the figure shows, the WDM LAN we study is possible to be an access network which is responsible for reaching the customer premises. By interconnecting STARNET and the WDM LAN, the increasing demand for the Internet traffic with a variety of types can be fulfilled.

Figure 7.1. Generic access network architecture

However, the WDM LAN is not the preferred choice of access network architecture by industry nowadays. The most popular architecture is Ethernet Passive Optical Network (EPON), which is more suitable than WDM LAN to be an access network because of the following three main aspects. Firstly, EPON deploys the point-to-multipoint architecture. This type of architecture is the best choice for access network due to the fact that it reduces the fiber deployment as well as the amount of optical transceivers. And secondly, EPON performs central bandwidth allocation algorithm in optical line terminal (OLT) rather than uses the distributed manner as WDM LAN performs, which makes the control of the communication between the customers to the backbone easier to implement. Last but not

least, WDM LAN needs many fixed transceivers or tunable transceivers which cause more money.

This dissertation dedicates to deal with the design and analysis of medium access control protocols for optical packet-switched WDM-based LANs/MANs built on the passive-star topology. The individual chapters of this work addressed the main issues of such networks listed above and their specific contributions can be summarized as follows.

In Chapter 4, three channel assignment algorithms and two message sequencing algorithms employed by reservation-based access protocols for passive-star coupled WDM systems relying on the CC-FT/TT-FR/TR architecture were introduced. The channel assignment algorithms address the problem of selecting an appropriate channel and a time slot on that channel to transmit a message. And, on the other hand, message sequence algorithms address the order of the messages in the network. LMS, CCS and CC-MSL are three novel channel assignment algorithms proposed. LMS is introduced to reduce the scheduling latency by taking the special characteristic of optical networks adequately into account. CCS is designed to reduce the penalty of tuning overhead by assigning the selected message to the channel that the transmitter of the source node has been tuned to whenever possible. And CC-MSL aims to reduce the scheduling latency as well as the penalty of tuning overhead. A simple comparison among these three algorithms is conducted. The result depicts that CCS works better than LMS when the traffic load is light due to its ability to avoid the retuning of the transmitters. However, when the traffic load is heavy, LMS outperforms CCS since scheduling latency becomes the major obstacle of the scheduling in this case. By combining the advantage of LMS and CCS, CC-MSL obtains the best performance in the comparison. On the other hand, MFTS and SMIT are two message sequencing techniques which aim to reduce the message delay from different points of view. MFTS algorithm sequences the messages based on the flow time of the messages, which consists of message waiting time and message transmission time. And SMIT tries to reduce or eliminate the unused time durations along the data channels. Both of these two algorithms

have taken consideration the delay caused by destination unavailability. However, MFTS cannot reduce the wasteful time slots along the data channels to the maximum extent. SMIT, on the other hand, considers the characteristic of the WDM optical networks more adequately so that it can achieve better performance. The main conclusion can be drawn from the comparison between these two algorithms is that the most efficient way to reduce the average message delay is to reduce or eliminate the additional delay caused by the destination unavailability. With the ability to reduce the wasteful time durations caused by destination unavailability, SMIT becomes the powerful and viable choice in MAC protocol design for passive star-coupled WDM networks. In addition, a simple comparison among different combinational algorithms is conducted. SMIT-CC-MSL, which combines the message sequencing algorithm SMIT and the channel assignment algorithm CC-MSL, obtains the best performance in the comparison.

In Chapter 5, two novel scheduling algorithms for providing quality of service (QoS) for single-hop passive-star coupled WDM optical networks were proposed and their performance was analyzed. DDS is an algorithm developed to handle real-time traffic. With the ability to consider the destination and channel availability of a WDM optical network, it can efficiently avoid blocking of longer messages in order to make more messages meet their deadlines resulting in less message loss rate, packet loss rate and throughput. CBPS is an algorithm to handle the traffic with and without time constraints. This algorithm regards message length, message laxity and static priority as the important factors on the costs of messages incurred in the network when considering the order of the messages. Therefore, both types of messages can be benefited when an integrated traffic is applied to the network. Furthermore, each factor on the cost of the messages is associated with one parameter. Therefore, CBPS was shown to be adaptable by the fact that adjusting the weight of these three factors through changing corresponding parameters will lead to different network performance. Based on this observation, one novel CBPS based QoS prediction structure was proposed. Given the QoS requirements in the form of delay or loss rate, it was shown

that it is possible to predict whether one combination of the parameters in CBPS can be found so that the QoS requirements can be fulfilled. With the QoS prediction, one novel framework for QoS service prediction in WDM optical networks is developed, which consists of four modules. *QoS Requirements Monitoring* detects the changes in the supplied QoS. *QoS Prediction* predicts whether the new QoS can be adapted to the current QoS. And *QoS Violation* detects violation. If the violation is detected, a module of *QoS Estimation* is used to make the decision whether the hopeless new traffic is refused or accepted with acceptable QoS requirements. It was shown that this scheme has the ability to perform at real-time, which provides a flexible mean to control network behavior as the surrounding environment changes.

Finally, in Chapter 6, we put forward two novel MILP based off-line scheduling algorithm for scheduling variable-length message transmission in single passive star-coupled WDM optical networks. In these two MILP models, the system constraints of the specific network, for example channel and receiver constraints, are represented by mathematical constraints. In this way, given a traffic request matrix R , whose elements represent the length or the deadline of the messages that must be transmitted from any source node s to any destination node d , a message sequence that guarantees the delivery of the requested traffic while minimizing the average delay or the loss rate for all the transmissions can be found subject to mathematical constraints. To evaluate the performance of the new algorithms, extensive experiments are conducted, which shows that with the ability to avoid the retuning of the transmitter and receivers as well as the ability to avoid channel collisions and receiver collisions, the new algorithms perform quite well in terms of average message delay or loss rate.

7.2 Recommendations for Future Research

7.2.1 Further Improvement on Current Results

As part of the future work, the design of efficient scheduling algorithms employed by MAC protocols for scheduling variable-length messages in single-hop passive-star coupled WDM optical networks is still a valuable direction. Although many protocols addressed channel assignment issue or message sequencing issue have been proposed in Chapter 4, the consideration of the special characteristic of WDM optical networks is not adequately enough. As Chapter 4 presents, SMIT is the algorithm with the best performance among the sequencing algorithms. However, it hasn't taken the idle time caused by the tuning overhead of the transceivers into account. On the other hand, CC-MSL is the algorithm that works best among the channel assignment algorithms, but it cannot avoid the retuning of the receivers. Therefore, more effort is still needed on how to develop new scheduling algorithms to reduce the negative impact of the system constraints on the performance of the messages.

Moreover, although two scheduling algorithms with the aim to obtain the optimal results are proposed in Chapter 6, these two algorithms cannot perform on-line due to the limitation of the computational capability nowadays. Therefore, heuristic approaches that have the ability to obtain the results that are close to the results of the optimal schedulers are in great need. These approaches are expected to be more efficient and easily utilized. In addition, these two algorithms are developed for only one kind of the traffic respectively. However, different kinds of traffic may have different performance requirements, i.e., some traffic requires low delay whereas other traffic requires low loss rate. Therefore, it is necessary to develop new MILP based models with multiple objectives so that two kinds of requirements (delay and loss rate) can be minimized simultaneously. Furthermore, the work can be extended to deal with optimization of a wide variety of schedules for any number of wavelengths and any transceivers tunability characteristics.

7.2.2 Potential Research Directions

1. The first potential research direction is to find an optimal scheduling policy for integrated traffic by using Markov Decision Processes (MDP). In contrast to the traditional networks, which are dedicated to a single application, today's networks are designed to integrate heterogeneous traffic types (voice, video, data) into one single network. As a result, optimal service scheduling policies that models many scenarios in telecommunications, such as access control to limited amount of communication channels, dynamic priority assignment between different traffic types and dynamic bandwidth allocation in the networks, are in great need. As an example, consider the problem where different kinds of traffic compete for the access to limited number of channels in the specified WDM optical network. One may wish to minimize the expected delay of non real-time traffic, but yet to impose bounds on the average delay or loss rate to the real-time traffic. To plan a "good" control policy for a given goal, MDP is a useful framework. MDP, also referred to as stochastic dynamic programs or stochastic control problems, is a model for sequential decision making when outcomes are uncertain. It consists of decision epochs, states, actions, rewards, and transition probabilities. Choosing an action in a state will generate a reward and determine the state at the next decision epoch through a transition probability function. A decision maker can influence the state by a suitable choice of some of the system's variable. Moreover, the actions applied to the system have a long-term consequence. That is, decisions made at the current epoch have an impact on decisions at the next epoch and so forth. Hence, good decision rules are needed to specify which actions should be chosen at any given epoch and state. Each control policy defines the stochastic process and values of objective functions associated with this process. And decision makers seek policies which are optimal in some sense. An analysis of such a model includes 1) providing conditions under which there exist easily implementable optimal policies; 2) determining how to recognize these policies; 3) developing and enhancing algorithms for computing them; and 4) establishing convergence of these

algorithms. With a MDP model, the optimal scheduling policy for integrated traffic in WDM optical networks can be obtained.

Furthermore, the system under study is distributed in nature where each separated node has to make its own decisions. In our system, although nodes do have the ability to communicate with each other, it is usually unrealistic for the nodes to communicate their local state information to all nodes at all times, because communication actions are usually associated with a certain cost. In our further research, we aim to propose an approach, in which each node can estimate the “local potential”. Thus the optimal policy for each node can balance the amount of communication such that the information is sufficient for proper coordination but the cost for communication does not outweigh the expected gain. The future research can focus on fully cooperative systems, where all nodes share the same goal of maximizing the total expected reward. Since nodes are distributed, a local MDP can be used to describe each agent’s state space and action space. To reflect the cooperative nature of the system, a global reward function is used to describe the relationship and dependency of the individual node’s states.

2. The second potential research direction is to serve multimedia applications. Recently, there is a rapid growth in the number of multimedia applications. And different multimedia applications require various classes of transmission service including the transmission of data, audio, and various types of video and image on the WDM optical networks. High-speed protocols including the protocols at the medium access control layer are needed to cater for the different requirements of the transmission of various multimedia applications.

Multimedia application contains a variety of media: data, graphics, images, audio and video. The transmission of the multimedia application is a kind of real-time and stream oriented communication. To characterize different multimedia applications, there are several criteria as follows. The first one is volume, which is a common denominator for defining different traffic rates such as Constant Bit Rate (CBR) and Variable Bit Rate (VBR). The

second is synchronization, which is relative time dimension to define time interval between the arrival and playout of synchronized objects. There are three kinds of synchronization, which are intra-media (isochronous) synchronization, inter-media (synchronous) synchronization and playout (asynchronous) synchronization. The third one is persistency. Based on this criterion, traffic can be divided into non-persistent live data (continuous stream) and persistent stored data (discrete media). Errors in non-persistent data are more perceivable than errors in persistent data, due to the temporal masking effect of moving objects as perceived by the human visual system. And the last one is QoS requirement, which differs from one media type to another. The quality of service required of a stream communication includes guaranteed bandwidth (throughput), delay and delay variation (jitter). And the quality of service of different kinds of media varies. On one hand, hard real-time traffic like voice and video requires stringent time delay and delay variance, but tolerates a small percentage of packet loss. On the other hand, soft or non real-time traffic like images graphics, text, and data requires no packet loss, but tolerates time delay. Therefore, to support different types of multimedia applications, different characteristics for them should be taken into account adequately.

To support multimedia application by MAC protocols has, currently, become a hot topic in the field of research on the WDM optical network. Many researchers have shown their interests on this issue. However, most of the protocols proposed are designed for certain type of traffic streams and few of them can serve a wide range of traffic streams typical in multimedia applications. As a result, to design a MAC protocol so as to efficiently serve a wide range of traffic streams typical in multimedia applications is a challenging research direction. The new protocols should consider various characteristics of different types of media and ensure the variety of QoS requirements of them.

3. The third potential research direction is to support multicasting. In the future high-speed networks, it is expected that a significant portion of the traffic will be multicast in nature. As

such, supporting multipoint traffic in WDM-based networks has received surging interest in the last few years, and many proposals were introduced to address this problem.

There are four challenges in the designing of MAC protocols to support multicasting over PSC-based WDM optical networks.

1) The first challenge is the high transmission rates of WDM networks. The very high transmission rate employed in optical networks will cause the high ratio of propagation delay to the packet transmission time. Therefore, new protocols should be able to achieve full coordination between the nodes before the transmission of the multicasting packets, which guarantees the success of the packet transmissions. In order to achieve this goal, control information will be exchanged first during pre-transmission coordination phase. Normally there are two ways for exchanging control information: out-of-band and in-band. In out-of-band control signaling, control and data flows use separate channels, while in in-band control signaling, control and data flows use the same wavelength.

2) The second challenge is the nature of WDM optical networks. In most of the systems that support multicasting, the node will be equipped with more than one tunable transmitters or receivers. Therefore, similar to the MAC protocols for unicast traffic, channel collision as well as receiver collision should be avoided. That is, no more than one message is permitted to transmit on one channel, and the number of the messages arriving to one node is not permitted to exceed the number of receivers at that node. In addition, multicasting is defined as point-to-multipoint, i.e., the information has to be transmitted from a source to multiple receivers. The set of nodes to which the multicast packet is directed is known as the multicast group. Therefore, when one multicasting packet arrives at the nodes in the multicast group, it has to be made sure that, for every node in the multicasting group, at least one receiver of it is ready to receive it. On the other hand, the negative impact of the transceivers must be taken into account when designing MAC protocols due to the limitation of the tuning time and the tuning range of the transceivers nowadays.

3) The third challenge is the dynamic feature of multicast traffic. Firstly, the membership of nodes in multicast group can change during session lifetime. Secondly, most of the services that require multicasting are dynamic in nature. As such, MAC protocols should be adaptive so that they are able to reflect the dynamic changes in traffic demands and achieve balance between the requirements of the different traffic types.

4) And the last challenge is to accommodate of different classes of service. This issue is becoming more important, especially with the demand to support differentiated service in today's network. The normal method to achieve this goal is to prioritize the station access to the medium, with which the multicast request from a station with higher priority may be considered first than that from a station with lower priority. Moreover, MAC protocols to benefit the messages with higher or lower priority simultaneously are in great need today.

Author's Publications

1. Papers (Accepted):

Journal Papers:

- [1] Maode Ma and Xiaohong Huang, "Adaptive Scheduling for Integrated Traffic on WDM Optical Networks", *Computer Networks*, vol. 44, no. 6, April 2004, pp. 773-788.
- [2] Xiaohong Huang and Maode Ma, "Efficient Message Scheduling for WDM Optical Networks with Minimizing Flow Time", *Journal of Communication and Networks*, vol. 6, no.2, June 2004, pp.147-155.
- [3] Xiaohong Huang and Maode Ma, "Message Scheduling in WDM Optical Networks with Reduced Transmitter Tuning Overhead", *IEEE Communication Letters*, vol.9, no. 1, January 2005, pp. 72-74.

Conference Papers:

- [4] Xiaohong Huang and Maode Ma, "Efficient Message Scheduling for WDM Optical Networks with Minimizing Flow Time", *Proceedings of IEEE International Conference on Software, Telecommunications and Computer Networks (SoftCOM'03)*, Split, Croatia, October 2003, pp. 636-640.
- [5] Xiaohong Huang and Maode Ma, "Message Scheduling in WDM Optical Networks with Reduced Transmitter Tuning Overhead", *Proceedings of IEEE International Conference on Communications (ICC'04)*, Paris, June 2004.
- [6] Xiaohong Huang and Maode Ma, "An Efficient Scheduling Algorithm for Real-Time Traffic on WDM Optical Networks", *Proceeding of IEEE International Conference on Communication Systems (ICCS'04)*, Singapore, September 2004, pp. 366-370.
- [7] Xiaohong Huang and Maode Ma, "Efficient Scheduling Algorithm with Reduced Message Delay for WDM Optical Networks", accepted by *IEEE International Conference on Networks (ICON'04)*, Singapore, November 2004.

2. Papers (Under Review)

Journal Papers:

- [8] Xiaohong Huang and Maode Ma, “An Efficient Scheduling Algorithm for Real-Time Traffic on WDM Optical Networks”, Submitted to *IEEE/OSA Journal of Lightwave Technology*.
- [9] Xiaohong Huang and Maode Ma, “Efficient Scheduling with Reduced Message Delay for Passive Star-Coupled WDM Optical Networks”, Submitted to *IEE Proceedings Communications*.

3. Papers (Under Submission)

- [10] Xiaohong Huang and Maode Ma, “QoS Prediction Scheme in Single-Hop Passive Star Coupled WDM Optical Networks”.
- [11] Xiaohong Huang and Maode Ma, “Two Mixed Integer Linear Programming Based Schemes for Passive Star-Coupled WDM Optical Networks.”

Bibliography

- [1] I. P. Kaminow, "Photonic Multiple-Access Networks: Topologies," *AT&T Technical Journal*, vol. 68, no. 2, pp. 61-71, March 1989.
- [2] P. S. Henry, "High-Capacity Lightwave Local Area Networks," *IEEE Communications Magazine*, pp. 20-26, October 1989.
- [3] Harry J. R. Dutton, *Understanding Optical Communications*, Prentice Hall PTR, 1999.
- [4] B. Mukherjee, "WDM-based Local Lightwave Networks- Part I: Single-Hop Systems," *IEEE Network*, May 1992, pp. 12-27.
- [5] B. Mukherjee, "WDM-based Local Lightwave Networks- Part II: Multi-Hop Systems," *IEEE Network*, July 1992, pp. 20-32.
- [6] M. S. Goodman, H. Kobrinski, M. P. Vecchi, R. M. Bulley, and J. L. Gimlett, "The LAMBDANET Multiwavelength Network: Architecture, Applications, and Demonstrations," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 6, Aug. 1990, pp. 995-1004.
- [7] F. J. Janniello, R. Ramaswami, and D. G. Steinberg, "A Prototype Circuit-Switched Multi-Wavelength Optical Metropolitan-Area Network", *IEEE/OSA Journal of Lightwave Technology*, vol. 11, no. 5/6, May/June 1993, pp. 777-782.
- [8] E. Hall, J. Kravitz, R. Ramaswami, M. Halvorson, S. Tenbrick, and R. Thosmen, "The Rainbow-II Gigabit Optical Network," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, June 1996, pp. 814-823.
- [9] T. K. Chiang, S. K. Agrawal, D. T. Mayweather, D. Sadot, C. Barry, M. Hickey and L. G. Kazovsky, "Implementation of STARNET: A WDM Computer Communications Network," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, June 1996, pp. 824-39.

- [10] I. Chlamtac, and A. Ganz, "Channel Allocation Protocols in Frequency-Time Controlled High Speed Networks," *IEEE Transactions on Communications*, vol.36, no.4, April 1988, pp.430-440.
- [11] G. N. Rouskas and M. H. Ammar, "Analysis and Optimization of Transmission Schedules for Single-Hop WDM Networks", *IEEE/ACM Transactions on Networking*, vol.3, no.2, April 1995, pp.211-221.
- [12] A. Ganz and Y. Gao, "Time-Wavelength Assignment Algorithms for High Performance WDM Star Based Systems", *IEEE Transactions on Communications*, vol.42, no.(2-3-4), February/March/April 1994, pp.1827-1836.
- [13] G. R. Pieris and G. H. Sasaki, "Scheduling Transmissions in WDM Broadcast-and-Select Networks," *IEEE/ACM Transactions on Networking*, vol.2, no.2, April 1994, pp. 105-110.
- [14] M.S. 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. 6, June 1996, pp. 923-934.
- [15] M. Azizoglu, R. A. Barry, and A. Mokhtar, "Impact of Tuning Delay on the Performance of Bandwidth-Limited Optical Broadcast Networks with Uniform Traffic," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, June 1996, pp. 935-944.
- [16] P. W. Dowd, "Random Access Protocols for High Speed Interprocessor Communication Based on An Optical Passive Star Topology", *IEEE/OSA Journal of Lightwave Technology*, vol.9, no.6, June 1991, pp.799-808.
- [17] A. Ganz and Z. Koren, "Performance and design evaluation of WDM stars", *IEEE/OSA Journal of Lightwave Technology*, vol. 11, no. 2, February 1993, pp.358-366.
- [18] I. M. I. Habbab, M. Kavehrad, and C.-E. W. Sundberg, "Protocols for Very High Speed Optical Fiber Local Area Networks Using A Passive Star Topology", *IEEE/OSA Journal of Lightwave Technology*, vol.5, no.12, December 1987, pp.1782-1794.

- [19] N. Mehravari, "Performance and Protocol Improvements for Very High-Speed Optical Fiber Local Area Networks Using a Passive Star Topology", *IEEE/OSA Journal of Lightwave Technology*, vol.8, no.4, April 1990, pp.520-530.
- [20] G. N. M. Sudhakar, M. Kavehrad, and N. Georganas, "Slotted ALOHA and Reservation ALOHA Protocols for Very High-Speed Optical Fiber Local Area Networks Using Passive Star Topology," *IEEE/OSA Journal of Lightwave Technology*, vol. 9, no. 10, October 1991, pp. 1411-1422.
- [21] G. N. M. Sudhakar, N. Georganas, and M. Kavehrad, "Multi-Control Channel Very High-Speed Optical Fiber Local Area Networks and Their Interconnections Using a Passive Star Topology," *Proceedings of IEEE GLOBECOM'91*, Phoenix, AZ, December 1991, pp. 624-628.
- [22] P. A. Humblet, R. Ramaswami, and K. N. Sivarajan, "An Efficient Communication Protocols for High-Speed Packet-Switched Multichannel Networks," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 4, May 1993, pp. 568-578.
- [23] F. Jia and B. Mukherjee, "The Receiver Collision Avoidance (RCA) Protocol for A Single-Hop Lightwave Network", *IEEE/OSA Journal of Lightwave Technology*, vol.11, no.5-6, May/June 1993, pp.1052-1065.
- [24] M.-S. Chen, N. R. Dono, and R. Ramaswami, "A Media-Access Protocol for Packet-Switched Wavelength Division Multiaccess Metropolitan Area Networks", *IEEE Journal on Selected Areas in Communications*, vol.8, no.6, August 1990, pp.1048-1057.
- [25] I. Chlamtac and A. Fumagalli, "Quadro-Star: A High Performance Optical WDM Star Network", *IEEE Transaction on Communications*, vol. 42, no. 8, August 1994, pp.2582-2591.
- [26] M. Chen and T.-S. Yum, "A Conflict-Free Protocol for Optical WDM Networks", *Proceedings of IEEE GLOBECOM'91*, December 1991, pp.1276-1291.

- [27] K.M. Sivalingam and Jie Wang, "Media Access Protocols for WDM Networks with On-Line Scheduling", *IEEE/OSA Journal of Lightwave Technology*, vol.14, no.6, June 1996, pp.1278-1286.
- [28] R. Chipalkatti, Z. Zhang, and A. S. Acampora, "Protocols for Optical Star-Coupler Network Using WDM: Performance and Complexity Study," *IEEE Journal on Selected Areas in Communications*, vol.11, no.4, May 1993, pp. 579-589.
- [29] K. Bogineni and P. W. Dowd, "A Collisionless Media Access Protocols for High Speed Communication in Optically Interconnected Parallel Computers," *Proceedings of SPIE*, vol. 1577, September 1991, pp. 276-287.
- [30] P. W. Dowd and K. Bogineni, "Simulation Analysis of A Collisionless Multiple Access Protocol for A Wavelength Division Multiplexed Star-Coupled Configuration," *Proceedings of 25th Annual Simulation Symposium*, Orlando, FL, April 1992.
- [31] K. Bogineni and P. W. Dowd, "A Collisionless Multiple Access Protocol for A Wavelength Division Multiplexed Star-Coupled Configuration: Architecture and Performance Analysis," *IEEE/OSA Journal of Lightwave Technology*, vol. 10, no. 11, November 1992, pp. 1688-1699.
- [32] K. M. Silvalingam and P. W. Dowd, "A Multilevel WDM Access Protocol for an Optically Interconnected Multiprocessor System," *IEEE/OAS Journal of Lightwave Technology*, vol. 13, no. 11, November 1995, pp. 2152-2167.
- [33] K. M. Sivalingam and P. W. Dowd, "A Lightweight Media Access Protocol for a WDM-Based Distributed Shared Memory System," *Proceedings of IEEE INFOCOM' 96*, 1996, pp. 946-953.
- [34] H. Jeon and C. Un, "Contention-based Reservation Protocols in Multiwavelength Optical Networks with a Passive Star Topology," *Proceedings, IEEE ICC*, June 1992, pp. 1473-1477.

- [35] J. H. Lee and C. K. Un, "Dynamic Scheduling Protocol for Variable-Sized Messages in a WDM-Based Local Network," *IEEE/OSA Journal of Lightwave Technology*, vol. 14, no. 7, July 1996, pp. 1595-1600.
- [36] F. Jia, B. Mukherjee, and J. Iness, "Scheduling Variable-Length Messages in a Single-Hop Multichannel Local Lightwave Network", *IEEE/ACM Transactions on Networking*, vol.3, no.4, August 1995, pp.477- 487.
- [37] A. Muir and J. J. Garcia-Luna-Aceves, "Distributed Queue Packet Scheduling Algorithms for WDM-Based Networks", *Proceedings of IEEE INFOCOM'96*, 1996, pp.938-945.
- [38] B. Hamidzadeh, Maode Ma, and M. Hamdi, "Message Sequencing Techniques for On-Line Scheduling in WDM Networks", *IEEE/OSA journal of lightwave Technology*, vol.17, no.8, August 1999, pp.1309-1319.
- [39] Maode Ma, B. Hamidzadeh, and M. Hamdi, "A Receiver-Oriented Message Scheduling Algorithm for WDM Lightwave Networks", *Computer Networks*, vol.31, no.20, September 1999, pp.2139-2152.
- [40] Bo Li, Ganz A., Krishna C.M., "An In-Band Signaling Protocol for Optical Packet Switching Networks" *IEEE journal on Selected Areas in Communications*, vol. 18, no. 10, October 2000, pp. 1876-1884.
- [41] Lee, J.H., "Persistent Reservation Protocol for Variable-Length Messages in WDM-based Local Networks", *Proceedings of IEEE Communications*, vol. 148, no. 2, April 2001, pp. 81-85.
- [42] Sakuta, M., Nishino, Y., Sasase, I., "Channel Reservation Protocol Using a Counter for Detecting a Source Conflict in WDM Single-Hop Optical Network with Non-Equivalent Distance", *Proceedings of IEEE International Conference on Communications*, vol.3, 2001, pp.707-711.

- [43] Hwa-Chun Lin and Pei-Shin Liu, "Dynamic Control Frames in Reservation-Based Packet Scheduling for Single-Hop WDM Networks", *Proceedings of IEEE International Conference on Parallel Processing*, 2003, pp. 87-95.
- [44] Papadimitriou G. I., Obaidat M.S., Pomportsis A.S., "Adaptive Protocols for Single-Hop Photonic Networks with Bursty Traffic", *Proceedings of IEEE International Conference on Parallel Processing*, pp. 227 – 231, 2001.
- [45] M. Jonsson, K. Borjesson, and M. Legardt, "Dynamic Time-Deterministic Traffic in a Fiber-Optic WDM Star Network", *Proceedings of Ninth Euromicro Workshop on Real Time Systems*, June 1997, pp.25-33.
- [46] Maode Ma, B. Hamidzadeh, and M. Hamdi, "Efficient Scheduling Algorithms for Real-Time Service on WDM Optical Networks", *Photonic Network Communications*, vol.1, no.2, August 1999, pp.161-178.
- [47] Maode Ma and M Hamdi, "A Scheduling Algorithm for Differentiated Service on WDM Passive Optical Access Networks", *Journal of Optical Networking*, vol.1, no. 11, November, 2002, pp. 386-396.
- [48] Hung-Ying Tyan, Jennifer C. Hou, Bin Wang, and Ching-chih Han, "On Supporting Temporal Quality of Service in WDMA-Based Star-Coupled Optical Networks", *IEEE Transactions on Computers*, vol.50, no.3, March 2001, pp.197-214.
- [49] Anthony C. Kam, Kai-Yeung Siu, Richard A. Barry, and Eric A. Swanson, "A Cell Switching WDM Broadcast LAN with Bandwidth Guarantee and Fair Access", *IEEE/OSA Journal of Lightwave Technology*, vol.16, no.12, December 1998, pp.2265-2280.
- [50] Maode Ma and M. Hamdi, "Providing Deterministic Quality-of-Service Guarantees on WDM Optical Networks", *IEEE Journal on Selected Areas in Communications*, vol.18, no.10, October 2000, pp.2072-2083.

- [51] B. Li and Y. Qin, "Traffic Scheduling in A Photonic Packet Switching System with QoS Guarantee", *IEEE/OSA Journal of Lightwave Technology*, vol.16, no.12, December 1998, pp.2281-2295.
- [52] Macro A. Marsan, Andrea Bianco, Emilio Leonardi, Alessandro Morabito, and Fabio Neri, "All-Optical WDM Multi-Rings with Differentiated QoS", *IEEE Communications Magazine*, February 1999, pp.58-66.
- [53] L. Elhanany, D. Sadot, "A Prioritized Packet Scheduling Architecture for Provision of Quality-of-Service in Tbits/sec WDM Networks", *Proceedings of IEEE International Conference on Communications*, New-Orleans, June 2000, vol.2, pp. 695-700.
- [54] J. Diao, P. L. Chu, "Packet Rescheduling in WDM Star Networks with Real-Time Service Differentiation", *IEEE Journal of Lightwave Technology*, vol. 19, no. 12, December 2001, pp. 1818 –1828.
- [55] K. Bengi, H.R. van As, "CONRAD: A Novel Medium Access Control Protocol for WDM Local Lightwave Networks Enabling Efficient Convergence of Real-Time and Data Services", *Proceedings of IEEE LCN'01*, 2001, pp. 468 – 476.
- [56] Jadwiga Indulska and Jason Richards, " A Comparative Simulation Study of Protocols for A Bus WDM Architecture", *Proceedings of International Conference on Networks*, 1995, pp.251-255.
- [57] W. Melody Moh, Teng-Sheng Moh, Yu-J. Chien, Joanna Wang, Ray-J. Wang, and Yi-W. Wang, "The Support of Optical Network Protocols for Multimedia ATM Traffic", *Proceedings. International Conference on Networks*, 1995, pp.1-5.
- [58] Nen-Fu Huang and Huey-Ing Liu, "Wavelength Division Multiplexing-Based Video-on-Demand Systems", *IEEE/OSA Journal of Lightwave Technology*, vol.17, no.2, February 1999, pp.155-164.
- [59] Lixin Wang, Maode Ma, Hamdi, M., "Efficient Protocols for Multimedia Streams on WDMA Networks", *IEEE/OSA Journal of Lightwave Technology*, vol.21, no.10, October 2003, pp. 2123-2144.

- [60] Jin Seek Choi, Golmie N., Su D., "A Bandwidth Guaranteed Multi-Access Protocol for WDM Local Networks," *Proceedings of IEEE ICC'00*, 2000, vol. 3, pp.1270 – 1276.
- [61] Maode Ma and M. Hamdi, "Providing Deterministic Quality-of-Service Guarantees on WDM Optical Networks", *IEEE Journal on Selected Areas in Communications*, vol.18, no. 10. October 2000, pp. 2072-2083.
- [62] Milan Kovacevic and Mario Gerla, "HONET: An Integrated Services Wavelength Division Optical Network", *Proceedings. IEEE ICC'94*, 1994, pp.1669-1674.
- [63] A. Yan, A. Ganz, and C. M. Krishna, "A Distributed Adaptive Protocol Providing Real-Time Services on WDM-based LANs," *IEEE Journal of Lightwave Technology*, vol.14, pp.1245-1254, June 1996.
- [64] S. French, *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. New York: Wiley, 1950.
- [65] Kenneth R. Baker, *Introduction to Sequencing and Scheduling*. New York: Wiley, 1974.
- [66] Moore, J. M., "An N Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs", *Management Science*, vol. 15, pp. 102-109, September 1968.
- [67] Jon M. Oeha and Fouad A. Tobagi, "Evaluation Scheduling Algorithms for Traffic with Heterogeneous Performance Objectives", *Proceedings of IEEE GLOBECOM'90*, 1990, vol.1, pp. 21-27.
- [68] Wu J.-L.C., Tien-Yu Huang, "Dynamic Priority Schemes to Improve the QOS in ATM Networks", *Proceedings of IEEE GLOBECOM'94*, 1994, vol2, pp. 1195-1199.
- [69] James K.W., Yun K.Y., "Supporting Quality of Service in a Terabit Switch", *Proceedings of IEEE Conference of Performance, Computing, and Communications*, Feb. 2000, pp. 55 –61.
- [70] Chen T.M., Walrand J., Messerschmitt D.G., "Dynamic Priority Protocols for Packet Voice", *IEEE Journal on Selected Areas in Communications*, vol.7, no.5, June 1989, pp. 632 –643.

- [71] Buttazzo G.C., Sensini F., “Optimal Deadline Assignment for Scheduling Soft Aperiodic Tasks in Hard Real-Time Environments”, *IEEE Transaction on Computers*, vol. 48, no.10, October 1999, pp. 1035 –1052.
- [72] Sung Won Lee, Dong Ho Cho, Yong Ki Park, “Improved dynamic weighted cell scheduling algorithm based on earliest deadline first scheme for various traffic of ATM switch”, *Proceedings of IEEE GLOBECOM’96*, 1996, vol.3, pp.1959 –1963.
- [73] Athanasios Papoulis and S.Unnikrishna Pillai, *Probability, Random Variables and Stochastic Processes*, New York: McGraw-Hill, 2002, pp.181-185.
- [74] Mitrani, I., P. J. B. King, “Multiprocessor Systems with Preemptive Priorities”, *Performance Evaluation*, 1981, pp. 118-125.
- [75] Gail, H.R., S. L. Hantler, B. A. Taylor, “Analysis of a Non-Preemptive Priority Multiserver Queue”, *Advances in Applied Probability*, vol.20, no.4, 1988, pp. 852-879.
- [76] Gail, H.R., S. L. Hantler, B. A. Taylor, “On Preemptive Markovian Queue with Multiple Servers and Two Priority Classes”, *Mathematics Operations Research*, vol. 17, no. 2, 1992, pp. 365-391.
- [77] Kao, E.P.C. and K.S. Narayanan, “Computing Steady-State Probabilities of a Nonpreemptive Priority Multiserver Queue”, *ORSA Journal on Computing*, vol.2, 1990, no. 211-218.
- [78] Kao, E.P.C. and S.D. Wilson, “Analysis of Nonpreemptive Priority Queues with Multiple Servers and Two Priority Classes”, *European Journal of Operational Research*, vol. 118, 1999, pp. 181-193.
- [79] Kella, O., and U. Yechiali, “Waiting Times in the Non-Preemptive Priority $M/M/c$ Queue”, *Communications in Statistics-Stochastic Models*, vol.1, 1985, pp. 257-262.
- [80] Wagner, D., “Analysis of a Multi-Server Model with Non-Preemptive Priorities and Non-Renewal Input”, in: J. Labetoulle, JW Roberts (Eds.), *the Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks*, Elsevier, Amsterdam, 1994.

- [81] Wagner, D., "Waiting Time of a Finite-Capacity Multi-Server Model with Non-Preemptive Priorities", *European Journal of Operational Research*, vol. 102, 227-241, 1997.
- [82] Davis, R.H., "Waiting Time Distribution of a Multi-Server Priority Queueing Model", *Operations Research*, vol. 14, 1966, pp. 133-136.
- [83] Kapadia, A.S., Kazmi, M.F., and Mitchell A.C., "Analysis of a Finite Capacity Non Preemptive Priority Queue", *Computers and Operations Research*, vol.11, no. 3, 1984, pp. 337-343.
- [84] A. Movaghar, "On queueing with customer impatience until the beginning of service", *Queueing Systems*, vol. 29, no. (2-4), 1998, pp. 337-350.
- [85] Leonard Kleinrock, *Queueing Systems*, vol. 1, *Theory*, John Wiley & Sons, New York, 1976, pp. 52-53.
- [86] Dimitris Bertsimas and John N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.
- [87] Mokhtar S. Bazaraa, and John J. Jarvis, *Linear programming and network flows*, John Wiley & Sons, New York, 1977.
- [88] I.S. Gopal and C.K. Wong, "Minimizing the Number of Switchings in a SS/TDMA system," *IEEE Transaction on Communications*, vol. COM-33, pp. 497-501, June 1985.
- [89] Marsan, M.A., Bianco, A., Leonardi, E., Neri, F., and Nucci, A., "Multihop packet scheduling in WDM/TDM networks with nonnegligible transceiver tuning times," *IEEE Transactions on Communications*, vol. 48, no. 4, April 2000, pp. 692 – 703.
- [90] S. Vavasis, "Nonlinear Optimization: Complexity Issues", Oxford University Press, New York, N.Y., 1991.
- [91] R. E. Gomary, "An Algorithm for the Mixed Integer Problem," *RAND Report P-1885*, Feb. 1960.
- [92] J.F. Benders, "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik*, vol. 4, pp. 238-253, 1962.