uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

u Ottawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Mushi Jin

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Network Design and QoS Provision in High-Speed Optical Switching Networks

TITRE DE LA THÈSE / TITLE OF THESIS

Oliver Yang

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Martin Bouchard | Alex Vukovic

Shikharesh Majumdar | Johnny Wong

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Network Design and QoS Provision in High-Speed Optical Switching Networks

Mushi Jin

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the PhD degree in Electrical Engineering

Computer Communication Network Research Lab
School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-49364-9
Our file Notre référence
ISBN: 978-0-494-49364-9

Abstract

Optical networking is a promising solution to support the ever-increasing Internet traffic. This thesis aims to investigate the important issues of topology design, resource sharing, differentiated service provision, flow control as well as the methodology for performance evaluation of optical networks.

We first build two analytical models in order to approximate the loss rate of an Optical Burst Switching (OBS) node more accurately than the traditional burst-based methods. Through the analysis of interacting bursts at the core nodes, we compare the loss ratio of different flows under various assembly parameters. We then propose a DiffServ scheme by adjusting the assembly parameters for different optical bursts. This scheme is suitable for a distributed OBS network without any requirement on centralized information or operation.

Next, we design a feedback-based OBS controller by assigning an exponential distribution to the burst assembly intervals. The loss rate of a flow can be controlled by adjusting the assembly timer. Based on this, we design and study a generic OBS controller for different assembly algorithms and signaling protocols that do not need global network information nor affect the network throughput. Some efficient predictor and adjuster models are proposed to implement the control process. Performance evaluation shows that this method is superior to other ordinary schemes based on the rate-adjustment method. Some critical issues on the system design and parameter settings are also investigated and discussed.

Finally, we revisit the fundamental star topology to develop an overlaid-star topological variation and address various implementation issues in such architecture. We model the node connections as an asymmetrical multiconnection three-stage Clos network, and propose a network flow model to solve the routing, wavelength assignment and time slot assignment problems under the time-division-multiplexing operation. We demonstrate via simulations that this overlaid-star topology can support flexible QoS provisioning that we are currently studying.

In summary, our work provides a technologically feasible solution to provide required QoS for future high-speed networks using available optical technologies.

Acknowledgment

I would like to express my sincere thanks to my supervisor, Professor Oliver W. W. Yang for his research supervision and guidance. I would also like to thank the members in the Computer Communication Network Research (CCNR) group in the School of Information Technology and Engineering (SITE) of the University of Ottawa, for their continuous cooperation and encouragement.

I would also like to acknowledge the financial support of an Ontario Graduate Scholarship and an admission/excellence scholarship from the University of Ottawa. The related research is supported in part by the Natural Sciences and Engineering Research Council (NSERC) and industrial partners through the Agile All-Photonic Networks (AAPN) Research Network.

Table of Contents

Abstract i
Acknowledgment ii
Table of Contents iii
List of Figures vii
List of Tables ix
Table of Acronyms x
Table of Symbols and Notations xii

Chapter 1 Introduction 1
1.1 Literature Review 1
1.1.1 Optical Networking 1
1.1.1.1 Network Topology 1
1.1.1.2 Switching Techniques 3
1.1.2 Optical Burst Switching 4
1.1.2.1 Burst Assembly 4
1.1.2.2 Differentiated QoS 6
1.1.2.3 Congestion Control 6
1.1.3 Optical Time-Division-Multiplexing 8
1.1.3.1 Time Slot Assignment 9
1.1.3.2 Clos Switches 9
1.2 Motivations 10
1.3 Objectives 12
1.4 Methodologies and Approaches 12
1.5 Contributions 14
1.6 Thesis Organization 15
1.7 Publications 15
Chapter 2 Network Models, Operations and Assumptions 17
2.1 Network Model 17
2.2 Node Model 18
2.2.1 Edge Node 19
2.2.2 Core Node 19
2.3 Network Operations 20
2.4 Simulation Model 22
2.4.1 Traffic Model 22
2.5 Performance Measures 23
2.6 Assumptions 24
Chapter 3 DiffServ Provision in OBS Networks 26
3.1 System Description and Model 26
3.2 Loss Analysis 27
3.2.1 JET Protocol 28
3.2.2 JETseg Protocol 28
3.2.3 The Less-accurate Burst-based Model 29
3.2.4 Verification 30

3.3 DiffServ Provision ... 33
3.3.1 Loss Analysis ... 33
3.3.2 DiffServ Algorithms ... 35
3.4 Performance Evaluation ... 37
3.5 Concluding Remarks ... 41
Chapter 4 Flow Control in OBS Networks: A Simple Approach ... 42
4.1 System Description ... 42
4.2 Analysis ... 43
4.2.1 Delay Analysis ... 44
4.2.2 Loss Analysis ... 44
4.3 Control Algorithm ... 49
4.4 Performance Evaluation ... 51
4.4.1 Verification of the Influence of the Assembly Timer ... 52
4.4.2 Static Networks ... 55
4.4.3 Dynamic Networks ... 57
4.4.4 Comparisons ... 59
4.5 Design Issues ... 60
4.5.1 Measurement Interval ... 61
4.5.2 Measurement Period ... 61
4.5.3 Adjustment Interval ... 62
4.5.4 Adjustment Rate ... 62
4.5.5 Range of the Loss Rate ... 63
4.5.6 Other Issues ... 63
4.6 Concluding Remarks ... 64
Chapter 5 Flow Control in OBS Networks: A Generic Approach ... 65
5.1 System Description ... 65
5.2 Analysis and Verification ... 66
5.2.1 Loss Analysis ... 67
5.2.1.1 Case of burst dropping ... 68
5.2.1.2 Case of burst-header dropping ... 69
5.2.1.3 Case of burst-tail dropping ... 69
5.2.2 Verification ... 71
5.2.3 Applications ... 74
5.3 Controller Design ... 74
5.3.1 Predictor Design ... 74
5.3.2 Adjuster Design ... 76
5.3.3 Predictor-Adjuster Combinations ... 77
5.4 Performance Evaluation ... 77
5.4.1 Static Networks ... 78
5.4.2 Dynamic Networks ... 79
5.4.3 Comparisons ... 80
5.5 Concluding Remarks ... 81
Chapter 6 APOSN Networks ... 83
6.1 Network Model ... 83
6.1.1 Connection Modeling ... 85
6.2 Architectural Features ... 88

6.2.1 Cost Analysis 89
6.3 OBS Operations 90
6.3.1 General Schemes 91
6.3.2 DiffServ Provision and Flow Control 92
6.4 OBS Performance 92
6.4.1 Basic Performance 93
6.4.2 Effect of MBS 95
6.4.3 Effect of MAD 95
6.4.4 Effect of Optical Switching Time 96
6.4.5 Effect of the Number of Wavelengths 96
6.4.6 Effect of Retransmission 98
6.4.7 DiffServ and Flow Control 98
6.5 OTDM Operations 100
6.5.1 Message Synchronization 102
6.5.2 Time-Wavelength Assignment 102
6.6 OTDM Performance 107
6.6.1 Basic Performance 107
6.6.2 Effect of MAD 110
6.6.3 Effect of MSD 110
6.6.4 Effect of the Number of Wavelengths 112
6.6.5 Effect of the Number of Iterations 112
6.7 Comparisons and Discussions 113
6.7.1 Comparison of OTDM and OBS 113
6.7.2 Network Scalability 115
6.7.3 Other Issues 116
6.8 Concluding Remarks 117
Chapter 7 Design Guidelines 118
7.1 Network Topology 118
7.2 Node Architecture 118
7.3 Operational Issues 119
7.4 Other Issues 119
7.4.1 Fairness 119
7.4.2 Time-Complexity 120
7.4.3 Scalability 120
7.4.4 Protection and Restoration 121
7.4.5 Optimization and Integration 121
Chapter 8 Conclusions 123
8.1 Future work 124
References 125

Appendix A: An OPNET Example 134
Appendix B: Burst-based Analysis 136
Appendix C: Delay Analysis 137
Appendix D: Delay Performance 139
Appendix E: Loss Rate and Throughput 141
Appendix F: Performance of the Simple Controller: Pareto Traffic 143

Appendix G: Performance of the Simple Controller: Effect of the Processing Time 145
Appendix H: Performance of the Simple Controller: Effect of the Interval Distributions..........147
Appendix I: Performance of the Generic Controller: Effect of the Processing Time..................148

List of Figures

Fig. 2.1. Optical network architecture 17
Fig. 2.2. Architecture of an edge node 18
Fig. 2.3. Architecture of a core node 20
Fig. 3.1. Packet-based view of the transmission scheme 26
Fig. 3.2. Comparison of two models: JET 30
Fig. 3.3. Comparison of two models: JETseg 32
Fig. 3.4. Network model for simulations 37
Fig. 3.5. Loss and delay of two classes of traffics 38
Fig. 3.6. Loss ratio of different classes of traffics 39
Fig. 4.1. Schematic transmission 42
Fig. 4.2. Flow analysis of JIT 45
Fig. 4.3. Flow analysis of JETtailSeg 47
Fig. 4.4. Network model for simulations 51
Fig. 4.5. Delay performance 52
Fig. 4.6. Loss rate and throughput performance of Flow 1-5 using JIT 53
Fig. 4.7. Loss rate and throughput performance of Flow 1-5 using JETtailSeg 54
Fig. 4.8. Controller performance using JIT 55
Fig. 4.9. Controller performance using JETtailSeg 57
Fig. 4.10. Controller performance using JIT 58
Fig. 4.11. Controller performance using JETtailSeg 58
Fig. 4.12. Network throughput using JETtailSeg 59
Fig. 4.13. Performance w.r.t. measurement interval t_m 60
Fig. 4.14. Performance w.r.t. measurement period t_p 61
Fig. 4.15. Performance w.r.t. adjustment interval t_a 62
Fig. 4.16. Performance w.r.t. adjustment rate ε 62
Fig. 5.1. Description of the control scheme 65
Fig. 5.2. Loss analysis of burst dropping 67
Fig. 5.3. Loss analysis of head dropping 69
Fig. 5.4. Loss analysis of tail dropping 69
Fig. 5.5. Network model for simulations 71
Fig. 5.6. Flow loss w.r.t. the assembly timer for Flow 1-5 72
Fig. 5.7. Performance of static networks 78
Fig. 5.8. Performance of dynamic networks under Poisson traffic 79
Fig. 5.9. Comparison of two schemes under Poisson traffic 81
Fig. 6.1. MAN connection of an APOSN 83
Fig. 6.2. WAN connection of an APOSN 84
Fig. 6.3. Transmission side of an edge node 85
Fig. 6.4. Connections of one core node 86
Fig. 6.5. Equivalent Clos network of APOSN 87
Fig. 6.6. Burst segmentation with header dropping 91
Fig. 6.7. Loss performance 93
Fig. 6.8. Delay performance 93
Fig. 6.9. Utilization and throughput performance 94
Fig. 6.10. Loss performance vs. MBS 94

Fig. 6.11. Delay performance vs. MBS 95
Fig. 6.12. Loss and delay performance vs. MAD 96
Fig. 6.13. Loss and delay performance vs. optical switching time 96
Fig. 6.14. Loss and delay performance vs. the number of wavelengths 97
Fig. 6.15. Loss performance w.r.t. the number of transmissions 97
Fig. 6.16. Delay performance w.r.t. the number of transmissions 98
Fig. 6.17. DiffServ performance 99
Fig. 6.18. Flow control performance 99
Fig. 6.19. Synchronization of message transmission 101
Fig. 6.20. The flow network corresponding to Fig. 6.5 network 103
Fig. 6.21. TWA operation in one slot period 105
Fig. 6.22. Loss performance 107
Fig. 6.23. Delay performance 108
Fig. 6.24. Utilization and throughput 108
Fig. 6.25. Scheduling of (a) Poisson traffic (b) Pareto Traffic 109
Fig. 6.26. Loss and delay performance 110
Fig. 6.27. Delay performance vs. MSD 111
Fig. 6.28. Loss performance vs. MSD 111
Fig. 6.29. Loss and delay performance vs. the number of wavelengths 111
Fig. 6.30. Loss performance vs. the number of iterations 112
Fig. 6.31. Utilization using OTDM and OBS 113
Fig. 6.32. Loss rate using OTDM and OBS 113
Fig. 6.33. ETC delay using OTDM and OBS 114
Fig. 6.34. Loss ratio w.r.t. optical switching time 114
Fig. 6.35. Loss ratio w.r.t. the number of wavelengths 115
Fig. 6.36. Loss vs. Load: comparison of two networks 115
Fig. A.1. An OBS network model 134
Fig. A.2. OBS node models 134
Fig. A.3. OBS process models 135
Fig. B.1. A burst-based model 136
Fig. D.1. Delay performance: Poisson packet arrival 139
Fig. D.2. 95% confidence interval of delay: Poisson packet arrival and fixed packet length 140
Fig. D.3. Delay performance: Pareto packet arrival and CAIDA packet length 140
Fig. E.1. Poisson packet arrival and fixed packet length using JIT 141
Fig. E.2. Poisson packet arrival and exponential packet length using JIT 141
Fig. E.3. Poisson packet arrival and exponential packet length using JETtailSeg 142
Fig. E.4. Pareto packet arrival and fixed packet length using JIT 142
Fig. F.1. Reducing loss rate: Pareto packet arrival and CAIDA packet length 143
Fig. F.2. Avoiding loss increasing: Pareto packet arrival and CAIDA packet length. 143
Fig. F.3. Reducing loss rate: Pareto packet arrival and CAIDA packet length 144
Fig. F.4. Avoiding loss increasing: Pareto packet arrival and CAIDA packet length 144
Fig. G.1. Controller performance: Poisson packet arrival 145
Fig. G.2. Controller performance: Pareto packet arrival 146
Fig. H.1. Controller performance 147
Fig. I.1. Controller performance in a P1-A2 network: Poisson packet arrival 148
Fig. I.2. Controller performance in a P1-A2 network: Pareto packet arrival 149

List of Tables

Table 4.1. Network Throughput (Gbps) 54
Table 5.1. Network Throughput (Gbps) 73

Table of Acronyms

		Section of first appearance
AAPN	Agile All-Photonic Network	1.1.1.1
APN	Articulated Private Networks	1.1.1.1
APOSN	All Photonic Overlaid-Star Network	1.2
AQM	Active Queue Management	1.1.2.3
ATM	Asynchronous Transfer Mode	1.1.2.3
BHP	Burst Header Packet	1.1.2.1
CAIDA	Cooperative Association for Internet Data Analysis	2.4.1
CS	Cell Size	6.5
DB	Data Burst	1.1.2.1
DiffServ	Differentiated Service	1.1.2.2
DSL	Digital Subscriber Line	1.1.1.2
ER	Edge Receiving side	6.1
ET	Edge Transmission side	6.1
ETC	Edge-to-Core	2.5
EWMA	Exponentially Weighted Moving Average	5.3.1
FCFS	First-Come-First-Serve	2.3
FCP	Feedback Control Packet	4.1
FDL	Fiber Delay Line	1.1.2.3
FIND	Future Internet Design	1.1.1.1
FRR	Forward Resource Reservation	1.1.1.2
GAIMD	General Addictive-Increase Multiple-Decrease	1.1.2.3
GENI	Global Environment for Networking Innovations	1.1.1.1
GMPLS	Generalized Multiprotocol Label Switching	1.1.2
GUI	Graphical User Interface	1.4
JET	Just-Enough-Time	1.1.2.3
JETheadSeg	Just-Enough-Time with burst-header segmentation	2.3
JETtailSeg	Just-Enough-Time with burst-tail segmentation	2.3
JETseg	Just-Enough-Time with burst segmentation	2.3
JIT	Just-In-Time	1.1.2.3
LQF	Longest Queue First	1.1.2.1
MAD	Maximum Assembly Delay	6.3.1

MAN	Metro-Area-Network	6.0
MBS	Maximum Burst Size	6.3.1
MPLS	Multiprotocol Label Switching	1.1.2
MSD	Maximum Scheduling Delay	6.3.1
NGN	Next Generation Network	1.1.1.1
NLR	National LambdaRail	1.1.1.1
NSF	National Science Foundation	1.1.1.1
OBF	Oldest Burst First	1.1.2.1
OBS	Optical Burst Switching	1.1.1.2
OCBS	Optical Composite Burst Switching	B.0
OTDM	Optical Time-Division-Multiplexing	1.1.1.2
OXC	Optical Cross-Connect	1.1.1.1
QoS	Quality of Service	1.1.1.1
PCwER	Proportional Control algorithm with Explicit Reduction-request	4.4.4
RWA	Routing and Wavelength Assignment	1.1.3
SONET	Synchronous Optical Network	1.1.1.1
TCP	Transport Control Protocol	1.1.2.3
TSA	Time Slot Assignment	1.1.3
TSOBS	Time-Slotted-Optical-Burst-Switching	1.1.3
TWA	Time-Wavelength Assignment	6.5
UCLP	User Controlled Light Paths	1.1.1.1
UDP	User Datagram Protocol	1.1.2.3
VOIP	Voice Over IP	1.1.2.3
VOQ	Virtual-Output-Queue	1.1.3.1
WAN	Wide-Area-Network	6.0
WDM	Wavelength-Division-Multiplexing	1.1.1.2
WLAN	Wireless Local Area Network	1.1.1.2

Table of Symbols and Notations

		Section of first appearance
δ	acceptable margin of the loss rate to be achieved	4.3
δ_i	remaining time from a request from ET#*i* arriving at the core to the nearest slot boundary	6.5.1
ε	adjustment rate to modify an assembly timer by the source edge node	4.3
θ	loss rate to be achieved	4.3
σ^2	variance of the burst length	3.3.1
λ	burst arrival rate per channel	3.2
λ_0	control wavelength	2.2.1
λ_i	payload wavelength (*i=1, ... W*)	2.2.1
λ_l, λ_l'	mean arrival rate of bursts to link *l* (bursts/sec)	4.2.2
$\lambda_{s,d}, \lambda_{s,d}'$	mean burst generation rate on *Flow s-d* (bursts/sec)	4.2
$\lambda_{s,d,l}, \lambda_{s,d,l}'$	mean arrival rate of *Burst(s,d,l)* (bursts/sec)	4.2.2
$\lambda^{(N)}$	mean output burst rate from an edge (bursts/sec)	C.0
$\lambda_{s,d}^P$	mean packet arrival rate at source-*s* going to destination-*d* (packets/sec)	4.2
μ	burst service rate per channel	3.2
μ_i	service rate of Class-*i* bursts	3.2
π_i	arrival rate of Class-*i* bursts	3.2
ρ	normalized traffic load	3.2
$\rho^{(N)}$	mean output traffic load from an edge	C.0
Λ_i	input traffic to ET#*i*	4.2.2
c_j	column node (processor) j	6.5.2
d, d'	destination edge node	4.2.1
f	number of ETs in a star network	6.1.1
$f_X(\cdot)$	probability density function (*pdf*) of a random variable *X*	4.2
g	number of ERs in a star network	6.1.1
l	link *l*	4.2.2
m_1	mean burst length	3.3.1
n	number of iterations	6.5.2
p_{avg}	mean loss rate over last *X* intervals	4.3
p_{busy}	probability of a channel being busy	3.3.1
$p_{predict}$	predictor result	5.3.1
$p_{s,d}, p_{s,d}'$	loss rate of *Flow s-d*	4.2.2
$p_{s,d}(i)$	loss rate of the last *i-th* interval, *i=0, 1, ...*	5.3.1
$p_{s,d,l}, p_{s,d,l}'$	mean loss rate of *Burst(s,d,l)*	4.2.2
$p_{s,d,l}(t)$	loss rate of *Burst(s,d,l)* at time *t*	4.2.2
r_1	residual length of a contended burst	3.3.1

r_i	row node (processor) i	6.5.2
s , s'	source node (processor)	4.2
$s-d$	burst flow from source edge s to destination edge d	4.2
(s,d,l)	bursts on *Flow* $s-d$ passing through link l	4.2.2
t	destination node (processor)	6.5.2
t_a	adjustment interval for the source node to modify an assembly timer	4.3
t_i	propagation delay from ET#i to the core node	6.5.1
t_{ij}	ij -th element in $\boldsymbol{T}$	6.5.2
t_m	interval for the destination edge node to measure the loss rate of a flow	4.3
$t_{\max}$	largest propagation delay from all ETs to the core node	6.5.1
t_p	period for the source edge node to calculate the loss rate of a flow	4.3
t_s	request processing time	6.5.1
t_{sw}	optical switching time of the core node	6.3.1
w_{EWMA}	weight for the EMWA predictor	5.3.1
w_i	weight Prd3	5.3.1
xct_i	flow on the arc from c_i to t	6.5.2
xrc_{ij}	flow on the arc from r_i to c_j	6.5.2
xsr_i	flow on the arc from s to r_i	6.5.2
A	assembly parameter	3.3.1
$A_l(t)$	number of bursts arrived to link l during period $[0, t]$	4.2.2
$A^P_{s,d}(t)$	number of packets arrived for *Flow s-d* during period $[0, t]$	C.0
$A_{s,d,l}(t)$	number of *Burst(s,d,l)* arrived to link l during period $[0, t]$	4.2.2
C	bandwidth of wavelength channels	3.2
C_j	sum of column j in $\boldsymbol{T}$	6.5.2
D_a	assembly delay of packets at the edge	4.2.1
D_q	queueing delay of packets at the edge	4.2.1
D_{sys}	total system delay of packets at the edge (=$D_a + D_q + D_t$)	4.2.1
D_t	transmission delay of packets at the edge	4.2.1
$F_X(\cdot)$	cumulative distribution function (*cdf*) of a random variable X	4.2
H_i	number of fibers from the core node to ER#i	6.1.1
$I_{s,d}$	interarrival time of bursts on *Flow s-d* (sec)	4.2
J	number of links	2.1
K_{fixed}	parameter for Adjuster#1/#3	5.3.2
K_i	number of fibers from ET#i to the core node	6.1.1
K_{prop}	parameter for Adjuster#2/#3	5.3.2
L	number of intervals for Predictor#1/3	5.3.1
$L^{(N)}$	length of bursts generated at a source (sec)	C.0
L_{cs}	cell size	6.5
L_i	length of Class-i bursts for the assembly process	3.2
$L_{s,d}$	length of bursts on *Flow s-d* (sec)	4.2.1
$L_{s,d,l}, L'_{s,d,l}$	length of *Burst(s,d,l)* (sec)	4.2.2

$L^{P}_{s,d}$	length of packets arrived at source-*s* going to destination-*d* (sec)	4.2
$L^{P\ (n)}_{s,d}$	total length of *n* packets arrived at source-*s* going to destination-*d* (sec)	C.0
M	number of core nodes	2.1
N	number of destination nodes	2.1
N_c	number of traffic class	3.2
N_e	number of edge nodes	3.2
N_{msd}	maximum scheduling delay (slots)	6.5
$P_{s,d}$	number of packets in a burst on *Flow s-d*	C.0
R_i	sum of row i in $\boldsymbol{T}$	6.5.2
R_{len}	ratio of Class#0 burst length to Class#1 burst length	3.4
S	number of ports of an core switch	6.2.1
$\boldsymbol{T}$	traffic matrix	6.5.2
T_{mad}	maximum assembly delay	6.3.1
T_{off}	offset time	2.3
$T_{s,d}$, $T'_{s,d}$	assembly timer for bursts on *Flow s-d*	4.2
W	number of wavelength channels for payload message	2.2.1
W_c	number of wavelength channels at the output side of a core node	3.2
W_e	number of wavelength channels at the output side of an edge node	3.2
$X_{s,d,l}$	arrival time of *Burst(s,d,l)*	4.2.2
$\overline{X}$	mean of random variable X	4.2
Z	number of intervals for calculating the average loss rate	5.3.1

CHAPTER 1
INTRODUCTION

Networking has become an essential part of our life. The advance of optical technologies is a major driving force for communication network evolution. The optical technology provides huge bandwidth than ever, which reduces the device expenditure and service cost, and enables new high-speed applications. We shall first review existing works in different areas which have motivated the present research in some important issues in optical networking, including topology design, resource allocation, differentiated service and congestion control. Then we shall present the objectives, methodologies and organization of this thesis.

1.1 Literature Review

The increasing volume of Internet traffic is always challenging the communication networks merely based on electronic switching and routing technology. Optical technologies are promising solutions to the high-speed networking required by the ever-increasing applications. These technologies and associated issues are reviewed in the following.

1.1.1 Optical Networking

A major concern on optical networking is the availability of optical components [BoJu97, MaKu03, LiGo99]. Due to the lack of feasible optical buffers and optical signal processors, the queue-based models and concepts in the electronic networking cannot be directly translated to the optical domain. As a result, some different viewpoints on network topology and switching scheme have been studied for optical applications as follows.

1.1.1.1 Network Topology

An appropriate topology is a primary factor affecting other issues in a network. There are three major types of topologies for optical transport networks: mesh (e.g., [IrMa98]), ring (e.g., [WuTs92]) and tree (e.g., [Yang94]). In a mesh topology, nodes are connected with many arbitrary inter-connections. The ring topology is widely used in optical networks, e.g., SONET (Synchronous Optical Network) rings, because it can offer a high efficiency in network failure protection. A simple physical tree topology requires the minimum number of links to interconnect the whole network, and can thus reduce the fiber expenditure in an optical network. Alternatively, the network topology can be classified into single-hop vs. multiple-hop [MuKh92a, MuKh92b]. Multiple-hop mesh networks allow an easy scalability, but it is also difficult to implement routing, resource allocation and traffic engineering. Although multiple-hop networks are commonly studied, single-hop networks are still playing an important role because of the simplicity in routing and possible elimination of complex wavelength or signal conversion. For example, the single-hop star configuration as a special form of a tree has an advantage of easy routing and configuration. Many research works have focused on the passive single-star optical network, but such passive network suffers from low efficiency and reliability and is usually suitable for access networks only, e.g., [PaLe04, ZhMo05, NaWe04].

When we consider the possible topology for the NGN (Next Generation Network), it can be a smooth evolution of the current topology or a redesigned architecture. A natural way is to interconnect existing systems. Then we have a multiple-hop meshed network and our remaining task is to find solutions to provide QoS (Quality of Service) for the applications. On the other hand, there are some works on future Internet architecture because of the concern that current Internet infrastructure might not be able to facilitate future services and applications. A handful of Internet pioneers are considering how to fix the network's shortcomings by starting again from scratch. America's NSF (National Science Foundation) has launched two initiatives. The first project, GENI (Global Environment for Networking Innovations), is a project to "enable the research community to invent and demonstrate a global communications network and related services that will be qualitatively better than today's Internet" [GENI06]. The second project, FIND (Future Internet Design), will examine how best to define the Internet architecture, principles and design for the needs of the future [FIND06]. The NLR (National LambdaRail) project [NLR06] is implemented to address a fully operational and national networking physical infrastructure in order to facilitate developing new networking technologies and capabilities.

In Canada, the UCLP (User Controlled Light Paths) project enables users to create, control and manage APNs (Articulated Private Networks) across multiple, independently managed domains [UCLP06]. A star-based Petaweb architecture [BlLe01, ViBe00] has been proposed to cover a nationwide distributed network using slowly-reconfigured OXCs (Optical Cross-Connect), but it lacks agility because of the slow reconfiguration-time. A further extension is the AAPN (Agile All-Photonic Network) research network which employs agile switching mechanism to connect edge nodes. The AAPN project focuses on active core nodes and trades more fibers for an increase in network reliabilities. It can further avoid complex routing, enforce network robustness and facilitate enabling technologies in the near future [BoCo04, MaVi06].

1.1.1.2 Switching Techniques

A communication network consists of three major elements: the transportation links, the switching nodes and the access networks to users. Currently, optical fibers, which have been the dominant transmission medium, are capable of providing a huge transmission capacity by using WDM (Wavelength-Division-Multiplexing) technology. There are also diversified choices of accessing methods, e.g., cable, DSL (Digital Subscriber Line) or WLAN (Wireless Local Area Network) to satisfy different services and applications. Therefore, the switching node needs to provide fast and flexible processing of messages to match the progress in the transmission links and to accommodate different access networks. However, electronic signal processing in the intermediate nodes still prohibits fast processing of optical messages. The electronic switching systems are bit-rate and protocol dependent; therefore, they are not able to keep pace with the growth in the capacity of the optical transmission. As we know, optical buffering and wavelength conversion are still infeasible in current applications although many proposals have been studied [ElMo00a, ElMo00b, MaMo93, LiSc03, StKl99]. The fabrication of a large-size fast optical switching device is another challenge that cannot be overlooked [GrDu03, PaPa03, DoFa03]. Therefore, sophisticated switching schemes are required in optical nodes.

Several technology candidates are applicable to all-optical switching [PaMa00], including circuit switching, packet switching and burst switching. In circuit switching, a switching path is set up before a payload message is sent out from a source node. This permanent connection is not released until the whole message has been transmitted successfully; therefore, this method generates a long latency and a low efficiency [StBa00]. Packet switching splits a payload message into sequential blocks, and uses a store-and-forward technique along physical paths that need not be pre-configured, e.g., [JoCh01, OmSi01, YaMu00]. However, it requires more processing functionalities on switching components, such as header processing and message storage. OBS (Optical Burst Switching) [Turn99, QiYo99, ChQi04, QiYo00, VeCh00, BaPe03, Gaug03] has combined and modified the previous two schemes to facilitate today's optical technology, but it is still unable to provide a good granularity due to its collision-based bursty characteristics. OTDM (Optical Time-Division-Multiplexing), e.g., [LiCh03, YuTo01, SeBe96], is a special WDM technique by combining these basic schemes, which is useful for highly-dynamic high-capacity photonic networks. The major challenges in an OTDM network are the routing, time slot assignment and wavelength assignment problems.

In general, OTDM technique provides good granularity and response in high-speed networks, and OBS technique enables fine flexibility and scalability, so they are considered as two most attractive techniques for WDM networks.

1.1.2 Optical Burst Switching

A lot of efforts have been made in optical burst switching [Turn99, QiYo99, XiVa00] as a viable solution for IP-over-WDM networks since it has combined the merits of traditional packet switching and circuit switching schemes, and modified them to facilitate enabling optical technologies. OBS networks use one-way signaling by letting a control header precede a payload burst and reserve the required switching bandwidth along the routing path. The core OBS network can avoid optical buffering by switching the payload bursts in the optical domain. Therefore, OBS has faster response than circuit switching and lower speed requirement on optical devices than packet switching. Furthermore, OBS scheme naturally fits the IP/MPLS (Multiprotocol Label Switching) framework under the GMPLS (Generalized MPLS) control [Qiao00, SaSh03, VlMo03]. Key design issues for the QoS provision in OBS networks include burst assembly, channel scheduling, signaling, contention resolution, congestion control, and differentiated service.

1.1.2.1 Burst Assembly

In an OBS network, the basic data blocks are DB (Data Burst), which is the aggregation of many packets, and BHP (Burst Header Packet), which is a small packet recording the DB's information. At the edge nodes, multiple packets are assembled into data bursts, while a BHP is formed based on the characteristics of each DB. A BHP precedes the corresponding DB to reserve a transmission path for the following DB. The DB is separated with the BHP by an offset-time. Since the bursts arise from the aggregation of packets and the burst duration affects network performance, the nature of the burst assembly algorithm is an important factor on the delay performance at the edge nodes, as well as the blocking probability performance at the core nodes. A typical approach for burst assembly can adopt either a timer or a threshold for the termination of burst assembly. The timer-based burst assembly algorithm only aggregates packets in a pre-defined time period, e.g., [GeCa00, LiAn04, RoGo04]. On the other hand, the threshold-based algorithm defines the allowable burst size, which can be either the burst length or the number of packets in a burst, e.g., [RoGo04, UeEr04]. When the traffic load is low, the threshold-based method may generate a long waiting delay for the packets in the beginning part of a burst. Although the timer-based method does not have this problem, the packets suffer from a long average delay and large burst length variance under a heavy traffic load. In summary, a single fixed-threshold might cause very poor loss and throughput performance. Consequently, some hybrid algorithms have been proposed to achieve a balance between these aspects, e.g., [VoZh02, YuCh02, RaOv04, TaAj03]. A further improvement is to use adaptive algorithms to respond to the network QoS demand and to adjust the assembly procedure dynamically based on the network environment and performance, e.g., [CaLi02, LiAn03, OhHo02]. All these methods have been examined regarding the network loss or throughput, but their analytical results do not capture the effect of the assembly parameters on congestion control.

Currently, all network applications generate packet traffic with different QoS requirements. On the other hand, a significant feature of OBS networks is to process the traffic at the burst level, each of which contains multiple packets. There are some literatures on traffic characteristics at the burst level that analyze different assembly algorithms and different packet inputs, e.g., [LuZe03, RoGo04, Laev02, KaOk05, DoGa01b, YuLi04]. It is shown that the interarrival process, length distribution and other features of the burst traffic are quite different from the packet input. As far as we know, there is no standard on the burst assembly yet, so the burst traffic may demonstrate more dynamic characteristics when the assembly parameter or the assembly algorithm varies. However, these literatures only focus on the operations at the edge nodes, but do not provide further analysis on possible effects related to the settings and operations at the core nodes.

After a burst becomes ready, a wavelength channel is chosen to transmit this burst and an appropriate time instant is chosen to send out this burst to solve the contention problem when several bursts are waiting for transmission on the same output wavelength channel. Some typical algorithms include OBF (Oldest Burst First), LQF (Longest Queue First) and random transmission algorithm [RaOv04]. The offset time is a key issue affecting channel scheduling. The offset time can be either fixed or variable. Different offset times can be used to support differentiated services [YoQi00], but they also cause a long delay and affect the fairness [ChHa01].

A BHP is usually sent out after a DB becomes ready. It can also be sent out earlier before the corresponding DB is ready in order to reserve an estimated-length of channel for the upcoming DB, e.g., the FRR (Forward Resource Reservation) method in [LiAn03]. In this way, the packet delay at the edge nodes can be reduced, but a complex algorithm is required.

1.1.2.2 Differentiated QoS

Since there is no optical buffer at the core nodes in OBS networks, the contentions among optical bursts always cause traffic loss, and the link bandwidths cannot be fully utilized. Therefore, it is a challenge to provide QoS for optical bursts, such as a required loss rate. There are many works on DiffServ (Differentiated Service) provision [BrJa94, BlBl98] in OBS networks, which includes, but is not limited to, assigning the packets in different positions of a burst and using burst segmentation [VoJu03, TaGu04], setting different offset times for multi-class traffics [YoQi00, FaFe02], using burst cluster transmission [TaKa06], or dropping low QoS bursts intentionally [ChHa01, ZhVo04]. More QoS proposals can be found in other papers [PrPr06, LoTu03, ArSu05, OzVe02]. However, many of these proposals just focus on the burst level performance, while the burst assembly process is not taken into account. For example, many of them use *M/M/k/k* queueing models for the analysis of optical bursts, which assumes an infinite number of incoming sources and does not consider the connectivity of an OBS node. Such model does not reflect the operation of a real node with a limited number of fiber links and wavelength channels. For example, the inefficiency of such a system due to the "streamline" effect has been studied in [PuSh06]. Moreover, although some literatures have talked about the burst assembly process when analyzing the burst performance, there is no further work on relating these two factors closely and distinguishing different types of bursts. Finally, it is impractical to have certain predefined global criteria/parameters in a real network or a dynamic network for the DiffServ provision.

1.1.2.3 Congestion Control

Typical signaling protocols for channel reservation at OBS core nodes are JIT (Just-In-Time) [WeMc00] and JET (Just-Enough-Time) [DoGa01a]. The JIT protocol allocates channels according to the arrival time of a BHP, while the JET protocol allocates a channel for the period of a DB only. The contention problem can be solved by wavelength conversion, FDL (Fiber Delay Line) buffering or re-transmission schemes [ZhVo05a]. To further improve the channel utilization, burst segmentation technique [DeEr02, VoJu02, VoTh03], which allows a burst to be truncated in order that a part of the burst can utilize the channel, has been studied. When employing any of these protocols in a large network, the link loss ratio can be determined by the reduced load fixed point approximation [Kell91, RoVu03]. Consequently, the loss ratio and the throughput of an OBS network are fixed once the offered load remains unchanged. However, this method does not distinguish the loss ratio of individual flows sharing the same link, so it cannot be used to implement congestion control on the flow level.

Congestion control mechanisms allow network systems to detect congestion and throttle the transmission rate to alleviate congestions. Since the loss is unavoidable in an OBS network due to the collision-based nature, a congested node would also have a poor loss rate performance. Using flow control to reduce the loss rate of optical burst flows is one OBS congestion control technique [AkBo05]. Since no buffer is used at core nodes, optical bursts are transmitted from end to end without queueing delay. Traditional congestion control methods are based on the queuing model in the intermediate nodes, so they are not suitable for solving the congestion control problem in a bufferless OBS network. There are two common congestion control methods for bufferless networks: deflection routing or feedback-based rate-adjustment. Deflection routing can be used to reroute some traffic to achieve required load balancing or lower loss in OBS networks, e.g., [ThVo03, KiKi02, HsLi02, OzFa02, TaOh03, ZaVu04], but it disrupts the packet sequence and can cause congestions on other links. As a result, it is uncontrollable and unstable.

The principle of feedback-based congestion control has been considered for the burst level by adjusting the sending-rate of bursts in order to reduce overlapping, e.g., [ZhVo05a, FaZh05, Wang03, DuAb04]. The fairness among bursts with different number of hops can be achieved by discarding bursts proactively before contention occurs [ZhBa04]. Max-min fair bandwidth allocation scheme is also considered to provide fair loss ratio among connections [LiCh05]. Feedback control can be combined with a flow-policing scheme to achieve fair transmission of flows, but it need global information of the network for controlling each flow [KiMu05]. There are more works [ZhVo05b, MaBo04, FaZh04, ShWu06] considering the adjustment of the burst sending-rate based on the feedback information to control the loss ratio of optical burst flows. An efficient way to control a system's performance is to use predictor to estimate its change trend and then dynamically adjust the parameters to modify its operation, e.g., the predictor-adjuster in [Atal06, FlHa00, FlJa93]. However, all these OBS control schemes use simple rate-adjustment mechanisms to modify the offered load only. Another proposal is to send a probe packet for reservation purpose and send the burst only if a successful feedback is received, which causes a long delay [PhCh04]. The GAIMD (General Addictive-Increase Multiple-Decrease) method is also adopted to adjust the sending-rate instead of simple adding/decreasing [JaFa05]. However, in all these proposals, although the loss rate can be reduced through reducing burst sending-rate, the network throughput also becomes lower. These proposals often require some nodes to have some global information of the whole network, which limits the scalability of these methods. Furthermore, as choking packets will increase the delay, reducing transmission-rate is not suitable for some applications, such as multimedia on UDP (User Datagram Protocol). Now there are applications, e.g., VOIP (Voice Over IP), that naturally prefer to adjust packet size in order to maintain a high packet rate and a sufficiently low delay [Good02]. The rate-adjustment method is not applicable in this type of application. Further, none of these proposals has provided a general model to connect the operations at the edge node to the core node, thus affecting the overall performance. Most control proposals have not been used in a dynamic scenario. No work has been found on the transient period, responsiveness or stability of feedback control yet. There are many works in these areas on TCP (Transport Control Protocol) or AQM (Active Queue Management) control, but there does not appear to be comparable work on OBS operations.

1.1.3 Optical Time-Division-Multiplexing

In an OTDM network, the time axis is divided into fixed-length time slots and the slots on different wavelength channels are time-synchronized. Each slot has a control packet that precedes the payload slot and reserves necessary wavelength channels along the routing patch and switching nodes. The control packets are usually transmitted on a separated control wavelength channel, while all other channels are used for the payload slots. The major challenge is the TSA (Time Slot Assignment) or scheduling problem, i.e., a TDM node needs to allocate the channel resources properly to achieve required performance when multiple slots contend for a limit number of output channels. An OTDM network is suitable for circuit switching in nature since the current optical technology is not able to process the channel scheduling promptly to retain the boundary of small optical slots. Therefore, the TSA problem is usually solved by traffic grooming through the RWA (Routing and Wavelength Assignment) procedure, e.g., [RaSi95, ZaJu00, ChWo04]. Typical RWA problem is NP-hard, which is not the topic of this thesis. To improve the efficiency of channel utilization and reduce delay, an OTDM network can work in the OBS mode, which is similar to the TSOBS (Time-Slotted-Optical-Burst-Switching) [RaTu03, LiXi05] except that the length of each payload message is fixed.

1.1.3.1 Time Slot Assignment

In an OTDM node, the reservation-based TSA is a practical choice to eliminate the requirement of optical buffers. In this case, the TSA problem is equal to the scheduling in a buffered switch. All slots are buffered in electronic domain at the edge nodes waiting for the TSA results, and the payload slots arriving at the same slot period can be processed simultaneously. To improve the efficiency of scheduling, the slots can be organized into fixed-length frames; then we can run the frame-based scheduling algorithm to allocate bandwidth among more slots at the same time. Considering the placement of buffers, we may classify the nodes into input-buffered, output-buffered, shared-buffered, intermediate-buffered and other combinations [KaHl87]. In regard to efficiency and implementation, the VOQ (Virtual-Output-Queued) architecture is believed to be most useful, which has received a lot of attention. Typical scheduling algorithms include *i*-SLIP, maximum-size-weighting, maximum-weight-weighting and so on [Mcke99, McAn96, McMe99, FaKa04]. A problem of these algorithms is that the complexity is very high and they are only suitable for a symmetrical architecture. Furthermore, the traffic matrix must satisfy a certain format for these algorithms, which might not always be feasible.

1.1.3.2 Clos Switches

Different types of interconnection topologies, such as Banyan, ShuffleOut, or Knockout [AlAl95, DeGi94, YeHl87], have been studied for switch architecture. Many TDM applications have adopted three-stage Clos switches [Clos53] as a useful architecture in computer communications because this architecture can take advantage of small switches to build a large-size switch, while keeping the number of switching elements to a moderate number. In a typical Clos switch, small-scale switching components are placed in three stages to form a larger-capacity switch with a reduced number of cross-points compared to a single-stage crossbar-switch. The switching components in the same stage are uniform. The numbers of switching components in the first and third stages are equal, and there is a single-channel connection between two connected small-switches. The scheduling and non-blocking issues in this type of symmetrical Clos switches have received a lot of attentions, e.g., [ChJi03, PuHa02, Hwan03, HwLi03].

There are many extensions to the basic Clos switches. For example, considering more than one link between two small-switches, the multiconnection Clos switches are studied in [Kiba95]. The multirate Clos switches are also studied when the traffic demand is neither normalized nor equal in each port [TuMe03, LiNg98, KaLi02, LiCh96]. This concept is very useful for the ATM (Asynchronous Transfer Mode) where variable small-bandwidth requirements are multiplexed to share a link. A more general Clos network is proposed to model more than one connection path between the source and the destination using asymmetrical multiconnections [Tham98, VaCh92, VaCh93]. Both network flow model and edge-coloring approach have been investigated to solve the routing and scheduling problem in this type of network, e.g., [Tham98, VaCh93]. The slots passing through a switch are modeled by a flow model. By assigning a number of flows between adjacent nodes according to the connection structure and traffic constraints, we can build up the switching matrix by retrieving the flow assignment results. However, these scheduling algorithms only consider a single switching node and assume the traffic matrix satisfy the non-blocking conditions, but are not suitable for widely-separated nodes. Furthermore, the high level of computing complexity is another obstacle in practical applications.

The above proposals on Clos switches are often based on electronic switching techniques which have flexible buffering and speedup capability to support the required non-blocking conditions or scheduling operations. With the increasing deployment of optical components, Clos switches are also considered in WDM networks, e.g., [OkYa03, ChDe03, NeRa01], but such systems lack feasibility and flexibility for real applications. The reason is that optical buffering and wavelength conversion are still immature in current applications [ElMo00b, MaMo93]. The fabrication of a large-size optical switching device is another challenge. Therefore, it is difficult to apply the above proposals to a general optical scenario.

1.2 Motivations

As reviewed above, the OBS mechanism is very flexible, scalable and practical in using enabling optical technologies. Therefore, we would like to first explore its potential for optical networking.

We have shown that existing OBS research often emphasizes a separate issue without considering other parameters when analyzing the QoS issues. The common approach of network performance analysis often assumes the traffic to be in the burst domain, while ignoring the burst assembly process. On the other hand, most researches on burst assembly often focus on the operations at the edge nodes only, but do not consider the possible effects on the settings and operations at the core nodes. Therefore, the inefficiency of the *M/M/k/k* model requires a more accurate method to analyze the performance of OBS nodes. It is not practical to achieve DiffServ provision by utilizing some global information in advance in a heterogeneous network. The inefficiency of these areas motivates us to relate the parameters for burst assembly in the edge nodes to the loss performance in the core nodes in order to allow choosing appropriate parameters and operations to provide differentiated loss ratio among burst flows.

As reviewed above, existing control algorithms/schemes leave much to be improved for a real OBS network as there are many issues in complexity, transient response and stability. None of these proposals has provided a general model to connect the operations at the edge node to the core node, thus affecting the overall loss performance. Since the network environment (especially OBS) is very dynamic, it would be desirable to develop adaptive schemes for QoS provisioning. Although there have been many works on TCP or AQM control, there does not appear to be comparable work on OBS operations.

The star topology is very efficient to connect the edge nodes in order to reduce the requirements on the switching speed and signal processing on optical components and to improve the robustness. To improve its reliability, we would like to consider its variation as our preferred choice of a possible new topology for future optical networks. Our next challenge is to implement this topology from existing infrastructure and to choose a suitable switching scheme for this architecture. The new topology should be able to provide a general framework which accounts for the geographical distribution of nodes, supports flexible switching methods and incorporates both quality of service and feasibility of technology.

There are also some interesting issues in the design of both the physical and logical network topologies. For one, most algorithms are only suitable for symmetrical node architecture because of their high complexity. Some scheduling algorithms only consider a single switching node and assume the traffic matrix satisfies the non-blocking conditions, but these algorithms are not suitable for widely-separated nodes. The Clos architecture as reviewed appears to have an advantage of building a large-size switch from smaller ones with arbitrary connection patterns, so we are interested in exploring the potential of this architecture to address all the above issues in the network level.

Finally, we are motivated to study OTDM an alternative solution as it is the scheme adopted by our AAPN research project for highly-dynamic high-capacity photonic networks. The major challenges in an OTDM network are solving the routing, time slot assignment and wavelength assignment problems. There is a need to design a low time-complexity solution to take advantage of the enabling optical technology.

1.3 Objectives

We are interested in general to study feasible topologies and efficient switching schemes to deliver messages fast and efficiently in order to achieve required QoS and fully exploit the resources in the NGN. Specifically, we want to do the following:

1) To investigate the difference of two network models for meshed OBS networks, one being based on the packet level and another one based on the burst level, in order to provide the DiffServ.
2) To design a simple flow control scheme which is capable of controlling the loss rate of a specific burst flow without sacrificing the overall throughput in meshed OBS networks.
3) To build a generic controller, a system with simplicity, scalability and efficiency for meshed OBS networks. We want to provide a flexible model for controlling the loss rate of optical burst flows using edge-based predictor-adjuster model.
4) To study the scheduling algorithm of a general Clos switch in all-photonic domain.
5) To propose and study an all-photonic architecture for the NGN that will make use of the OBS and the OTDM techniques.

1.4 Methodologies and Approaches

We would like to investigate the difference in performance between the burst-based and packet-based model. We shall first use the probability and queueing theories to analyze the burst assembly procedure in an OBS network in order to build a mathematical model on the burst flow properties. Unlike other burst-based analysis, we shall emphasize on a packet-based system. After studying the inefficiency of the traditional burst-based scheme, we shall evaluate assembly algorithms under different signaling protocols at a core node. Through comparison, we want to relate the assembly parameters to the loss performance of different burst flows. We would like to provide a possible application on DiffServ provisioning.

We then propose several controller prototypes and evaluate the parameter combinations. We combine the burst aggregation strategies, the queueing analytical methods and the congestion control schemes, in order to set up a framework to choose appropriate models and operations to control the flow loss and delay performance in multiple-hop OBS networks. Our design covers major burst assembly algorithms and signaling protocols. We develop QoS schemes through adaptive burst assembly, which is capable of controlling the loss rate of a burst flow without affecting the throughput of the whole network. We are interested in the relationship of different parameters at edge and core nodes. We want to implement our congestion control using the predictor-adjuster algorithm in order to have a good scalability. This scheme will be evaluated for its capability to avoid the computing-complexity of existing OBS control schemes.

We also revisit the star topology as an implementation of future all-photonic networks by considering the All Photonic Overlaid-Star Network (APOSN) variation. We shall consider active stars to achieve faster and more flexible switching than passive star networks. We shall investigate how this topology would support the implementation of both OTDM and OBS methods without any optical buffers or wavelength conversion by using some universal node architecture with microsecond switching capability. We would like to provide an asymmetrical multiconnection Clos network model to implement resource scheduling issues. We want to find out the capability of this model in handling the routing, wavelength assignment and contention resolution problems. Under this model, we adopt parallel processing schemes to accelerate wavelength/time slot assignment and analyze the time-wavelength assignment problem when OTDM is adopted. When OBS is applied in this model, we investigate the influence of signaling protocols and major system parameters. We also apply the DiffServ and flow control scheme we have designed in the APOSN. Furthermore, our work has been applied in a practical network example being implemented, i.e., the AAPN project.

Throughout the study, we do not want to limit our proposed mechanisms for some limited traffic types only, so our analysis will focus on finding out the influences of different parameters by approximations and comparisons, which is good enough as a reference for our design.

We shall use mathematical analysis to support our algorithms and run software simulations to verify our proposals. Since there is no specific software suitable for the simulations of optical networks being studied in this thesis, we have developed some simulation models by using OPNET Modeler [OPEN06]. We have chosen OPNET Modeler because it provides a user-friendly GUI (Graphical User Interface) environment for network modeling and simulation, along with a comprehensive library of detailed protocols, applications and vendor devices. Its discrete event simulation kernel allows one to design and study protocols, devices and behaviors of communication networks with good flexibility and scalability. We make use of the hierarchical models of OPNET to simulate our APOSN network and the OBS and the OTDM operations. We also develop new process models or upgrade existing models to implement our feedback control.

When we run simulations, we examine a wide range of typical parameters and their combinations, and five different seeds are adopted for each case in order to obtain an average result. Each simulation run is based on one million bursts or slots, and 95% confidence intervals have been obtained. A typical simulation run takes about 30 to 100 minutes depending on the algorithm being studied. Our performance evaluation covers different scenarios in a network. Representative results are presented in each section, while additional results can be found in the appendices. All simulations are executed under Windows XP system and Microsoft Visual C++ 6 environment.

1.5 Contributions

The following are the contributions on several QoS issues of optical networking.

1) The relationship of burst assembly parameters and the loss ratio of burst flows in meshed OBS networks: we investigated this relationship by analyzing the loss performance as a function of some typical node parameter settings, and examining the difference between the packet-based or burst-based models. Our study covers major assembly algorithms and signaling protocols and their variations.

2) An edge-based DiffServ scheme for OBS networks: under this scheme, an OBS system is always fully utilized without affecting the traffic load injected to the network. Our scheme can be applied to different burst assembly algorithms. The proposed scheme includes different steps throughout the traffic transmission process in an OBS network. Thus, our proposal is a practical, simple and efficient relative QoS solution.

3) Several loss predictors and assembly adjusters for the implementation of a traffic controller: our generic control algorithm can be implemented under both static and dynamic network environments and using typical signaling protocol. A flow can be controlled without knowledge of other flows in a network. Unlike conventional congestion control methods, our method does not affect the overall throughput of the OBS network.

4) A single-hop APOSN topology: this topology can be applied to any geographical network and does not require any optical buffers or wavelength converters throughout the network, which are considered to be impractical in current applications. This proposal has also been part of the AAPN research project.

5) An asymmetrical multiconnection Clos architecture: this framework is adopted to support the APOSN network and solve the resource sharing problem. Both OTDM and OBS switching schemes are applied in this topology under similar configurations. Effects of major parameters on typical performance criteria are investigated, which enables us to optimize the system settings and integrate system design.

To the best of our knowledge, there is no existing work that is able to fulfill the same QoS tasks in meshed OBS networks through edge-based parameters and operations only, nor work in the optical networking area using agile overlaid-star as the basic topology for core networks and providing an efficient resource sharing scheme based on agile cores at the same time.

1.6 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 describes the network topology and node architecture for our research. Chapter 3 proposes several algorithms to provide differentiated loss among different classes of optical bursts. Chapter 4 designs a feedback-based control mechanism for optical burst flows in a meshed optical network based on a simple adjustment scheme of the bursts assembly process. Chapter 5 proposes a generic control scheme for optical burst switching networks. Chapter 6 presents an APOSN topology and discusses the OBS and OTDM operations in it. We study the system performance regarding major parameters. Chapter 7 summarizes some design guidelines based on the proposals in this thesis. Chapter 8 concludes our research work. Some supplemental materials are provided in the appendices.

1.7 Publications

The followings are the related publications.

[1] Mushi Jin and Oliver Yang, "Analysis and Design of Edge-Based Controllers Supporting Absolute QoS for Optical Bursts", to appear in *Journal of Optical Networking*.

[2] Mushi Jin and Oliver Yang, "APOSN: Operation, Modeling and Performance Evaluation", to appear in *Computer Networks*.

[3] Mushi Jin and Oliver Yang, "Provision of Differentiated Performance in Optical Burst Switching Networks Based on Burst Assembly Processes", submitted to *Computer Communications*, Sept 2006, under revision.

[4] Mushi Jin and Oliver Yang, "Fast Scheduling of Asymmetrical Multi-connection Clos Switches and Its Application in AAPN Networks", submitted to *IET Communications*, Jan 2007.

[5] Mushi Jin and Oliver Yang, "Enhancing Differentiated Loss Performance in Optical Burst Switching Networks", submitted to *IEEE/Createnet Broadnets 2007 (Workshop on Optical Burst Switching*), Apr 2007.

[7] Mushi Jin and Oliver Yang, "Flow Control in Optical Burst Switching Networks by Using Adaptive Burst Assembly", submitted to *IEEE JSAC*, May 2006.

[7] Mushi Jin and Oliver Yang, "An Improved Analysis on the Loss Performance of Optical Bursts", to appear in *IEEE AINA 2007*, May 21-23, 2007, Niagara Falls, Canada.

[8] Mushi Jin and Oliver Yang, "A Traffic Shaping Heuristic for Congestion Control in Optical Burst-Switched Networks", *Proc. IEEE LCN 2006*, Nov 14-16, Tampa, FL, USA, pp.529-530.

[9] Mushi Jin and Oliver Yang, "On Controlling the Loss Rate in Optical Burst Switching Networks", *Proc. IEEE/Createnet Broadnets 2006 (Workshop on Optical Burst Switching)*, Oct 02-04, 2006, CA, U.S.A.

[10] Akbar Ghaffar Pour Rahbar, Mushi Jin, Choudhury A. Al Sayeed and Oliver W.W. Yang, "APOSN Performance Under the MAN and WAN Environments", *Proc. IEEE CCECE 2006*, May 07-10, 2006, Ottawa, Ontario, Canada, pp.1211-1214.

[11] Mushi Jin and Oliver Yang, "A TDM Solution for All-Photonic Overlaid-Star Networks", *Proc. CISS 2006, Session FP-06: Optical Communications and WDM Networks*, Mar 22-24, 2006, Princeton, New Jersey, USA.

[12] Mushi Jin and Oliver Yang, "An integrated architecture enabling different resource sharing schemes for AAPN networks", *Proc. Photonics North, Session: Photonic Applications in Devices and Communication Systems*, Sept 12-14, 2005, Toronto, Ontario, Canada.

[13] Mushi Jin, Oliver Yang, Y. Zhang, A. G. P. Rahbar and W. Yang, "Applying Time-Division-Multiplexing in Star-based Optical Networks", *Proc. IEEE CCECE 2005*, May 1-4, 2005, Saskatoon, Saskatchewan, Canada, pp.1141-1144.

[14] Mushi Jin and Oliver Yang, "Parameter Choices and Their Effects on the Performance of the OBS Networks", *Proc. APOC 2004*, Nov 7-11, 2004, Beijing, China, pp.672-681.

CHAPTER 2
NETWORK MODELS, OPERATIONS AND ASSUMPTIONS

In this chapter we provide the network model, node architecture and assumptions on the optical networks being studied in this thesis. We outline the general principle of message transmission. Architecture models for edge and core nodes are also presented which are the foundations of further studies on switching schemes.

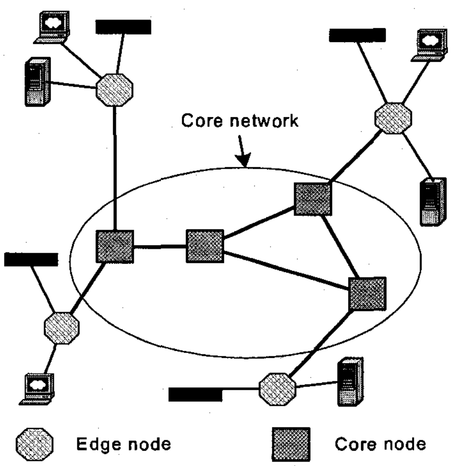
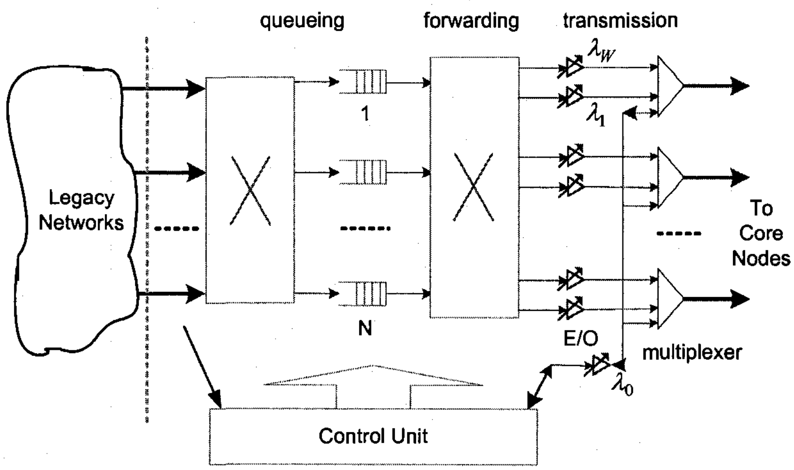


Fig. 2.1. Optical network architecture

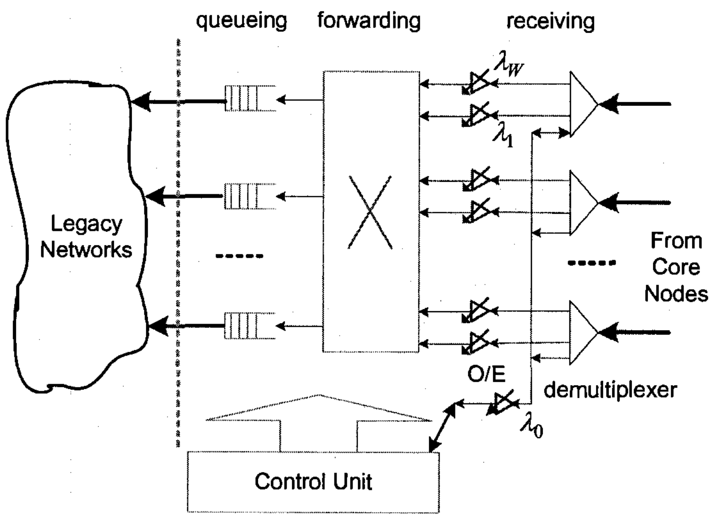
2.1 Network Model

Fig. 2.1 is a general network architecture we consider in our research. This architecture consists of N edge nodes which are responsible for packet aggregation and dissemination between legacy packet networks and optical core networks, and M core switching nodes which are responsible for header regeneration, message switching and other tasks. The core nodes are interconnected to form a core network, which transmits the payload message in pure optical domain. The network has J fiber links. The routing path from a source edge to a destination edge is a subset of $(1,2,...,J)$. Each fiber contains several wavelength channels, and each channel is equipped with a transmitter at the transmission side and a receiver at the receiving side. One wavelength channel is dedicated for the control header packet and feedback packet transmission between two adjacent nodes and other channels are used for payload messages.

When packet messages from different legacy access networks, such as ATM, Ethernet or SONET, arrive at an edge node (called the source edge node), they are first buffered electronically. Those destined to the same destination are combined to form a burst or a slot. A control packet is formed by recording the information of the payload burst or slot, e.g., the length, the starting instant of the burst/slot, the source and destination information. Then this header packet is sent out optically earlier than the payload, transmitted on a separated wavelength channel from the payload message. The control header packets will be transformed into electronic domain at each core node. When feedback flow control is enabled, a feedback control packet is used, which is processed in electronic domain at core nodes too.



(a) Transmission side



(b) Receiving side

Fig. 2.2. Architecture of an edge node

2.2 Node Model

The general architecture for the edge nodes and core nodes in the Fig. 2.1 network is described as follows.

2.2.1 Edge Node

Referred to Fig. 2.2a, the general architecture of the transmission side of an edge node constitutes three stages: queueing, forwarding and transmission. The queueing stage aggregates the traffic trying to enter the core networks and sorts them based on their destinations. This stage consists of N VOQ buffers, each of which stores the traffic to one of the N destination addresses. The forwarding stage removes a message from its electronic buffer and submits it to an electronics/optical converter. The transmission stage performs the electronics/optical conversion and sends out the messages as optical signals. The stages before and after the VOQs can be implemented by fast electronic switches. All operations of these stages are under the control of an electronic control unit which is also responsible for time synchronization and control packet generation.

Each edge node needs to receive messages from core nodes as well. The receiving side is implemented by an architecture similar to the transmission side, but messages are transmitted in the reverse direction as shown in Fig. 2.2b. After an edge receives an optical message from the core node, it performs optics/electronics conversion first and then delivers it to appropriate destinations. We assume the transmission parts are capable of forwarding any ready message in VOQs to the transmission stage, and receiving parts capable of forwarding all newly-arrived optical messages to the queueing stage at any time. Since the operations are equivalent, our work only focuses on analyzing sending messages from the transmission side throughout the core, while the message processing at the receiving side is not covered. In both transmission and receiving sides, one wavelength channel (λ_0 in our diagrams) is used for the control message. Each fiber can have up to W wavelength channels ($\lambda_1 \dots \lambda_w$ in our diagrams) for payload transmission, while the actual number depends on the traffic demand and the topology design.

2.2.2 Core Node

Fig. 2.3 depicts the basic architecture of a core node without wavelength conversion. The major part is the W switching fabrics, each of which switches messages of the same wavelength. Payload messages are demultiplexed into switching fabrics, while control packets are sent to the control unit where the optics-electronics conversion and electronic processing are performed. Using the information included in the control packets, the control unit modifies the status of the switching fabric at the beginning epoch of each message. A new control packet is also generated after updating the routing information. Then new control packets and switched payload messages are combined by multiplexers and sent out through the output ports. The switching fabric requires a non-blocking architecture using any fast switching components. For examples, acoustic-optic switch devices have a switching range of micro-seconds or the faster LiNbO3 switches [Kart02] have a range of nano-seconds. When wavelength conversion is allowed, some converters are added at the interface part and one large switching fabric is enough for a core node.

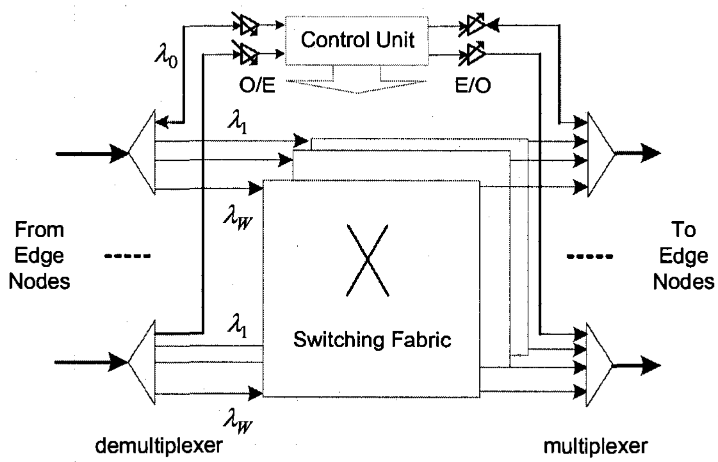


Fig. 2.3. Architecture of a core node

2.3 Network Operations

OBS is the major bandwidth sharing technique in this research. At the edge nodes, the burst assembly process is affected by two parameters, the timer and the threshold. Both parameters are used singly or jointly, depending on the assembly process adopted. When the adopted parameter reaches its maximum value, the assembly procedure is terminated and the newly-arrived packets are assigned to a new burst. The DBs in the same buffer are sent out in FCFS (First-Come-First-Service) order. The BHP precedes the corresponding DB by a fixed offset time (T_{off}) to reserve a transmission path for the following DB. After a burst becomes ready, a wavelength channel among all possible DB channels is chosen randomly with an equal probability. If the chosen channel becomes free in less than the offset time, the BHP packet is sent out immediately and then the DB after an offset time. Otherwise, the BHP is sent out with an offset time before the channel becomes free, and the DB is sent out immediately after the channel becomes available.

At the core nodes, the BHPs are processed in FCFS order and BHPs have no loss (it is normal is an OBS network). The wavelength channel that becomes free at the earliest time among the output link is assigned to the newly arrived burst. Our network can choose either one of these popular signaling protocols for contention resolution:

1) JIT protocol: in this protocol, a DB is dropped if no channel is free at the time the corresponding BHP arrives.
2) JET protocol: in this protocol, a DB is dropped if the channel is busy at the DB's arrival time.
3) JETseg (JET with burst segmentation) protocol: in this protocol, the overlapping part of a burst is dropped, while its remaining part is still transmitted.

The JETseg protocol has two variations: JETtailSeg (JET with burst segmentation and tail dropping) and JETheadSeg (JET with burst segmentation and head dropping). The first one drops the tail part of the existing DB from the instance of the new DB arrival, while the second drops the head part of the newly-arrived DB. The segmentation technique is more efficient than the original OBS scheme, but requires more processing capability at the nodes.

At the core nodes, due to the limitation of current optical technology, the switching time, which is the time needed to change the status of an optical switch, is not a negligible parameter. When a core switch assigns a channel to the traffic, the channel does not become available immediately until a delay interval that is equal to the switching time. If an optical message arrives before the switch becomes ready, it has to be dropped since we assume no optical buffer in the core nodes. On the other hand, the processing of one header packet will delay the processing of other header packets that arrive during its processing interval and require the same channel for their payload messages. If a header packet has not been processed until its data payload has arrived, the corresponding data payload has to be dropped.

We use fixed source routing in our design. The DBs from a source edge node to a destination edge node (formed in the same buffer) comprise a flow whose routing path includes a sequence of links and core nodes. Assuming wavelength conversion capability, a flow on each link can be transmitted simultaneously by different wavelengths. Otherwise, a flow can only choose the same wavelength on different links. Each link can carry different flows at the same time. Note that the source and destination of a flow are always edge nodes.

Our OTDM can be seen as a special case of OBS. The general operation scheme is the same as the OBS. The major differences include two aspects. One is that all slots have a same length during the assembly process, and the other one is that the control unit needs to synchronize the transmission of all slots in different channels. We will use OTDM in the APOSN as required by the AAPN implementation, where we do some comparisons and trade-off studies.

To summarize, when a message arrives at a source edge node, it is buffered electronically. A routing path is then set up using some signaling information (which is contained in the header packets used by the OBS or the OTDM operations) before the payload is sent out. The payload remains at optical domain until it is received at a destination edge node where it is converted back to electronic domain for delivering to the legacy network connected to the destination edge node.

2.4 Simulation Model

We use OPNET Modeler to implement the network and node models as described in Sections 2.1 and 2.2. OPNET simulation captures all these naturally in a hierarchical manner. At the highest level, we have the network models in which different nodes are connected by links and suitable interfaces. Below the network model are the node models that represent different node elements as described previously. The detailed operations (as in Section 2.3) are captured at the lowest level by the process models. Specific operations of protocols, queues or algorithms are captured through finite state machines. Each state in a process model can utilize customized C/C++ code or OPENT library functions for functioning programming.

Finally, a link model is developed to model the transmission path between two adjacent nodes. Different packets and messages can be implemented in the packet model and allowed to "flow" through the implemented simulation network. An example of the network, node and process models can be found in Appendix A.

2.4.1 Traffic Model

The traffic model for our study is based on the packet level. The packet arrival is either a Poisson process or a Pareto process. The first one has been a general model for the network modeling for a long time [KaMo04], while the later one is often used to model the self-similarity of Internet traffic [CrBe97, Gord95] nowadays. Considering the popularity of Internet applications, we adopt the packet size distribution from the measured Internet traffic by CAIDA (Cooperative Association for Internet Data Analysis) [CAID06]. That is, the packet size shall have a distribution according to the measurements by which 46% of the packets are 40 bytes long, 18% of the packets are 552 bytes long, 18% of the packets are 576 bytes long, and 18% of the packets are 1500 bytes long. This distribution makes our design to be more practical as it captures the realistic Internet environment. For comparison, we shall also test other distributions. Since our OBS and OTDM schemes aggregate many packets into bursts or slots, the difference of these distributions does not affect the designed algorithms as shown later.

We study a network in either a static or a dynamic status when our controllers are studied. In a static network, the input loads remain unchanged and the loss rate of a flow is stable. In a dynamic network, the inputs vary and the loss performance of a flow fluctuates as well.

2.5 Performance Measures

Several performance measures are used in this thesis: the average packet delay, the loss ratio, the utilization, the throughput and so on. These measurements are calculated separately for a class of traffic, for a flow, for the whole network or for all traffics in total. Their definitions are as follows.

A packet experiences three types of delay at an edge node: the assembly delay, the queueing delay and the transmission delay. From the instant that a burst or slot receives its first packet to the instant that this burst or slot stops to accept any more packets, all packets belonging to this burst or slot are buffered in the queue. This buffering time is defined as the assembly delay imposed on packets. A ready burst or slot has to wait for an offset time before being sent out. Moreover, if the required channel is occupied by some other message when a burst or slot is ready for transmission, it has to wait longer in the buffer. These two factors contribute to the queueing delay. The transmission delay is the time duration from transmitting the first bit to the last bit of the burst or slot, which is decided by the length of the payload burst or slot. The system delay is defined to be the interval between the arrival time of a packet at an edge node until the successful transmission (from the edge node) of the slot or the burst it belongs to, which is the total time in an edge node and is the sum of these three delays. We also consider the ETC (edge-to-core) delay in an APOSN network, i.e., the interval between the arrival time of a packet at an edge node until when the slot or the burst it belongs to arrives the core node successfully, which includes both the system delay and the propagation delay.

The loss ratio is the total number of dropped packet bits to the total number of arrived packet bits. The throughput is the aggregated transmitted capacity on all payload wavelength channels. The channel utilization is the transmitted capacity over the total channel capacity in all payload channels.

When we evaluate the control process, we consider the convergence, i.e., the length of the transient time period for the loss rate to reach the required range, and the stability, i.e., the variation of the loss rate along the timeline. A shorter transient period means a faster convergence, and less loss variation indicates better stability.

2.6 Assumptions

The following general assumptions are used for the remainder of this thesis unless otherwise specified.

1) No optical buffering for data in the optical network: this is due to the constraint of existing optical technology. The payload messages are kept in optical domain throughout the transmission, while the control packets are transmitted in optical domain and converted to electronic domain at each node.

2) The header processing time or core switching times are negligible: this is because these times are much smaller than the payload length, and the controlling part should not be a bottleneck in a realistic high-speed network. It also simplifies the analysis of a specific QoS issue, even though we can show by simulations that this assumption can be relaxed.

3) Control packets have no loss: this is because in a practical network, the control packets are considered to be critical and their transmission should not be hindered by any bottleneck, and enough resources should have been allocated for their transmission during network design process to start with.

4) The packet arrival process of each traffic stream is independent of each other: this is generally the case because the packets are generated by different users and applications independently.

5) All packets have an *i.i.d.* (independent identically distributed) length distribution as commonly used, and independent of the arrival process.

6) The arriving traffic load to an edge is normalized with respect to the capacity (measured in *bps*) of all wavelength channels joining that edge and a core node: this normalization will allow us to make meaningful comparison among different network configurations.

7) All channels carry the same traffic load and have the same bandwidth: this assumption is reasonable in a core network being studied in this thesis because it allows us to reflect a

real network's capability of balancing its traffic in order to achieve fair and efficient resource utilization.

8) The probability that a packet is destined to one destination edge node is proportional to the number of wavelength channels carrying payload traffic from the core nodes to this destination. This is a common approach to balance the network traffic distribution in order to utilize the resource efficiently.

9) Fairness is not considered: this is because we focus on schemes that can guarantee the performance of critical applications. Therefore, not all users are treated equally. On the other hand, we have also demonstrated the impact on other users is minimal.

CHAPTER 3
DIFFSERV PROVISION IN OBS NETWORKS

We start our work with the study on the loss performance in meshed OBS networks. This chapter intends to show that the performances of optical bursts are dramatically different when considering burst assembly process at the edge nodes and the connection pattern of the core nodes. We first develop some analytical equations that can model the loss performance more accurately than traditional methods. We then propose a scheme to support differentiated loss performances for optical burst flows in JETseg networks.

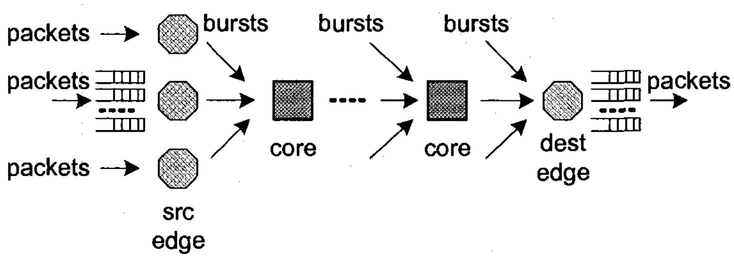


Fig. 3.1. Packet-based view of the transmission scheme

3.1 System Description and Model

The transmission of a burst flow can be represented by Fig. 3.1, which includes a series of links and nodes from the source edge node to the destination edge node. The traffic arrives at the edge nodes in the packet domain, but becomes bursts when arriving at the core nodes and the intermediate links. Define a burst-traffic class as bursts with the same priority assembled in the same buffer at an edge node. Since an optical burst consists of multiple packets and the packets are the basic unit in real applications, Fig. 3.1 depicts the more accurate system model that includes both types of traffics for a complete transmission (this is different from the burst-based model to be discussed in Section 3.2.3 and Appendix B).

We consider two signaling protocols at the core nodes in this chapter, i.e., JET or JETseg. Each signaling scheme has a different loss rate which is the major performance concern in our study. The loss rate in JET can be measured at the burst, packet or bit level, but only at the packet or bit level in JETseg. We will use burst loss in JET and bit loss in JETseg in our later analysis. Since we have assumed independent packet arrival and an *i.i.d.* packet length distribution, the loss measurement under other levels is the same in each type of network. When burst assembly process is considered, the system delay of incoming packets at the edge nodes is another important parameter in our study.

3.2 Loss Analysis

To conduct loss analysis, we first use the accurate packet-based OBS model to carry out queueing analysis on the first core node along the routing path in Fig. 3.1. The loss rate is analyzed as a function of some important parameters, and then compared to the burst-based model.

Let ρ be the total normalized offered load and W_c be the number of output channels. Let N_e be the number of edge nodes and W_e be the number of output wavelength channels at each edge node. For analytical simplicity, we assume the input traffic load to be uniformly distributed to all edge nodes, and the buffers at the source edges are large enough that they would never overflow (this assumption will be relaxed in later simulations). The input packet traffic belongs to one of N_c classes with equal probabilities, and packets of each class are assembled in a different buffer. The assembly process in Buffer#*i* is determined by the allowable burst length of L_i bits ($i = 0,...,N_c - 1$) and the offset time is T_{off} second. Let C (bps) be the bandwidth of a wavelength channel, then the average burst service time becomes $\frac{1}{\mu_i} = \frac{L_i}{C}$ second for Class#*i* bursts. Let a generated burst choose the output channels of an edge node randomly. The bursts of same class are treated equally by the core node when assigning the wavelength channels. Therefore, the average arrival rate for Class#*i* bursts from each input channel to an edge node is $\pi_i = \frac{\rho \cdot C}{N_c \cdot N_e \cdot L_i}$ bursts/second. Since the burst assembly at each buffer is implemented independently, it is reasonable to assume the burst arrival to the core node follows a Poisson process for a large W_e, an average arrival rate of $\lambda = \frac{N_e \cdot \sum_{i=1}^{N_c} \pi_i}{W_e}$ burst/second per input channel and an average service time of $\frac{1}{\mu} = \frac{\sum_{i=1}^{N_c} 1/\mu_i}{N_c}$ second per burst. Obviously, the propagation delay from the edges to the core does not affect the characteristics of the input bursts at a core node. If the number of input channels at the core node is fewer than the number of output channels, $N_e W_e \le W_c$, there is no loss at all. In other word, a loss only occurs when $N_e W_e > W_c$. We analyze two protocols: the JET or the JETseg.

3.2.1 JET Protocol

Since there is no optical buffer at a core node for the optical bursts and there are only a limited number of input channels, the behavior of a JET core node can be estimated by using an $M/M/W_c/W_c/N_eW_e$ queueing model [Klei75], in which there are N_eW_e customers, W_c servers and W_c storage spaces. In this queueing system, the probability of having k simultaneous bursts is

$$p_k = \begin{cases} \dfrac{\binom{N_eW_e}{k}\cdot\left(\dfrac{\lambda}{\mu}\right)^k}{\sum_{i=0}^{W_c}\binom{N_eW_e}{i}\cdot\left(\dfrac{\lambda}{\mu}\right)^i}, & 0 \le k \le W_c < N_e \cdot W \\ 0, & otherwise \end{cases} . \tag{3.1}$$

Compared to the traditional *M/M/k/k* model, our equation has taken into account the connectivity of the core node and the properties of optical bursts caused by the assembly process. Since there are only W_c servers in our queueing system, the loss rate is approximately equal to the probability that all W_c servers are busy, i.e.,

$$p_{JET} = p_{W_c} = \begin{cases} \dfrac{\binom{N_eW_e}{W_c}\cdot\left(\dfrac{\lambda}{\mu}\right)^{W_c}}{\sum_{i=0}^{W_c}\binom{N_eW_e}{i}\cdot\left(\dfrac{\lambda}{\mu}\right)^i}, & N_e \cdot W_e \ge W_c \\ 0, & N_e \cdot W_e < W_c \end{cases} . \tag{3.2}$$

3.2.2 JETseg Protocol

When the JETseg protocol is adopted, we take the number of output channels of the core node to be infinite in order to approximate the loss performance. To facilitate the loss approximation, we extend the OCBS analysis in [NeRo02], by making W_c of those channels corresponding to the truly-existing channels while others are virtual ones. When a burst arrives at the core node, it is assigned to a real channel first. If all real channels are busy, then it is assigned to a virtual channel. Once a burst in a real channel has finished its transmission, we let the remaining part of another burst in the virtual channels (if exists) to take over that real channel, thus effectively representing the burst segmentation process. Overall, we can now capture the utilization of all real channels when the bursts (or partial bursts) can transmit successful, and at the same time capture the loss performance of the bursts (or partial bursts).

Using the above method, the core node can then be approximated as an $M/M/\infty//N_eW_e$ queue, in which there are N_eW_e customers and infinite servers. Each server corresponds to a wavelength channel, either a real or a virtual one. The probability that k $(k \le N_e \cdot W_e)$ bursts appear at the input of the core node at the same instance is equal to the probability of having k busy servers in such queueing system [Klei75].

$$p_k = \begin{cases} \dfrac{\binom{N_eW_e}{k} \cdot \left(\dfrac{\lambda}{\mu}\right)^k}{(1+\lambda/\mu)^{N_e \cdot W_e}}, & 0 \le k \le N_e \cdot W_e \\ 0, & otherwise \end{cases} . \tag{3.3}$$

The loss will occur when the number of the busy servers is more than the number of the real servers. Then the loss rate of such a JETseg OBS system can be approximated by

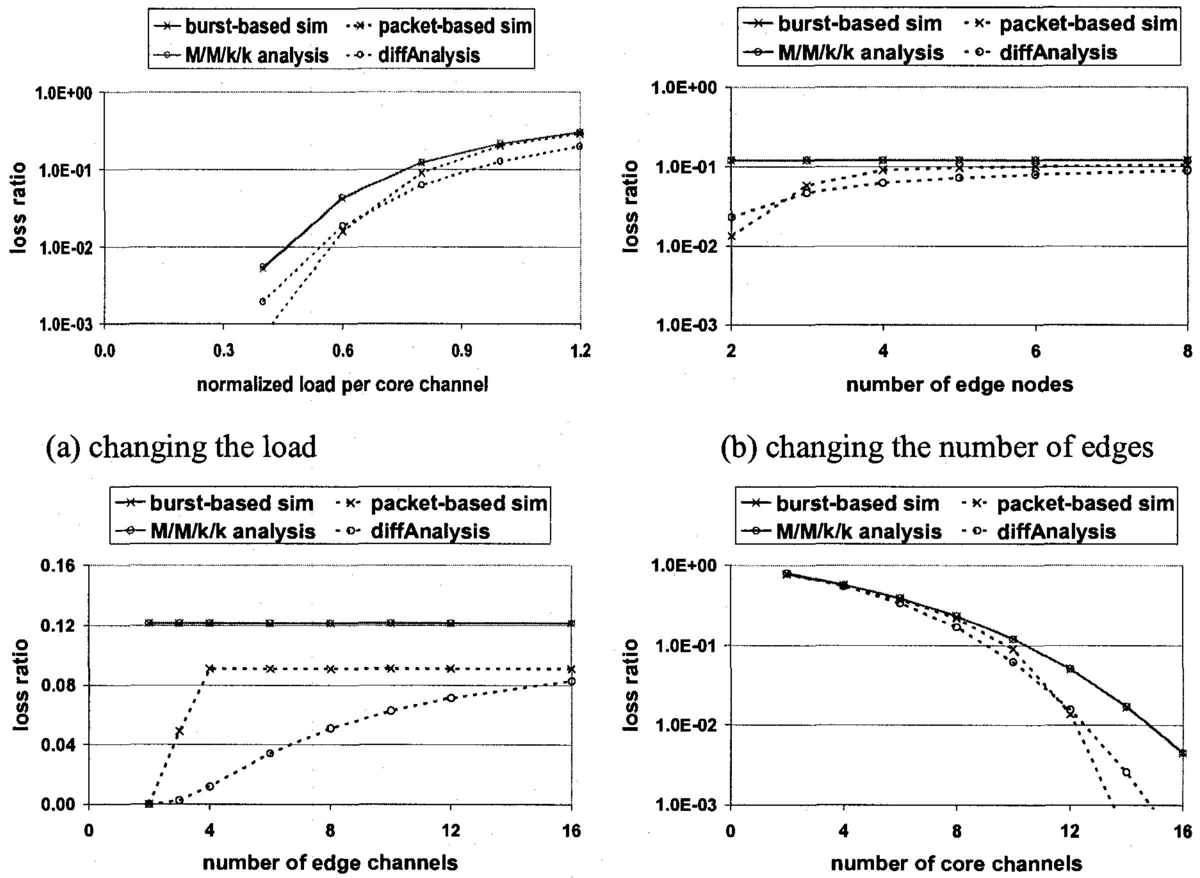
$$p_{JETseg} = \begin{cases} \sum_{k=W_c+1}^{N_e \cdot W_e} \left(\dfrac{k-W_c}{k} \cdot p_k\right) = \sum_{k=W_c+1}^{N_e \cdot W_e} \left(\dfrac{k-W_c}{k} \cdot \dfrac{\binom{N_eW_e}{k} \cdot \left(\dfrac{\lambda}{\mu}\right)^k}{(1+\lambda/\mu)^{N_e \cdot W_e}}\right), & N_e \cdot W_e \ge W_c \\ 0, & N_e \cdot W_e < W_c \end{cases} . \tag{3.4}$$

Compared to (B.2) in Appendix B for the burst-based JETseg model, our analytical equation (3.4) for the packet-based model has taken into account the connectivity of the core node and the assembly process at the edge nodes. Thus, we expect our analytical results to model a real system more accurately.

The above analysis applies to a core node which accepts traffics from the source edge nodes directly. If the edge nodes in Fig. 3.1 are replaced by core nodes and the offered traffic (corresponding to λ and μ in Equations 3.1 to 3.4) is replaced with optical bursts, Eqn. (3.4) is still valid in modeling an intermediate core node along the routing path. Therefore, our analysis is a good estimation of the performance of any OBS core node. Moreover, if the burst assembly process is determined by other parameters, e.g., a timer or the allowable number of packets in a burst, our analytical process is also applicable. As a result, our analysis represents typical OBS operations.

3.2.3 The Less-accurate Burst-based Model

Different from our previous analysis, most existing OBS works adopt a burst-based model, which does not take the burst assembly process into account, but assumes the offered traffic to be in optical bursts already. As reviewed in Chapter 1, this method does not match the real OBS networks. The analytical equations for the loss rate of this less accurate model are summarized in Appendix B. On the other hand, our previous model considers the assembly process at the edge node, in which the burst traffic characteristic is determined by the packet arrival characteristics and the assembly algorithm at the buffers. This system is supposed to be an accurate model in a real OBS network. We will show the improvement of our method in the next section.



(a) changing the load (b) changing the number of edges

(c) changing the number of edge channels (d) changing the number of core channels

Fig. 3.2. Comparison of two models: JET

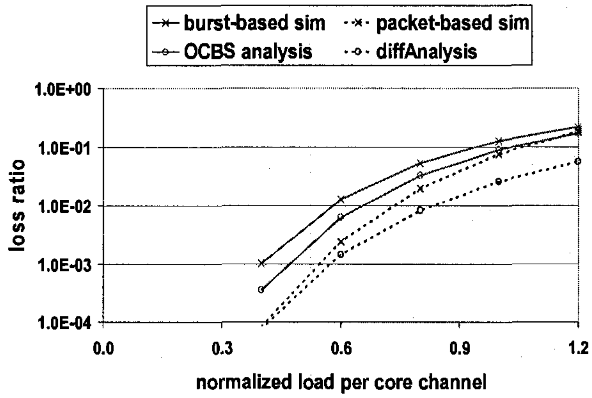
3.2.4 Verification

We now compare the performances of the two models. The basic parameters are as follows: $L_i = 600Kbits$, $N_e = 4$, $N_c = 4$, $W_e = 10$, $W_c = 10$, $C = 1Gbps$, $T_{off} = 100us$ and $\rho = 8$. The default normalized load is 0.5 per channel. We run packet-based simulations and use (3.2, 3.4) for the Fig. 3.1 model. Since our analysis is different from the conventional ones, we use "diffAnalysis" to represent our analytical results. We also run burst-based simulations and use (B.1, B.2) for the Fig. B.1 model in Appendix B in order to make comparisons. Assume the average burst length in the burst-based system is equal to the assembled burst in the packet-based system. The burst arrival of the burst-based model is assumed to be Poisson as done in many literatures. Our simulations are based on one million bursts and five runs using different seeds.

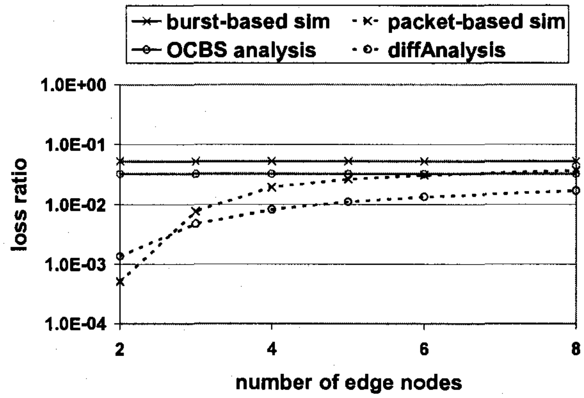
We first examine the performance of the Fig. 3.1 model using the JET protocol. In Fig. 3.2a, it shows that both the burst-based simulation and the *M/M/k/k* analysis have a close loss performance when compared with the real packet-based simulation but only when the normalized load per channel is 1 or higher. At a high load, the input is very dense, thus the system is more like an *M/M/k/k* queue than our analytical method indicated by "diffAnalysis". As the load decreases below 1, the burst-based methods generate a significant difference from the packet-based simulation. On the other hand, the performance of our "diffAnalysis" has a better approximation in the low-load range. Since it is not practical to have a normalized load above 1 in a real network, our analysis matches a real system more closely. Note that our analysis does not capture the streamline transmission of adjacent bursts, so it over-estimates the loss rate at a lower load range. But it is still a better method than the burst-based one.

Fig. 3.2b shows that neither the burst-based simulation nor *M/M/k/k* analytical model can reflect the change on the number of edge nodes. On the other hand, the packet-based simulation and our diffAnalysis do capture the effect on the loss performance, especially for a small number of edges. In a real network, the node degree is usually small, so only the packet-based model and our analysis represent the actual case.

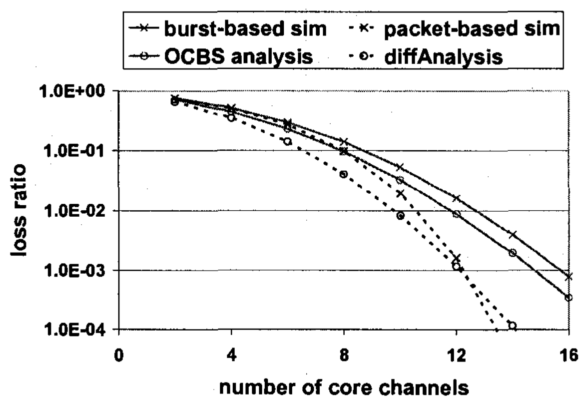
With the development of the WDM technology, the number of wavelength channels is on the rise. Fig. 3.2c demonstrates such impact on the loss performance. When the number of edge channels is fewer than four, the average normalized load on each channel is high (>0.66), so the assembled bursts usually have to wait for an available channel for transmission. The loss rate is low in such a system because the input bursts on different channels of the core node are sort of "time-synchronized" through queueing. With an increasing number of edge channels per node (four or more), the average load on each channel becomes lower than 0.5 and all assembled bursts can easily find an available channel at the edge node for transmission immediately. So the burst transmission process to the core node becomes more random, thus generates a higher loss. It is interesting to see that any further increases beyond four wavelengths has not further increased the "randomness" in the burst transmission and therefore does not affect the loss performance further. It generates a stable loss along with four or more channels per edge node. It is also demonstrated in Fig. 3.2c that the burst-based system does not reflect the possible change of edge channels as only a horizontal line is seen. In comparison, our analysis shows the effect of the number of edge channels on the loss rate in the small edge-channel range. Similarly, our analysis agrees better with the results from the packet-based simulation in Fig. 3.2d, while the burst-based model generates a huge difference regarding the number of core channels. Thus our analysis is more useful for the multiple-wavelength WDM networks.



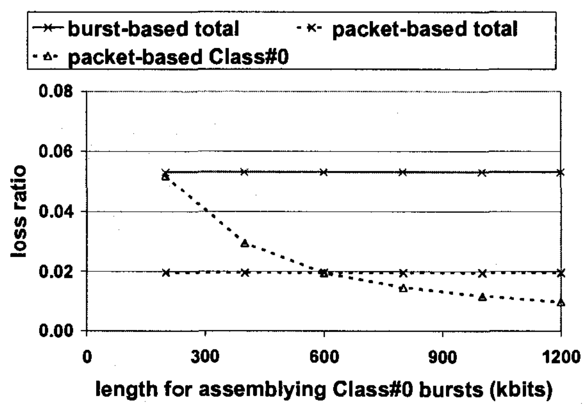
(a) changing the load



(b) changing the number of edge nodes



(c) changing the number of core channels



(d) changing the burst length

Fig. 3.3. Comparison of two models: JETseg

We then examine the performance of the two models using the JETseg protocol. In Fig. 3.3a, it shows again that the burst-based simulation and the *OCBS* analysis are not a good model for the packet-based system when the normalized load per channel is in the normal operating range (load<1) in a real network. The change in the number of the edge nodes is not reflected by the burst-based method in Fig. 3.3b neither, while that change affects the real packet-based system dramatically and our analysis can capture this change, practically when the number of edge nodes is small. As we know, a real node always has a small node-degree (normally less than 4), so our analysis in Fig. 3.3b reflects the effect of the node-degree on the loss performance more actually than the burst-based methods. In a WDM network, a large number of core channels imply a small normalized-load per channel, which cannot be represented by the burst-based system as shown in Fig. 3.3c, while our analysis demonstrates a better matching.

In all of above analysis and simulations, we have used the same load and assembly algorithm for all bursts as most research work, thus all bursts passing through the core node have the same loss rate. However, if we apply a different assembly parameter for the buffer to generate Class#0 bursts at Edge#1, while keeping all other parameters unchanged, an interesting phenomenon is observed in Fig. 3.3d in that while the overall loss rate of the whole network remains unchanged for either burst-based or packet-based simulations, the loss rate of Class#0 bursts from Edge#1 is a decreasing function of its assembly length. For example, if we increase its assembly length from 200Mbits to 800Mbits, its loss rate would decreases from 5% to 1.5%. We can see that neither the conventional burst-based simulation nor our proposed simulation is a good model for this class of bursts because both of them only consider the total loss rate of all classes. Moreover, since the offered load is constant in Fig. 3.3d and the overall loss performance has not changed, the loss of other classes of bursts must change. In other words, the loss performances of different classes of bursts can be varied by the assembly parameter of some class.

3.3 DiffServ Provision

We can take advantage of the previous observation to design a DiffServ scheme. This feature is now analyzed in a JETseg OBS network, i.e., a class of bursts with a larger assembly parameter can generate a lower loss rate.

3.3.1 Loss Analysis

We still use the packet-based model in Fig. 3.1 and use the same assumptions as in the previous section except that the Class#0 bursts have a larger assembly parameter than other classes.

We would like to find out the relative loss ratio of Class#0 bursts and Class#*i* bursts ($i = 1, \ldots, N_c - 1$). When a Class#*i* burst arrives at a core node, a loss will occur *iff* all wavelength channels are busy. There are two possible scenarios: 1) if the ending time of the newly-arrived burst is earlier than the earliest available time of all channels, then the whole burst is dropped; otherwise 2) only the header part of the new burst is dropped (we can also drop the tail part of the existing burst. In this case, the analytical method is similar to the following presentation, but is omitted here.). For these two scenarios, the loss of the new burst is determined by the residual length of the pervious burst on that channel: a longer residual length causes a higher loss for the new burst.

The probability that a new burst find all W_c output channels busy at an epoch is $p_{busy} = \sum_{k=W_c}^{N_e W_e - 1} \binom{N_e W_e - 1}{k} \cdot \left(\frac{\lambda}{\mu}\right)^k \cdot \left(1 - \frac{\lambda}{\mu}\right)^{N_e W_e - 1 - k}$.

Since the total load is fixed and each class has the same load, the probability that a channel is occupied by any class of burst is equal to p_{busy} / N_c. It can be seen that this probability is determined by the normalized load and the total number of channels. Furthermore, the change of the assembly process does not affect this probability (i.e., the chance of contention between any two bursts of different classes), so the probability that a Class#*i* burst finds a channel occupied by a Class#0 burst does not change even if they are assembled by different assembly parameters.

On the other hand, the relative dropping probability of the Class#*i* burst to Class#0 is affected by the residual length of the Class#0 burst occupying the channel, i.e., if the residual length is longer, the dropping part of the Class#*i* burst is longer and vice versa. In order to understand more on the effect of an assembly parameter *A* (*A* can be either a timer or a threshold) of Class#0 bursts on this dropping rate, we start our investigation by examining the arrival process. Due to the independent packet arrival process and the *i.i.d.* packet size distribution, it is easy to see that the number of packets in a Class#0 burst and the length of a Class#0 burst are proportional to *A*. Thus, the mean m_1 and the variance σ^2 of the length of Class#0 bursts are also proportional to the assembly parameter according to the central limited theory [Leon94], i.e., $m_1 \propto A$ and $\sigma^2 \propto A$. By considering these two factors in the residual length equation $r_1 = \frac{m_1}{2} + \frac{\sigma^2}{2m_1}$ [Klei75], one sees that $r_1 \propto A$. In other words, the residual length of the Class#0 burst is approximately proportional to and therefore a linear function of its assembly parameter. So a larger assembly parameter will cause a longer residual length for the Class#0 burst, which in turn will cause more newly-arrived Class#*i* burst to be dropped, i.e., a higher loss of Class#i. Similarly, this principle can be applied to all other classes of bursts. Since the total system loss does not change according to our analysis in Section 3.2, the Class#0 bursts will have a lower loss. In summary, by using a larger assembly parameter for the Class#0 bursts, we can achieve a lower loss rate for this class of traffic at the price of an increasing loss rate for other classes. This is the basics to provide DiffServ among different classes of traffics.

The above analysis only adopts a single timer or threshold for burst assembly, which can cause a high delay or large length variation when the offered load is too low or too high. To alleviate this problem, we employ a supplemental constraint. For example, we can add a lower and an upper bound on the burst length in the timer-based algorithm. These length constraints have to be far away from the average length in the timer period. Thus, the timer still places the major role and the other ones just avoid extraordinary bursts, which have a tiny probability, so the overall burst traffic still satisfies the above relationships. For simplicity, we shall emphasize the basic parameter only, while not explicitly mention the supplemental ones.

So far, our analysis applies to a core node which accepts traffics from the source edge nodes directly. If the edge nodes are replaces by core nodes, it can be seen that our analysis is still valid to approximate an intermediate core node along the routing path by replacing the offered traffic with optical bursts. In this case, the relationship of the bursts and the assembly parameter is an approximation. Therefore, our analysis represents typical OBS operations and its effectiveness will be verified by later simulations.

3.3.2 DiffServ Algorithms

We have designed three algorithms to provide DiffServ among optical bursts in a JETseg OBS network. The first algorithm utilizes the differentiated assembly parameters for each class of bursts based on our previous observation and analysis. The second one is a preemptive scheme. Finally, we use our differentiated scheme as a supplemental solution to the second algorithm to enhance the DiffServ provision.

Alg1: Differentiated burst assembly algorithm

This algorithm applies our previous analysis results directly in DiffServ provision. The differentiated loss rate is determined by the operations at the edge nodes.

1) Assign the same buffer for all packets to the same destination with the same priority at a source edge node.
2) Use a different assembly parameter for each class of bursts at the source edge, i.e., a higher-

priority class uses a larger parameter.

3) Treat the bursts of all priorities equally at the core node.

For example, if we use the burst length as the threshold of the assembly process, we can set $L_0 > ... > L_{N_c-1}$ at the source edge. Class#0 has the highest priority and Class# $(N_c - 1)$ has the lowest priority. Then the loss rate of a higher-priority class is always lower than a lower-priority class. If other parameters are adopted for the burst assembly process, we can achieve different loss ratio in a similar way. This scheme implements DiffServ through operations at the source edges only and does not affect the parameters of the core nodes, so it has a good scalability. Moreover, in a distributed network, it is difficult for an edge node to know the information of other parts of the network in a short time. In this situation, our algorithm can still be employed to achieve a DiffServ immediately.

Alg2: Preemptive channel assignment algorithm

This algorithm uses the preemptive method at the core nodes, in which a higher-priority burst can take over the channel occupied by a lower-priority burst in cases of conflicts, but not vice versa. Under the JETseg operation, a burst chooses a channel as follows when it arrives at a core node.

1) Search all channels occupied by the bursts of the same class to find a channel that can transmit the whole burst without burst segmentation. If there are multiple channels available, one of them is chosen randomly. Stop if chosen successfully.
2) Choose the earliest available channel occupied by the bursts of lower classes. Transmit the new burst as a whole or after segmentation. Stop if the new burst can be transmitted.
3) Choose the earliest available channel occupied by the higher-priority classes (including the same class). Transmit the new burst as a whole or after segmentation.

After the above operations, a channel is guaranteed to be assigned to a new burst. It can be seen that our preemptive algorithm does allow pre-emptive of higher-priority traffics while it also allows burst segmentation in order to utilize the channel efficiently. We first assign channels among the same-priority bursts in order to reduce the inter-class effectuations. We then allow a burst to use the channels that are occupied by other classes of bursts, from low priority to high priority. Therefore, our scheme can utilize the channels efficiently and treat the traffics based on their priorities as well. For a given network, this scheme will provide a fixed difference once the system is settled. Different form Alg1, our Alg2 is based on the operations at the core nodes.

Alg3: Hybrid algorithm

Although Alg2 can utilize the channels efficiently, it still does not utilize differentiated assembly schemes. In order to provide further differentiated loss ratio in Alg2, we can apply our Alg1 at the edge nodes to enhance the loss difference, i.e., letting the higher-priority bursts use a larger assembly parameter. Our DiffServ scheme includes operations at both the edge nodes and the core nodes.

1) At the source edge nodes, the higher-priority bursts use a larger assembly parameter than the lower ones.
2) At the core nodes, the channel assignment follows the same rule as in Alg2.

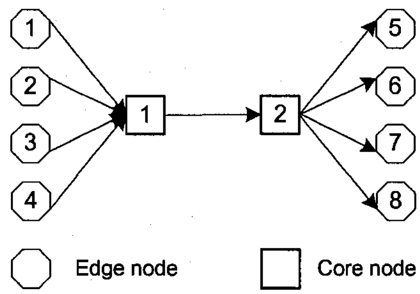
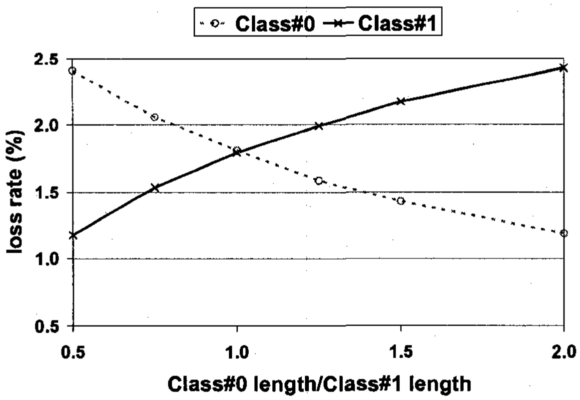


Fig. 3.4. Network model for simulations

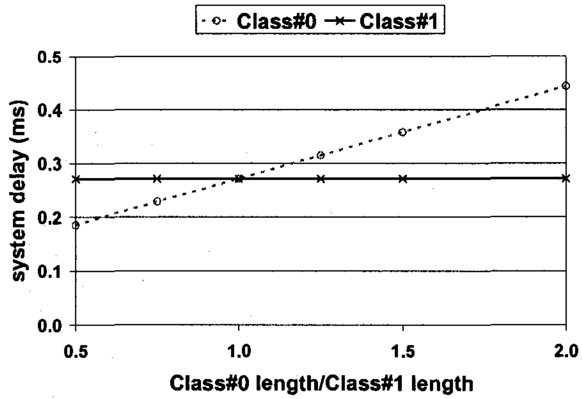
3.4 Performance Evaluation

We evaluate our DiffServ mechanisms in the ten-node network in Fig. 3.4. This network includes eight edge nodes and two core nodes. Edge nodes 1-4 are the sources and Edge nodes 5-8 are the destinations. Each edge node receives the same traffic load and the same traffic classes. Unless specified otherwise, the following parameter settings are used.

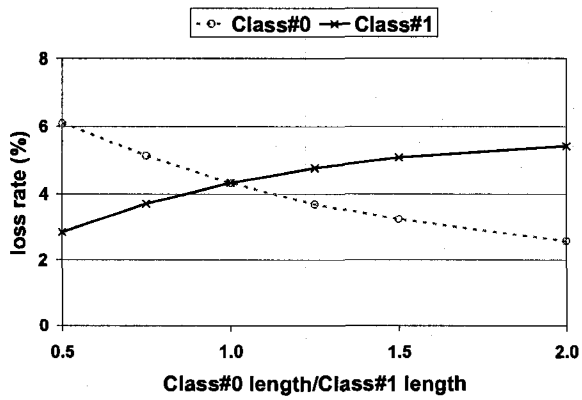
- The packet arrival is a Poisson process
- The Hurst value is equal to 0.85 if Pareto packet arrival process [Gord95] is used
- The packet size distribution follows CAIDA measurements [CAID06]
- The distance between an edge and a core node is 200 km (0.01 ms delay)
- The distance between two core nodes is 1000 km (0.05 ms delay)
- The bandwidth is 10 Gbps per wavelength channel
- The offset time is 100 us
- The switching time of the core switches is 1 us
- The processing time of each burst header is 100 ns
- The size of each buffer is 200M bytes
- The normalized load per channel at Core node 1 is 0.7
- The number of wavelength channels between an edge and a core is 8
- The number of wavelength channels between two core nodes is 24



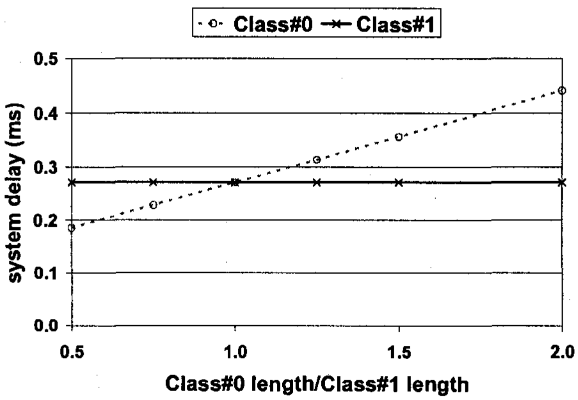
(a) Loss: Poisson packet arrival



(b) Delay: Poisson packet arrival



(c) Loss: Pareto packet arrival

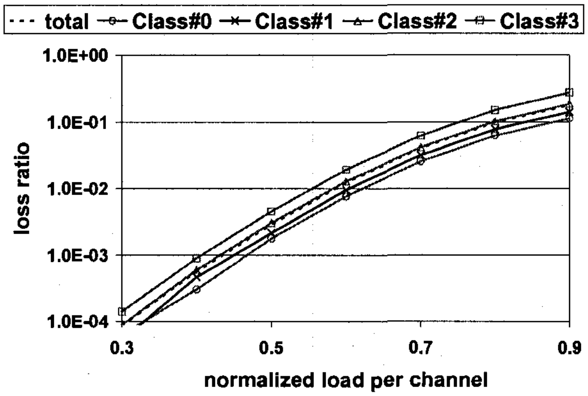


(d) Delay: Pareto packet arrival

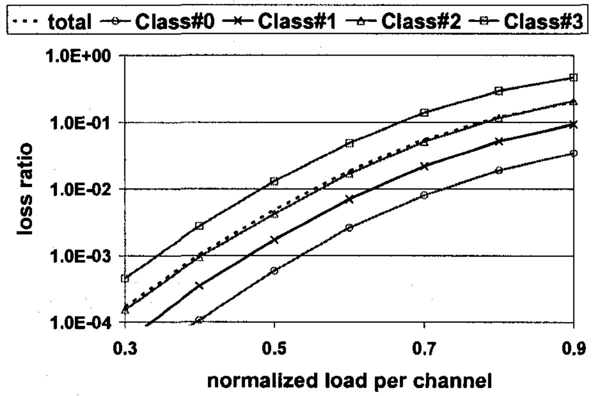
Fig. 3.5. Loss and delay of two classes of traffics

To verify the effectiveness of our DiffServ principle through the burst assembly process (Alg1), we assume there are only two traffic classes and we use burst length as the threshold of the assembly process. The length for assembling Class#1 bursts is set to be 1Mbits while the length for Class#0 bursts is allowed to vary. Fig. 3.5a depicts the loss ratio of two classes of optical bursts with regard to different length ratio R_{len}. The loss rate of Class#0 is a decreasing function of R_{len} while that of Class#1 is an increasing function. Thus Class#0 bursts have a lower loss rate if we let them have a longer assembly length than Class#1 bursts. For example, if Class#0 bursts have a length of 50% of Class#1 bursts, they have a loss rate of 6.1% while Class#1 bursts 2.8%. If we increase Class#0 bursts to 1.5 times length of Class#1 bursts, their loss ratio become 3.2% and 5% respectively. This agrees well with our analysis in Section 3.3.1, i.e., the loss rate of Class#0 bursts is reduced when its assembly length increases. At the same time, Class#1 bursts have fewer chances to be transmitted, thus have a higher loss. In summary, the effect of DiffServ on two classes of bursts is seen by the opposite behavior in the two loss curves as the ratio of their assembly lengths changes from 0.5 to 2.

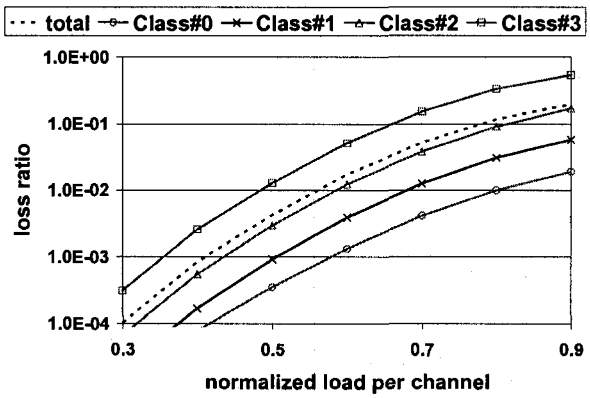
Fig. 3.5b shows that a longer burst length will cause a higher system delay. If we double the length for Class#0 bursts from 1Mbits to 2Mbits, its system delay increases from 0.28ms to 0.44ms. Considering the propagation delay, we can say that the effect on the total end-to-end delay is not significant. Moreover, since the assembly process for the Class#1 bursts is not affected, its delay performance remains stable.



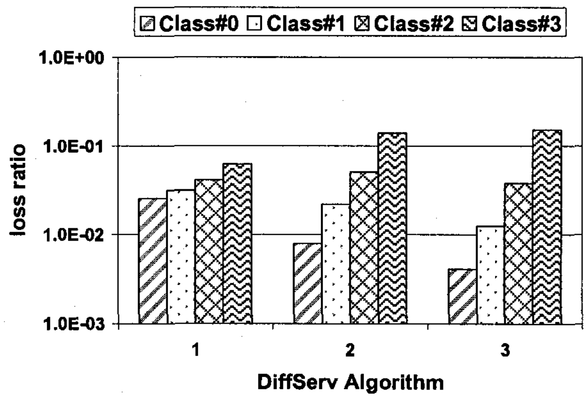
(a) Alg1



(b) Alg2



(c) Alg3



(d) load=0.7

Fig. 3.6. Loss ratio of different classes of traffics

Currently, the Pareto process is often adopted to model the self-similarity property of the Inter traffic. We also compare the loss ratio of the two classes of bursts when the packet arrival is a Pareto process. It is shown in Fig. 3.5c that our previous observations on the Poisson traffic still pertain except a higher loss caused by the more bursty nature of the Pareto distribution. The effect of the system delay on the end-to-end delay is also insignificant in Fig. 3.5d. Summing up, it is a feasible solution to provide DiffServ in the burst loss through adjusting the burst assembly parameters.

Figs. 3.6a-c depict the loss ratio of four burst traffic classes along with the normalized traffic load from 0.3 to 0.9 per input channel at the first core node. The loss rate is an exponentially increasing function of the normalized load in all cases. The assembly lengths of the bursts for Alg1 and Alg3 are 1.5Mbits, 1.2Mbits, 900Kbits and 600Kbits respectively from the highest-priority Class#0 to the lowest-priority Class#3. In Fig. 3.6a, our Alg1 always generates a lower loss for a higher-priority class, which confirms the effectiveness of our scheme based on different assembly parameters again. When we set all classes to have the same assembly length of 1Mbits and use our proposed preemptive scheme Alg2, it also provides a good DiffServ for four different traffic classes as shown in Fig. 3.6b. Fig. 3.6c demonstrates that our hybrid Alg3 algorithm is able to further differentiate the loss ratio among these classes than a single scheme of Alg1 or Alg2. This can be seen from the loss rate comparison in Fig. 3.6d as explained below.

Fig. 3.6d takes a look from a different angle by fixing the offered load at 0.7 per channel and comparing the loss performance of the four classes of bursts. We can see that the loss difference between two burst classes is improved by introducing differentiated assembly lengths. For example, in Alg1, Class#0 bursts have a 2.6% loss rate and Class#3 6.2%. In Alg2, Class#0 bursts have a 0.8% loss rate and Class#3 14.1%. Therefore, the loss difference of the Class#0 and the Class#3 bursts is about 1:2.5 and 1:17 respectively when only Alg1 or only Alg2 is used, while this difference becomes 1:38 in the hybrid Alg3 (Class#0 0.4% and Class#3 15.3%). If we look at the loss difference of any other two classes, it is enhanced in Alg3 than in Alg1 (or Alg2) too. Since the Alg1 only needs to adjust the parameters at the source edge node, it can be adopted easily as a supplemental method to enhance the QoS requirements when other DiffServ scheme (such as Alg2) cannot provide a required loss difference. Therefore, our proposed mechanism of differentiated assembly algorithm is a flexible solution.

Since each burst class has a different length for its assembly process, its system delay will exhibit different performance as in Fig. 3.6. However, this difference has a little effect on the total end-to-end delay as demonstrated before, which is not repeated here. Our DiffServ schemes also work well for other packet arrival and packet length distributions. We have tested other burst assembly processes using either a timer of the allowable number of packets in a burst. Although not shown here, all results have confirmed that our DiffServ schemes are valid and efficient.

3.5 Concluding Remarks

In this chapter, we have investigated some important parameters such as the node degree, the connectivity and the assembly parameters that would affect the analysis and modeling of OBS networks. We have demonstrated that the ignorance of these parameters would generate an obvious disagreement with a real network. We have proposed some analytical models to take into account these parameters in order to remedy the inefficiency of traditional methods. Based on our observations on the flow loss performance and the assembly parameters, we then design a new DiffServ scheme for JETseg OBS networks. By adjusting the assembly parameters at the source edge node, we can achieve a desired loss-difference between optical burst flows. By integrating our design with the preemptive scheme, we can enhance the flexibility of performance differentiation through the operations at the edge node only. Our design does not affect the operations at the core node and does not require any system information of other parts in advance, so it has a good flexibility and scalability in the heterogeneous Internet. Our simulations have shown that it works successfully by itself or by the combination with the preemptive scheme.

CHAPTER 4
FLOW CONTROL IN OBS NETWORKS: A SIMPLE APPROACH

In addition to DiffServ provision, flow control is another important QoS requirement in OBS networks. This chapter designs an edge-based flow control[1] scheme to achieve a guaranteed loss performance for optical burst flows. We start from the analysis of characteristics of burst flows that are assembled by different algorithms discussed in the last chapter. We then investigate the relationship of the assembly parameters and the loss performances of different flows. Finally, we propose a simple control method by arbitrarily setting the characteristics of burst flows in order to facilitate our controlling on the loss performance. We would like to obtain an efficient and easy implementation approach.

4.1 System Description

At the source edge node of our OBS networks, the bursts are formed within intervals that have an exponential distribution and a mean called the assembly timer. In other words, every time after a burst is assembled a new timeout value (according to an exponential distribution but with bounds) is assigned to the new interval to assemble the next burst. Our network can choose one of two popular signaling protocols, JIT or JETtailSeg, for channel assignment at the core nodes.

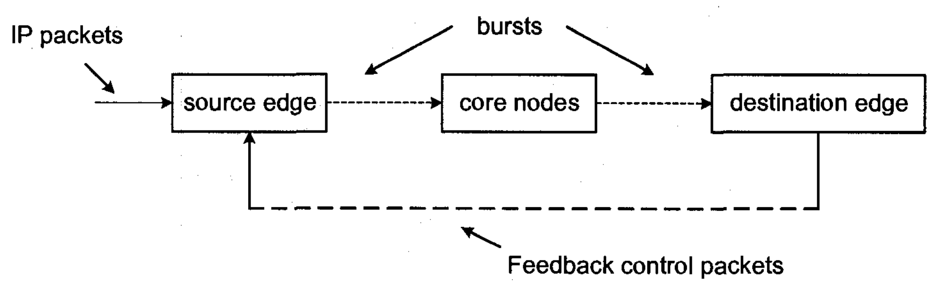


Fig. 4.1. Schematic transmission

In order to control the loss rate of a flow in the acceptable range, we employ a feedback-based control scheme by extending the above basic operations. That is, even if a DB is dropped at a core node, its BHP is still transmitted to the destination edge which computes the loss rate by

[1] Traditional flow control usually means managing the data rate on a link. As discussed in the Literature Review of this thesis, flow control here refers to the loss rate control of optical burst flows, where a burst flow can pass multiple links from the source node to the destination node without intermediate buffering.

comparing the received bits and the original length information in BHPs. The destination node measures the loss rate of a flow every t_m interval by comparing the received message length to the original length included in the BHPs and sends a FCP (Feedback Control Packet) which contains the measured results to the source edge node. The transmission model of a burst flow can be represented by Fig. 4.1. Based on the information in FCPs and the QoS requirement, the source edge node takes appropriate actions on the burst assembly procedure in order to control loss rate of a burst flow. The details will be presented in Section 4.3.

4.2 Analysis

We shall investigate the effects of the assembly timer on the flow loss and the delay, from which we develop the details of the controlling process. Assume there are N destinations and W output channels for a source node s, and each packet has an equal probability going to any destination. Throughout the analysis of this section, we shall use $f_X(\cdot)$ to represent the probability density function (*pdf*) of a random variable X, $F_X(\cdot)$ the cumulative distribution function (*cdf*), $\overline{X}$ the mean value and X' the new value of X after changing the assembly parameter.

Let us focus on the assembly procedure in one buffer which stores bursts of *Flow s-d* for any destination edge node d. The assembly timer for this particular *Flow s-d* is $T_{s,d}$ second. Let $\lambda_{s,d}^P$ (packets/sec) be the average packet arrival to *Flow s-d* and $L_{s,d}^P$ (measured in second) be *i.i.d.* packet length variable. According to the operation scheme described in Section 4.1, the burst generation on *Flow s-d* can be approximated by a Poisson process[2] with an average rate of $\lambda_{s,d}$ bursts/sec and we have $\lambda_{s,d} = 1/T_{s,d}$. Thus, the burst interarrival time $I_{s,d}$ has a *pdf* of $f_{I_{s,d}}(t) = \lambda_{s,d} e^{-\lambda_{s,d} t}$ $(t \geq 0)$ and its mean is $\overline{I_{s,d}} = T_{s,d}$. The total assembled bursts in N buffers form a Poisson process going to W output channels, a known property that the addition of independent Poisson processes is still a Poisson one [Leon94]. Since neither the JIT nor the JETtailSeg protocol would modify the interarrival time of two adjacent bursts at a bufferless core node[3], the

[2] If there is no packet arrival during an interval, there is no burst being generated. This case is rare since the assembly timer is much larger than the packet interarrival time. So our Poisson approximation is still valid and useful for the design of our control method, which is verified by later simulations.

[3] If a burst is dropped, we assume a virtual burst of zero length is still transmitted. Thus, all bursts, including the real ones and the virtual ones, still hold the same interarrival times on all links of a flow. This is reasonable in a network with a moderate load and a small loss. Therefore, our approximation of Poisson burst arrival is possible. For the same reason, we approximate the length distribution on each link by the original one at the source node, which is also verified to be effective by later simulations.

total burst arrival to and departure from any core node satisfies the same Poisson distribution. Note that this is a very important consideration in the original design of our control scheme. First, the Poisson distribution can be studied by many analytical models. Second, since this Poisson property is determined by the assembly timer, the assembly operation at edge nodes can be related to the performance at core nodes easily. Third, the packet delay can be varied by using an adaptive assembly timer according to the QoS requirement. Although other distribution is also possible, the use of exponential distribution is for simplicity and tractability of our analysis.

4.2.1 Delay Analysis

In order to calculate the delay of packets, we need to know the *burst length* $L_{s,d}$ (measured in second). However, it is difficult to find out its distribution since the actual distribution of packet arrival or packet length is unknown. Here we estimate the relationship of the delay performance and the assembly timer. Since the bursts are the aggregation of many IP packets, the burst length approximately has a linear change with the interval of burst assembly. The *assembly delay* D_a will vary in the same trend as the assembly interval, which means $\overline{D_a}$ has a linear relationship with the assembly timer as well. Furthermore, when we examine the generated bursts going to output channels, it can be approximated by an *M/M/W* queueing system. The *queueing delay* D_q and the *transmission delay* D_t in such a system have been widely studied in many literatures (e.g. [Klei75]). In this queueing system, the average values of these two delays have a linear relationship along with the average burst length. Summing up, the average *system delay* D_{sys} a packet has at the edge node can be approximated by a linear function of the assembly timer.

In Appendix C, we have analyzed the delay performance under Poisson packet arrival. Through further simulation study in Section 4.4.1, it can be seen our above approximation is valid for different distributions of packet arrival and packet length: the average delay changes linearly along with the assembly timer. Therefore, we can utilize this property to control the delay performance at edge nodes.

4.2.2 Loss Analysis

As reviewed in Section 1.1.2, the loss ratio of all links in an OBS network can be uniquely determined through some iteration processes [RoVu03]. It is interesting to see that the link loss and network throughput are only dependent on the traffic load in an OBS network by using this method. However, this method did not distinguish the flows sharing the same link. Here we demonstrate that the flows on the same link suffer different loss ratio if they have different assembly parameters.

Consider $Burst(s,d,l)$, the bursts on $Flow\ s-d$ arriving to link l and let $\lambda_{s,d,l}$ (bursts/sec) be its average arrival rate. In a real OBS network with a low loss probability, we have approximately

$$\lambda_{s,d,l} = \lambda_{s,d} = 1/T_{s,d}. \tag{4.1}$$

Let $L_{s,d,l}$ (sec) be the random variables for the length of $Burst(s,d,l)$ and $X_{s,d,l}$ be the burst arrival time, then $f_{X_{s,d,l}}(x) = \frac{1}{t}$, $0 < x \le t$. Denote the loss rate of $Burst(s,d,l)$ on link l by $p_{s,d,l}$. If l is not on the route, $\lambda_{s,d,l} = p_{s,d,l} = 0$. Assuming independence of losses among all links, the loss rate of $Flow\ s-d$, $p_{s,d}$, can be approximated by

$$p_{s,d} = 1 - \prod_{\forall\, l \in (1,\ldots J)} (1 - p_{s,d,l}). \tag{4.2}$$

Before we provide the flow control mechanism, we shall present two theorems below related to the effect of the assembly timer in the proposed burst assembly algorithm.

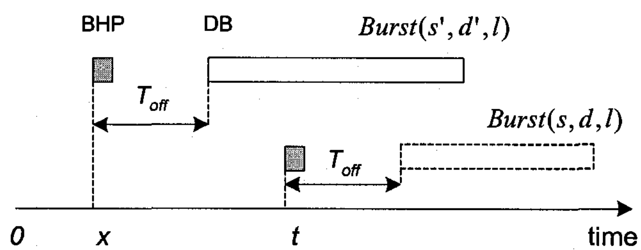


Fig. 4.2. Flow analysis of JIT

Theorem 1: The loss rate of a flow is a decreasing function of its assembly timer in an OBS network using the JIT protocol.

Proof:

The loss of a flow arises from two scenarios: the conflicts of the bursts in the same flow and from different flows. Since the dropping probability of Poisson traffic is fixed for a given load [Leon94], the conflicts in the same flow are fixed. Then we only need consider the change of the conflicts from different flows.

First consider a simple case of two flows, $Flow\ s-d$ and $Flow\ s'-d'$, contending for link l,

($l \in (1,2,...,J)$) that has only one wavelength channel. As shown in Fig. 4.2, when the BHP of $Burst(s,d,l)$ arrives at time t $(t > 0)$, this burst will be dropped *iff* the BHP of earlier $Burst(s',d',l)$ has arrived at time x ($x < t$) and its reserved length is longer than $t-x$, i.e., $(L_{s',d',l} + T_{off}) > (t-x)$. Let $A_{s,d,l}(t)$ be the number of $Burst(s,d,l)$ arrived during a period t. Since the arrival is Poisson, we have $\Pr[A_{s,d,l}(t) = n] = \frac{(\lambda_{s,d,l}t)^n}{n!} e^{-\lambda_{s,d,l}t}$. Due to the independence of the burst arrival time and burst length distributions (Section 2.6), the loss rate of $Burst(s,d,l)$ at time t becomes (using the conditioning and unconditioning technique)

$$\begin{aligned}
p_{s,d,l}(t) &= \Pr[Burst(s',d',l) \textit{ is longer than } (t-x-T_{off}) | \textit{no other } Burst(s',d',l) \textit{ arrives during } t-x] \\
&\quad \cdot \Pr[\textit{no other } Burst(s',d',l) \textit{ arrives during } t-x | Burst(s',d',l) \textit{ arrives at time } x] \\
&\quad \cdot \Pr[Burst(s',d',l) \textit{ arrives at time } x,\ x < t] \\
&= \int_{x=0}^{t} \Pr[(L_{s',d',l} + T_{off}) \geq (t-x) \mid A_{s',d',l}(t-x) = 0] \cdot \Pr[A_{s',d',l}(t-x) = 0] \cdot f_{X_{s',d',l}}(x) \cdot dx \\
&= \int_{x=0}^{t} \Pr[(L_{s',d',l} + T_{off}) \geq (t-x)] \cdot \Pr[A_{s',d',l}(t-x) = 0] \cdot f_{X_{s',d',l}}(x) \cdot dx \\
&= \int_{x=0}^{t} e^{-\lambda_{s',d',l}(t-x)} \cdot (1 - F_{L_{s',d',l}}(t-x-T_{off})) \cdot \frac{1}{t} \cdot dx
\end{aligned} \tag{4.3}$$

Similarly, we can obtain the loss rate of $Burst(s',d',l)$ at time t

$$\begin{aligned}
p_{s',d',l}(t) &= \Pr[Burst(s,d,l) \textit{ is longer than } (t-x-T_{off}) | \textit{no other } Burst(s,d,l) \textit{ arrives during } t-x] \\
&\quad \cdot \Pr[\textit{no other } Burst(s,d,l) \textit{ arrives during } t-x | Burst(s,d,l) \textit{ arrives at time } x] \\
&\quad \cdot \Pr[Burst(s,d,l) \textit{ arrives at time } x,\ x < t] \\
&= \int_{x=0}^{t} e^{-\lambda_{s,d,l}(t-x)} \cdot (1 - F_{L_{s,d,l}}(t-x-T_{off})) \cdot \frac{1}{t} \cdot dx
\end{aligned} \tag{4.4}$$

If we now increase the assembly timer of $Flow\ s-d$ from $T_{s,d}$ to $T'_{s,d}$ while keeping the load constant, the arrival rate of $Burst(s,d,l)$ will decrease according to (4.1), i.e., $\lambda'_{s,d,l} < \lambda_{s,d,l}$, and consequently

$$e^{-\lambda'_{s,d,l}(t-x)} > e^{-\lambda_{s,d,l}(t-x)},\quad t > x > 0. \tag{4.5}$$

Since a longer assembly timer allows more packets to be aggregated into the burst for a fixed load, the burst length $L'_{s,d,l}$ tends to be longer. Thus, for a given value y, $\Pr[L'_{s,d,l} > y] > \Pr[L_{s,d,l} > y]$. Due to $F_{L_{s,d,l}}(y) = 1 - \Pr[L_{s,d,l} > y]$, the *cdf* functions satisfy

$$F_{L'_{s,d,l}}(y) < F_{L_{s,d,l}}(y),\quad y > 0. \tag{4.6}$$

By combing (4.5) and (4.6), we now obtain

$$e^{-\lambda'_{s,d,l}(t-x)} \cdot (1 - F_{L'_{s,d,l}}(t-x-T_{off})) > e^{-\lambda_{s,d,l}(t-x)} \cdot (1 - F_{L_{s,d,l}}(t-x-T_{off})),\quad t > x > 0, \tag{4.7}$$

which yields

$$\int_{x=0}^{t} e^{-\lambda'_{s,d,l}(t-x)} \cdot (1-F_{L'_{s,d,l}}(t-x-T_{off})) \cdot \frac{1}{t} \cdot dx > \int_{x=0}^{t} e^{-\lambda_{s,d,l}(t-x)} \cdot (1-F_{L_{s,d,l}}(t-x-T_{off})) \cdot \frac{1}{t} \cdot dx \,. \tag{4.8}$$

Substituting (4.8) in (4.4), we obtain the relationship $p'_{s',d',l}(t) > p_{s',d',l}(t)$ for any instance *t>0*. Since the mean loss rate is a time-average of instantaneous ones, we have $p'_{s',d',l} > p_{s',d',l}$, which means the loss rate of $Burst(s',d',l)$ become higher with an increasing $T_{s,d}$. According to (4.1), the loss rate of $Flow\ s'-d'$ increases.

For a flow with a given load, a higher loss rate causes a lower throughput. Thus, the throughput of $Flow\ s'-d'$ decreases in this case. As mentioned before, the total network throughput is fixed for a given load, so the throughput of $Flow\ s-d$ will become higher. Since the input load of $Flow\ s-d$ does not change, the loss rate of $Flow\ s-d$ ($p_{s,d}$) drops.

The above results can be extended to more general cases where there are multiple flows or multiple wavelength channels. When considering the effect of multiple flows, the aggregated burst arrival still forms a Poisson process (a known property of independent Poisson distributions). So Eqns. (4.3) and (4.4) can be reused to arrive at the same conclusion. When considering multiple channels on link *l*, each channel has the same behavior. Thus, the effect of the assembly timer is same to all channels. As a result, Theorem 1 is valid for a general case. ■

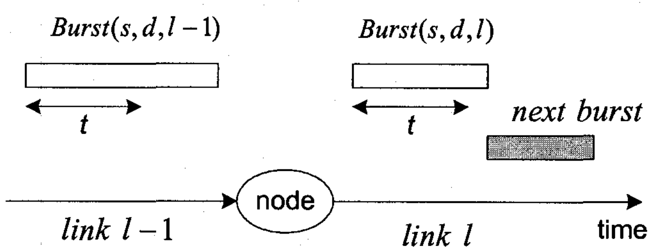


Fig. 4.3. Flow analysis of JETtailSeg

Theorem 2: The loss rate of a flow is a decreasing function of its assembly timer in an OBS network using JETtailSeg protocol.

Proof:

In our JETtailSeg protocol, a burst can be truncated by another burst that arrives before its ending time. We start from the scenario that $Flow\ s-d$ contends for link l that has only one wavelength channel as shown in Fig. 4.3. The length of $Burst(s,d,l)$, $L_{s,d,l}$, is longer than *t* (*t>0*) *iff* this burst is longer than t on the previous link (assuming to be *l-1*, corresponding to $Burst(s,d,l-1)$), and no other bursts (including all flows sharing this link) have arrived to link l during this t period. The first condition guarantees the burst length and the second condition ensures the burst has not being dropped or truncated before time t. Being Poisson, the probability that n bursts have arrived to link l during a period t is represented by $\Pr[A_l(t)=n]=\frac{(\lambda_l t)^n}{n!}e^{-\lambda_l t}$. Then we have

$$\begin{aligned}\Pr[L_{s,d,l}\geq t] &= \Pr[Burst(s,d,l)\ is\ longer\ than\ t]\\ &= \Pr[no\ burst\ arrives\ to\ link\ l\ in\ t\ period \mid Burst(s,d,l-1)\ is\ longer\ than\ t]\\ &\qquad \cdot \Pr[Burst(s,d,l-1)\ is\ longer\ than\ t]\\ &= \Pr[A_l(t)=0 \mid L_{s,d,l-1}\geq t]\cdot \Pr[L_{s,d,l-1}\geq t] = \Pr[L_{s,d,l-1}\geq t]\cdot \Pr[A_l(t)=0] = e^{-\lambda_l t}\cdot(1-F_{L_{s,d,l-1}}(t))\end{aligned}$$

We can also obtain the *cdf* of $L_{s,d,l}$ from the above equation, i.e.,

$$F_{L_{s,d,l}}(t)=\Pr[L_{s,d,l}<t]=1-\Pr[L_{s,d,l}\geq t]=1-e^{-\lambda_l t}\cdot(1-F_{L_{s,d,l-1}}(t)),\ t>0\,.$$

Then the loss rate of $Burst(s,d,l)$ becomes

$$\begin{aligned}p_{s,d,l} &= 1-\frac{E[burst\ length\ on\ link\ l]}{E[burst\ length\ on\ link\ l-1]} = 1-\frac{\overline{L_{s,d,l}}}{\overline{L_{s,d,l-1}}} = 1-\frac{\int_{t=0}^{\infty} t\cdot f_{L_{s,d,l}}(t)\cdot dt}{\int_{t=0}^{\infty} t\cdot f_{L_{s,d,l-1}}(t)\cdot dt}\\ &= 1-\frac{\int_{t=0}^{\infty}[1-F_{L_{s,d,l}}(t)]\cdot dt}{\int_{t=0}^{\infty}[1-F_{L_{s,d,l-1}}(t)]\cdot dt} = 1-\frac{\int_{t=0}^{\infty} e^{-\lambda_l t}\cdot[1-F_{L_{s,d,l-1}}(t)]\cdot dt}{\int_{t=0}^{\infty}[1-F_{L_{s,d,l-1}}(t)]\cdot dt}\end{aligned} \tag{4.9}$$

If we now increase the assembly timer of *Flow* $s-d$ from $T_{s,d}$ to $T_{s,d}^{'}$ while fixing the load, the arrival rate of $Burst(s,d,l)$ will decrease according to (4.1), i.e., $\lambda_{s,d,l}^{'}<\lambda_{s,d,l}$. Since the total burst arrival rate to link l is the sum of all possible bursts sharing this link $\lambda_l=\sum_{\forall s-d}\lambda_{s,d,l}$, the burst arrival rate to link l becomes lower, $\lambda_l^{'}<\lambda_l$, and consequently

$$e^{-\lambda_l^{'}t}>e^{-\lambda_l t},\ t>0\,. \tag{4.10}$$

Similar to the previous analysis of (4.6), the *cdf* functions satisfy

$$(1-F_{L_{s,d,l}^{'}}(t))>(1-F_{L_{s,d,l}}(t)), \tag{4.11}$$

and $e^{-\lambda_l^{'}t}(1-F_{L_{s,d,l}^{'}}(t))>e^{-\lambda_l t}(1-F_{L_{s,d,l-1}}(t)),\ t>0\,.$ (4.12)

By integrating (4.11) and (4. 12), we have

$$\int_{t=0}^{\infty}(1-F_{L_{s,d,l}^{'}}(t))\cdot dt>\int_{t=0}^{\infty}(1-F_{L_{s,d,l}}(t))\cdot dt, \tag{4.13}$$

and $\int_{t=0}^{\infty} e^{-\lambda_l' t}(1-F_{L'_{s,d,l}}(t))\cdot dt > \int_{t=0}^{\infty} e^{-\lambda_l t}(1-F_{L_{s,d,l-1}}(t))\cdot dt$, $t>0$. (4.14)

Substituting (4.13) and (4.14) in (4.9), we can see that both the numerator and the denominator will increase after increasing $T_{s,d}$ to $T'_{s,d}$. However, the numerator will increase faster than the denominator because of the coefficient $e^{-\lambda_l' t} > e^{-\lambda_l t}$, which means

$\frac{\int_{t=0}^{\infty} e^{-\lambda_l' t}\cdot[1-F_{L'_{s,d,l-1}}(t)]\cdot dt}{\int_{t=0}^{\infty}[1-F_{L'_{s,d,l-1}}(t)]\cdot dt} > \frac{\int_{t=0}^{\infty} e^{-\lambda_l t}\cdot[1-F_{L_{s,d,l-1}}(t)]\cdot dt}{\int_{t=0}^{\infty}[1-F_{L_{s,d,l-1}}(t)]\cdot dt}$. Consequently, $p'_{s,d,l} < p_{s,d,l}$. According to (4.2), the loss rate of *Flow* $s-d$ will decrease eventually, i.e., $p'_{s,d} < p_{s,d}$.

It is also straightforward to extend the above results to a general case where there are multiple wavelength channels or multiple flows in the same way as we have explained in the JIT analysis. Therefore, our conclusion is always valid. ■

In the above analysis, the header processing time and the optical switching time are not taken into account. Since the control traffic has a small volume in an OBS network, the FCFS principle still holds if the header processing time is considered. Moreover, the switching time of the optical core nodes is often fixed, so its effect is equivalent to subtracting each burst by a short fixed-length. One can see that neither of these two parameters would affect the effectiveness and conclusion of our analysis on the flow loss and the assembly parameter. Another assumption of wavelength conversion can be relaxed too. If no wavelength conversion is allowed, the number of available channels for a flow is reduced to the number of available fibers. This change does not affect our analysis.

4.3 Control Algorithm

The above two theorems are the basics of our control scheme to be implemented at the source edge node. Assume the source edge node wants to keep the loss rate of a flow to be θ and the acceptable margin is δ. If the measured loss rate is out of this marginal range around θ, the source node needs to take some actions to rectify it. The source node interprets the feedback FCPs of a flow by implementing following algorithm.

for each arrival of feedback control packets
    if (current_time – last_time > t_a*)*
        calculate the average loss rate p_{avg} *over past* t_p *period*
        if $p_{avg} > \theta\cdot(1+\delta)$

multiply the assembly timer by $(1+\varepsilon)$

else if $p_{avg} < \theta \cdot (1-\delta)$

multiply the assembly timer by $(1-\varepsilon)$

last_time= current_time

In the above operation, t_a is the minimal waiting time between two consecutive adjustments of the assembly timer and ε is the adjustment rate of the assembly timer for every modification. The average loss rate, p_{avg}, is the mean of all measured loss ratio during past t_p period (t_p is assumed equal to a multiple of t_m to facilitate calculations). Since the adjustment is made periodically, it is well-known from the control theory that such a system might fall into an oscillation state. We set t_a to be larger than the round-trip propagation delay in order to avoid oscillation on the adjustment before its effect is reflected in the next measurement. Furthermore, we set $t_a \geq t_m$ so that the source node reacts to each FCP for no more than once.

The above algorithm can be applied in either a static or dynamic network. In a static network, the input loads remain unchanged and the loss rate of a flow is stable. If we want to reduce this loss rate, we can apply our control scheme to do it. In a dynamic network, the inputs vary and the loss performance of a flow fluctuates as well, where our control scheme can be used to keep the loss rate of a flow in a stable range. For both applications, the control task is implemented by the edge nodes only, while the core nodes can adopt either JIT or JETtailSeg protocol.

Our burst assembly intervals are designed to have an exponential distribution. If the generated interval is large, the system delay will become high. Conversely, shorter intervals will generate shorter DBs and more BHPs, which impact a pressure on the BHP processing capability. Therefore, we impose an upper and a lower bounds on the generated assembly intervals for the considerations of delay requirement and system capacity. This means the real burst interarrival is actually a truncated-Poisson process. Since these bounds (between a few microseconds and a few milliseconds) are normally far from the assembly timer (around a few hundred microseconds) in our network, the probability of truncation is usually assumed to be negligible (according to the properties of exponential distributions) as done in our previous analysis. This is supported by our simulation results in the next section where the truncation does not affect the effectiveness of our flow control scheme.

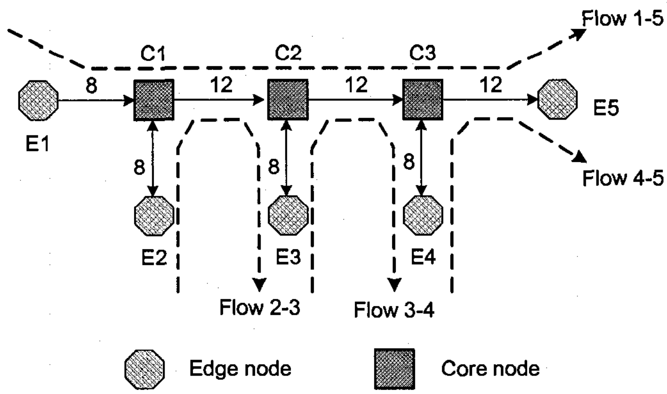
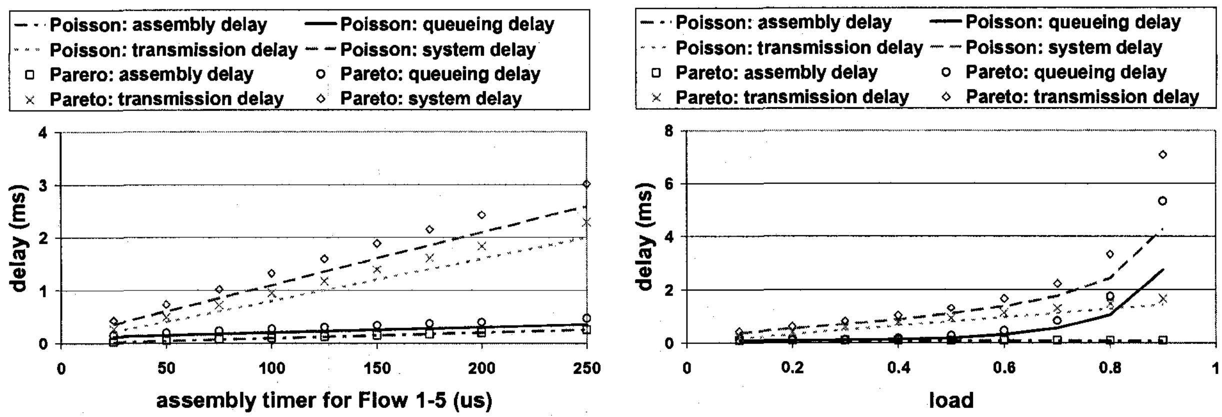


Fig. 4.4. Network model for simulations

4.4 Performance Evaluation

We shall use the network in Fig. 4.4 to verify our analysis and flow control mechanism. This network consists of five edge nodes E1-E5 and three core nodes C1-C3. Four burst flows are transmitted including *Flow* 1-5 (E1 -> E5), *Flow* 2-3 (E2 -> E3), *Flow* 3-4 (E3 -> E4) and *Flow* 4-5 (E4 -> E5), each of which takes the routing path along the dashed lines. The solid lines refer to the links for DB transmission and the number beside the link indicates the amount of wavelength channels for DBs. The control channels are not shown in the diagram. Unless specified, the same set of parameter values as in Section 3.4 is used. The distance between adjacent nodes is 200km, the bandwidth is 1Gbps, each source edge receives same traffic load 0.5, the assembly timer is 100us, and the lower and upper bounds of assembly intervals are 5us and 5ms respectively. We have obtained the 95% confidence interval for our performance measures. An example is shown in Fig. 4.7 and more examples can be found in Appendix D. Since they are usually small, they are often omitted for clarity reason.

The loss rate of burst flows, the packet system delay at edge nodes and the throughput are major performance concerns in our design, which can be measured for either a flow or the whole network. In Section 4.4.1, we use simulations to confirm the relationship of the delay and throughput performance (with respect to the assembly timer) established in Section 4.2. Next, we apply our flow control mechanism from Section 4.3 in two applications, a static network (Section 4.4.2) and a dynamic network (Section 4.4.3). For each case, we consider two possible choices, i.e., using either JIT or JETtailSeg protocol at the core nodes.



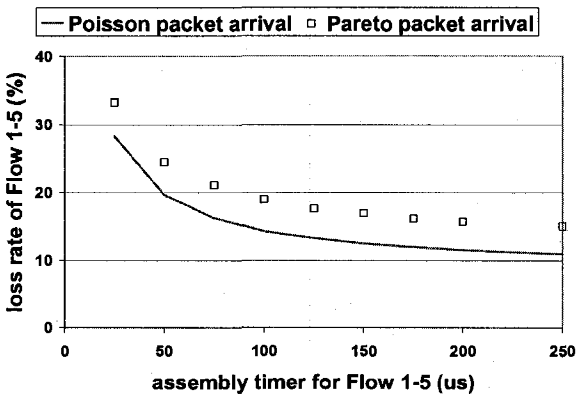
(a) Delay vs. assembly timer of *Flow* 1-5 (b) Delay vs. load

Fig. 4.5. Delay performance

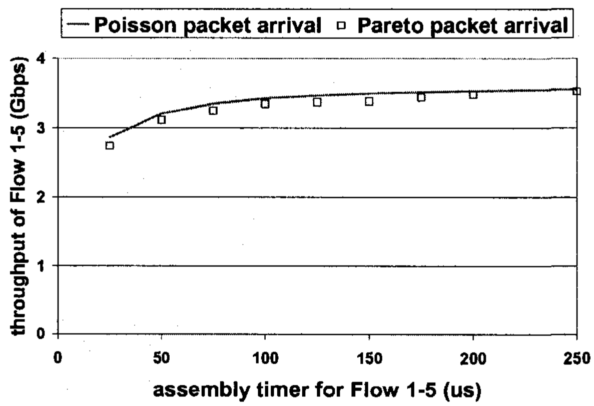
4.4.1 Verification of the Influence of the Assembly Timer

We first examine the delay performance of packets in *Flow* 1-5 in Fig. 4.5. Fig. 4.5a shows that, when packet arrival is a Poisson process, the total system delay increases linearly from 0.6ms to 2.6ms as the assembly timer of *Flow* 1-5 increases from 50us to 250us. All other types of delays also exhibit a linear relationship with the assembly timer, which agrees with our approximation in Section 4.2. Special mention is the performance under Pareto packet arrival. Fig. 4.5a shows that all delays under Pareto packet arrival change in the same way as the Poisson arrival case, but with larger values, which can be explained by the more bursty nature of the self-similar traffic. Fig. 4.5b illustrates the delay change as a function of the traffic load. Let us examine the Poisson packet arrival first. Since we choose the same assembly timer for the full load range, the average assembly delay is almost a constant of 0.1ms. The transmissions delay has a linear increasing with the load because the burst length has an almost linear relationship with the load. The queueing delay increases exponentially from 0.1ms to 2.7ms along with the traffic load from 0.1 to 0.9. This observation also agrees with our previous approximation. The Pareto packet arrival introduces a longer delay due to the bursty nature of Pareto distribution. These delay values are independent of the signaling protocol adopted for core nodes and all results are a proof that the system delay performance can be controlled by the assembly timer.

Our previous analysis is not limited to some special packet types only (e.g., Poisson packet arrival, exponential or fixed packet length distribution), instead we focus on finding out the effects or relationship of different parameters by approximation and comparison, which is good enough as a reference for the design of our control scheme. In light of this fact, we have similar observations under different traffic inputs: Poisson packet arrival, Pareto packet arrival, fixed packet length and exponentially-distributed packet length. For all scenarios, we use same average packet arrival rate and same average packet length in order to make comparisons. Since bursts are the aggregation of many IP packets, the differences of these distributions are smoothed at the burst level. Thus, all scenarios exhibit very close performances regarding to the load and the assembly timer. Please see Appendix D for details.



(a) Loss rate

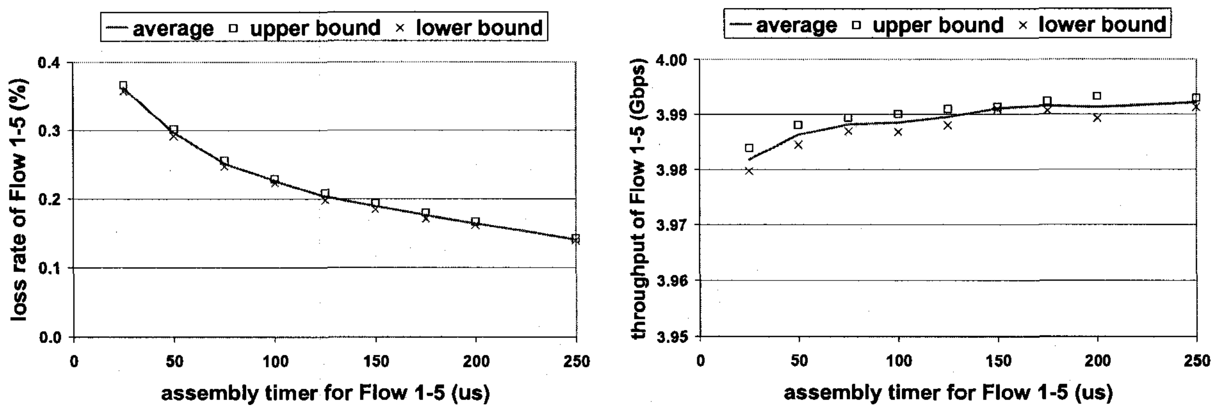


(b) Throughput

Fig. 4.6. Loss rate and throughput performance of Flow 1-5 using JIT

Fig. 4.6 depicts the loss ratio and throughput of *Flow* 1-5 when its assembly timer changes in a JIT OBS network. It can be seen that the loss rate of *Flow* 1-5 decreases significantly with the increasing of the assembly timer. Consider the case of Poisson packet arrival. *Flow* 1-5 has a loss rate of 14% and a throughput of 3.42Gbps when the assembly timer is 100us. If we double the assembly timer to 200us, the loss rate decreases to 11.5% and the throughput increases from 3.42Gbps to 3.55Gbps. A similar observation applies to Pareto packet arrival, but it suffers a higher loss due to the bursty nature. The simulation results have endorsed our theorems in an earlier section, i.e., the loss rate of a flow is a decreasing function of its assembly timer.

The effect of the assembly timer in a JETtailSeg OBS network is illustrated in Fig. 4.7. When the assembly timer is 100us, *Flow* 1-5 has a loss rate of 0.22%. If we double the assembly timer to 200us, the loss rate decreases to 0.16%. Since the offered load does not change when we adjust the assembly timer, a lower loss rate means a higher throughput for this flow. Accordingly, the throughput of *Flow* 1-5 increases from 3.988Gbsp to 3.992Gbps. These observations have supported our previous theorems again. For both loss and throughput, the 95% confidence intervals are very small.



(a) Loss rate (b) Throughput

Fig. 4.7. Loss rate and throughput performance of Flow 1-5 using JETtailSeg

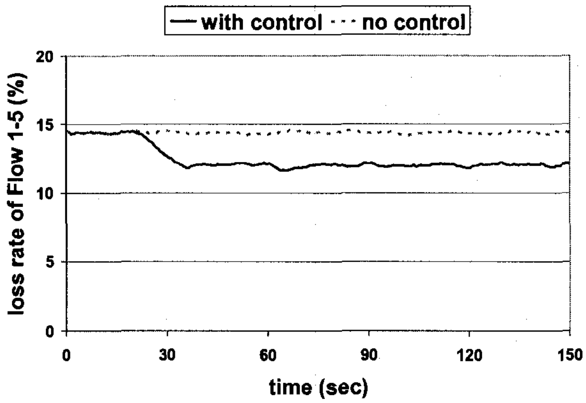
We have studied the effect of other packet arrival processes and packet length distributions. They are summarized in Appendix E and one can see the assembly timer again exhibits same effects as the ones showed in Figs. 4.6 and 4.7.

Table 4.1. Network Throughput (Gbps)

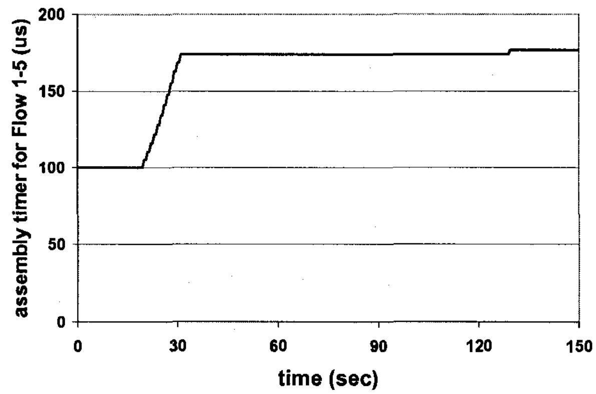
Assembly Timer for *Flow* 1-5 (us)	25	50	75	100	125	150	175	200	250
JIT throughput	13.09	14.16	14.55	14.75	14.86	14.94	14.99	15.04	15.09
JETtailSeg throughput	15.96	15.97	15.97	15.97	15.98	15.98	15.98	15.98	15.97

Table 4.1 demonstrates our statement in Section 4.2.2, i.e., the overall network throughput is not affected by the change of the assembly timer of a flow. One can see that the network throughput is almost constant around 15.97Gbps when JETtailSeg is used no matter the assembly timer of *Flow* 1-5 increases or decreases. Therefore, the adjustment of the assembly timer of a flow will not suffer the overall throughput of the network. Even though the throughput has not been studied analytically, this observation supports our argument again: our timer adjustment scheme is superior to the ordinary rate-based scheme in terms of the overall network throughput. On the other hand, the network throughput increases from 13.09Gbps to 15.09Gbps along with the assembly timer of *Flow* 1-5 from 25us to 250us when using JIT protocol. This phenomenon

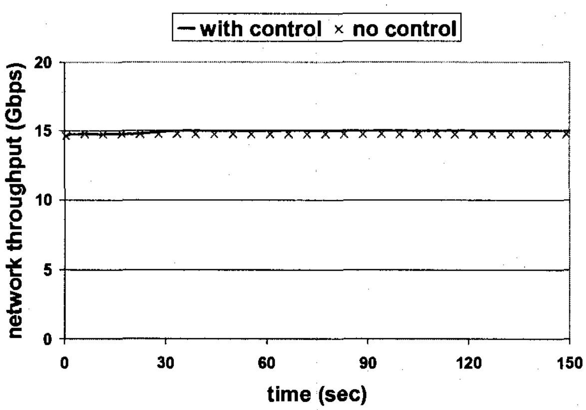
seems to be inconsistent with our argument that the overall throughput should not be affected. Remember that argument is based on a large loss network, while we have adopted a relatively simple network model in Fig. 4.4. On the other hand, we have demonstrated that the increasing of the assembly timer can reduce a flow's loss rate as expected. Therefore, the improvement of the network throughput is an additional benefit indeed. It is an advantage of our design when compared to other mechanism as shown later.



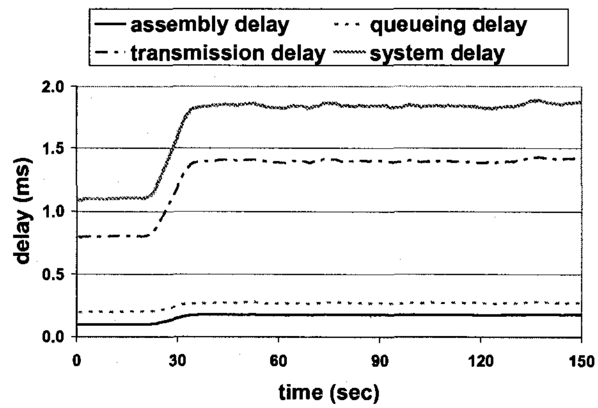
(a) Loss rate of *Flow* 1-5



(b) Assembly timer for *Flow* 1-5



(c) Network throughput



(d) Delay at E1

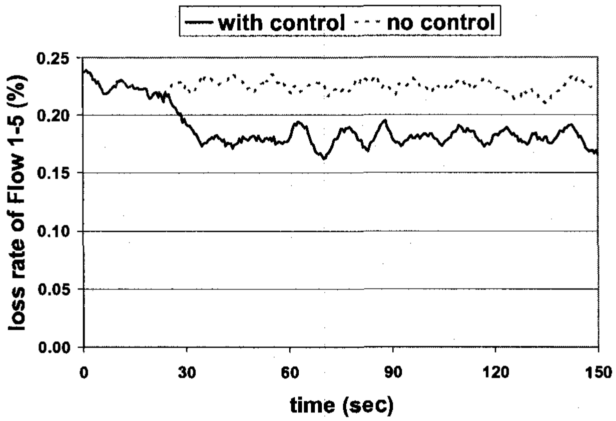
Fig. 4.8. Controller performance using JIT

4.4.2 Static Networks

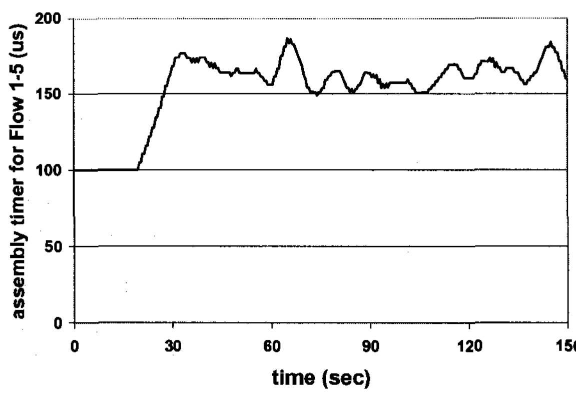
We then examine the feedback control scheme on *Flow* 1-5 when the network is static, i.e., the input load is a constant for each flow. For the JIT case, we set θ to be 12%. For the JETtailSeg case, we set θ to be 0.18%. These are the values that produce about 80% of the loss rate without congestion control in each case. Assume the allowable margin δ is equal to 5% and adjustment rate ε to 5% for both cases. The edge node will wait t_a=1s between two consecutive assembly timer adjustment. The destination node E5 measures the loss rate of *Flow* 1-5 every t_m=0.5s and the measurement period for the source node is t_p=5s. These three parameter values are chosen based on the requirement discussed in Section 4.3, and we shall examine different parameter values in Section 4.5. We enable the control process at 20 second after the simulation in order to compare the performance change before and after the control operation.

Fig. 4.8 compares the performance with and without the feedback control under the JIT protocol. It can be seen from Fig. 4.8a that the loss rate of *Flow* 1-5 starts at about 14.5% in the beginning, and decreases after the control process is activated at t=30 second. It finally converges to 12% as required. The assembly timer for *Flow* 1-5 also increases from 100us to about 170us and remains around 170us after time 40 second as in Fig. 4.8b. It confirms that our timer-based control mechanism can achieve required loss rate successfully. With the increase of the assembly timer, the delay at the edge node increases as well as we have analyzed before, which is demonstrated in Fig. 4.8d. Our method does not introduce a severe increase on the delays and all types of delays become stable after the control process is stable. According to our analysis, the network throughput should remain unchanged during all processes. An interesting result is shown in Fig. 4.8c, in which the overall network throughput changes from 14.7Gbps to 15Gbps after activating the control process, which seems to be inconsistent with our argument that the overall throughput should be a constant. As discussed before, this small discrepancy does not affect controlling the loss rate which is the primary objective of our mechanism.

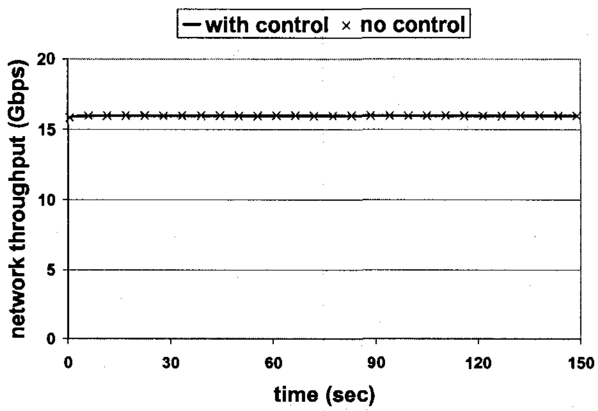
Fig. 4.9 depicts our control process on the same network using JETtailSeg protocol. In Fig. 4.9a, the loss rate of *Flow* 1-5 is about 0.22%. After the feedback control starts to work at t=20 second, it generates a transient period of about 15 seconds until the loss rate stays around 0.18% as required. In Fig. 4.9b it also takes about 15 seconds for the assembly timer changing from 100us to around 160us. After time 35 second, both the loss rate and the assembly timer remain around above values respectively. It is again a strong evidence that our proposed method is a successful control mechanism. In Fig. 4.9c the overall network throughput keeps almost no change at 16Gbps before and after the control operation which is consistent with our previous analysis. The delay performance (delay definitions can be found in Section 2.5) at the source edge varies with the changing assembly timer in Fig. 4.9d which can be easily explained by the illustration in Section 4.4.1. Comparing to the JIT results in Fig. 4.8, the control process in the JETtailSeg system generates more ripples. This is probably due to the burst segmentation technique which causes the characteristics of burst length distribution to become more irregular which in turn introduces more fluctuations on the system.



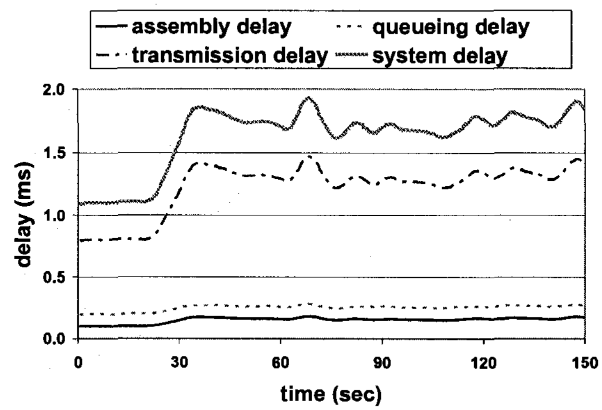
(a) Loss rate of *Flow* 1-5



(b) Assembly timer for *Flow* 1-5



(c) Network throughput



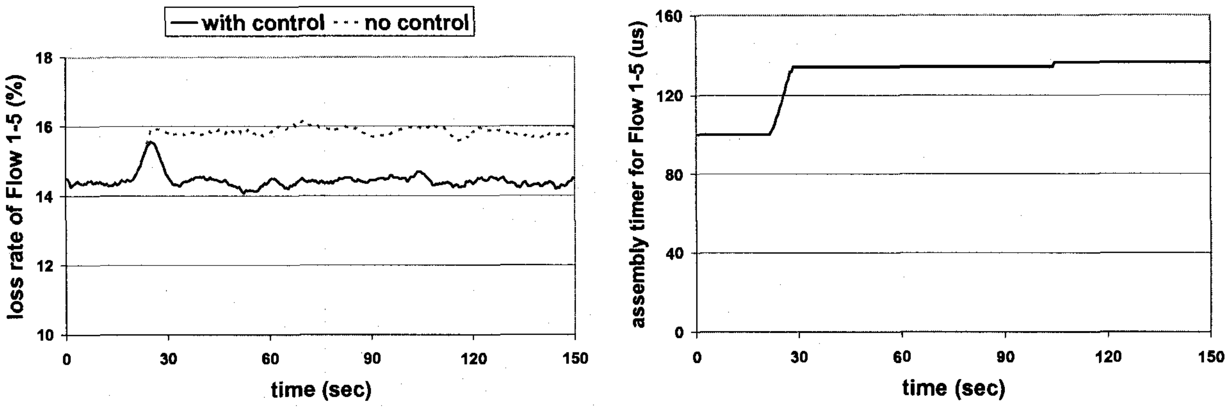
(d) Delay at E1

Fig. 4.9. Controller performance using JETtailSeg

Our control scheme also works well for other packet arrival processes and packet length distributions. Some of these results are shown in Appendix F and they all confirm our control process is independent of the distributions of packet arrival or packet length.

4.4.3 Dynamic Networks

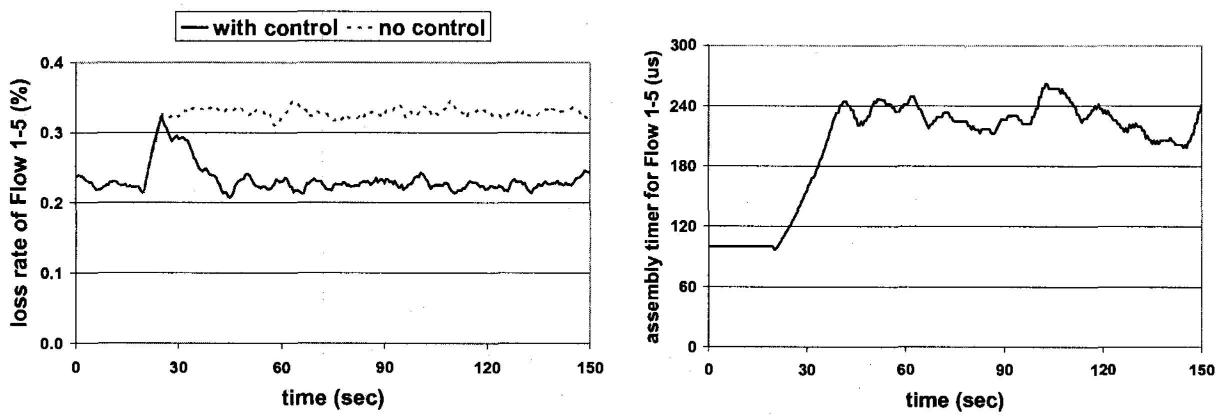
We now test how our control mechanism keeps the loss rate of *Flow* 1-5 stable in dynamic networks. We still use the same network settings except that the load of *Flow* 2-3, 3-4 and 4-5 will increase from 0.5 to 0.53 at *t*=20 second. Obviously, the loss rate of *Flow* 1-5 will increase since more conflicts occur at a higher load. We will use our control process to remove this disturbance. Our objective is to keep the loss rate of *Flow* 1-5 stable all the time and keep the allowable margin δ to be 5%.



(a) Loss rate of *Flow* 1-5 (b) Assembly timer for *Flow* 1-5

Fig. 4.10. Controller performance using JIT

When the JIT protocol is used, it shows the loss rate of *Flow* 1-5 along with the timeline in Fig. 4.10a. The loss rate starts at around 14.5% and increases to 16% eventually if no control mechanism is adopted. If our feedback control scheme is enabled, the loss rate remains around 14.5% after a convergence period for about 13 seconds. This process is implemented by the adjustment of the assembly timer. From Fig. 4.10b we can see that the assembly timer for *Flow* 1-5 increases from 100us before t=20 second to about 135us after t=33 second. Then the system stays at a stable status. Summing up, our feedback control scheme successfully keeps the loss rate of *Flow* 1-5 in its original range.



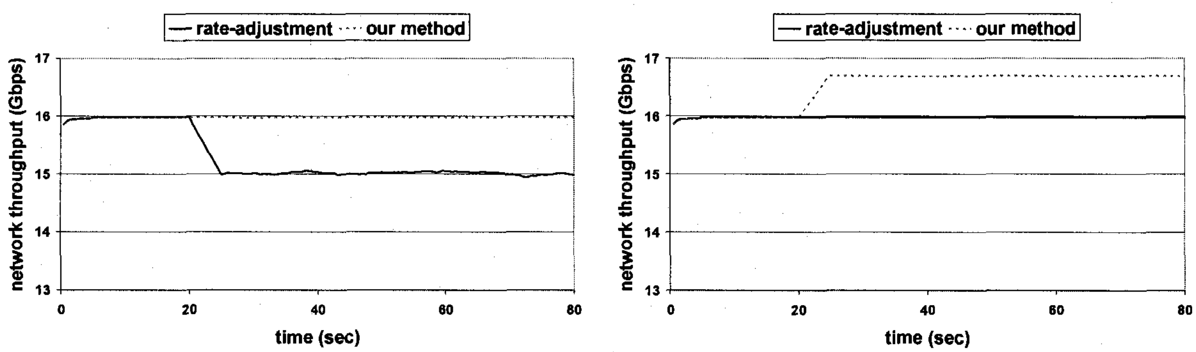
(a) Loss rate of *Flow* 1-5 (b) Assembly timer for *Flow* 1-5

Fig. 4.11. Controller performance using JETtailSeg

Fig. 4.11a shows the loss rate of *Flow* 1-5 along with the timeline when JETtailSeg protocol is used. The loss rate starts at around 0.22% before t=20 second and increases to 0.33% eventually if no control mechanism is adopted. If our feedback control scheme is enabled, it takes about 25 seconds for the loss rate to converge back to 0.22%. This process is implemented by the adjustment of the assembly timer for *Flow* 1-5 from 100us before t=20 second to about 230us after t=45 second as illustrated in Fig. 4.11b. Then the system stays at a stable status. Note that a lower loss implies a higher throughput and the system delay is proportional to the assembly timer as demonstrated in static networks, although they are not repeated here. Summing up, our feedback control scheme successfully keeps the loss rate of *Flow* 1-5 in its original range.

Similar to the static networks, our control scheme also works well in a dynamic network with other packet arrival or packet length distributions, some of which are shown in Appendix F. All results have again confirmed the independence of our control process to these parameters.

So far, we have assumed a zero switching time and header processing time in order to facilitate the controller design. In a realistic case, they are not negligible. It is easy to see that our previous analysis in Section 4.2 based on approximation and comparison should still be valid to relate the loss rate to the assembly timer. We have run some simulations to verify our control algorithm by using non-zero switching time and header processing time. Since the behavior of the network is similar, we do not repeat them here. Instead, one can see Appendix G for those results.



(a) Static network (b) Dynamic network

Fig. 4.12. Network throughput using JETtailSeg

4.4.4 Comparisons

We compare our designed method with the PCwER (Proportional Control algorithm with Explicit Reduction-request), which is a typical scheme for controlling the loss rate in OBS networks [FaZh05]. In this rate-adjustment scheme, when the offered load exceeds a threshold, a core node will notify each edge to reduce the transmission rate to him. Then each edge node has to buffer its overloaded traffic in order to affect the loss control at the core. In the comparison, we use the same settings as in the previous two sections where the control process starts from t=20 second.

Both methods are capable of implementing the loss control as previous sections, so they are not repeatedly depicted. However, our method has an obvious advantage in terms of the overall network throughput. As shown in Fig. 4.12a, our method remains 15.9Gbps throughput all the time in a static network, whereas the PCwER method (indicated as rate-adjustment in the diagram) decreases it to 15Gbps eventually. In a dynamic network, our method provides a higher throughput of 16.7Gbps after the control operation than 16Gbps of the PCwER method in Fig. 4.12b. Other scenarios demonstrate the same conclusions and are omitted here. Furthermore, we want to point out that, in the PCwER scheme and other rate-adjustment schemes reviewed in Section 1.1.2.3, a core node needs some global network information beforehand to calculate parameters for the control operation, which is not required by our method. So our method is more flexible and scalable.

4.5 Design Issues

Based on our experience on the analysis and simulations, we shall examine some important parameters using same static network settings and JETtailSeg protocol as before. Then we shall discuss a few critical issues related to the design and operation of our control scheme.

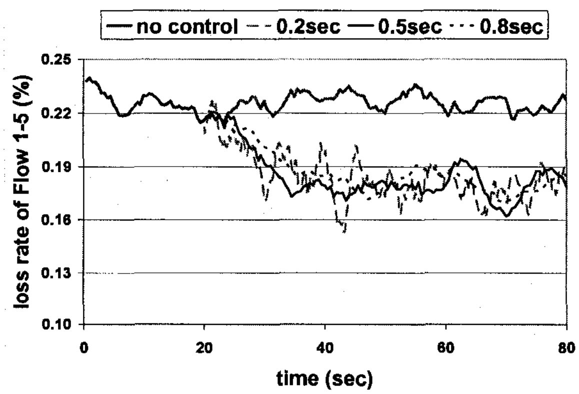


Fig. 4.13. Performance w.r.t. measurement interval t_m

4.5.1 Measurement Interval

The measurement interval t_m determines how often the destination node calculates the loss rate of a flow. Fig. 4.13 is an example that shows the change of loss rate when t_m is set to 0.2, 0.5 and 0.8 second. One can see that a short interval of 0.2 second reaches the required loss rate in about 10 seconds, while an interval of 0.5 second needs 15 seconds and an interval of 0.8 second needs 20 seconds of transient period. But the interval of 0.2 second also generates the largest variation, and the longest interval of 0.8 second generates the fastest convergence.

In general, if this interval is long, the response converges slowly. Although a short interval can make the destination to respond faster, too many FCPs may be generated to cause large oscillations.

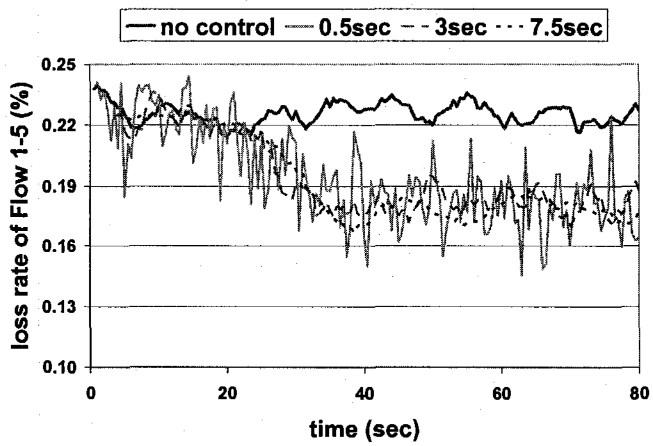


Fig. 4.14. Performance w.r.t. measurement period t_p

4.5.2 Measurement Period

The measurement period of the loss rate, t_p, affects how accurate the node is aware of the flow loss. If this period is long, the consecutive loss calculation has a small difference which may cause a slow convergence. On the other hand, a short period may cause large variation on the loss calculation which forces the system to be unstable. Fig. 4.14 depicts the control process regarding to three different measurement periods, 0.5, 3 and 7.5 seconds. It is shown that all three settings can achieve the desired loss rate. However, the shortest period of 0.5 second generates very large variation of the flow loss, and the longest period of 7.5 seconds generates the most stable loss rate, but the longest convergence period as well.

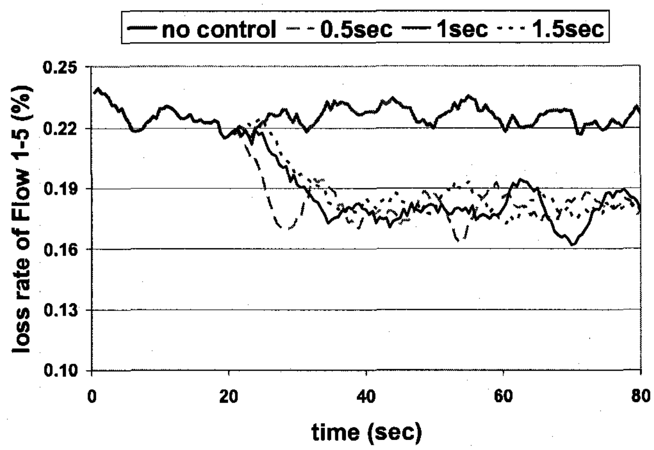


Fig. 4.15. Performance w.r.t. adjustment interval t_a

4.5.3 Adjustment Interval

The adjustment interval t_a determines how frequent the source edge responds to the adjustment demand. Fig. 4.15 depicts the effects of three intervals, 0.5 second, 1 second and 1.5 second. A t_a of 0.5 second costs about 7 seconds to converge to the required loss range, a t_a of 1 second about 10 seconds, and a t_a of 1.5 second about 12 seconds. Therefore, a shorter t_a is preferred to achieve a short transition period. On the other hand, a larger t_a generates less oscillation.

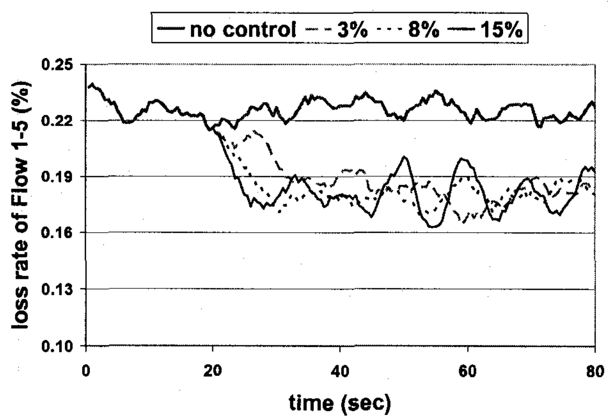


Fig. 4.16. Performance w.r.t. adjustment rate ε

4.5.4 Adjustment Rate

The adjustment rate ε determines how much adjustment the source edge reacts to the adjustment demand. From Fig. 4.16, we can see that a large adjustment rate of 15 percent generates the shortest convergence period and the highest variation before reaching the desired loss rate, while a small rate of 3 percent generates the longest convergence period and the least variation.

4.5.5 Range of the Loss Rate

The desired loss rate cannot be set too small due to two reasons. The first one is that it might affect the operation of other flows. Since our control scheme is designed to achieve a desired loss rate for a specific optical burst flow, while not considering the effect on other flows, it is possible that several flows have individual requirements at the same time, but the system cannot meet all of them due to the limit of the throughput capacity. The second reason is the constraints on the assembly interval. These constraints are due to the delay constraint imposed by an application. If a flow's loss rate cannot meet the demand for a long time even if the control process keeps working, we have to assume that the source node is limited by its bounds of assembly intervals and we need to rethink the design.

4.5.6 Other Issues

Beside the desirable operation and performance obtained by our controller, it is also important to note that in our flow control operation, a flow does not need any information of other flows in the network. Therefore, our controller is quite scalable in the heterogeneous Internet, which has been demonstrated by our many simulations (not shown). Furthermore, a source edge node can adjust the assembly timer without affecting the offered loads of other flows and therefore the overall network throughput. Consequently, our control process can keep the loss rate of the specified flow at a desired level, while avoiding the normal problem in other rate-adjustment control methods in OBS networks that the network throughput is suffered.

The use of exponential distribution in our design is for simplicity and tractability of our analysis. Our objective is to design a scheme that works, while the Poisson distribution owns good properties to enable us to fulfill this task. Other distributions have been tried, but Poisson is used to demonstrate the feasibility analytically. It is interesting to know that the interval distributions need not be exponential. For examples, Appendix H shows that improvement can be achieved even with normal and lognormal distributions. On the other hand, we have some intricate mathematics that we have not been able to obtain a closed-form solution even with exponential distributions (which have a nice memoryless property). Therefore, we have to resort to approximations with our analysis below

It is also significant that a simple but effective scheme can be derived from a simple observation and comparison between the measured loss rate and the required one. Although more sophisticated schemes might be used to have a better estimation of the change trend in order to achieve an explicit adjustment, more complicated and expensive components or more time-consuming algorithms might be required as well. As we know, the OBS scheme is thought to be suitable for the enabling technology without complex devices or time-consuming algorithms. Therefore, our proposed control method fits OBS networks exactly. This effective scheme appears to escape the attention of researchers on its application in terms of flow loss control.

All results in this chapter are used to demonstrate the effectiveness of our proposed control scheme and the effects of different parameters, while they may not have a very low loss rate, a minimum convergence period or minimum oscillation. All parameters being studied in this section can cause either a shorter convergence period or a more stable response, but never at the same time. Since the design of a control process depends on the tradeoff of QoS requirements, this observation has allowed us to examine different packet arrival processes, packet length distributions and signaling protocols in either static or dynamic network settings, which exhibits similar effects. Due to the space limit, those results are not shown here.

4.6 Concluding Remarks

This chapter has designed an effective flow control mechanism based on adaptive burst assembly in OBS networks. The burst assembly algorithm enables a good understanding of burst characteristics and efficient flow control. By adjusting the assembly timer of a flow, we can achieve the required loss performance in either static or dynamic networks using either JIT or JETtailSeg protocol. The control of a flow is implemented by the source and destination edge nodes without any global knowledge of the network or the core nodes. Our method has overcome this shortcoming of conventional control methods which adopt rate-based congestion control algorithms and suffer lower network throughput in an OBS network. Our performance evaluation results have demonstrated the validity and effectiveness of our algorithm. Some important issues related to our design and operations have been discussed as well.

CHAPTER 5
FLOW CONTROL IN OBS NETWORKS: A GENERIC APPROACH

The control scheme in Chapter 4 is limited to some specific burst assembly algorithm and signaling protocols. We now extend this control concept to more general cases. We shall design several predictor and adjuster models to implement the flow loss control. We then study different combinations of these predictors and adjusters. We shall examine the controller performance under different scenarios.

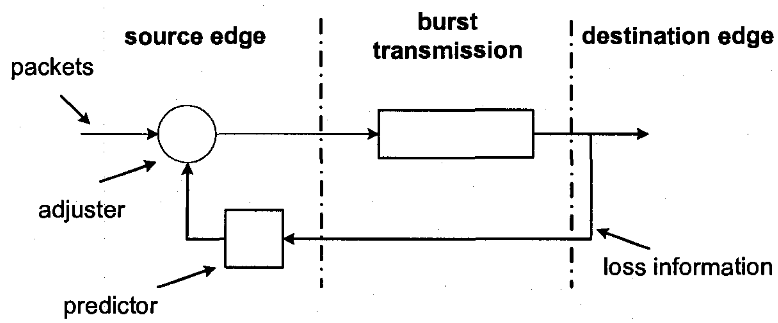


Fig. 5.1. Description of the control scheme

5.1 System Description

The basic system is the same as described in Section 4.1, i.e., the destination node sends FCPs to the source periodically about the traffic it has received. Consider the assembly procedure in one buffer at edge node *s*, which stores bursts of *Flow* $s-d$ to be sent to destination *d*. Let A represent the assembly parameter, i.e., the timer or threshold for burst assembly, and $Burst(s,d)$ represent the bursts in *Flow* $s-d$. The burst assembly is determined by A which starts immediately from the arrival of the first packet in a burst. It terminates when A reaches a predefined limit, and a new burst starts when another packet arrives. One of the following typical assembly algorithms A1-A3 can be implemented at the edge nodes.

A1: Timer-based: here A is the time threshold in seconds.

A2: MaxPktNum -based: the threshold A=MaxPktNum, which is the number of packets allowed in a burst.

A3: MaxBurstLen-based: the threshold A=MaxBurstLen, which is the length bits of a burst allowed.

Since the loss rate of a flow is directly related to the signaling protocol and contention resolution scheme at the core nodes, any one of the two signaling protocols, JETheadSeg (P1) and JETtailSeg (P2) as discussed in Section 2.3, will be sued for their efficiency.

Fig. 5.1 is an extended model of the basic one in Fig. 4.1 for the burst flow controller at the source edge. The destination node measures the received bits of a flow every t_m seconds, and sends a FCP to the source edge node. In our network, the source edge knows the propagation delay of each flow originating from it (it is measured at the initial state of a network) and keeps a record of the sending burst-length during each t_m interval. It then calculates the loss rate of that period by comparing the FCP information and the record. The source edge interprets the FCPs as follows. First, the predictor estimates the loss rate in the next interval based on a number of past FCPs. Then the adjuster will modify the burst assembly process in order to steer the loss rate to the target range. Since the adjustment of one flow will affects the performances of other flows, we cannot reduce the loss rate of a flow indefinitely to zero, but can only adjust it to within an acceptable range. At the same time, we assume other flows have certain loss tolerance caused by this control process. Our proposed mechanism is used to control the loss performance of a protected flow in case other flows misbehave or network settings are changed.

The three basic assembly algorithms A1-A3 only adopt a single timer or threshold, which can cause a high delay or large length variation when the offered load is too low or too high. To alleviate this problem, we employ a supplemental constraint as done in the previous chapter. That is, we add a lower and an upper bound on the burst length in the timer-based algorithm. These length constraints are far away from the average length in the timer period. Thus, the timer still places the major role and the other ones just avoid extraordinary bursts, which have a tiny probability, so the overall burst traffic still satisfies the above relationships. For simplicity, we shall emphasize the basic parameter only, while not explicitly mention the supplemental ones later.

5.2 Analysis and Verification

We shall investigate the effects of the assembly parameters on the flow loss by analyzing the traffic characteristics of burst flows. The influences of different parameters are approximated, compared and utilized in our controller design.

5.2.1 Loss Analysis

Let $\lambda_{s,d}$ (bursts/sec) be the average generation rate, and $L_{s,d}$ (sec) be the random variable for the length of $Burst(s,d)$. Also let $Burst(s,d,l)$ be the bursts on *Flow* $s-d$ arriving to link *l*, $\lambda_{s,d,l}$ (bursts/sec) be its average arrival rate, and $L_{s,d,l}$ (sec) be its length variable. According to the assumptions of independent packet arrival and *i.i.d.* packet length distribution, the number of packets in a burst and the length of a burst are proportional to the assembly parameter *A* (from the central limit theory [Klei75]). Although the contention resolution may modify the burst interarrival, this effect is tiny since the loss rate is low in a moderately loaded transport network. We can see that

1) If *A* decreases, $\lambda_{s,d}$ and $\lambda_{s,d,l}$ increase, while $L_{s,d}$ and $L_{s,d,l}$ decrease.
2) If *A* increases, $\lambda_{s,d}$ and $\lambda_{s,d,l}$ decrease, while $L_{s,d}$ and $L_{s,d,l}$ increase. (5.1)

Assume there are *N* destinations and *W* output channels for source node *s*. Considering the independent assembly processes of each flow, the generated bursts from the output channels of an edge node to all destination nodes can be approximated by an *M/G/W* queueing system. The delay performance of such a system has been analyzed in [Klei75]. Since the burst length in a flow is related to *A*, the system delay of packets at the edge nodes is proportional to *A* consequently.

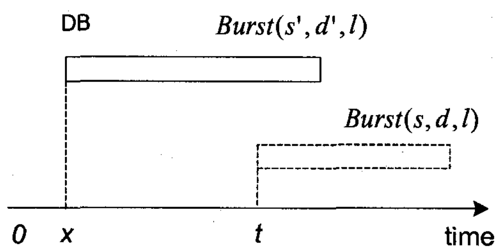


Fig. 5.2. Loss analysis of burst dropping

We shall analyze the effect of the assembly parameter on the loss rate of a flow when different protocols are adopted by the network. As discussed in Section 4.2.2, we only need consider conflicts from different flows, which can be divided into three types: a whole burst is dropped, the header part of a burst is dropped or the tail part of a burst is dropped. We shall analyze these three cases respectively.

Denote the loss rate of $Burst(s,d,l)$ on link *l* by $p_{s,d,l}$. Assuming independence of losses among all links, the loss rate of *Flow* $s-d$, $p_{s,d}$, can be approximated from

$$p_{s,d} = 1 - \prod_{\forall\, l \in (1,\dots J)} (1 - p_{s,d,l}) \,. \tag{5.2}$$

5.2.1.1 Case of burst dropping

In our P1 protocol, a burst can be dropped if there is no free channel before its ending time. Let us consider the case when two bursts from *Flow* $s-d$ and *Flow* $s^{'}-d^{'}$ are contending for a wavelength channel on link l ($l \in (1,2,\dots,J)$). When *Burst* (s,d,l) arrives at time t ($t>0$), it finds the channel is occupied by *Burst* $(s^{'},d^{'},l)$ who has arrived at time x ($x<t$). Referring to Fig. 5.2, *Burst* (s,d,l) will be dropped *iff* $x<t$ and $L_{s^{'},d^{'},l} > (t-x)$. Let $A_{s,d,l}(t)$ be the number of *Burst* (s,d,l) arrived during period t, $\Pr[A_{s,d,l}(t)=n]$ be the probability that n bursts on *Flow* $s-d$ have arrived to link l during a period t, and $X_{s,d,l}$ be the burst arrival time. Since the burst arrival time and the burst length distribution are independent of each other, the loss rate of *Burst* (s,d,l) at time t can be obtained as follows (using the conditioning and un-conditioning technique)

$$\begin{aligned}
p_{s,d,l}(t) &= \Pr[\mathit{Burst}(s^{'},d^{'},l) \text{ is longer than } t-x \,|\, \text{no other } \mathit{Burst}(s^{'},d^{'},l) \text{ arrives during } t-x] \\
&\quad \cdot \Pr[\text{no other } \mathit{Burst}(s^{'},d^{'},l) \text{ arrives during } t-x \,|\, \mathit{Burst}(s^{'},d^{'},l) \text{ arrives at time } x] \\
&\quad \cdot \Pr[\mathit{Burst}(s^{'},d^{'},l) \text{ arrives at time } x,\ x<t] \\
&= \int_{x=0}^{t} \Pr[L_{s^{'},d^{'},l} \ge (t-x) \,|\, A_{s^{'},d^{'},l}(t-x)=0] \cdot \Pr[A_{s^{'},d^{'},l}(t-x)=0] \cdot f_{X_{s^{'},d^{'},l}}(x) \cdot dx \\
&= \int_{x=0}^{t} \Pr[L_{s^{'},d^{'},l} \ge (t-x)] \cdot \Pr[A_{s^{'},d^{'},l}(t-x)=0] \cdot f_{X_{s^{'},d^{'},l}}(x) \cdot dx \,.
\end{aligned} \tag{5.3}$$

Similarly, we can obtain the loss rate of *Burst* $(s^{'},d^{'},l)$ at time t

$$p_{s^{'},d^{'},l}(t) = \int_{x=0}^{t} \Pr[L_{s,d,l} \ge (t-x)] \cdot \Pr[A_{s,d,l}(t-x)=0] \cdot f_{X_{s,d,l}}(x) \cdot dx \,. \tag{5.4}$$

If we now increase the assembly parameter of *Flow* $s-d$ from A to $A^{'}$, the following results are generated.

1) The arrival rate of *Burst* (s,d,l) will decrease, $\lambda^{'}_{s,d,l} < \lambda_{s,d,l}$, according to (5.1). Consequently, $\Pr[A_{s,d,l}(t-x)=0]$ increases, i.e., $\Pr[A^{'}_{s,d,l}(t-x)=0] > \Pr[A_{s,d,l}(t-x)=0]$.
2) The burst length will increase, $\Pr[L^{'}_{s,d,l} \ge (t-x)] > \Pr[L_{s,d,l} \ge (t-x)]$, according to (5.1).
3) The arrival of *Burst* $(s^{'},d^{'},l)$ is not affected.

Considering the above results, we obtain

$$\int_{x=0}^{t} \Pr[L^{'}_{s,d,l} \ge (t-x)] \cdot \Pr[A^{'}_{s,d,l}(t-x)=0] f_{X^{'}_{s,d,l}}(x)dx \ > \int_{x=0}^{t} \Pr[L_{s,d,l} \ge (t-x)] \cdot \Pr[A_{s,d,l}(t-x)=0] f_{X_{s,d,l}}(x)dx \ ,$$

so $p'_{s',d',l}(t) > p_{s',d',l}(t), t \ge 0$, which in turn yields $p'_{s',d',l} > p_{s',d',l}$. According to (5.2), the loss rate of *Flow* $s'-d'$ increases. Since other settings of the network do not change, *Flow* $s-d$ will have more chances to utilize the channel. Finally, $p_{s,d}$ (the loss rate of *Flow* $s-d$) becomes lower.

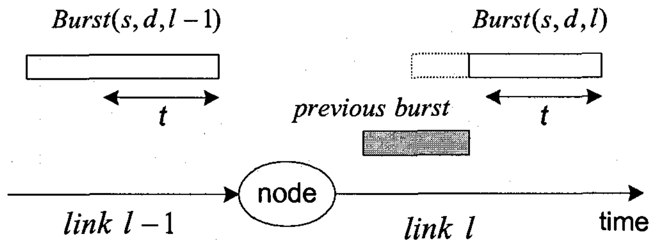


Fig. 5.3. Loss analysis of head dropping

5.2.1.2 Case of burst-header dropping

In our P1 protocol, a burst can be truncated by another burst that arrives before its starting time. If $Burst(s,d,l)$ is truncated by dropping its header part as shown in Fig. 5.3, its length $L_{s,d,l}$ also affects the loss rate. We can use similar method for the next case to find out the effect of the assembly parameter. Here we adopt a simpler approximation. Suppose the assembly parameter is increased for $Burst(s,d,l)$. Then $L_{s,d,l}$ tends to have a larger value. Assume the previous burst on link-*l* is the same, the dropping part on $Burst(s,d,l)$ does not change. However, the dropping portion of $L_{s,d,l}$ will become smaller, which generates a lower loss. From (5.2), the loss rate of *Flow* $s-d$ will decrease eventually, $p'_{s,d} < p_{s,d}$.

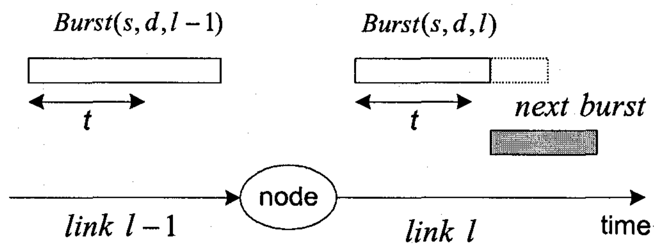


Fig. 5.4. Loss analysis of tail dropping

5.2.1.3 Case of burst-tail dropping

In our P2 protocol, a burst can be truncated by another burst that arrives before its ending time. The length of $Burst(s,d,l)$, $L_{s,d,l}$, is longer than t (*t>0*) *iff* this burst is longer than t on the previous link (assuming to be *l-1,* corresponding to $Burst(s,d,l-1)$), and no other bursts (including all flows sharing this link) has arrived to link *l* during this *t* period, as illustrated in Fig.

5.4. The first condition guarantees the burst length and the second condition ensures the burst has not being dropped or truncated until time t. Then we have

$$\begin{aligned}\Pr[L_{s,d,l} \geq t] &= \Pr[Burst(s,d,l) \textit{ is longer than } t] \\ &= \Pr[Burst(s,d,l-1) \textit{ is longer than } t \mid \textit{no burst arrives to link } l \textit{ in } t \textit{ period}] \\ &\qquad \cdot \Pr[\textit{no burst arrives to link } l \textit{ in } t \textit{ period}] \\ &= \Pr[L_{s,d,l-1} \geq t \mid A_l(t)=0] \cdot \Pr[A_l(t)=0] = \Pr[L_{s,d,l-1} \geq t] \cdot \Pr[A_l(t)=0]\end{aligned}$$

We then obtain the *cdf* of $L_{s,d,l}$ from the above equation.

$$F_{L_{s,d,l}}(t) = \Pr[L_{s,d,l} < t] = 1 - \Pr[L_{s,d,l} \geq t] = 1 - \Pr[L_{s,d,l-1} \geq t]\Pr[A_l(t)=0] = 1 - [1 - F_{L_{s,d,l-1}}(t)]\Pr[A_l(t)=0]\,,\ t > 0\,.$$

The loss rate of $Burst(s,d,l)$ becomes

$$\begin{aligned}p_{s,d,l} &= 1 - \frac{E[\textit{burst length on link } l]}{E[\textit{burst length on link } l-1]} = 1 - \frac{\int_{t=0}^{\infty} t \cdot f_{L_{s,d,l}}(t) \cdot dt}{\int_{t=0}^{\infty} t \cdot f_{L_{s,d,l-1}}(t) \cdot dt} \\ &= 1 - \frac{\int_{t=0}^{\infty} [1 - F_{L_{s,d,l}}(t)] \cdot dt}{\int_{t=0}^{\infty} [1 - F_{L_{s,d,l-1}}(t)] \cdot dt} = 1 - \frac{\int_{t=0}^{\infty} [1 - F_{L_{s,d,l-1}}(t)] \cdot \Pr[A_l(t)=0] \cdot dt}{\int_{t=0}^{\infty} [1 - F_{L_{s,d,l-1}}(t)] \cdot dt}\end{aligned} \tag{5.5}$$

If we increase the assembly parameter A of *Flow* $s-d$, the arrival rate of $Burst(s,d,l)$ will decrease accordingly, $\lambda'_{s,d,l} < \lambda_{s,d,l}$. Since the total burst arrival rate to link l, λ_l, can be approximated as the sum of all possible bursts sharing this link (i.e., $\lambda_l = \sum_{\forall s-d} \lambda_{s,d,l}$), the burst arrival rate to link l becomes lower, $\lambda_l' < \lambda_l$, and consequently

$$\Pr[A_l'(t)=0] > \Pr[A_l(t)=0]\,,\ t > 0\,. \tag{5.6}$$

A larger assembly parameter allows the burst length $L'_{s,d,l-1}$ to have a larger value. Thus, for a given y, we have $\Pr[L'_{s,d,l-1} > y] > \Pr[L_{s,d,l-1} > y]$, $(y > 0)$. From $F_{L_{s,d,l-1}}(y) = 1 - \Pr[L_{s,d,l-1} > y]$, the *cdf* functions satisfy

$$F_{L'_{s,d,l-1}}(y) < F_{L_{s,d,l-1}}(y)\,. \tag{5.7}$$

By combing (5.6) and (5.7), we obtain

$$(1 - F_{L'_{s,d,l}}(t)) > (1 - F_{L_{s,d,l}}(t))\,, \tag{5.8}$$

and $\Pr[A_l'(t)=0] \cdot (1 - F_{L'_{s,d,l-1}}(t)) > \Pr[A_l(t)=0] \cdot (1 - F_{L_{s,d,l-1}}(t))$, $t > 0$. (5.9)

Substituting (5.8) and (5.9) into (5.5), we can see that both the numerator and the denominator will increase after increasing A. However, the numerator will increase faster than the denominator because of the coefficient $\Pr[A_l'(t)=0] > \Pr[A_l(t)=0]$. Consequently, $p_{s,d,l}$ will decrease, $p'_{s,d,l} < p_{s,d,l}$. From (5.2), the loss rate of *Flow* $s-d$ will decreases, i.e., $p'_{s,d} < p_{s,d}$.

In the above analysis, the header processing time and the optical switching time are not taken into account. Since the control traffic has a small volume in an OBS network, the FCFS principle still holds if the header processing time is considered. Moreover, the switching time of the optical core nodes is often fixed, so its effect is equivalent to subtracting each burst by a short fixed-length. One can see that neither of these two parameters would affect the effectiveness and conclusion of our analysis on the flow loss and the assembly parameter. Another assumption of wavelength conversion can be relaxed too. If no wavelength conversion is allowed, the number of available channels for a flow is reduced to the number of available fibers. This change does not affect our analysis.

In summary, all three cases have shown that the loss rate of a flow is a decreasing function of its assembly parameter. Since our previous analysis is based on comparisons and assumptions only and does not produce any closed-form equation, we shall verify our analysis through simulations in the next section in order to provide a strong support for our controller design. It can be seen that in addition to the loss rate of burst flows, the system delay of a packet at edge nodes and the effective throughput are also major performance concerns in our design.

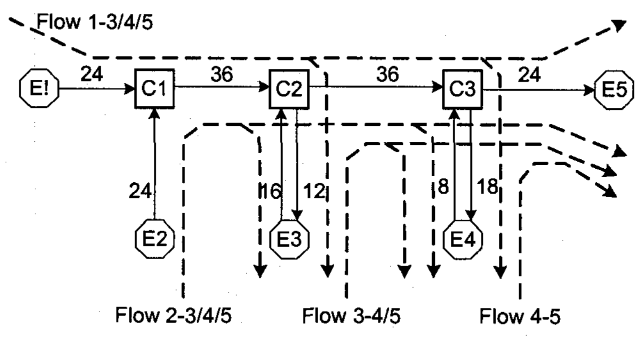
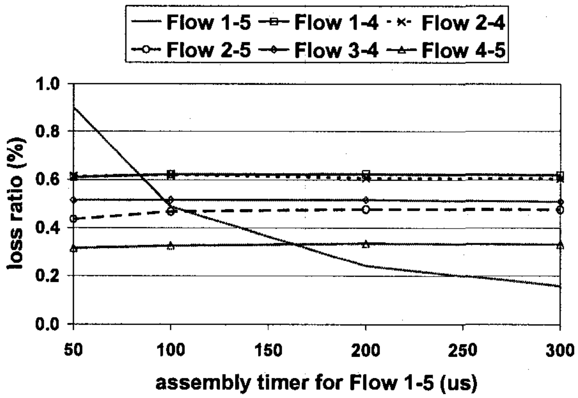


Fig. 5.5. Network model for simulations

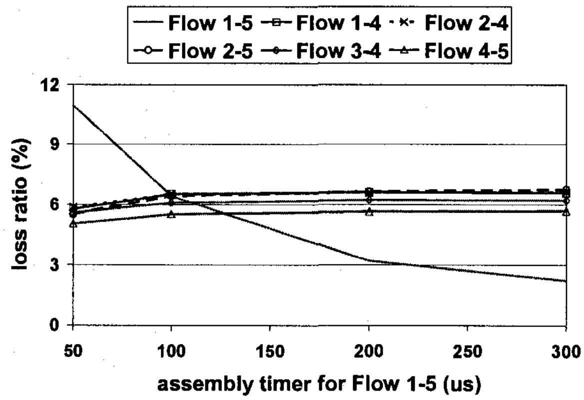
5.2.2 Verification

We shall use the network in Fig. 5.5 to verify our analysis. This network consists of five edge nodes E1-E5 and three core nodes C1-C3. The solid lines refer to the links for DB transmission and the numbers beside the link indicate the amount of wavelength channels for DBs which is chosen randomly. The control channels are not shown in the diagram. Nine burst flows are studied including *Flow* 1-3/4/5 (E1 -> E3, E4, E5), *Flow* 2-3/4/5 (E2 -> E3, E4, E5), *Flow* 3-4/5 (E3 -> E4, E5) and *Flow* 4-5 (E4 -> E5), each of which takes the routing path along the dashed lines. We will focus on *Flow* 1-5 to examine our analysis. Except the following basic parameters, other settings are the same as in Section 4.4.

- The assembly timer is 100us for A1
- The MaxBurstLen is 560Kbits for A2
- The MaxPktNum is 140 for A3
- The traffic load is 0.7



(a) Poisson traffic



(b) Pareto traffic

Fig. 5.6. Flow loss w.r.t. the assembly timer for Flow 1-5

Fig. 5.6 depicts the loss ratio of different flows in a P1-A1 network when the assembly timer for *Flow* 1-5 changes. Let us first examine the performance under the Poisson traffic in Fig. 5.6a. When the assembly timer is 100us, *Flow* 1-5 has a loss rate of 0.49%. If we double the assembly timer to 200us, the loss rate decreases to 0.25%. Therefore, the effect of the assembly timer on the flow loss is very significant. On the other hand, since the change of *Flow* 1-5 only affects some hops of other flows, the loss ratio of those flows are affected by a very smaller portion only. All measured results are consistent with our approximation.

Due to the self-similarity of Internet traffic, we also test the loss performance with the Pareto packet arrival process. As seen in Fig. 5.6b, a similar observation applies to the same P1-A1 network under Pareto packet traffic except a higher loss rate caused by more bursty nature of the Pareto distribution. When the assembly timer is 100us, *Flow* 1-5 suffers 6.3% loss, which decreases to 3.2% along with a timer of 200us. Other flows suffer slightly more losses along with an increasing timer for *Flow* 1-5, while the change is not very significant in the diagram. All simulation results have endorsed our previous analysis. Since the offered load does not change when we adjust the assembly parameter for *Flow* 1-5, a lower loss rate means a higher throughput for a flow. We have verified that other assembly algorithms and signaling protocols have similar performances as the ones in Fig. 5.6.

Table 5.1. Network Throughput (Gbps)

a) Algorithm A1: Poisson traffic	Assembly Timer for *Flow* 1-5 (us)	**50**	**100**	**200**	**300**
	P1	50.052	50.075	50.090	50.097
	P2	50.046	50.070	50.085	50.092
b) Algorithm A2: Poisson traffic	MaxBurstLen for *Flow* 1-5 (Kbits)	**280**	**560**	**1120**	**1680**
	P1	50.338	50.347	50.351	50.352
	P2	50.299	50.312	50.318	50.319
c) Algorithm A3: Poisson traffic	MaxPktNum for *Flow* 1-5	**70**	**140**	**280**	**420**
	P1	50.120	50.142	50.157	50.162
	P2	50.115	50.137	50.152	50.157
d) Algorithm A3: Pareto traffic	MaxPktNum for *Flow* 1-5	**70**	**140**	**280**	**420**
	P1	47.815	47.792	47.921	47.973
	P2	48.290	48.330	48.409	48.448

We let the source edge node adjust the assembly parameters only, while keeping the offered load of each flow at 0.7. Table 5.1 shows the network throughput regarding different assembly parameters (the assembly timer, MaxBurstLen or MaxPktNum) for *Flow* 1-5. Take Table 5.1a for example, when P1-A1 is adopted, the network throughput remains around 50.07Gbps when the assembly timer for *Flow* 1-5 changes from 50us to 300us. Tables 5.1b and 5.1c also demonstrate that the throughput of the whole network will not degrade along with an increasing MaxBurstLen or MaxPktNum when the input traffic follows Poisson. Finally, Table 5.1d depicts the network throughput using the assembly algorithm A3 under Pareto packet arrival, in which the changing of the throughput is almost indistinguishable too. Summing up, from all these observations plus others not shown, we can say that the efficiency of the whole system is not sacrificed when reducing the loss rate of a specific flow.

5.2.3 Applications

Our feedback control mechanism based on above observations can be used to provide a required flow loss by modifying the assembly parameter: the loss rate of a flow becomes lower when its assembly parameter increases. The first application is to maintain the loss rate of a flow in case of performance downgrading in a dynamic network. In this scenario, if the monitored loss rate exceeds the acceptable range, the source edge node increases the assembly parameter of this flow until the loss rate remains in the acceptable range. On the other hand, the loss rate of a flow is stable in a static network, so we can apply our control scheme to reduce the loss rate to a desired value in order to meet the demand of certain applications in this type of network. These functions would be implemented and studied by the flow controllers designed in the next section.

5.3 Controller Design

Two important components in our control scheme, the predictor and the adjuster as shown in Fig. 5.1b, can be implemented in different formats. We have designed several predictor and adjuster models as follows. These models have a low time-complexity to meet the requirements of OBS networks. They are also effective as demonstrated in our simulations later on.

5.3.1 Predictor Design

The predictor is used to estimate the loss rate based on the past feedback information, which is then used as a reference for the adjustment of the assembly parameter. Therefore, a good predictor should have a precise accuracy, i.e., a close match of the predicated loss rate and the actual loss rate in the future. We have designed three types of predictors in this thesis.

As mentioned before, every FCP is used to calculate the loss rate in an interval by comparison with the recorded burst lengths. Assume the loss rate of each interval to be $p_{s,d}(i), i = 0,1,\dots$, where i is the number of past intervals counting from the current one (so the latest loss rate is denoted by $p_{s,d}(0)$). When the next FCP arrives, the sequences of all past loss rate will shift by one. We use a moving window t_p to observe the performance. Define $Z = \lfloor t_p / t_m \rfloor$. Let $p_{avg} = \frac{1}{Z}\sum_{i=0}^{Z-1} p_{s,d}(i)$ be the average loss rate and $p_{predict}$ represent the prediction result.

Predictor #1 (Prd1): Using the average loss rate over last L intervals

This method is achieved by simply setting $p_{predict} = \frac{1}{L}\sum_{i=0}^{L-1} p_{s,d}(i)$. Here the choice of L is critical in determining the accuracy of the prediction. There is a clear trade-off between measuring the loss rate over a short period of time and responding rapidly to changes in the available bandwidth, versus measuring over a longer period of time and getting a signal that is much less noisy. If L is large, $p_{predict}$ is more accurate in terms of the long term average, but is insensitive to the instantaneous change. If L is small, $p_{predict}$ can reflect the instantaneous change more accurately, but may cause large variations on the loss calculation which forces the system to be unstable. A network operator can choose an appropriate L based on his experience. Another problem is that all intervals are treated equally in this method. However, the bursts in different intervals have different effects on the later bursts, which can not be reflected by this method. The major advantage of this predictor is its simplicity.

Predictor #2 (Prd2): Using EWMA (Exponentially Weighted Moving Average) loss rate

This method has been widely adopted in the TCP congestion control [FlJa93, Jaco88], which has been proved to be very successful. Let w_{EWMA} to be the weight. Then the next loss rate can be approximated by $p_{predict} = (1 - w_{EWMA}) \cdot p_{avg} + w_{EWMA} \cdot p_{s,d}(0)$, which is equal to a low-pass filter. The weight value affects the influence of past bursts. A large w_{EWMA} can reduce the error on the mean calculation, but a small w_{EWMA} can reduce the damage of the fluctuation. Thus, it is critical to choose an appropriate w_{EWMA} from past experience.

Predictor #3 (Prd3): Using the weighted loss rate over last L intervals

The loss rate in this method is calculated over the last L loss ratio which are generated from the FCP packets, i.e., $p_{predict} = \frac{\sum_{i=1}^{L} w_i \cdot p_{s,d}(i)}{\sum_{i=1}^{L} w_i}$. The weights are calculated by [FlHa00]

$$w_i = \begin{cases} 1, & 0 \le i < L/2 \\ 1 - \dfrac{i - L/2}{L/2+1}, & L/2 \le i < L \end{cases}.$$

It is dependent on L to achieve the accuracy of prediction. A large L generates a stable response, but costs a longer time to become stable. A small L generates a fast response, but is less resilient to the possible fluctuations of the measured loss. Furthermore, the latest bursts have more influences on the future loss, which is reflected by the difference of the chosen weights. Thus, this predictor is also expected to have a better performance than Prd1.

5.3.2 Adjuster Design

Based on the prediction result and the required loss rate θ, the assembly parameter A can be adjusted by different ways. We have designed three types of adjusters.

Adjuster #1 (Adj1): Fixed-rate adjustment

The assembly parameter is adjusted by a fixed-rate K_{fixed} each time based on the comparison of the predictor result and the desired loss rate.

1) If $p_{predict} > \theta$, $A \leftarrow A \cdot (1 + K_{fixed})$.

2) If $p_{predict} < \theta$, $A \leftarrow A \cdot (1 - K_{fixed})$.

The coefficient K_{fixed} can be determined by the experience. Since the adjustment rate is fixed for all scenarios, when the predicted loss rate becomes closer to θ, this adjustment may be too large to cause oscillations. When the predicted loss rate is far from θ, this adjustment may be too slow and generate a lagging response. The advantage of this method is its simplicity.

Adjuster #2 (Adj2): Proportional adjustment

By comparing the predictor result and the desired loss rate, we adjust the assembly parameter by $A \leftarrow A \cdot (1 + K_{prop} \cdot \frac{p_{predict} - \theta}{\theta})$, where K_{prop} is a constant. This method treats the adjustment according to the difference of the future loss and the desired one. When this difference is large, the adjustment is larger. When this difference is small, the adjustment becomes smaller too. Thus, this method is assumed to have a better response than Adj1, but it has a little bit higher complexity. Note that the coefficient K_{prop} can be determined by the experience too.

Adjuster #3 (Adj3): Fixed+Proportional adjustment

This adjuster is a combination of the previous two, i.e., Adj1 and Adj2. The assembly parameter is adjusted by both a fixed rate K_{fixed} and a proportional part K_{prop}.

1) If $p_{predict} > \theta$, $A \leftarrow A \cdot (1 + K_{fixed} + K_{prop} \cdot \frac{p_{predict} - \theta}{\theta})$.

2) If $p_{predict} < \theta$, $A \leftarrow A \cdot (1 - K_{fixed} + K_{prop} \cdot \frac{p_{predict} - \theta}{\theta})$.

This method uses a fixed-rate to accelerate the adjustment, and a proportional adjustment to reflect the difference of the future loss and the target loss as well. In this way, we expect the response to be both fast and stable by combining the advantages of Adj1 and Adj2. The coefficients can be determined in the same way as in Adj1 and Adj2.

5.3.3 Predictor-Adjuster Combinations

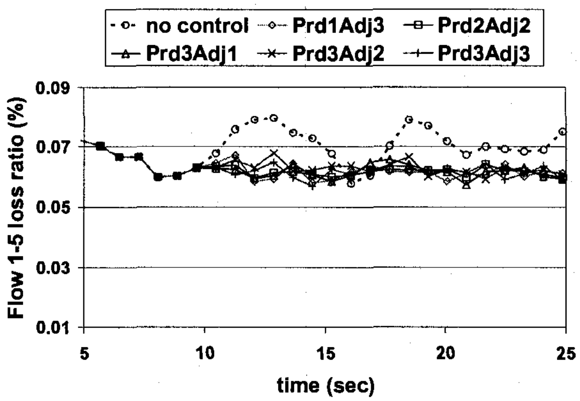
Our complete OBS flow controller includes a predictor and an adjuster, which changes its operations at periodic intervals. It is known in control theory that such on-off mechanism may lead to oscillations that can exhibit complex and even chaotic behavior. Therefore, the predictor and the adjuster need to coordinate each other in order to achieve a quick and stable response. So far, there is no obvious evidence of a preferred combination since each of them appears to work successfully as demonstrated in the next section. However, by steering the parameters carefully based on experiences, some combinations may have a little bit better response than others.

An important fact to consider is that the source node does not respond to an FCP until its last adjustment has passed at least t_a seconds in order to avoid oscillations. Obviously, a shorter t_a is preferred to achieve a shorter transition period. On the other hand, a larger t_a generates less fluctuation. The measurement interval t_m determines how often the destination node calculates the performance of a flow. If this interval is long, the response is slow. On the other hand, a short interval causes the destination to respond promptly, but send out too many FCPs, which may generate larger oscillation. A controller design depends on the tradeoff of QoS requirements to choose either a larger or a smaller one. When the loss rate is very close to the target value, a small change may cause the controller oscillation. To prevent this problem, we set a bound by allowing the loss rate within a small range of δ. When the predicted loss rate falls within this range of the target loss, there is no adjustment operation.

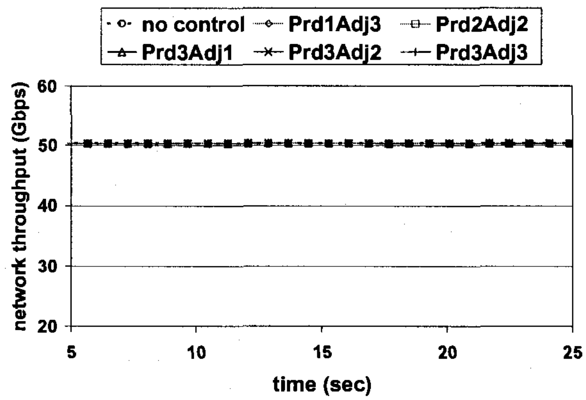
We want to emphasize that each application has some constraints on the delay performance which limits the tunable range of the assembly parameter. This will be reflected in a flow not achieving its target loss rate after the control process keeps updating the assembly parameter for a long time. In this case, we assume other mechanism or network entity exists and is expected to take over to solve the problem.

5.4 Performance Evaluation

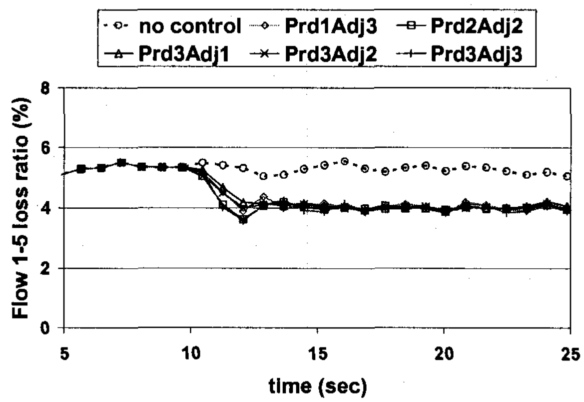
We shall apply our flow controllers to the network in Fig. 5.5 to examine the behavior of the controller. The basic parameters are the same as those in Section 5.2.2. In light of the discussions in Section 5.3.3, we have determined the following good set (not optimal) of parameters based on our experience. The edge node waits at least t_a=50ms between two consecutive adjustments, all performance measures are recorded every t_m=50ms, the measurement window is t_p=2sec, and the allowable margin is $\delta = 3\%$. Our control process will be imposed on *Flow* 1-5. One can see that other flows represent different influences in a real network, therefore, *Flow* 1-5 is a typical case in a general network. Our control scheme works well for different scenarios, although only some of them are shown here due to a limited space. Based on our experience, we set $Z = 40$, $L = 6$, $w_{EWMA} = 0.5$, $K_{fixed} = 0.05$ and $K_{prop} = 0.05$.



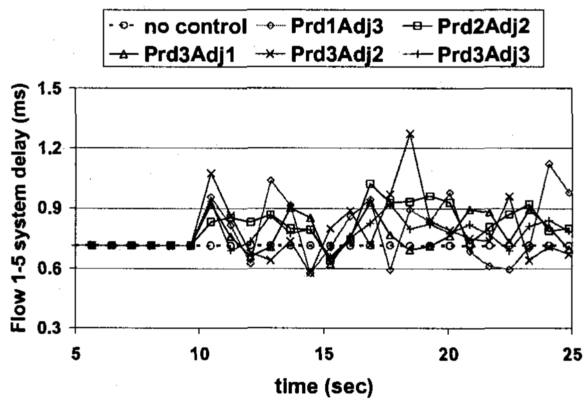
(a) Loss ratio: Poisson traffic



(b) Network throughput: Poisson traffic



(c) Loss ratio: Pareto traffic



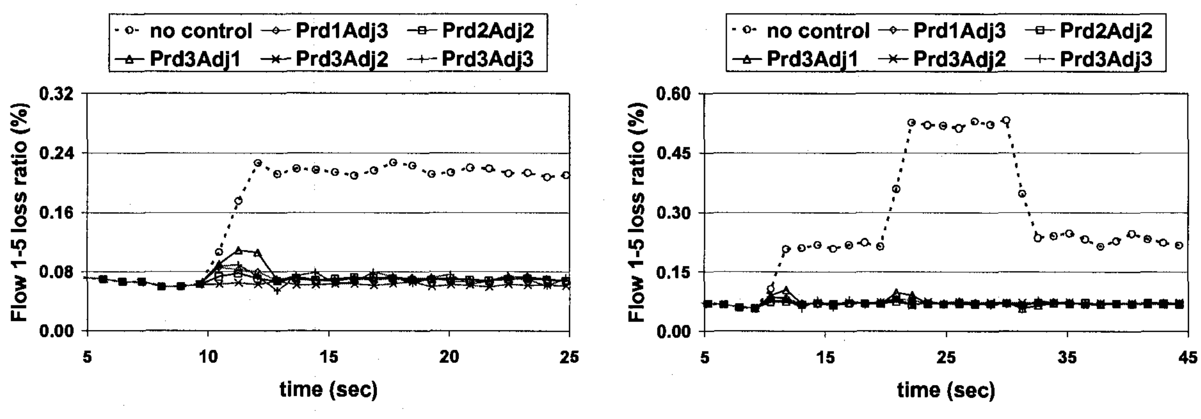
(d) System delay: Poisson traffic

Fig. 5.7. Performance of static networks

5.4.1 Static Networks

Fig. 5.7 demonstrates the performance of the control scheme on *Flow* 1-5 in a static network, i.e., when the input load is a constant for each flow. The assembly algorithm and the signaling protocol are A2 and P1 respectively. We enable the control process at t=10 second in order to compare the performance change before and after the control process. For the case using Poisson traffic, we set $\theta = 0.06\%$. For different controllers, Fig. 5.7a shows that our design can successfully reach the target loss rate for *Flow* 1-5 with a good convergence and stability. Fig. 5.7b depicts how the network throughput is hardly affected by the control operations at all. In Fig. 5.7d, the system delay increases from 0.7ms to around 0.8ms on the average after the control process is activated. Some controllers, e.g., Prd3Adj2, generate a longer delay than other controllers occasionally. This is probably due to a large parameter used in the adjuster. However, if we take into account the propagation delay, the average end-to-end delay of any controller increases by about 1% only, so our control mechanism does not affect the delay performance significantly.

When the input packet traffic follows a Pareto arrival process, a higher loss will occur due to the more bursty nature. As shown in Fig. 5.7c, *Flow* 1-5 has about 5.6% loss in such a network. We then set the target loss θ to be 4%. Similar to the Poisson case in Fig. 5.7a, our controllers also implement the control task successfully. The throughput and delay performances are also similar to the Poisson case, which are omitted here. All other combinations of different protocols, assembly algorithms, predictors and adjusters can accomplish the control task successfully with a good convergence and stability, and demonstrate similar performances as well.



(a) Load change (b) Load change + channel failure

Fig. 5.8. Performance of dynamic networks under Poisson traffic

5.4.2 Dynamic Networks

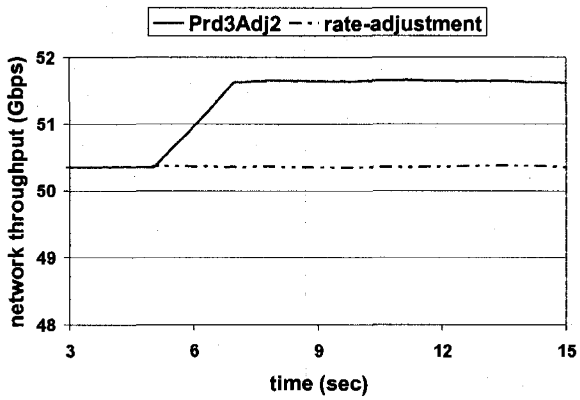
Next we shall test how our control mechanism keeps the loss rate of *Flow* 1-5 stable when a network using P1-A2 is offered a varying load. We still use the same settings as in the static case except that at t=10 second the input traffics at E2/E3/E4 will increase their load from 0.7 to 0.72. In a bufferless OBS network, a small load change would cause a large loss fluctuation. This can be seen in the "no control" case in Fig. 5.8a, that the loss rate of *Flow* 1-5 increases from 0.07% to 0.22% since more conflicts occur at a higher load. We then use our control process to remove this disturbance by setting the allowable margin δ to 3%. It can be seen that different controllers based on our design are capable of keeping the loss rate around 0.07% for all the time.

Fig. 5.8b examines a more complicated dynamic case. In addition to E2/E3/E4 increasing their load from 0.7 to 0.72 at t=10 second, they also decrease it back to 0.7 again at t=30 second, and one wavelength channel between C1 and C2 breaks down at t=20 second. It can be seen that the loss rate of *Flow* 1-5 changes more dramatically under this situation if no controller is imposed, while our controllers are able to remove the fluctuations and achieve a stable loss rate for all the time. Other choices of the controllers also work accurately and the increase of the system delay due to the adjustment of the assembly parameter is also small as in Fig. 5.7d, which is not shown here.

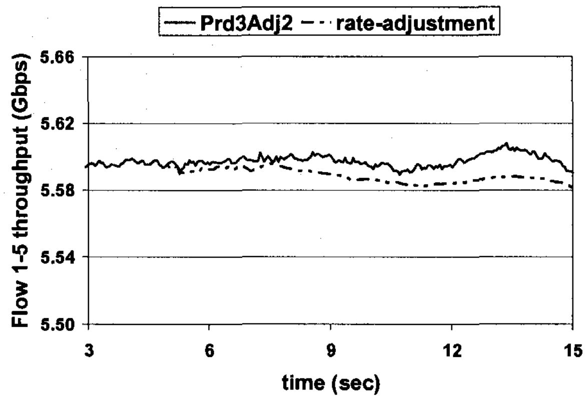
We have also tested the cases when the optical switching time and process time of control packets are non-zero. All results show that our controllers work effectively. These similar results are summarized in Appendix I for clarity reason.

5.4.3 Comparisons

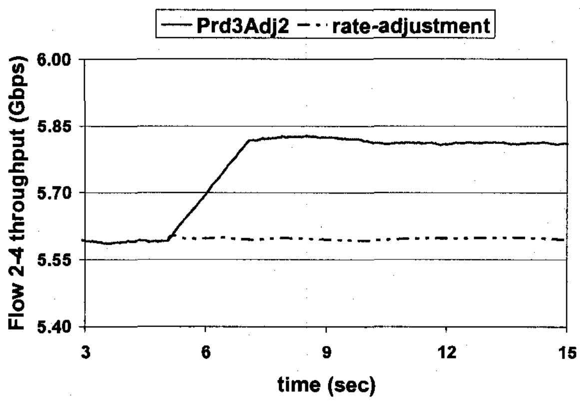
Here we make a comparison of our method with a rate-adjustment scheme, PCwER in [FaZh05], in a P1-A2 network. The reason for picking this scheme has been discussed in 4.4.4. The basic parameters for our simulation are the same as before except that the load of E2/E3/E4 is changed from 0.7 to 0.73 at t=5 second. Since we have demonstrated that different controllers have same capability, we shall only use the Prd3Adj2 controller to make a comparison. In Fig. 5.9a, our controller generates a higher network throughput than the rate-adjustment scheme. It is understandable because the rate-adjustment method chokes the input traffic at the source edge, which causes a lower throughput in a bufferless system. For the same reason, the throughput of individual flows will be suffered by it, as shown in Fig. 5.9b and 5.9c, whereas our design can achieve a higher throughput performance. Because the rate-adjustment scheme stores the overloaded traffic at edge nodes, the system delay will increase almost linearly as time evolves, while our method generates a stable delay, as demonstrated in Fig. 5.9d. If we run the simulation for a longer time, the system delay of the rate-adjustment scheme will remain at a large value when the buffers become full. On the other hand, the buffers used in our scheme have never overflowed during all simulations. Since our method transmits as much traffic as possible and the delay is controllable under the adjustment of the assembly parameter, the advantage of our design is significant. We have compared other scenarios using different network settings and controllers. The results are similar and omitted here.



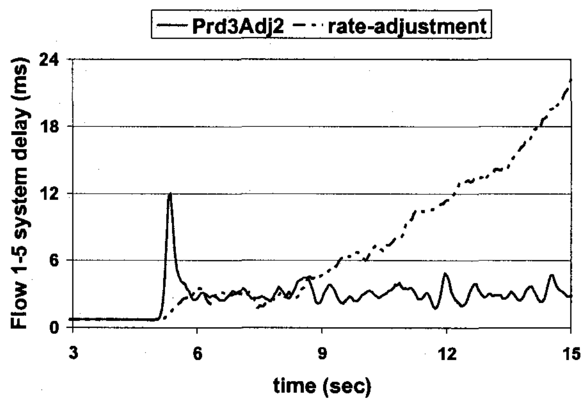
(a) Network throughput



(b) *Flow* 1-5 throughput



(c) *Flow* 2-4 throughput



(d) *Flow* 1-5 system delay

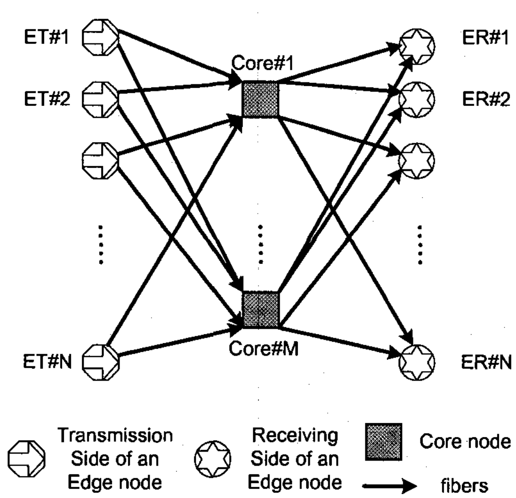
Fig. 5.9. Comparison of two schemes under Poisson traffic

5.5 Concluding Remarks

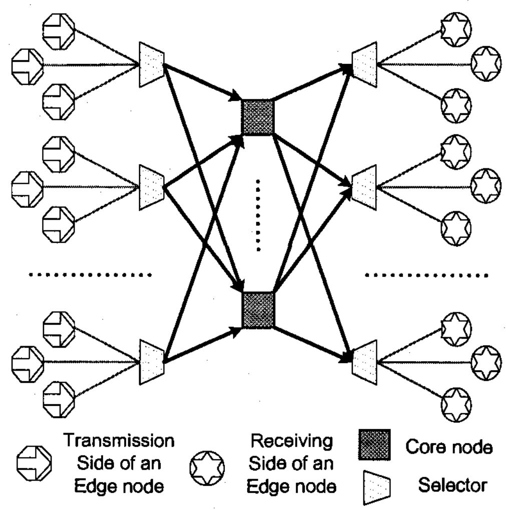
We have proposed some effective design on the flow controllers for optical bursts based on adjusting the burst-assembly process at the edge nodes only. By adjusting the assembly parameters of a flow, we can achieve the required loss performance in both static and dynamic networks. We have designed several efficient loss predictors and burst adjusters, and examined the performance of different scenarios. The advantages of our proposal are: (1) all control operations are edge-based, and it is simple and easy to implement the controller without considering the reservation or signaling details at the core nodes; (2) the control process does not affect the overall network throughput; (3) it works well to different traffics, assembly algorithms and signaling protocols; (4) the influence on the end-to-end delay is very small, (5) it is not affected by the change of core nodes, and so on. Although based on heuristic, our controllers appear to be scalable from our many simulations as the computations are not affected significantly when using a larger number of nodes or wavelengths. On the other hand, our design is used for the control of one flow only, whereas further study is required for multiple flow control, and for the theoretical stability limit. In summary, our scheme has a good scalability to meet the demand of different sizes of networks and provides good flexibility to fit varying networks. These features demonstrate that our controller is superior to the conventional OBS control methods which only adopted rate-adjustment algorithms at the burst level. They will be again considered in a more specific implementation in the next chapter.

CHAPTER 6
APOSN NETWORKS

So far, our work has focused on meshed networks using the OBS scheme. As we have discussed in Chapter 1, it is possible to design a new network architecture in order to have more flexibility on the switching operations and QoS provision. We shall use the APOSN network (also considered for the AAPN project) to implement and evaluate our OBS in both Metro-Area-Network (MAN) and Wide-Area-Network (WAN) networks. In addition we shall illustrate that the OTDM operation is also suitable in APOSN networks. A parallel-processing method is presented to solve the time slot and wavelength assignment problem when the OTDM operation is adopted. Performance evaluation of the APOSN architecture is performed to reveal the effects of different parameters.



(a) Basic architecture



(b) Alternative architecture

Fig. 6.1. MAN connection of an APOSN

6.1 Network Model

We consider an overlaid-star network consisting of M core nodes and N edge nodes as depicted in Fig. 6.1a. This APOSN topology is a special case of the general network in Fig. 2.1 by reducing the number of hops between two edges to one core node only. Each edge node consists of two parts: the ET (Edge Transmission) part and the ER (Edge Receiving) part. Each ET or ER can have a number of direct fiber links to a core node, and each fiber link contains up to $W+1$ wavelength channels. An edge node can be connected to more than one core node depending on the location, physical resources, traffic demand, budget plan and other factors, and the input traffic to a destination node is randomly distributed to all possible output channels. Since a core node is only responsible for transferring messages from a group of ETs to a group of ERs, its operation can be considered as being independent of other core nodes. Therefore, the topology in Fig. 6.1a can be effectively divided into M independent single-star networks.

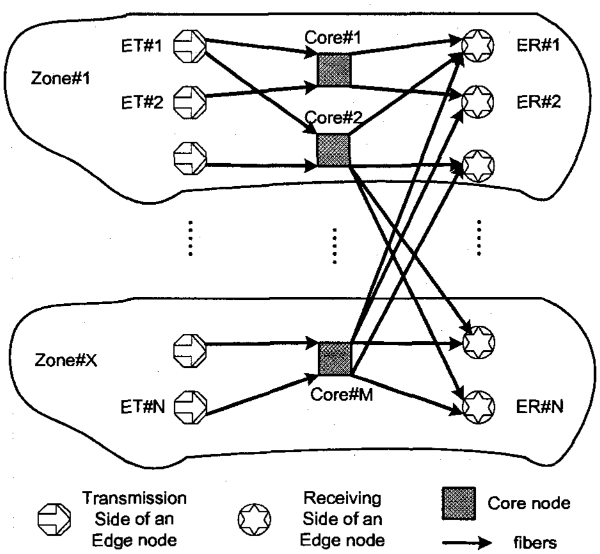


Fig. 6.2. WAN connection of an APOSN

In a WAN configuration, we shall group one core node and its nearby edge nodes into a zone as in Fig. 6.2 (effectively like a MAN) so that the edge nodes and the core node in the same zone are only separated by a short distance. All ETs rely on the core node to transmit to the ERs within the same zone. Depending on the zone size and traffic demand, it is also possible to have more than one core node within the same zone.

To maintain full connectivity within a WAN, at least one core node in each zone must be arranged to transmit traffics to some ERs outside (such arrangement is a topology optimization problem). Similar to the MAN topology within the same zone, each core node responsible for inter-zone communication is also working independently, which means the whole network can again be considered as consisting of M independent single-star networks. It is easy to see that Fig. 6.1a is a special case of Fig. 6.2 when the number of zones reduces to one.

Figs. 6.1a and 6.2 only illustrate the transmission of payload messages. If an ET or ER has a direct fiber connection to a core, one wavelength channel (not shown in the diagrams) can be dedicated to the transmission of control messages between the ET (or the ER) and the core node, while other channels are used for payload messages. The control channel between an edge node and a core is bi-directional in order to enable the scheduling or retransmission control (it can be implemented by two channels transmitting messages in two directions respectively), and it is also used to implement time synchronization when necessary.

An alternative configuration is to add some selectors as concentrators between a core and edge nodes as in Fig. 6.1b. The traffics from several edge nodes are selectively grouped under a selector that has direct connections to core nodes. In this way, the size of a core node, in terms of the number of switching ports, can be reduced dramatically. There can be two types of operations on the ET side. One choice is to assign some wavelength channels for each ET and different wavelength channels are multiplexed at the selectors. In this case, a selector can be a passive coupler. The second choice is for the system to work in a slotted mode only, i.e., a selector assigns the slots among ETs in a Round-Robin order so that the traffics from several edge nodes can pass through the selector sequentially. As a mirror operation, the selectors on the ER side are demultiplexers. When we look at the traffic going to a core node, there is no difference for the two connections in Figs. 6.1a and 6.1b. In a WAN network, we can use selectors in a similar way. Since each selector is a passive device, its function is equivalent to a directly connected edge node and we shall use the term of edge node only for the following switching and scheduling analysis without losing generality.

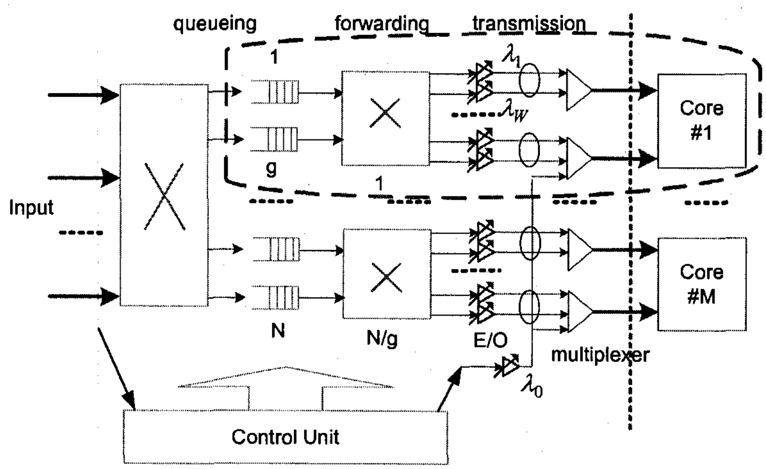


Fig. 6.3. Transmission side of an edge node

6.1.1 Connection Modeling

The basic architecture for ET and ER is shown in Fig. 2.2. We have pointed out before that the APOSN in Figs. 6.1 and 6.2 can be considered as M independent single-star networks. So we can group the components on an ET side according to whether they handle the traffic to the same core node. For example, the dotted box in Fig. 6.3 consists of the components involved in the traffic transmission to Core#1. There are g VOQs in the box, each corresponding to a destination ER. Other components in the edge node involve the connections to other core nodes. All edge nodes connected to the same core node use same wavelength for the control message transmission (for simplification, we use λ_0 to denote the control channel in our diagram, although each star can employ a different control wavelength.).

Since the ETs are capable of forwarding any ready messages in the VOQs to the transmission stage and the ERs capable of forwarding all newly-arrived optical messages to the queueing stage at any time as mentioned in Section 2.2.1, the messages after the core nodes will not be lost in an APOSN. Therefore, we only need to analyze messages transmitted from an ET to the core, while ignoring the message processing at the ER side. Each fiber can have up to W wavelength channels ($\lambda_1 \ldots \lambda_w$) for payload transmission, while the actual number depends on the traffic demand and the topology design.

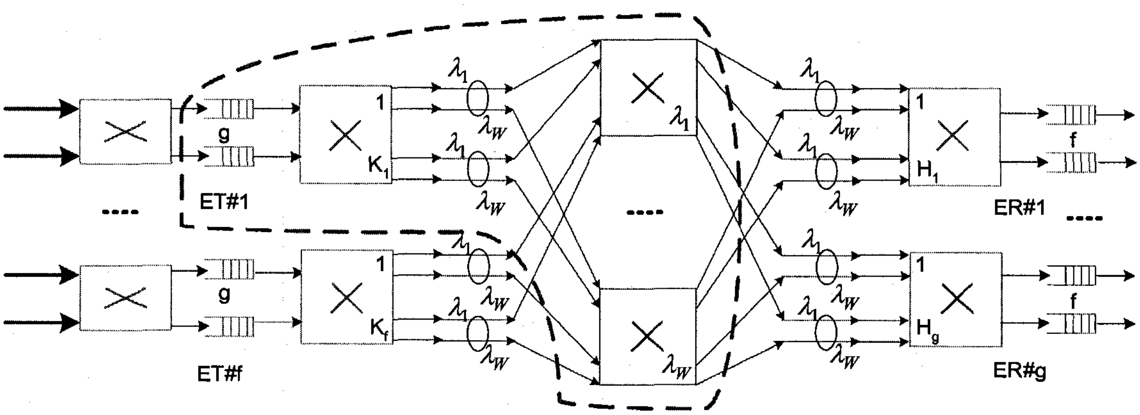


Fig. 6.4. Connections of one core node

By further integrating the architecture of the edge node and the core node, the complete connection related to one core node is shown in Fig. 6.4. This core node is responsible for transferring messages from f ETs to g ERs. ET#i has K_i ($1 \le i \le f$) fiber links going to the core node and ER#j has H_j ($1 \le j \le g$) fiber links coming from the core node. Only g VOQs in each ET are related to this core node and other VOQs need not be considered. The dotted box refers to the same components in the dotted box of Fig. 6.3. In other words, VOQ#j in the dotted box buffers traffic destinated for ER#j. The controlling part is omitted here as it does not affect the payload part.

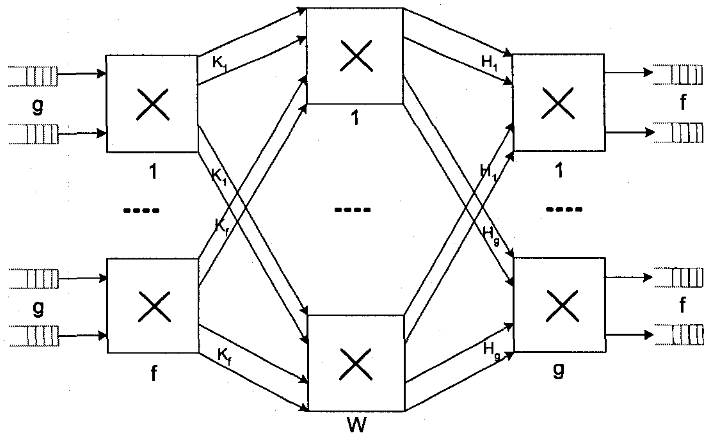


Fig. 6.5. Equivalent Clos network of APOSN

By removing the redundant parts, we can simplify the connections of one core node to a three-stage multi-connection Clos network as shown in Fig. 6.5. The first stage corresponds to the ETs, the second stage to the core node and the third stage to the ERs. The adjacent stages are connected by wavelength channels and separated by different distances. When we solve the routing problem in such a network, we need to find out an appropriate wavelength channel to carry the input message and send out the message at an appropriate time instant. Therefore, the wavelength assignment and time assignment problems can be solved simultaneously by solving the routing problem in this asymmetrical network. According to our topology design, this principle applies to both MAN and WAN networks.

For other core nodes, the connection pattern is similar, but involves different edge nodes. The number of fiber links between an ET and a core node can also be different, and each fiber can have a different number of wavelength channels as well. Therefore, the equivalent connection of the whole network consists of *W* independent asymmetrical multi-connection Clos networks (in which each wavelength channel is thought to be a link). The traffic at an ET is assumed to be randomly distributed to all possible output channels, so we only need to focus on the analysis of the switching operation in one core node, which represents the whole network behavior.

If selectors are used, the above Clos model is still effective. In such case, the first stage or the third stage does not correspond to a specific edge node. If a group of edge nodes are sharing a selector sequentially in slotted mode, only one among this group is selected to represent the first (third) stage in Fig. 6.5 at a time, normally in a Round-Robin manner. If each edge node is assigned with a few wavelength channels in the selector exclusively, the first (third) stage in Fig.

6.5 can be seen as a super edge node including all ETs (ERs) sharing the selector.

6.2 Architectural Features

Our overlaid-star topology has the following advantages in terms of the topological design of optical networks. Some cost analysis is also provided at the end of this section.

First, our design can avoid the use of wavelength conversion and optical buffering. Current technologies of wavelength conversion and optical buffering are still immature, and it is also a big obstacle to integrate optical components in a small space at a reasonably low cost. Since the transmission path between two edge nodes in an APOSN passes through only one core node, we can simply use pure optical switching in the core node while using electronic buffering at the edge nodes. Along with the edge-based scheduling mentioned later, our design is a reasonable and practical solution for the near future.

Secondly, a reservation-based scheduling can be used to improve the utilization of wavelength channels. Since the distances from the ETs to a core node can be kept short in a zone (usually within the coverage of a MAN), the corresponding round-trip propagation delay becomes small. Consequently, a reservation-based or coordination-based scheduling algorithm can be used to improve the efficiency. This is different from the conventional opinion that the optical networks in circuit-switching mode have a low efficiency and a long latency. By using active switches in the core node, our star topology has more flexibility and capacity than passive star networks.

Thirdly, our overlaid-star topology can much alleviate the serious reliability problem of a single-star due to the central node failure because an edge node in an APOSN network can easily reroute the traffic to another star (through another core node). The failure detection can be implemented by monitoring the optical signals on the transceiver side. Some spare capacity is also provided on each link to protect (reroute) the traffic of a failed link. This protection and restoration scheme is simple, flexible and efficient when dealing with failures in wavelength channel, fiber link or core node, whereas ordinary protection mechanisms are usually more complex in determining the spare capacity and re-routing the affected traffic. One can also see that the APOSN topology provides a good reliability. For example, the terminal-pair reliability of multi-center tree networks has been analyzed in [YaMa88], and extended to multi-level star networks in [LiYa04]. Our APOSN network can be seen as a special case of those topologies, so its reliability can be analyzed similarly.

Fourthly, this topology provides a good scalability as the network size grows physically in the sense that it is easy to add an edge node to a component star. Furthermore, this topology reduces the number of fiber segments, so it is easy to expand an APOSN network quickly, and to provision various protection and restoration schemes.

Finally, a network of our design can reduce the operating expense which is a big and difficult concern for carriers. The operating expense includes, but not limited to, network planning, network management (e.g., installation, path provisioning, and alarm collection), network expansion (e.g., system capacity, and number of nodes), system maintenance (e.g., hardware and software systems, and different platforms), and so on. It can be seen that these expenses would last for a long time. By avoiding expensive and complex components in our APOSN network, it is easier to maintain and manage a network of our design with a limited budget.

In summary, considering the above discussions along with the cost analysis provided below, one can see that an innovation of the network architecture can be a fundamental solution to the capability to provision QoS for a given budget. Our design enables service providers to reduce costs in these areas, and to provide better services consequently.

6.2.1 Cost Analysis

We now provide some analysis on the cost issues related to the settings of an APOSN network, which would help us to understand the advantages of our design. Consider the case that a core node transmits traffics from f ETs to g ERs and each fiber has the same number of wavelength channels, W, for payload messages. Let S be the number of ports in each core switch, $\Lambda_i\ (i=1,\ldots f)$ be the input traffic to ET#i, and ρ be the allowable traffic load on each channel between an ET and a core node (we set $\rho<1$ in order to provide some spare capacity for protection). Without complicating the argument, we let the connections be symmetric, i.e., the numbers of ETs and ERs are the same in a star, and each edge node has the same number of wavelength channels so that $f=g$ and $K_i=H_i\ (i=1,\ldots f)$. Then under a uniform traffic distribution, the following relationships should be satisfied:

$W\cdot\sum_{i=1}^{f}K_i\le S$,

$\Lambda_i\le M\cdot W\cdot K_i\cdot\rho$,

$\sum_{i=1}^{N} \Lambda_i \leq M \cdot W \cdot S \cdot \rho$.

The set of inequalities above have considered the size limitation of core switches, the outgoing traffic load of edge nodes and the offered traffic load to core nodes. It is shown that the number of core nodes is dependent on the number of edge nodes, traffic load, available wavelength and protection requirement. If the network is extremely irregular, e.g., the number of wavelengths in each fiber is different, or the size of each core switch is different, then the above conditions should be further elaborated, and the complexity of the problem becomes higher because more constraints are involved.

Our design can save the number of transceivers compared to other topologies with the same capacity. In our study a transceiver means either a transmitter or a receiver, which is assumed to have same cost. We also assume each fiber contains same number of wavelength channels. The total number of transceivers in a single-star is equal to $4 \cdot \sum_{i=1}^{f} K_i \cdot W$. Then the total number of transceivers for the whole APOSN becomes $4WM \cdot \sum_{i=1}^{f} K_i$. On the average, each ET has $\frac{WM}{N} \cdot \sum_{i=1}^{f} K_i$ output wavelength channels. With the estimation above, we can now make a comparison to other topologies such as a multiple-hop network with N edge nodes. To provide the same connection capacity, we also allow each edge node to have $\frac{WM}{N} \cdot \sum_{i=1}^{f} K_i$ output channels on the average. In general, each source-destination path now needs to pass through one or more core nodes, and so by considering an average length of 2-hops (which is a conservative estimation on the low-side), we need $\frac{WM}{N} \cdot \sum_{i=1}^{f} K_i \cdot N \cdot 6 = 6WM \cdot \sum_{i=1}^{f} K_i$ transceivers for the whole network since each routing path now includes two core nodes and six transceivers on the average. Thus by comparison, our design can obviously reduce the number of transceivers. Please see Section 6.7.2 for further discussions.

6.3 OBS Operations

We first consider OBS operations in an APOSN network. Then we apply the DiffServ and flow control scheme we have designed in Chapters 3-5 in it. Considering the topology characteristics of our network design, we shall analyze the OBS method in the same star network in Fig. 6.5 which represents all switching operations in the overlaid-star architecture.

6.3.1 General Schemes

Referring to the architecture in Fig. 6.3, the packets destined to the same ER are stored in the same VOQ and form a data burst. The high priority packets are put in the tail portion of a burst, while low priority packets are put in the front portion of a burst. The idea is to discard low priority packets in case of conflicts. The assembly procedure is determined by two threshold parameters: a timer of T_{mad} s which is called the MAD (Maximum Assembly Delay), and the MBS (Maximum Burst Size) which is the maximum number of packets in a burst. These two parameters have same functions as the assembly parameters used in Chapters 3-5. Whenever one of these two thresholds is reached, the burst assembly is terminated and a BHP packet is formed.

We also consider the re-transmission scheme as a possible choice to improve the throughput, but it is not necessary as shown later. After a DB is sent out from an ET, a copy of this burst is kept in the VOQ. If the core node sends back an acknowledgement saying the burst has been scheduled successfully, the copy of the DB is removed. Otherwise, this DB is scheduled to retransmit by repeating the above scheduling scheme. Note that a DB can only stay in the VOQ for the MSD (Maximum Scheduling Delay) period before it is removed even if it is not scheduled.

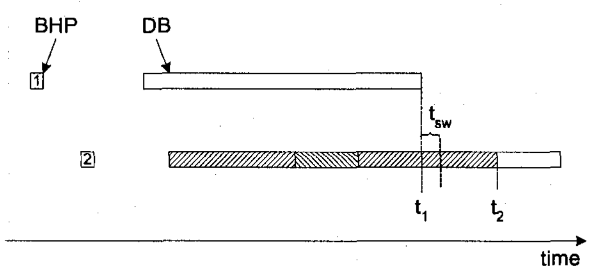


Fig. 6.6. Burst segmentation with header dropping

Either JETheadSeg or JETtailSeg protocol can be employed in the core nodes because they generate the same effect through our simulation study. We shall only provide the analysis of the JETheadSeg as shown in Fig. 6.6. Here the earliest available channel among all output fiber links is occupied by *Burst* 1 until time t_1 with *Burst* 2 arriving earlier than t_1. Considering the overhead of the switching time t_{sw}, then only the message after $t_1 + t_{sw}$ in *Burst* 2 can be transferred. If the truncated packet at $t_1 + t_{sw}$ does not end until t_2, then only the message after t_2 is useful. The core node sends back an acknowledgement to the ET to indicate if this burst is scheduled or not in case burst retransmission is enabled.

The operations described above do not make use of selectors. When selectors are adopted, some wavelength channels have to be assigned to each edge node exclusively as mentioned before. Note that the selectors are transparent to the core nodes under the operation of the transmission and scheduling, so the performance should not be affected when compared to no-selector configuration.

6.3.2 DiffServ Provision and Flow Control

The VOQs in our APOSN architecture can be used to implement both the DiffServ Provision and Flow Control. In the case of DiffServ scheme developed in Chapter 3, the number of VOQs is expanded so that each class of traffic is assembled in a unique buffer. By applying different assembly parameters in these assembly processes corresponding to different VOQs, we can achieve differentiated loss ratio for different flows. Note that both schemes (from Chapter 3 and the present chapter) can support multiple classes. We shall use two classes only in later simulations to verify the effectiveness of our proposals.

In the case of flow control schemes developed in Chapters 4 and 5, we need to expand the number of VOQs in the same way. The FCPs need to inform the source edge about the loss rate of the controlled flow. By adjusting the parameters of the controller based on our experiences, a desired loss rate can be achieved for a burst flow (see Section 6.4.7).

6.4 OBS Performance

We would like to evaluate the effect of some typical parameters on the delay, utilization, throughput and loss performance of our overlaid-star architecture. We shall use an 8x8 star network to evaluate our OBS method. The basic parameter settings are the same as previous chapters except the following ones.

- ET to core distance: 2, 4, 10, 16, 20, 40, 100, 200 km respectively
- number of fibers from ET to core: 1, 2, 3, 4, 1, 2, 3, 4 respectively
- number of fibers from core to ER: 1, 2, 3, 4, 1, 2, 3, 4 respectively
- number of wavelengths: 4 wavelengths for payload traffic in each fiber
- traffic classes: 60% high priority (Class#0), 40% low priority (Class#1)
- traffic load: 0.5
- Bandwidth of each wavelength channel: 10 Gbps
- maximum assembly delay for burst assembly: 1 ms
- maximum burst size: 100 packets
- maximum scheduling delay: 5 ms

- BHP processing time: 100 ns per header
- burst re-transmission: not used

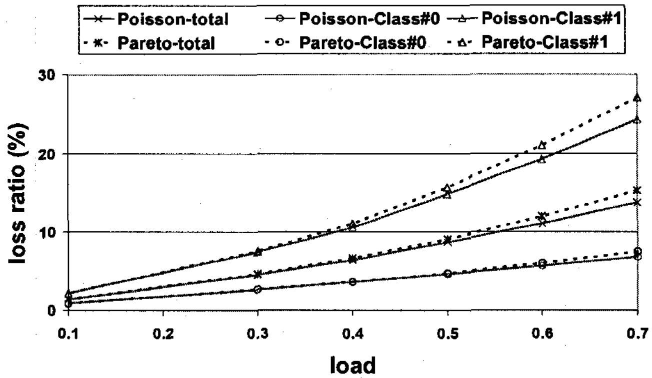


Fig. 6.7. Loss performance

6.4.1 Basic Performance

We first check the loss rate of total Poisson traffic in Fig. 6.7. It can be seen that our OBS scheme generates less than 10 percent loss when traffic load is lower than 0.5. In the full load range, the loss rate increases almost linearly with increasing traffic load. Since messages are not synchronized, conflicts occur frequently and the loss rate becomes significant at high load. Because we treat two classes of traffic with different priority, the Class#0 Poisson traffic generates almost half loss rate than the total Poisson traffic, while Class#1 one generates almost twice loss rate compared to the total loss of the two classes of traffic. The Pareto traffic shows similar loss trend, but with a higher loss rate.

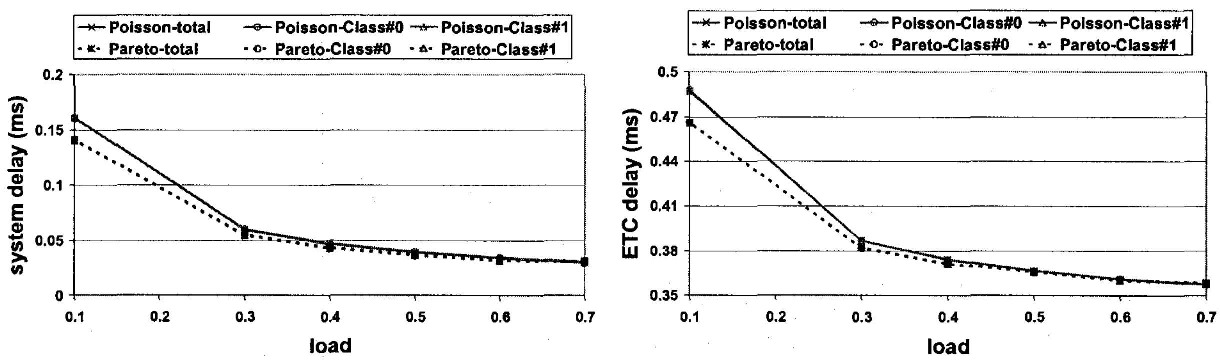


Fig. 6.8. Delay performance

From Fig. 6.8, we can see that the system delay introduced by our OBS scheme is pretty short (less than 0.2ms). Both Class#0 and Class#1 Poisson traffic do not demonstrate a significant difference on the system delay at a given load. As the load increases from 0.1 to 0.3, the burst assembly becomes faster, which decreases the system delay. As the load increases further, this acceleration becomes less effective because more bursts contend for transmission, and then cause the delay-decreasing less significant as in the first half. The propagation delay in our network is fixed and much longer than the system delay, so the ETC delay of the same Poisson traffic exhibits a same curve as the system delay, while the ETC value is larger. The Pareto traffic generates similar performance characteristics. The bursty Pareto traffic forms bursts faster, thus the delay is smaller than that of Poisson traffic at load range of 0.1 to 0.5. This difference diminishes at higher traffic load.

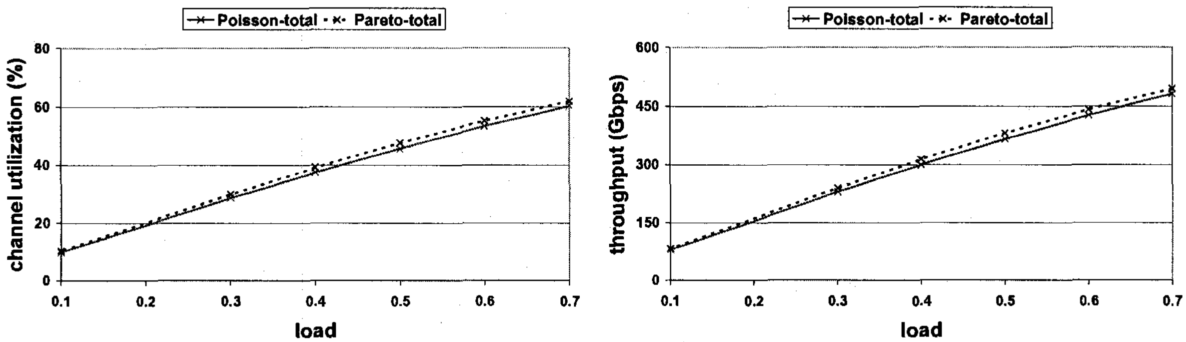


Fig. 6.9. Utilization and throughput performance

The channel utilization and aggregated throughput of Poisson traffic have a linear relationship with the load change in Fig. 6.9, which demonstrates our OBS system is stable in the full load range. When the offered load is 0.2, the channel utilization is about 20%. When the offered load becomes 0.7, the channel utilization is about 60%. Therefore, our OBS scheme has a pretty good support to the Poisson traffic. When Pareto traffic is applied, the performance is very close.

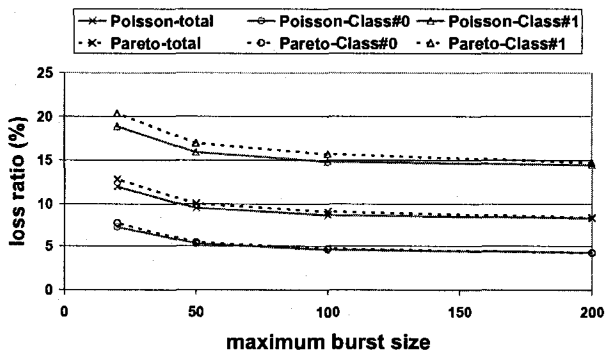


Fig. 6.10. Loss performance vs. MBS

6.4.2 Effect of MBS

By increasing the burst size (in terms of the number of packets in a burst), the channel waste due to the optical switching time is reduced, which means less loss rate as shown in Fig. 6.10. The total Poisson traffic reduces the loss from 12% to 9% as the MBS changes from 20 to 100. However, this improvement is limited after the burst size becomes larger than 100. The Pareto traffic has similar performance except for a higher loss rate when the MBS is less than 100 too.

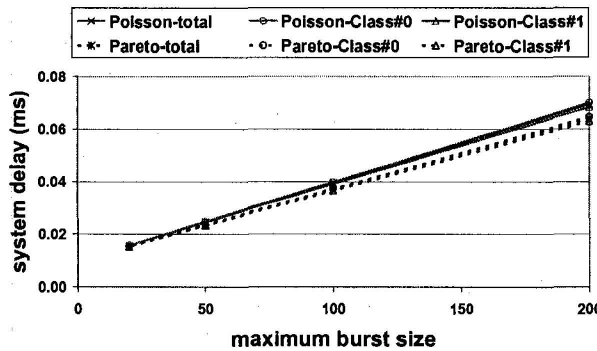


Fig. 6.11. Delay performance vs. MBS

On the other hand, increasing burst size has a bigger impact on the system delay since the packet waiting time increases linearly with the change of burst size, which is demonstrated in Fig. 6.11. The system delay of Poisson traffic changes linearly from 0.015ms to 0.07ms when the MBS changes from 20 to 200 packets. Pareto traffic also has a linear increasing system delay. In general, if more packets are put into one burst, the number of switching operations will be reduced, but the packets will experience longer assembly delay. However, the assembly delay will be reduced if fewer packets are assembled into a burst, but the control and switching operation will occur more frequently. Therefore, a balance is needed between these two parameters.

6.4.3 Effect of MAD

We shall examine the performance of Class#0 Poisson traffic as MAD changes in Fig. 6.12. As MAD increases from 0.2ms to 0.5ms, its system delay increases from 0.036ms to 0.039ms because lower MAD assemblies bursts faster. If MAD increases further, the MBS takes effect before the MAD expires during the burst assembly procedure. Thus, a higher MAD does not generate different loss rate. On the other hand, the MAD has little effect on the burst scheduling and segmentation, so Class#0 traffic suffers similar loss rate regarding to different MADs. For the same reason, Pareto traffics show similar performance trend. The difference between Poisson and Pareto traffic can be explained in the same way as in Fig. 6.8.

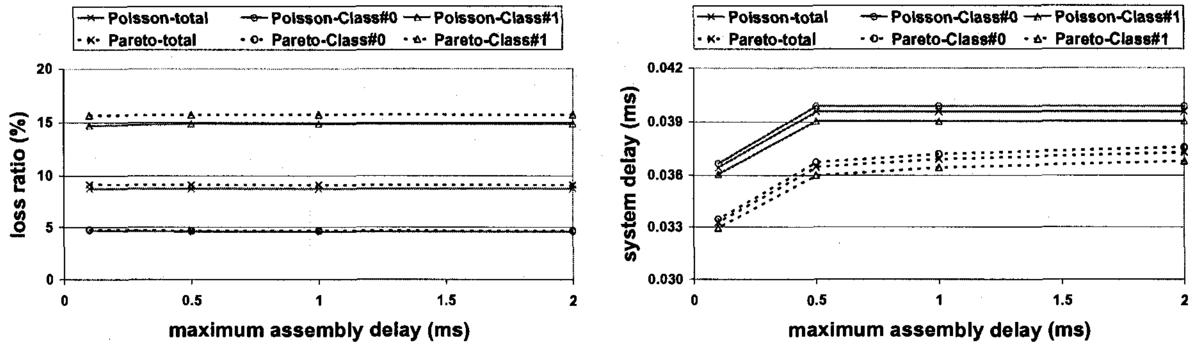


Fig. 6.12. Loss and delay performance vs. MAD

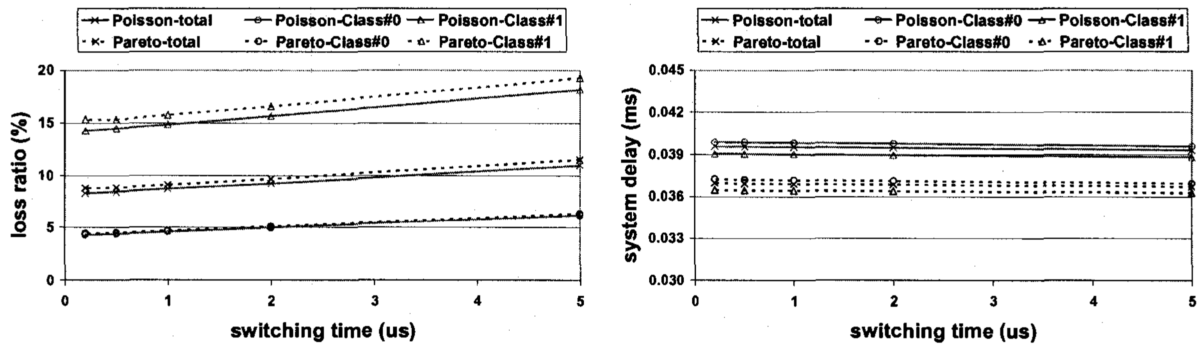


Fig. 6.13. Loss and delay performance vs. optical switching time

6.4.4 Effect of Optical Switching Time

When the switching time changes between 200ns to 5us in Fig. 6.13, the total packet loss ratio of Poisson traffic increases slightly and linearly from 8 to 12 percent and the system delay remains almost unchanged at 0.04ms. The reason lies in that the channel is wasted and no message is transmitted during the period when a switch changes its status. On the other hand, the switching time has no effect on the burst assembly and edge transmission, thus will not affect the system delay. All traffic types follow the same rule and demonstrate same performance trend. The difference among different traffic types can be explained in the same way as previous parameters.

6.4.5 Effect of the Number of Wavelengths

When we scale a network by increasing the number of wavelength channels from 1 to 8, the loss rate of Class#0 Poisson traffic remains almost the same 5 percent in Fig. 6.14a. It can be explained by the equal probability that a channel will be occupied under the same traffic load. At the same time, the high priority Poisson traffic experiences 0.13 ms of system delay in the one-wavelength case and 0.028ms delay in the eight-wavelength case in Fig. 6.14b. The reason is the latter case give more chances that a burst to find an available channel. The Class#0 Pareto traffic has similar performances, while both Class#1 traffics suffer higher loss when more wavelengths are employed. From Fig. 6.14c we can see the increasing in the total loss is quite small.

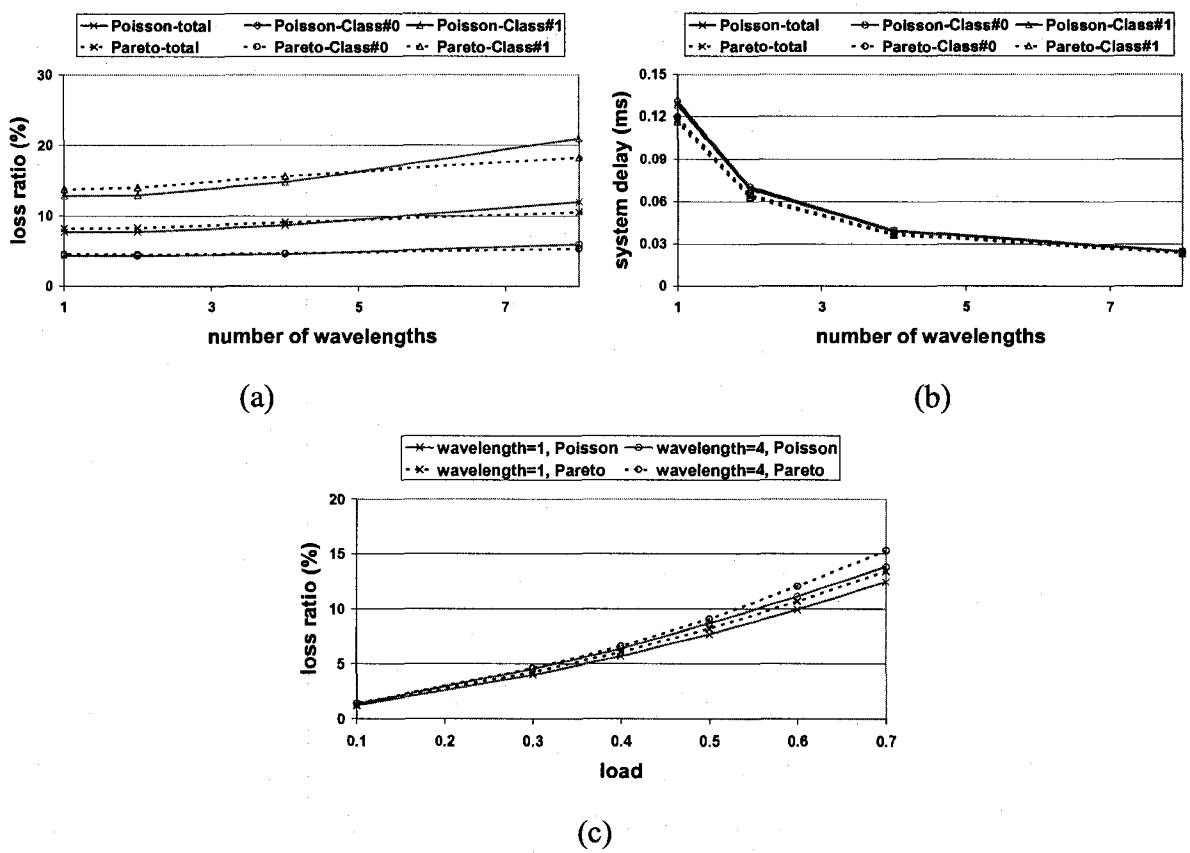


Fig. 6.14. Loss and delay performance vs. the number of wavelengths

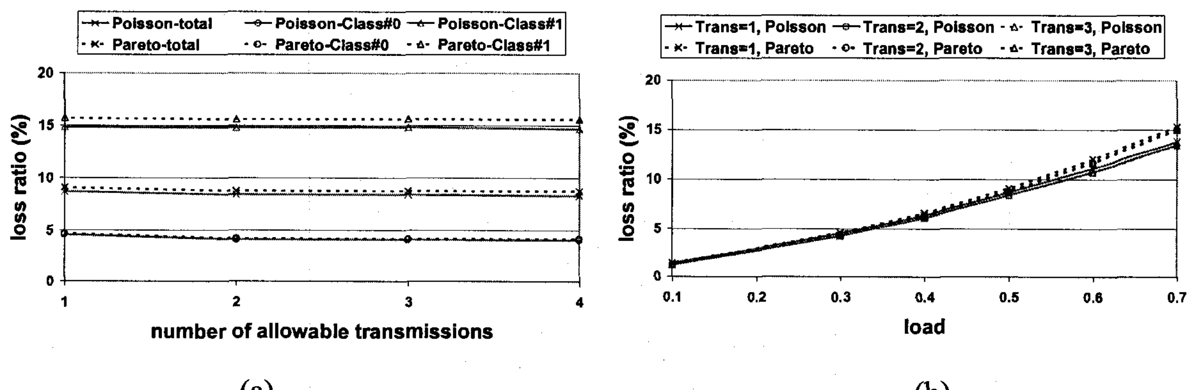


Fig. 6.15. Loss performance w.r.t. the number of transmissions

6.4.6 Effect of Retransmission

When the total number of transmissions is either 1, 2 3 or 4, the loss rate of Class#0 Poisson traffic remains around 4.9 percent in Fig. 6.15a. We can say that when retransmission is enabled, the loss performance has almost no change to high-priority Poisson traffic. This conclusion also applies to different traffic loads as in Fig. 6.15b. Other traffics are insensitive to the retransmission too. Since burst segmentation is enabled, most bursts at least have a chance to transmit part of them. Only a few bursts are wholly dropped, which need to be retransmitted. Therefore, there is no need to employ complicated retransmission scheme.

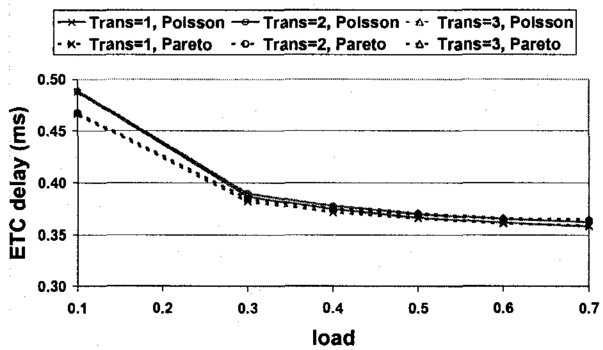


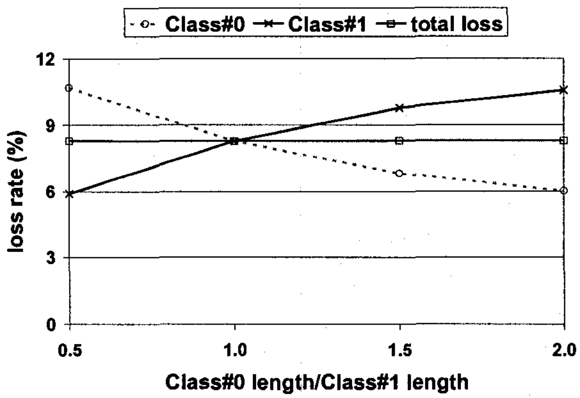
Fig. 6.16. Delay performance w.r.t. the number of transmissions

When we consider the delay of total Poisson traffic in Fig. 6.16, we can see there is no difference on the ETC delay on a given traffic load when different number of transmissions is used. Pareto traffic demonstrates no difference too. It confirms again that the retransmission is not necessary which means we can make the system simple by reducing the buffer requirement and avoiding the retransmission arrangement at edge nodes.

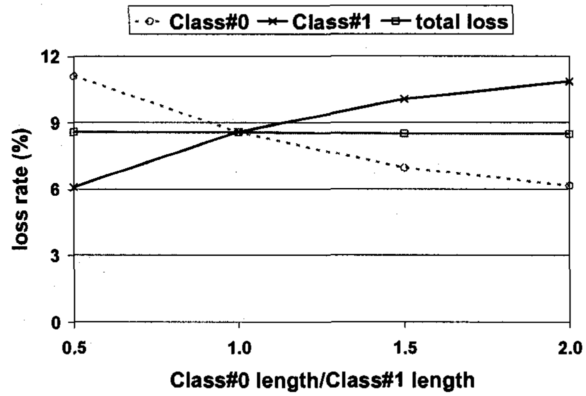
6.4.7 DiffServ and Flow Control

We shall examine how our DiffServ and flow control mechanisms designed in Chapters 3 to 5 perform in a burst-switched APOSN network. We let each class of traffic to be 50% of the total traffic and the same class of traffics is assembled in an individual buffer. The MBS is 200 packets and the MAD is 1ms. We modify the MBS to achieve the DiffServ and flow control. Other parameters are the same as in the beginning of Section 6.4.

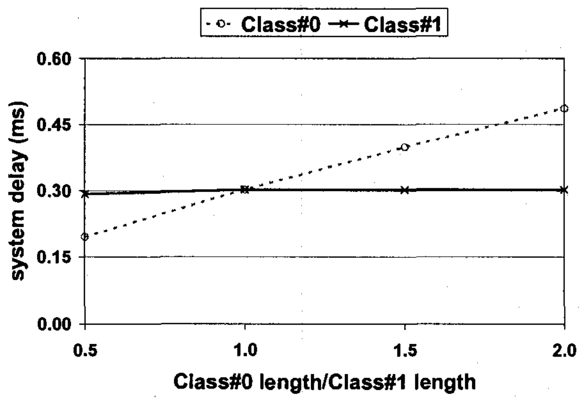
Fig. 6.17 illustrates how the loss ratio and the system delays of two classes of traffics change. It can be seen that the differentiated loss ratio and system delay exhibit a same trend as those in Section 3.4. Our DiffServ scheme can generate a lower loss for a class of traffic that has a larger assembly parameter. On the other hand, the influence on the system delay is small and controllable. When considering the total end-to-end delay, we see that the effect is very small. All results demonstrate that our design is suitable for the APOSN.



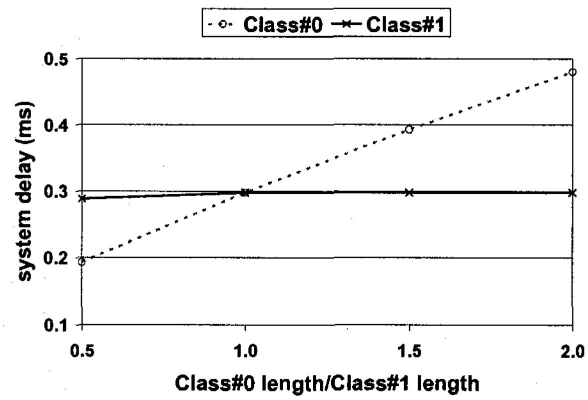
(a) Loss: Poisson traffic



(b) Loss: Pareto traffic

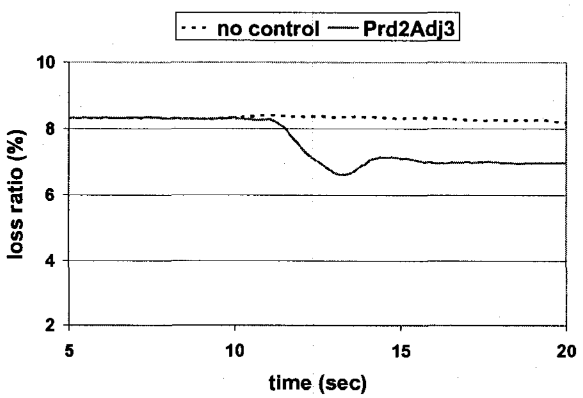


(c) Delay: Poisson traffic

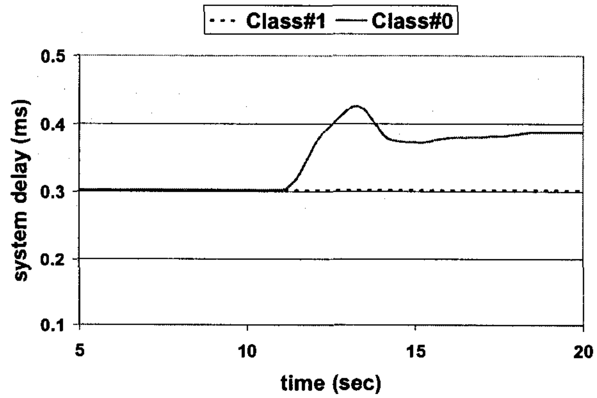


(d) Delay: Pareto traffic

Fig. 6.17. DiffServ performance



(a) Loss ratio



(b) System delay

Fig. 6.18. Flow control performance

Next we test the performance of the flow controllers designed in Chapter 5 in an APOSN. We use the Prd2Adj3 controller as an example to reduce the loss rate of Class#0 bursts under Poisson packet arrival. Fig. 6.18a depicts the loss rate of Class#0 traffic is reduced from 8.3% to 7% after activating the control from t=10 second. The controller exhibits a good convergence and stability. The system delay in Fig. 6.18b is increased from 0.3ms to around 0.39ms, which has a little effect on the total end-to-end delay. Other scenarios generate similar performance, but are not shown here. In summary, our flow controllers designed in Chapter 5 also fit the APOSN networks.

6.5 OTDM Operations

In Figs. 6.1 and 6.2, each core node only needs to coordinate the traffic coming from the transmission sides of the edge nodes in one star. Similar to the OBS operation, we only need to analyze message transmission in such a star network when using the OTDM.

Under the OTDM operation, the time axis of each channel is sliced into continuous equal-length slots. Each slot consists of two parts: a fixed-length cell containing payload messages, and an empty interval equal to the switching time of the core optical switching fabric. All ETs are aware of the propagation delay between any ET and the core node in the same star. We let the propagation delay between each edge and the core be equal to a multiple of the slot period so that the slots in different channels can be time-synchronized (it can be done by adjusting the fiber length).

When messages from legacy networks arrive at an ET, they are aggregated into fixed-length cells. A cell consists of a variable number of packets of varying lengths. If a packet is longer than a cell, this packet is split into several cells. The assembly procedure is determined by two parameters: the timer threshold MAD of T_{mad} sec, and the CS (Cell Size) of L_{cs} bits. The CS is the maximum size of a cell. The cell assembly process is terminated whenever anyone of these two parameters is exceeded. Therefore, a cell may be just partially filled because of the MAD constraint. Once a cell is ready, the edge node stores it in the corresponding VOQ first and sends a request to the core node through the control channel. After receiving an acknowledgement, the edge node sends out the message cell on the assigned wavelength channel and at the assigned time slot. If no acknowledgement is received, that slot is empty and wasted. The waiting time of a cell also has a maximum allowable value called the MSD (Maximum Scheduling Delay) of

N_{msd} slots. If the MSD time expires before a cell is scheduled to transmit, the corresponding cell will be dropped at the edge node.

At the core node, control packets are converted to their electrical format in order to form a traffic matrix. Using an operation called the time slot assignment or time slot scheduling, a core node distributes the wavelength channels by assigning the time slots to different requests based on its knowledge of the network resources and the traffic matrix, and then sends back the scheduling information (a grant or acknowledgement signal) to the edge nodes. If a request cannot be met at the time when it arrives at the core node, the core node adds it to the traffic matrix during next time slot in order to give that request another chance to be scheduled. If the MSD time expires before a request being scheduled, the core node will remove this request from the traffic matrix. Unlike the random access used in OBS, the controlled access of OTDM has no collision loss, but at the expense of other performance, such as a longer delay.

Note that we only deal with time slots when we talk about the scheduling problem in OTDM operations, we shall use a "time slot" to mean a "message cell" in the OTDM scheduling discussed later. According to Fig. 6.5, the time slot assignment also implies the wavelength assignment, so we shall use TWA (Time-Wavelength-Assignment) in latter analysis.

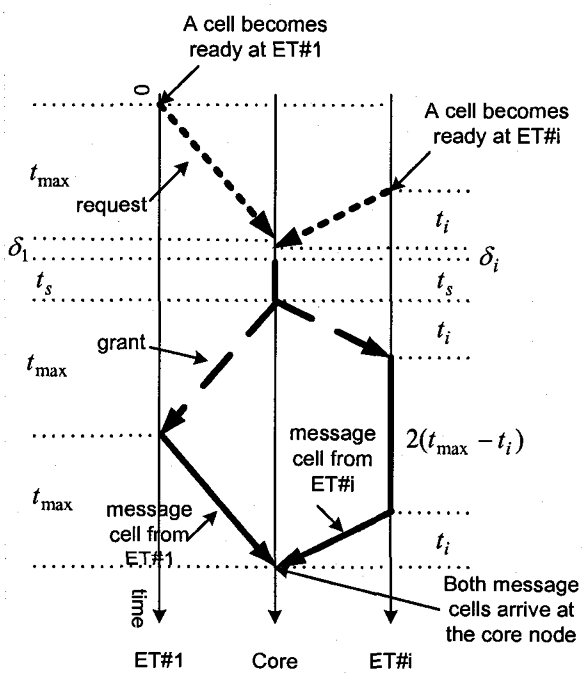


Fig. 6.19. Synchronization of message transmission

If selectors are used and some specific wavelength channels are assigned to each edge node, then a selector becomes transparent to the scheduling operations because each edge node can now communicate with the core node directly. If each edge node shares the channels in a Round-Robin manner, a core node needs to keep a copy of the waiting slots from each edge node, and then selects the corresponding record to generate a traffic matrix during each scheduling step. In a long run, the average offered load to a core node is expected to be the same no matter which configuration is adopted for the selectors.

6.5.1 Message Synchronization

Since each edge node may have a different distance to the core node, it is important to synchronize the message slots in a star in order to implement the centralized TWA. Referring to Fig. 6.19, let t_i $(i=1...f)$ be the propagation time from ET#i to the core, and t_{max} be the longest propagation time from the ETs in the referred star to the core node (assume from ET#1). Both are measured in the number of time slot periods. Let us denote the time that a cell becomes ready at ET#1 as time *0*. A request sent by ET#1 at t=0 and another request sent by ET#i at $t = t_{\max} - t_i$ will arrive at the core in the same time slot which includes the time instant t_{max}. The remaining time from the first request's arrival to the end of the slot boundary is δ_1 and the remaining time from the second request's arrival to the slot boundary is δ_i. The core node collects all requests during this slot period to form a traffic matrix, spends one slot of time (t_s time) to implement the TWA, and sends the results (grant signal) to the ETs immediately after the processing slot. Once ET#1 receives the grant signal, it sends out the cell immediately. After ET#i receives the grant signal, it waits $2(t_{\max} - t_i)$ before sending out the cell. Both cells from ET#1 and ET#i arrive at the core at $t = 3t_{\max} + t_s + \delta_1$ when the core node sets up the status of the switching fabric using the previous TWA results. Therefore, the overhead is $2t_{\max} + t_s + \delta_i$ for all cells from the instant when they become ready at ET#i to the instant when they are sent out from this edge node if their request is granted immediately after the scheduling. Otherwise, a cell has to wait further until it is scheduled in a later time slot eventually or dropped due to timing out (MSD exceeded). Since the ETs and the core node are designed to be separated by a short distance in the same zone, the waiting time for the scheduling can be controlled within an acceptable range. Thus, our centralized operation scheme will not introduce a severe delay.

6.5.2 Time-Wavelength Assignment

Here we solve the routing problem, i.e., the TWA problem, of the multiconnection three-stage

Clos network in Fig. 6.5. Since ETs are capable of forwarding any ready messages in VOQs to the transmission stage and ERs capable of forwarding all newly-arrived optical messages to the queueing stage at any time, it is equivalent to the case that the switches in the first stage have fan-out capability and the switches in the last stage have fan-in capability. The switches in the center stage have neither fan-in nor fan-out capability because they are all-photonic switching fabrics. Based on the concept of network-flow model and parallel scheduling in the Clos network [VaCh92], we propose a parallel scheduling algorithm suitable for the Fig. 6.5 network. Our algorithm concerns the separation of the nodes and keeps the time-complexity acceptable by using parallel processors and controllable iterations. All operations are implemented by the control unit at the core node.

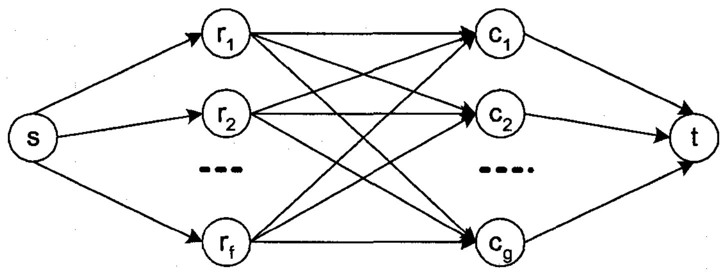


Fig. 6.20. The flow network corresponding to Fig. 6.5 network

First of all, we build a flow network in Fig. 6.20 in which the nodes are connected by directional arcs. This network is used to model the operations of multiple parallel processors. Each row node r_i $(i=1...f)$ corresponds to a switch in the first stage and each column node c_j $(j=1...g)$ corresponds to a switch in the third stage in the Clos network of Fig. 6.5[4]. We define s to be the source node and t to be the destination node. The traffic propagates between adjacent nodes in the arc's direction, and the amount of the traffic load (here it refers to the number of time slots) is defined to be the flow on the arc. The flow from node s to r_i is xsr_i, the flow from node r_i to c_j is xrc_{ij} and the flow from node c_j to t is xct_j. The flows going out a row or column node should not exceed the flows going into the same node. Suppose the traffic matrix is $\boldsymbol{T}=[t_{ij}]$ ($1 \le i \le f, 1 \le j \le g$), where t_{ij} represents the traffic requirement (number of waiting message slots) from Switch Module #i in the first stage to Switch Module # j in the last stage (in Fig. 6.5. Let R_i denote the sum of row i and C_j denote the sum of column j in $\boldsymbol{T}$

[4] The names "row" and "column" are used for the related elements in the traffic matrix to be processed by the processors, but not the positions as shown in Fig. 6.20 itself.

($1 \le i \le f, 1 \le j \le g$). We shall map the traffic matrix into the flow network to obtain *W* flow matrix $[xrc_{ij}]$ and each matrix corresponds to the switching matrix of a switching fabric in the core node.

We employ $f+g+2$ processors to implement the TWA in a parallel mode. Each processor locates in a node in Fig. 6.20, and is denoted by the same symbol as its node. These processors are divided into four types: the source processor (s), the destination processor (t), the row processors (r_i) and the column processors (c_j) ($1 \le i \le f, 1 \le j \le g$). Processor s and t contain a copy of the traffic matrix $\boldsymbol{T}$ and processors r_i and c_j contain the flow information on their adjacent arcs respectively. Each processor is aware of the change of the flows on any adjacent arc immediately and can exchange information with any adjacent node (the transmission path is not shown in the diagram). During each slot period, the processors implement the following operations and exchange information along the arcs.

<u>Source Processor</u> s:

1) At the beginning of a time slot, form a traffic matrix $\boldsymbol{T} = [t_{ij}]$ by counting all waiting requests and removing the expired requests (MSD exceeded). t_{ij} represents the number of message slots waiting to go from ET#i to ER#j (referring to Fig. 6.5).
2) At the end of a time slot, calculate $R_i = \sum_{j=1}^{g} t_{ij}$, and set $xsr_i = \min\{K_i, R_i\}$. The value of xsr_i is sent to processor r_i.

<u>Destination Processor</u> t:

1) At the beginning of a time slot, form a traffic matrix $\boldsymbol{T} = [t_{ij}]$ by counting all waiting requests and removing the expired requests (MSD exceeded). t_{ij} represents the number of message slots waiting to go from ET#i to ER#j (referring to Fig. 6.5).
2) At the end of a time slot, calculate $C_j = \sum_{i=1}^{g} t_{ij}$, and set $xct_j = \min\{H_j, C_j\}$. The value of xct_j is sent to processor c_j.

<u>Row Processor</u> r_i (*i=1...f*):

1) Receive the value of xsr_i from processor s.
2) Randomly assign xsr_i flow to some arcs starting from r_i. The flow assignment should satisfy $xrc_{ij} \le t_{ij}$ and $\sum_{j=1}^{g} xrc_{ij} \le xsr_i$.

<u>Column Processor</u> c_j (*j=1...g*):

1) Receive the value of xct_j from processor t.

2) Count the incoming flows to node c_j. If $\sum_{i=1}^{f} xrc_{ij} \geq xct_j$, randomly reduce xrc_{ij} on some arcs going to c_j until $\sum_{i=1}^{f} xrc_{ij} = xct_j$. The reduction should satisfy $xrc_{ij} \geq 0$.
3) Update the traffic matrix: $t_{ij} \leftarrow (t_{ij} - xrc_{ij})$. Send new t_{ij} (*j=1...g*) to Processor r_i.

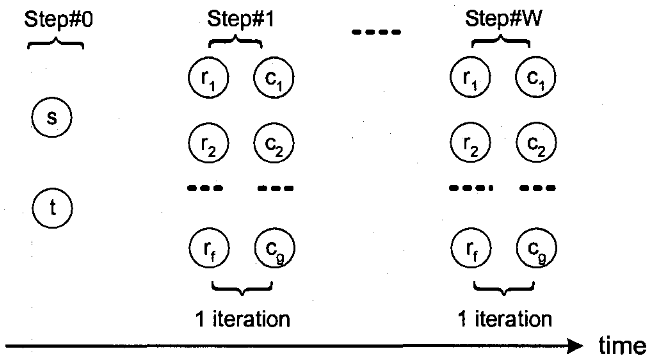


Fig. 6.21. TWA operation in one slot period

The overall operation of all processors in one slot period is depicted in Fig. 6.21, which includes *W+1* sequential steps. In each step, the processors in a vertical column work independently and simultaneously.

1) In Step #0, the source processor *s* builds the traffic matrix and initializes the flows leaving the source node; at the same time, the destination processor *t* also builds the traffic matrix and initializes the flows going into the destination node.
2) In Step #1, all row processors, $r_1 ... r_f$, execute the operations described above simultaneously, and then all column processors, $c_1 ... c_g$, execute the described operations simultaneously as well. The operations of all row and column processors consist of an iteration.
3) From Step #2 to Step #W, repeat the operations as in Step #1.

After each step of Steps #1 to #W, the flows on the arcs between the row and column nodes become the switching matrix for one wavelength switch. After one slot period, all *W* switching matrixes are formed and sent to the ETs. The flow xrc_{ij} $(1 \leq j \leq g)$ in a general Step #k means that ET#i can send xrc_{ij} slots to ER#j using wavelength #k (*k=1... W*). When an edge node receives scheduling results, it sends out waiting cells in the specified wavelength channels and time instants (referring to Fig. 6.19). The cells in the same VOQ are served in FCFS manner to maintain the sequence of the message packets. If different VOQs are set for different classes of traffic, the edge node will send out the high priority traffic first to ensure QoS. Obviously, the routing (choosing a fiber), wavelength assignment and time slot assignment problems are solved simultaneously. The core node will adjust the status of the switching fabrics at appropriate slot time using the synchronization scheme in Fig. 6.19.

By allowing multiple processors to work in a parallel mode (as shown in Fig. 6.21), the time complexity is much reduced. If time permitting, we can execute more iterations of the operations on row and column processors in each step to improve the throughput since each iteration will add more flows on the arcs from the previous iteration. In this case, the complexity of each step is multiplied by the number of iterations.

The time complexity of our parallel scheduling is the sum of the calculation time of all steps in Fig. 6.21, including the communication overhead among different processors. In Step#0, the complexity of Processor s and t is $O(f \cdot g)$. The operation of Processor r_i has the complexity of $O(\max(g, H_i))$ and the operation of Processor c_j has the complexity of $O(\max(f, K_j))$. Thus, the total complexity of our parallel scheduling is $O(f \cdot g + W \cdot (\max(g, H_i) + \max(f, K_j))$. If n iterations are enabled in each step of Processor r and c operations, the total time complexity becomes $O(f \cdot g + n \cdot W \cdot (\max(g, H_i) + \max(f, K_j))$. Considering the number of switches at the first and third stages is often in the same order and it is usually larger than the number of links of a small switch, we obtain the approximate time complexity of $O(f^2 + n \cdot W \cdot f)$. For a network in which the number of switches is in the order of tens, it is possible to implement the proposed scheduling in one time slot period in microsecond using matured electronic technology.

If the traffic belongs to different classes, the higher-priority slots will be transmitted first while the lower-priority ones wait for the next TWA turn. One can see that our TWA algorithm can support multiple classes of traffics, although only two classes are used to verify our proposal in later simulations.

If selectors are used and some specific wavelength channels are assigned to each edge node, then a selector becomes transparent to the scheduling operations because each edge node can now communicate with the core node directly. If each edge node shares the channels in a Round-Robin manner, a core node needs to keep a copy of the waiting slots from each edge node, and then selects the corresponding record to generate a traffic matrix during each scheduling step. In a long run, the average offered load to a core node is expected to be the same no matter which configuration is adopted for the selectors.

6.6 OTDM Performance

We shall examine the effect of different parameters. We use the same basic parameters as the OBS operation in Section 6.4 except for the following ones.

- number of VOQs in each ET: 16 (each VOQ stores one class of traffic to one destination ER)
- maximum assembly delay for slot assembly: 100 us
- maximum scheduling delay (time out): 300 slots
- slot length: 10 us (9 us for data cell, 1us for switching time)
- iteration in each scheduling step: 1

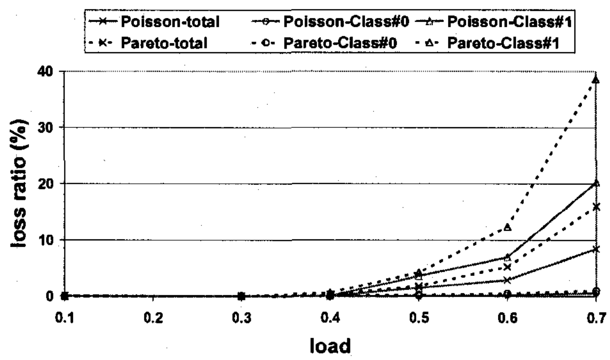


Fig. 6.22. Loss performance

6.6.1 Basic Performance

Fig. 6.22 illustrates the average loss ratio of different traffic types. No packet loss is observed when the Pareto traffic load is less than 0.3 and when the Poisson traffic load is less than 0.4. With the increasing of traffic load, the loss rate increases almost exponentially. For example, the total Poisson traffic has about 3 percent loss at load of 0.5, and about 9 percent loss at load of 0.7. When traffic load is no more than 0.5, the total loss is less than 5 percent for both Poisson and Pareto traffic, and the high priority traffic (Class#0) suffers no more than 10^{-4} loss. Considering the advantages of our topology and scheduling design, we can say our design provides a good transmission system. The Pareto traffic has a worse performance than the Poisson traffic, which can be explained by the more bursty naturals introduced by the Pareto distribution.

The delay performance is demonstrated in Fig. 6.23. For the Poisson traffic, the average system delay of all packets ("Poisson-total" in the diagram) decreases slightly as the load increases from 0.1 to 0.4. This can be explained in that the major delay is caused mainly by the waiting time of slot assembly, and higher traffic load would therefore accelerate the assembly procedure. When the traffic load increases further (e.g. 0.4), the waiting time for the TWA operation now becomes the major factor and more packets have to wait longer time to be scheduled under a high load. Other traffic types exhibit a similar behavior. Since the longest propagation time between edges and the core is 1ms (200km), the waiting time of each packet must be longer than $2 \times 1ms + 10us$ (Ref: Fig. 6.19). All curves in Fig. 6.23 have demonstrated that all traffic types match this conclusion in the simulated load range. The ETC delay is equal to the system delay plus the propagation delay. Thus, it exhibits the same trend except the delay value is higher than the system one at each load point and traffic type. For all traffic types and full simulated load range, the ETC delay is always less than 3 ms.

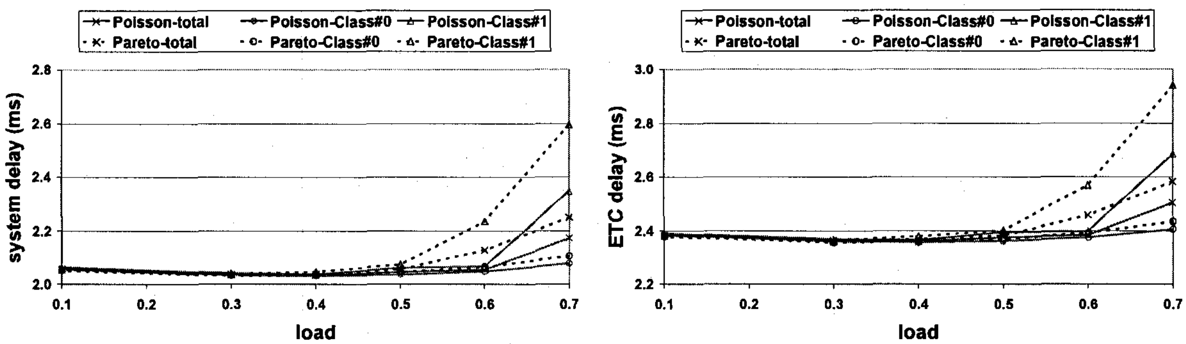


Fig. 6.23. Delay performance

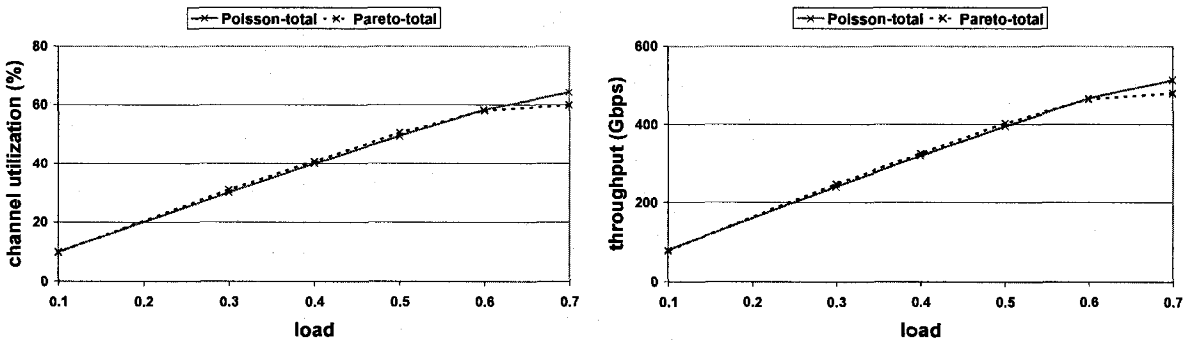


Fig. 6.24. Utilization and throughput

Fig. 6.24 shows that the channel utilization and total throughput increase linearly along with the offered traffic load. Both Poisson traffic and Pareto traffic achieve similar performance. When the load is 0.2, the utilization is almost 20 percent. When the load is 0.6, the utilization is very close to 60 percent. Thus, the designed topology and scheduling method can utilize the link resources efficiently disregard the change of input traffic.

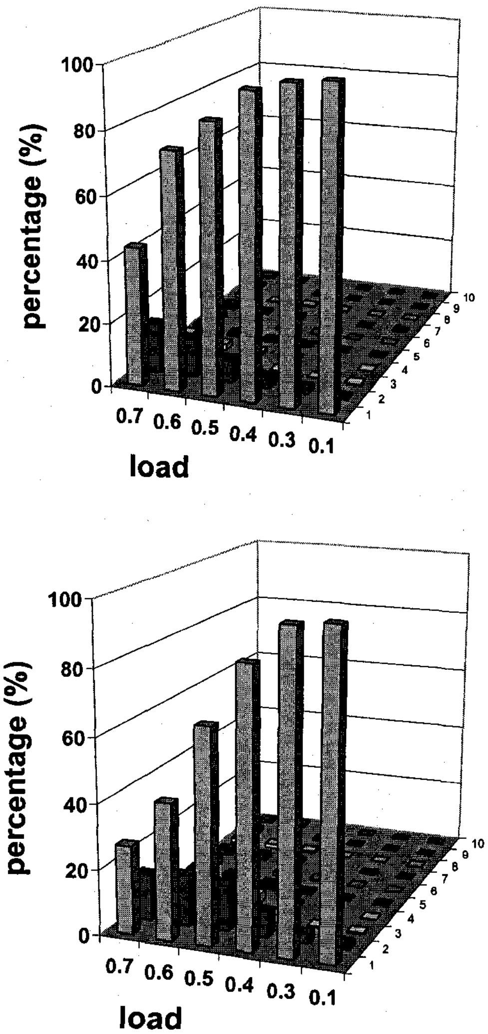


Fig. 6.25. Scheduling of (a) Poisson traffic (b) Pareto Traffic

In our TWA algorithm, a request may have to wait for some slot time to be scheduled after it arrives at a core node. According to our parameter settings, the longest round trip distance is 200 time slots, while each cell is allowed to stay in the edge node for 300-slots time. Therefore, each request has 100 chances to be scheduled in the core node. Fig. 6.25 depicts the percentage of requests that are scheduled and transmitted successfully as a function of the number of slots they have been waiting in the core node. With the increasing of traffic load, more requests have to wait for a longer time, but most of them, either Poisson or Pareto traffic, are processed within ten time slots; thus the TWA scheduling does not create significant extra delay. This observation complies with the delay performance in Fig. 6.23 and it also demonstrates the efficiency of our TWA algorithm.

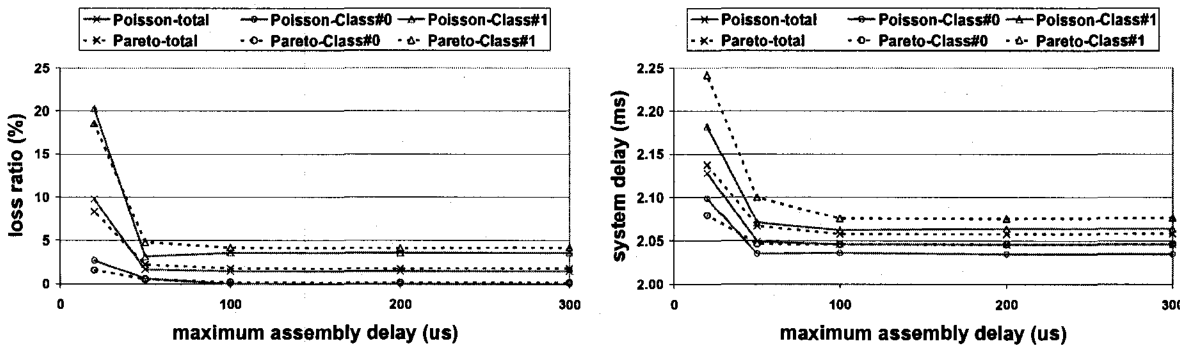


Fig. 6.26. Loss and delay performance

6.6.2 Effect of MAD

Fig. 6.26 depicts the loss and delay performance when the MAD increases from 20us to 300us. If we focus on the average loss of total Poisson traffic, the MAD does not affect its loss significantly when it becomes much longer than the slot period, say larger than 50us. Because a large MAD takes little effect on the cell assembly, the system becomes insensitive to it. A small MAD forces most cells to be partially filled, and then too many cells are formed at edge nodes. The system cannot meet the requirement of so many message cells, thus has to drop many of them. Therefore, we observe a sharp increase in loss rate at low MAD range (below 50us). For the same reason, the packets have to wait for longer time if more message cells are generated, which results in a longer system delay in the low MAD range. Other traffic types show a similar behavior as the Poisson traffic. Pareto traffic generates worse performance than Poisson traffic, and Class#0 traffic generates better performance than Class#1 traffic, which can be attributed to the bursty characteristics of Pareto distribution and the priority assigned to the Class#0 traffic.

6.6.3 Effect of MSD

The average system delay of all Poisson packets increases almost linearly when the MSD changes from 220 slot periods to 400 slot periods in Fig. 6.27. In our parameter setting, the MAD has little effect on the total delay comparing to the much longer round trip time, so the delay a packet is imposed in an edge node has a nearly-linear relationship with the MSD. Other traffic types and classes demonstrate similar delay trend, but the actual values are either higher or lower depending on its characteristics or priority which has been explained beforehand.

The MSD decides how many chances a request can be processed until it is scheduled. The longer the MSD is, the more chance a request can be processed. In our simulation study, each request has 100 chances to be processed. In Fig. 6.28, the Class#0 Poisson traffic exhibits a loss rate from 0.3 percent to 0.02 percent as the MSD increases from 220 to 400 slot periods. Having a MSD longer than 250 slots does not have much more improvement on the loss performance. This observation is supported by the results in Fig. 6.25 since most requests are processed within a small number of time slots after arriving at the core node. The high-priority traffic is the major target of a network transmission, so a MSD around 250 slot periods is good enough for Poisson traffic. As for Pareto traffic, similar observation applies.

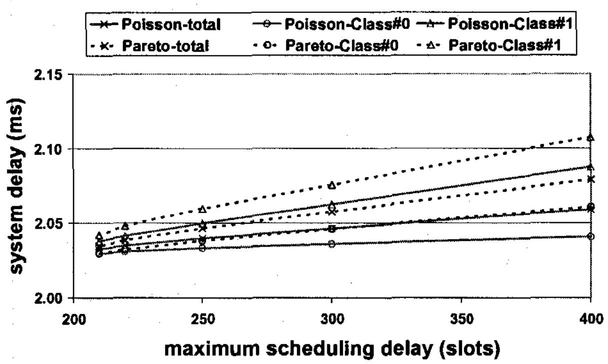


Fig. 6.27. Delay performance vs. MSD

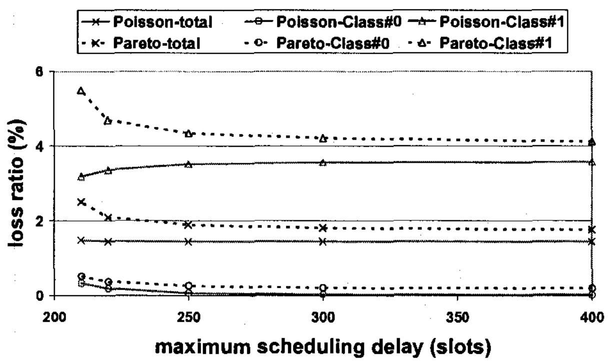


Fig. 6.28. Loss performance vs. MSD

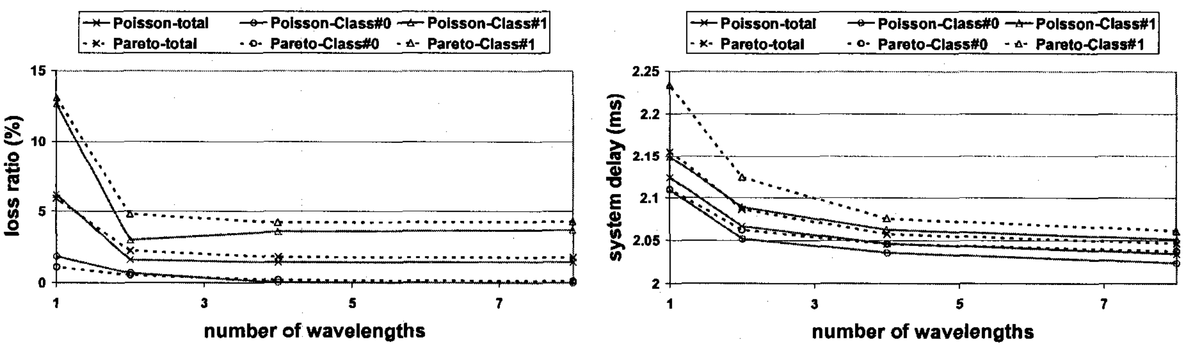


Fig. 6.29. Loss and delay performance vs. the number of wavelengths

6.6.4 Effect of the Number of Wavelengths

With an increasing traffic demand, we can add more wavelengths to an exiting network. We check the effect of this addition in Fig. 6.29. The total Poisson traffic generates a much lower loss rate, from 6 percent to 2 percent, as we increase the number of wavelength from 1 to 2. The system delay of all Poisson traffic decreases from 2.12ms to 2.07ms as well. If we continue to increase the number of wavelength, the Poisson traffic shows little difference on the loss and delay performance. All other traffic types demonstrate same performance changes. This phenomenon can be explained that the later case gives the traffic request more chances by sharing wavelengths. But further scaling of the number of wavelengths has no improvement.

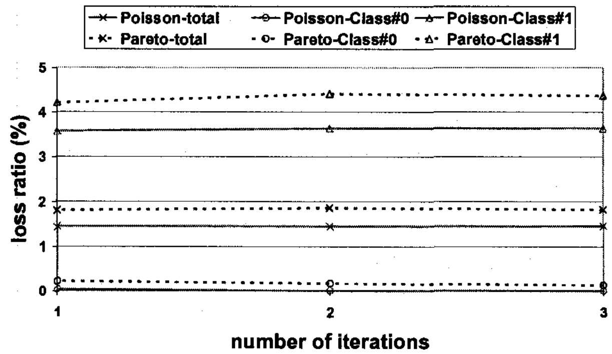


Fig. 6.30. Loss performance vs. the number of iterations

6.6.5 Effect of the Number of Iterations

Our parallel scheduling scheme is a heuristic way which adopts only one iteration in each step of the TWA operation. If time permitting, more iterations can be executed to schedule as more requests as possible. However, Fig. 6.30 shows that this increase does not generate improvement. The total Poisson traffic suffers almost the same 1.5 percent loss when 1, 2 or 3 iterations are applied in each TWA step. Other data types do not show different loss neither when using different iterations. Therefore, we do not have to employ complicated components to implement multiple iterations.

We have also simulated other parameter combinations, which demonstrate that all parameters exhibit similar effects as shown in this section. In all simulation scenarios, the VOQs have never overflowed. Since the 200 Mbytes buffer adopted in our design is a mature technology, we can say that the buffer size is not a constraint in our overlaid-star network.

6.7 Comparisons and Discussions

We compare the OTDM and the OBS operations in the same network in order to find out the pros and cons of each scheme. These results can further help our network design. We shall use the same 8x8 star network to evaluate our methods. The basic parameter settings are the same as previous sections.

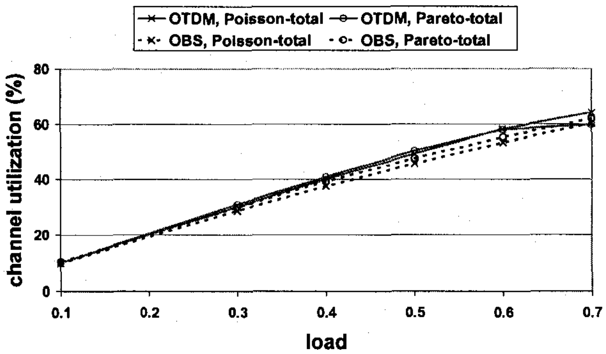


Fig. 6.31. Utilization using OTDM and OBS

6.7.1 Comparison of OTDM and OBS

The channel utilization under different traffic loads is illustrated in Fig. 6.31. In the low load range (<0.3), the difference of two schemes is negligible. With increasing traffic load, the OTDM achieves about 3 to 5 percent higher utilization than the OBS most of the time. Despite the small difference, it is noticed that both schemes achieve very good utilization for all simulation scenarios. Thus, our design can accommodate both OTDM and OBS schemes and both of our designed schemes can be a candidate switching method in the applications.

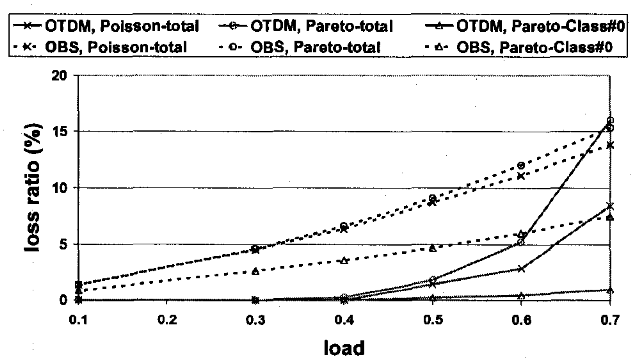


Fig. 6.32. Loss rate using OTDM and OBS

We then compare the loss ratio in Fig. 6.32. For both Poisson and Pareto traffics, the OTDM performs better in terms of loss rate. For example, the OTDM loss rate of Class#0 Pareto traffic is less than 1 percent, while the OBS loss of Class#0 Pareto traffic is almost 5 percent when the load is 0.5. Even though the OTDM generates a little bit more total loss than the OBS in the high load range (say 0.7), the OTDM can provide better performance to the high-priority traffic in the whole load range.

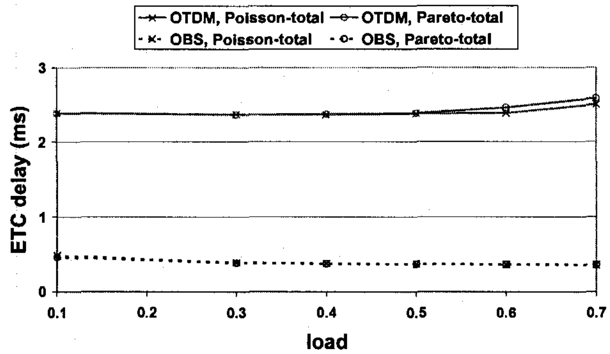


Fig. 6.33. ETC delay using OTDM and OBS

Fig. 6.33 shows that the OBS is superior to the OTDM in term of the delay performance because the OBS system sends out the ready bursts within a short time without waiting for the reservation acknowledgement. For a given load and traffic type, the OTDM generates about 2ms longer delay than the OBS operation. Therefore, each scheme has its pros and cons. On the other hand, the performance may demonstrate a different trend if a different scheduling algorithm is adopted. The final choice is likely to be a compromised consideration between network resources and performance requirement.

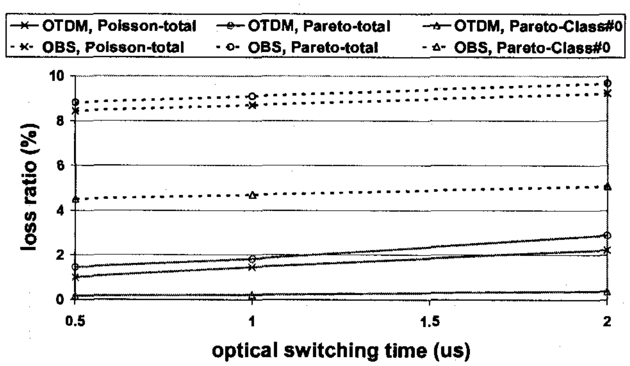
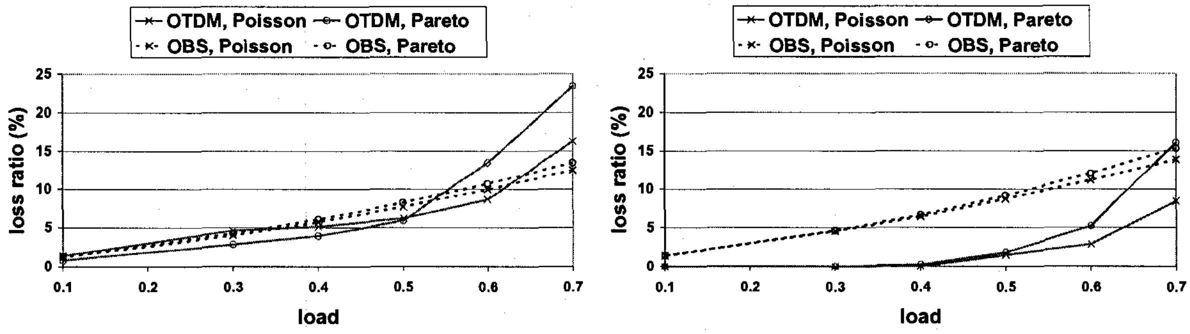


Fig. 6.34. Loss ratio w.r.t. optical switching time

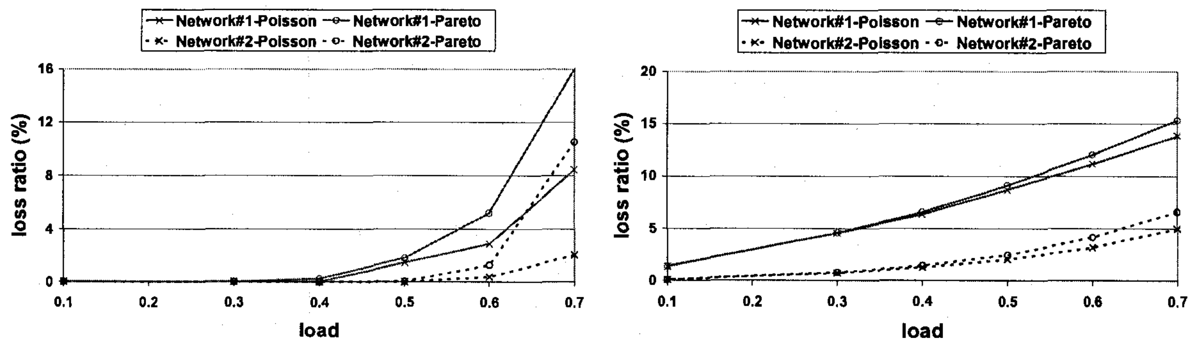
In both OTDM and OBS operations, during the period a switch changes its status, the channel is wasted and no message is transmitted. As the switching time changes from 0.5 to 2 us in Fig. 6.34, the packet loss ratio increases linearly too for all traffic types. When the switching time of core switches is equal to 1us, the system based on our parameter settings generates less than 5 percent loss to the high-priority traffic using either switching scheme. The OTDM system demonstrates much better performance since the OBS is collision-based.



(a) One wavelength (b) Four wavelengths

Fig. 6.35. Loss ratio w.r.t. the number of wavelengths

Fig. 6.35 illustrates the performance for single-wavelength and multi-wavelength networks. In the single-wavelength network, the OBS and the OTDM generates similar loss ratio when the load is lower than 0.5. In the higher load range, the OBS becomes better than the OTDM due to lower loss. In the four-wavelength network, the OTDM always generates lower loss than the OBS in the whole load range 0.1-0.7, which is generally the practical range for safe network operations in the industry. These results can be a reference for our network design too.



(a) Using OTDM (b) Using OBS

Fig. 6.36. Loss vs. Load: comparison of two networks

6.7.2 Network Scalability

When more and more applications emerge, we may meet the increasing traffic demands by two ways: either increasing the number of fibers or increasing the number of wavelength channels in a fiber. Fig. 6.36 depicts the loss ratio of two star networks using the same parameters in the previous section except that (1) in Network #1 there are (1, 2, 3, 4, 1, 2, 3, 4) fibers between the ETs and the core node and each fiber consists of four wavelengths; (2) in Network #2 there are (2, 4, 6, 8, 2, 4, 6, 8) fibers between the ETs and the core node and each fiber consists of two wavelengths. Then these two networks have the same transmission capacity in terms of the total number of wavelength channels. It is demonstrated that Network #2 has a significant improvement in terms of the average loss rate. However, the costs of these two networks are different because one network needs more fibers, and it also employs more optical switching components to support fewer wavelengths, whereas another one needs fewer fibers, and requires fewer switching components that can support more wavelengths. Therefore, when we consider expanding existing networks, the final choice depends on the available resources and the budget. The analysis in Section 6.2.1 would provide some guidelines in this aspect as well.

6.7.3 Other Issues

A network should handle the traffic in the same class equally. The treatment of messages going to the same destination and having the same QoS requirement should behave in the same manner as time evolves. In our OTDM scheme in the APOSN network, we schedule the higher priority traffics earlier than the lower ones. In our OBS scheme, we put high priority traffics at the tail of a burst in order to have a better chance to survive even after burst segmentation. Therefore, our design can provide good differentiated QoS. Our evaluation results have verified that Class#0 traffic always performs much better than Class#1 type.

With increasing demands, an APOSN need to support more nodes or traffics. Since our designed network keeps the payload in the optical domain throughout the core nodes, the switching and routing operation is transparent to the message itself. Then both OTDM and OBS are capable of supporting new applications or data types. Since our OTDM scheduling algorithm is a heuristic approach with a low time complexity and our OBS scheme only requires a very low processing capability too, the control unit can accommodate the increasing electronic messages without difficulty even when some new edge nodes are inserted to a star network. Our evaluation on the performance of different classes of traffics has supported the good scalability of our design.

Finally, we would like to emphasize that the discussed performance criteria have intrinsic relationships. For example, high throughput and fairness are contradictory goals in the design of scheduling algorithms. As a result, some forms of tradeoff between the throughput and fairness should be exploited. The performance can also become worse in case of failures due to a higher traffic load shifting to the remaining working links. We may face similar dilemma when analyzing other performance criteria, so the final choice depends on the tradeoff that is acceptable among different factors.

6.8 Concluding Remarks

In this chapter, we have proposed a feasible solution to implement an APOSN. We have designed the network topology and node architecture for different nodes. A Clos network model has been generalized. A network flow model is adapted to schedule OTDM time slots. Under this model, we apply parallel processing to accelerate wavelength/time slot assignment and analyze the time-wavelength assignment problem. Several parameters have been analyzed in terms of their effect on the major performance. When the OBS is applied in this model, we investigate the influence of signaling protocols and major system parameters. Several parameters affecting OBS operations have also been investigated by using the same Clos model. We have also tested our design in Chapters 3 to 5 in an APOSN. It is shown that both OTDM and OBS achieve good throughputs under all load range. We have compared these two operation schemes and presented some tradeoff analysis in their performance evaluation. It is demonstrated that our design is a feasible implementation that account for both the available technologies and the requirement of increasing applications. Our design has a good scalability, throughput and robustness by exploiting the existing optical technology. Note that we have set a high traffic load to test the effectiveness of our proposal. In a real network with a lower load, the performance should be better than what we have tested. Therefore, the overlaid-star architecture can provide an efficient and feasible solution for future high-speed networks using the currently available optical technologies.

CHAPTER 7
DESIGN GUIDELINES

We have proposed several algorithms on the DiffServ provision, flow control and architecture design of optical networks. Some design issues on each scheme have been provided in Chapters 3 to 6. Our design is based on approximations or heuristics, so we cannot provide accurate parameters for all cases. Instead, we shall summarize our design and provide some guidelines on some important issues about the overall system settings and operations under certain typical scenarios.

7.1 Network Topology

We have considered two network topologies, i.e., a mesh network and an APOSN. A meshed topology enables fine flexibility, scalability and robustness, but it is difficult and complicated to provide required performances. On the other hand, an APOSN network in Figs. 6.1 and 6.2 is more regular and controllable. We have chosen the overlaid-star topology using unidirectional transmission of the payload messages and bi-directional transmission of the control messages. With the progress of optical components, we might have more choices on the topology of future all-optical networks. However, the high-cost is a big obstacle to build a new network. Thus, the mesh topology by interconnecting existing networks is more practical in the near future. To facilitate all possible technology development and applications in a long term, we need to be prepared to both possibilities that have been studied in this thesis.

7.2 Node Architecture

The same architecture in Figs. 2.2 and 2.3 can be used for the edge nodes and the core nodes respectively under different topologies and operations. Most components for the OTDM and the OBS schemes can be the same except one needs to provide more computation capacity on the control unit and synchronization function for the OTDM operation. In all our simulation scenarios, the VOQs have never overflowed. Since the buffers adopted in our design can be implemented by mature technology, we can say that the buffer size is not a constraint in our overlaid-star network. The switching time of the optical fabrics affects the efficiency of the system because no message can be transmitted when a switch tunes its status. As shown in Fig. 6.34, the change of the switching time has a small impact on the APOSN performance. By comparing the results in Sections 4.4/5.4 and Appendix G/I, we find our control scheme is always valid for different switching times, except a lower efficiency as the time increases. In summary, a universal architecture as in Figs. 2.2 and 2.3 has provided a lot of flexibility for choosing different operations. This idea is also helpful for future design of other network topologies or operation schemes.

7.3 Operational Issues

We have chosen slottless OBS and "slotted OBS" (OTDM) for our networks. In a meshed optical network, the OBS scheme enables end-users to operate the high-speed optical system using the relatively-slower electronic control units. It also avoids complex scheduling algorithms. Our flow controller for a meshed network is based on the analysis of a specific flow, without considering other flows sharing the same link resources. Our DiffServ provision only needs to make a balance of the edge-based parameters among many flows, thus it is not a challenge for a large network. On the other hand, in a regular APOSN topology, we have designed an efficient heuristic for the OTDM operation. According to our comparison in Section 6.7 (an example is demonstrated in Fig. 6.35), when the packet arrival is a Pareto process in a single-wavelength APOSN, the OTDM is our choice to achieve a lower loss rate when the load is less than 0.55, while the OBS is preferred at a higher load. Moreover, the performance may demonstrate a different trend if some parameters or traffic characteristics have changed. The final choice is a combined consideration of network resources and performance requirements, while the study of this thesis is a useful reference.

7.4 Other Issues

There are more issues in the network design. Here are some typical ones. The final design of a network is a comprehensive consideration of different factors.

7.4.1 Fairness

A network should handle the traffic in the same class equally. The treatment of the message going to the same destination and having the same QoS requirement should behave in the same manner with the time going on. In an OBS network, the burst contains many packets and the source has no knowledge of the following transmission and processing. Thus, it is difficult to distinguish the traffic with different QoS requirements. Even though we have designed some useful schemes to provide required QoS for an optical burst flow, it also affects other flows as shown in Fig. 5.6. In an OTDM network, it is easier to assign a priority to certain messages through the scheduling algorithms. Thus, the OTDM technique is able to achieve fairness more easily, but requires more expensive devices. In summary, the ability of fairness provision is determined by the switching scheme we choose.

7.4.2 Time-Complexity

Each switching method consumes some computation power to arrange the message transmission. The time for the algorithm calculation and processor communication determines the time-complexity. The proposed TWA algorithm in Section 6.5.2 for the OTDM operation in an APOSN always has a number of computation steps. Since our TWA scheme is a heuristic method and the number of iterations is controllable, our OTDM operation is then easy to implement. Our OBS scheme just sends out the burst in a random way without waiting any information of the network. Then the OBS has a lower time-complexity. Our proposed DiffServ and flow controllers in Chapters 3 to 5 are based on some simple operations at the edge nodes only, which do not involve any complex algorithm or time-consuming operations. Therefore, all proposals in this thesis have a low time-complexity. In other words, the time-complexity is not a limitation when we choose either scheme being studied in this thesis.

7.4.3 Scalability

With increasing users and applications, the scalability problem becomes a critical point in the network design. The scalability involves in two aspects, the ability to support higher number of switch elements, and the ability to support the new emerging applications. As for a meshed network, each node works in sort of distributed manners. Neither our flow control mechanisms nor our DiffServ schemes require any global information for a specific flow, so it is not a problem in terms of scalability. It is also easy to scale a meshed network by adding more links or nodes without affecting our designed QoS schemes. For example, the results in Fig. 5.6 show that the control on *Flow* 1-5 has very small impact on other flows, which means our controller still works if the network scales. Moreover, taking the APOSN in Fig. 6.36 as an example, we can easily scale it by adding more fibers while achieving a lower loss rate at the same time. In this case, it is a better choice to scale a network by trading fibers than wavelengths.

Since the complexity of our OTDM TWA algorithm is linear to the number of nodes as discussed in Section 6.5.2, it is able to finish the processing of the increased requirements in one slot period. On the other hand, the OBS-based network has pretty low processing task too, which means it can accommodate the development of a network without difficulty. The payload is kept in the optical domain throughout the core nodes, so the switching and routing is transparent to the message itself. Then both the OTDM and the OBS are capable of supporting new applications or data types easily.

Summing up, the schemes and algorithms being proposed in this thesis would not introduce any difficulty for system scaling. Although based on heuristic, our design appears to be scalable from our many simulations. In other words, the DiffServ and flow control schemes through edge-based operation, and the OTDM TWA using parallel processors in this thesis (Chapters 3 to 6) are good choices for network design.

7.4.4 Protection and Restoration

A star network has a serious problem with the central node failure, while this problem does not exist in our APOSN topology. A core node failure only affects the operation of one star, while other star networks would not be disturbed. Since an edge node has connections to multiple core nodes, we can easily reroute the traffic to another star (through another core node). This protection and restoration scheme is simple and efficient.

In a meshed network, some sophisticated schemes are expected to detect the link or node failure, calculate the protection path and reroute the affected traffic. Thus, a lot of operations are involved and more nodes would be affected than in an APOSN network. We need to determine the best restoration route for each wavelength demand, given the network topology, the capacities, existing routes and other factors, and determining working and backup routes for each wavelength path to minimize network cost. This is not the topic in this thesis.

In summary, a simple topology of APOSN is more robust than a meshed one.

7.4.5 Optimization and Integration

Finally, we would like to emphasize that the discussed performance criteria have intrinsic relationships. When several parameters are modified at the same time, some form of tradeoff between these parameters should be exploited. We may face similar dilemma when analyzing other combined performance criteria, so the final choice depends on the tradeoff that is acceptable among different parameters. A possible solution is to find a good choice by using some optimization methods. Another possible solution is to find out some integrated schemes by incorporating the advantages of different choices. However, these works often introduce a high complexity which may forbid its applicability. On the other hand, the proposals in this thesis have taken advantage of some simple schemes so that they are practical solutions. We shall talk about some possible improvements in the next chapter.

CHAPTER 8
CONCLUSIONS

In this research, we have proposed a more accurate traffic arrival model (than the traditional *M/M/k/k* model) and their formulas to calculate the loss rate in OBS nodes. An edge-based DiffServ scheme has been designed to achieve differentiated loss ratio among burst flows without affecting the operation at the core nodes. It has a good flexibility and scalability in the heterogeneous Internet, and works successfully alone or when integrated with the preemptive scheme.

We have also studied the interaction of optical burst flows under different assembly parameters. Our analysis has led to the design of a feedback-based controller consisting of several effective loss predictors and burst adjusters. Our schemes utilize the information and operations at the edge nodes only, but without any global network information or pre-defined global criteria as required in many existing schemes. The control process does not affect the overall network throughput and its influence on the end-to-end delay is shown to be very minimal under different traffic conditions, assembly algorithms and signaling protocols. These features demonstrate that our controller is superior to the conventional OBS control methods with only rate-adjustment algorithms at the burst level. Same as our DiffServ scheme, our flow controller also has a good scalability and flexibility. Therefore, our work can provide an efficient QoS solution for OBS networks.

Our OBS QoS schemes are then applied in a star-based APOSN topology. This new architecture is a feasible choice of future optical networks due to its reduced cost and system operation complexity. We have provided systematic evaluations on the parameters and operations of the APOSN networks. We have examined the loss, delay and throughput performance under the OBS operation. Moreover, an efficient TWA algorithm has been developed for the OTDM operation in the APOSN based on an asymmetrical multiconnection Clos model. Our APOSN network has integrated the availability of the network components and the requirement of traffic demands. Thus, this topology is an efficient choice for future all-optical networks by using enabling technologies.

In summary, our research has investigated the important issues of topology design, differentiated service provisioning, flow control, resource sharing and network performance evaluation methods in optical networking. All designs and proposal have been analyzed and verified successfully by simulations.

8.1 Future work

All of our proposals have been verified to be effective. Improvements are still possible. Many can be found in the design guidelines where we do not have a definite answer, or due to the lack of technologies. For example, the flow controller for OBS networks can be combined with scheduling algorithms at the core nodes to improve the convergence and the stability of the control process. When multiple flows are using control operations at the same time, it is worth investigating the interactions among them. Our generic controller contains a predictor and an adjuster, whose parameters are determined by the experience of users. It is possible to find out some optimal choice of these parameters by considering the network settings, e.g., the parameters Z and L for the estimation of loss rate. It is also an interesting topic to examine the DiffServ or flow control schemes in an un-balanced or asymmetrical network.

When designing the APOSN architecture, a possible choice is to use multiple-layer stars to cover a large area network. This architecture is supposed to have better structure than the arbitrary mesh networks and further coverage than the single-layer star being studied in this thesis. We have only studied two switching techniques (i.e., OBS or OTDM) in our networks. It is also possible to build a hybrid scheme by partitioning the system into two halves each of which working in a different mode in order to take advantage of both schemes. A scheme to integrate the operations is also attractive.

REFERENCES

[AkBo05] N. Akar and H. Boyraz, "Differentiated abr: a new architecture for flow control and service differentiation in optical burst switched networks", *Proc. ONDM 2005*, Feb 7-9, 2005, Milan, Italy, pp.11-18.

[AlAl95] M. Alimuddin, H. M. Alnuweiri and R. W. Donaldson, "The fat banyan ATM switch", *Proc. IEEE INFOCOM '95*, Apr 2-6 1995, Boston, MA, U.S.A., pp.659-666.

[Amst83] S. Amstutz, "Burst switching--An introduction", *IEEE Communications Magazine*, vol.21, no.8, pp.36-42, Nov 1983.

[ArSu05] P. Argibay-Losada, A. Suarez-Gonzalez, M. Fernandez-Veiga, et al., "From Relative to Observable Proportional Differentiation in OBS Networks", *Proc. CoNEXT'05*, Oct 24-27, 2005, Toulouse, France, pp.115-123.

[Atal06] B. S. Atal, "The history of linear prediction", *IEEE Signal Processing Magazine*, vol.23, no.2, pp.154-161, Mar 2006.

[BaPe03] T. Battestilli and H. Perros, "An introduction to optical burst switching", *IEEE Communications Magazine*, vol.41, pp.s10-s15, Aug 2003.

[BlBl98] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services", *Network Working Group, RFC 2475*, 1998.

[BlLe01] F. J. Blouin, A. W. Lee, A. J.M. Lee and M. Beshai, "Comparison of two optical-core networks", *Journal of Optical Network*, vol.1, no.1, pp.56-65, Jan 2001.

[BoCo04] G.v. Bochmann, M.J. Coates, T. Hall, L. Mason, R. Vickers and O. Yang, "The Agile All-Photonic Network: An architectural outline", *Proc. The 22nd Symposium on Communications*, Queen's University, Kingston, Canada, 2004.

[BoJu97] M.S. Borella, J.P. Jue, D. Banerjee, B. Ramamurthy and B. Mukherjee, "Optical components for WDM lightwave networks", *Proceedings of the IEEE*, vol.85, no.8, pp.1274-1307, Aug 1997.

[BrJa94] R. Braden, V. Jacobson and S. Shenker, "Integrated Services in the Internet Architecture: An Overview", *Network Working Group, RFC 1633*, 1994.

[CAID06] Packet Length Distributions, http://www.caida.org/analysis/AIX/plen_hist/.

[CaLi02] X. Cao, J. Li, et al., "Assembling TCP/IP packets in optical burst switched networks", *Proc. IEEE GLOBECOM 2002*, Nov 17-21, 2002, Taiwan, China, pp.2808-2812.

[ChDe03] H. J. Chao, K-L. Deng and Z. Jing, "PetaStar: A Petabit Photonic Packet Switch," *IEEE Journal on Selected Areas in Communications*, vol.21, no.7, pp.1096-1112, Sept 2003.

[ChHa01] Y. Chen, M. Hamdi and D. H.K. Tsang, "Proportional QoS over OBS Networks", *Proc. IEEE GLOBECOM 2001*, Nov 25-19, 2001, San Antonio, Texas, U.S.A., pp.1510-1514.

[ChJi03] H.J. Chao, Z. Jing and S.Y. Liew, "Matching algorithms for three-stage bufferless Clos network switches", *IEEE Communications Magazine*, vol.41, no.10, pp.46-54, Oct 2003.

[ChQi04] Y. Chen, C. Qiao and X. Yu, "Optical burst switching: a new area in optical networking research", *IEEE Network*, vol.18, no.3, pp.16-23, May-Jun 2004.

[ChWo04] A. Chen, A.K.-S. Wong and C.-T. Lea, "Routing and time-slot assignment in optical TDM networks", *IEEE Journal on Selected Areas in Communications*, vol.22, no.9, pp.1648-1657, Nov 2004.

[Clos53] C. Clos, "A Study of Non-Blocking Switching Networks", *Bell Sys. Tech. J.*, pp.406-24, Mar 1953.

[CrBe97] M.E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: evidence and possible causes", *IEEE/ACM Transactions on Networking*, vol.5, no.6, Dec 1997, pp.835-846.

[DeEr02] A. Detti, V. Eramo and M. Listanti, "Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)", *IEEE/OSA Journal of Lightwave Technology*, vol.20, no.2, pp.154-165, Feb 2002.

[DeGi94] M. Decina, P. Giacomazzi and A. Pattavina, "Multistage shuffle networks with shortest path and deflectionrouting for high-performance ATM switching: the closed-loop shuffleout", IEEE Transactions on Communications, vol.42, no.11, pp.3034-3044, Nov 1994.

[DoFa03] P. D. Dobbelaere, K. Falta and S. Gloeckner, "Advances in Integrated 2D MEMS-Based Solutions for Optical Network Applications", *IEEE Communications Magazine*, vol.41, pp.s16-s23, May 2003.

[DoGa01a] K. Dolzer, C. Gauger, et al., "Evaluation of Reservation Mechanisms for Optical Burst Switching", *Int. J. Electro. Commun*, 55(2001), no.1, pp.1-8, Jan 2001.

[DoGa01b] K. Dolzer and C. Gauger, "On Burst Assembly in Optical Burst Switching Networks - A Performance Evaluation of Just-Enough-Time", *Proc. 17th International Teletraffic Congress*, Salvador, Brazil, Sept 24-28, 2001.

[DuAb04] P. DU and S. ABE, "TCP performance analysis of optical burst switching networks with a burst acknowledgment mechanism", *Proc. 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications*, Aug 29-Sept 1, 2004, Beijing, China, pp.621-625.

[ElMo00a] J. M. H. Elmirghani and H. T. Mouftah, "All-Optical Wavelength Conversion: technologies and Applications in DWDM Networks", *IEEE Communications Magazine,* vol.38, no.3, pp.86-92, Mar 2000.

[ElMo00b] J. M. H. Elmirghani and H. T. Mouftah, "Technologies and architectures for scalable dynamic dense WDM networks", *IEEE Communications Magazine,* vol.38, no.2, pp.58-66, Feb 2000.

[FaFe02] P. Fan, C. Feng, et al., "Investigation of The Time-Offset-Based QoS Support with Optical Burst Switching in WDM Networks", *Proc. IEEE ICC 2002*, Apr 28-May 2 2002, New York, U.S.A., pp.2682-2686.

[FaKa04] M. Fayyazi, D. Kaeli and W. Meleis, "Parallel Maximum Weight Bipartite Matching Algorithms for Scheduling in Input-Queued Switches", *Proc. 18th International Parallel and Distributed Processing Symposium (IPDPS'04) 2004*, April 26-30, 2004, New Mexico, Mexico.

[FaZh04] F. Farahmand, Q. Zhang and J.P. Jue "A Feedback-Based Contention Avoidance Mechanism for Optical Burst Switching Networks", *Proc. Broadnets 2004 (Workshop on Optical Burst Switching)*, Oct 25, 2004, San Jose, CA, U.S.A..

[FaZh05] F. Farahmand, Q. Zhang and J.P. Jue, "A Closed-Loop Rate-based Contention Control for Optical Burst Switched Networks", *Proc. IEEE GLOBECOM 2005*, Nov 28- Dec 2, 2005, St. Louis, MO, U.S.A., pp.1989-1993.

[FIND06] FIND, http://find.isi.edu/.

[FlHa00] S. Floyd, M. Handley, J. Padhye and J. Widmer, "Equation Based Congestion Control for Unicast Applications", *Proc. SIGCOMM'00*, Aug 28-Sept 1, 2000, Stockholm, Sweden, pp.43-56.

[FlJa93] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol.1, no.1, pp.397-413, Aug 1993.

[Gaug03] C. M. Gauger, "Trends in Optical Burst Switching", *Proc. SPIE 5247 (ITCOM 2003)*, Sept 7-11 2003, Orlando, FL, U.S.A., pp.115-125.

[GeCa00] A. Ge, F. Callegati and L. S. Tamil, "On Optical Burst Switching and Self-Similar Traffic", *IEEE Communications Letters*, vol.4, no.3, pp.98-100, Mar 2000.

[GENI06] GENI Home, http://www.geni.net/.

[Good02] B. Goode, "Voice over internet protocol (VoIP)", *Proceedings of the IEEE*, vol.90, no.9, pp.1495-1517, Sept 2002.

[Gord95] J. Gordon, "Pareto process as a model of self-similar packet traffic", *Proc. IEEE GLOBECOM 1995*, Nov 13-17, 1995, Singapore, pp.2232-2236.

[GrDu03] J. Gripp, M. Duelk, J.E. Simsarian, et al., "Optical switch fabrics for ultra-high-capacity IP routers", *IEEE Journal of Lightwave Technology*, vol.21, no.11, pp.2839-2850, Nov 2003.

[HsLi02] C. Hsu, T. Liu and N. Huang, "Performance Analysis of Deflection Routing in Optical Burst-Switched Networks," *Proc. IEEE INFOCOM 2002*, Jun 23-27, 2002, New York, NY, U.S.A, pp.66-73.

[HuDo03] G. Hu, K. Daher and C. Gauger, "Does Burst Assembly Really Reduce the Self-Similarity?", *Proc. OFC 2003*, Mar 23-28, 2003, Atlanta, Georgia, U.S.A., pp.124-126.

[Hwan03] F. K. Hwang, "A Survey of Nonblocking Multicast Three-Stage Clos Networks", *IEEE Communications Magazine*, vol.41, no.10, pp.34-37, Oct 2003.

[HwLi03] F. K. Hwang, S. Liaw and L. Tong, "Strictly Nonblocking Three-Stage Clos Networks With Some Rearrangeable Multicast Capability", *IEEE Transactions on Communications*, vol.51, no.11, pp.1765-1767, Nov 2003.

[IrMa98] R. R. Iraschko and M. H. MacGregor, "Optimal Capacity Placement for Path Restoration in STM or ATM Mesh Survivable Networks", *IEEE Transactions on Networking* vol.6 no.3, pp.325-335, Jun 1998.

[IzAr02] M. Izal and J. Aracil, "On the influence of self-similarity on optical burst switching traffic", *Proc. IEEE GLOBECOM 2002*, Nov 17-21, 2002, Taiwan, China, pp.2308-2312.

[Jaco88] V. Jacobson. "Congestion avoidance and control," *Proc. SIGCOMM'88*, Aug 16-19, 1988, Stanford, California, U.S.A., pp.314-329.

[JaFa05] M. James and D. Fayek, "Improved Flow Control in Optical Burst Switched Networks for Healthcare Applications", *Proc. HEALTHCOM 2005*, Jun 23-25, 2005, BU.S.A.n, Korea, pp.61-66.

[JoCh01] A. Jourdan, D. Chiaroni, E. Dotaro, et al., "The Perspective of Optical Packet Switching in IP-Dominant Backbone and Metropolitan Networks", *IEEE Communications Magazine,* vol.39, no.3, pp.136-141, Mar 2001.

[KaHl87] M. Karol, M. Hluchyj and S. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch", *IEEE Transactions on Communications*, vol.35, no.12, pp.1347-1356, Dec 1987.

[KaLi02] W. Kabacinski and F. Liotopoulos, "Multirate Nonblocking Generalized Three-Stage Clos Switching Networks", *IEEE Transactions on Communications*, vol.50, no.9, pp.1486-1494, Sept 2002.

[KaMo04] T. Karagiannis, M. Molle, M. Faloutsos and A. Broido, "A nonstationary Poisson view of Internet traffic", *Proc. IEEE INFOCOM 2004*, Mar 3-11, 2004, Hongkong, China, pp.1558-1569.

[KaOk05] B. Kantarci, S. Oktug and T. Atmaca, "Analyzing the Effects of Burst Assembly in Optical Burst Switching under Self-similar Traffic", *Proc. the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop, 2005*, July 17-22 2005, Lisbon, Portugal, pp.109-114.

[Kart02] S. V. Kartalopoulos, *DWDM: Networks, Devices, and Technology*, Wiley-IEEE Press, 2002.

[Kell91] F.P. Kelly, "Loss networks", *Annals of Applied Probability*, vol.1, no.3, pp.319-378, 1991.

[Kiba95] W. Kabacinski, "On non-blocking switching networks for multichannel connections," *IEEE Transactions on Communications*, vol.43, no.2/3/4, pp.222-224, FebApr 1995.

[KiKi02] S. Kim, N. Kim and M. Kang, "Contention Resolution for Optical Burst Switching Networks Using Alternative Routing", *Proc. IEEE ICC 2002*, Apr 28-May 2 2002, New York, NY, U.S.A., pp.2678-2681.

[KiMu05] S. Kim, B. Mukherjee and Mi. Kang, "Integrated Congestion-Control Mechanism in Optical Burst Switching Networks", *Proc. IEEE GLOBECOM 2005*, Nov 28- Dec 2, 2005, St. Louis, MO, U.S.A., pp.1973-1977.

[Klei75] L. Kleinrock, *Queueing Systems*, Volume 1: Theory, New York, Wiley Interscience, 1975.

[Laev02] K. Laevens, "Traffic characteristics inside optical burst switched networks", *Proc. SPIE 4874 (Opticomm 2002)*, Jul 30, 2002, Boston, MA, U.S.A., pp.137-148.

[Leon94] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd version, Addison-Wesley, 1994.

[LiAn03] J. Li, N. Ansari and T. J. Ott, "FRR for Latency Reduction and QoS Provisioning in OBS Networks", *IEEE Journal on Selected Areas in Communications*, vol.21, no.7, pp.1210-1219, Sept 2003.

[LiAn04] J. Li and N. Ansari, "The impact of the burst assembly interval on the OBS ingress traffic characteristics and system performance", *Proc. IEEE ICC 2004*, Jun 20-24 2004, Paris, France, pp.1559-1563.

[LiCh03] S. Y. Liew, and H. J. Chao, "On Slotted WDM Switching in Bufferless All-Optical Networks", *Proc. of the 11th Symposium on High Performance Interconnects (HOTI'03)*, New York, NY, U.S.A., Aug 20-22, 2003, pp.96-101.

[LiCh05] Y. Liu, K. Chua and G. Mohan, "Achieving Max-Min Fairness in WDM Optical Burst Switching Networks", *Proc. HPSR 2005*, May 12-14, 2005, Hong Kong, China, pp.187-191.

[LiCh96] F. K. Liotopoulos and S. Chalasani, "Semi-Rearrangeably Nonblocking Operation of Clos Networks in the Multirate Environment", *IEEE Transactions on Networking*, vol.4, no.2, pp.281-291, Apr 1996.

[LiGo99] L. Y. Lin, E. L. Goldstein and R. W. Tkach, "Free-Space Micromachined Optical Switches for Optical Networking", *IEEE Journal of Selected Topics in Quantum Electronics*, vol.5, no.1, pp.4-9, Jan/Feb 1999.

[LiNg98] S. Liew, M.-H. Ng and C. Chan, "Blocking and Nonblocking Multirate Clos Switching Networks", *IEEE Transactions on Networking*, vol.6, no.3, pp.307-318, Jun 1998.

[LiSc03] L. Li, S. D. Scott and J. S. Deogen, "A Novel Fiber Delay Line Buffering Architecture for Optical Packet Switching", *Proc. GLOBECOM 2003*, Dec 1-5, 2003, San Francisco, CA, U.S.A., pp.2809- 2813.

[LiXi05] O. Liang, T. Xiansi, M. Yajie and Y. Zongkai, "A Framework to Evaluate Blocking Performance of Time-slotted Optical Burst Switched Networks", *Proc. IEEE LCN 2005*, Nov 15-17, 2005, Sydney, Australia, pp.258-267.

[LiYa04] Z. Liu and O. Yang, "Terminal-Pair Reliability Analysis of Overlaid-Star Networks", *Proc. CCCT 2004*, Aug 14-17, 2004, Austin, Texas, U.S.A., pp.406-411.

[LoTu03] K. Long, R. S. Tucker and C. Wang, "A new framework and burst assembly for IP DiffServ over optical burst switching networks", *Proc. IEEE GLOBECOM 2003*, Dec 1-5, 2003, San Francisco, CA, U.S.A., pp.3159-3164.

[LuZe03] J. Luo, Q. Zeng, H. Chi, Z. Zhang and H. Zhao, "The Impacts of Burst Assembly on the Traffic Properties in Optical Burst Switching Networks", *Proc. ICCT 2003*, Apr 9-11 2003, Beijing, China, pp.521-524.

[MaBo04] A. Maach, G. Bochmann and H. Mouftah, "Robust Optical Burst Switching", *Kluwer Telecommunication Systems Journal*, 27:2-4, pp.115-131, 2004.

[MaKu03] X. Ma and G. Kuo, "Optical switching technology comparison: optical MEMs vs. other technologies", *IEEE Communications Magazine*, vol.41, pp.s16-s24, Nov 2003.

[MaMo93] F. Masetti, P. Gavignet-Morin, D. Chiaroni and G. Da Loura, "Fiber Delay Lines Optical Buffer for ATM Photonic Switching Applications", *Proc. IEEE INFOCOM 1993*, Mar 28-Apr 1, 1993, San Francisco, CA, U.S.A., pp.935-942.

[MaVi06] L. Mason, A. Vinokurov, N. Zhao and D. Plant, "Topological design and dimensioning of Agile All-Photonic Networks", *Computer Networks*, vol.20, no.2, pp.268-287, Feb 2006.

[McAn96] N. McKeown, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch", *Proc. IEEE INFOCOM 1996*, Mar 24-28, 1996, San Francisco, CA, U.S.A., pp.296-302.

[Mcke99] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches", *IEEE/ACM Transactions on Networking*, vol.7, no.2, pp.188-201, Apr 1999.

[McMe99] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch", *IEEE Transactions on Communications*, vol.47, no.8, pp.1260-1267, Aug 1999.

[MuKh92a] B. Mukherjee, "WDM-based local lightwave networks. I. Single-hop systems", *IEEE Network*, vol.6, no.3, pp.12-27, May 1992.

[MuKh92b] B. Mukherjee, "WDM-based local lightwave networks. II. Multihop systems", *IEEE Network*, vol.6, no.3, pp.20-32, May 1992.

[NaWe04] K. Naik, D. Wei, et al., "A Reservation-Based Multicast Protocol for WDM Optical Star Networks", *IEEE Journal on Selected Areas in Communications*, vol.22, no.9, pp.1670-1680, Nov 2004.

[NeRa01] A. Neukermans and R. Ramaswami, "MEMS Technology for Optical Networking Applications", *IEEE Communications Magazine*, vol.39, no.1, pp.62-69, Jan 2001.

[NeRo02] M. Neuts, Z. Rosberg, H. Vu, J. White and M. Zukerman, "Performance Analysis of Optical Composite Burst Switching", *IEEE Communications Letters*, vol.6, no.8, pp.346-348, Aug 2002.

[NLR06] National LambdaRail, http://www.nlr.net/.

[OhHo02] S. Oh, H. Hong and M. Kang, "A Data Burst Assembly Algorithm in Optical Burst Switching Networks", *ETRI Journal*, vol.24, no.4, pp.311-322, Aug 2002.

[OkYa03] E. Oki, N. Yamanaka, K. Nakai and N. Matsuura, "Multi-stage switching system using optical WDM grouped links based on dynamic bandwidth sharing", *IEEE Communications Magazine*, vol.41, no.10, pp.56-63, Oct 2003.

[OmSi01] M. J. O'Mahony, D. Simeonidou, D. K. Hunter and A. Tzanakaki, "The Application of Optical Packet Switching in Future Communication Networks", *IEEE Communications Magazine*, vol.39, no.3, pp.128-135, Mar 2001.

[OPEN06] OPNET Modeler, http://www.opnet.com/products/modeler/home.html.

[OzFa02] T. Ozugur, F. Farahmand and D. Verchere, "Single-Anchored Soft Bandwidth Allocation System with Deflection Routing for Optical Burst Switching", *Proc. HPSR 2002*, May 26-29, 2002, Kobe, Hyogo pref. Japan, pp.257-261.

[OzVe02] T. Ozugur and D. Verchere, "Congestion Control Mechanism Using LSP Differentiation for Label-Switched Optical Burst Networks", *Proc. IEEE Workshop on IP Operations and Management 2002*, Oct 29-31, 2002, Dallas, Texas, U.S.A., pp.31-36.

[PaLe04] S. Park, C. Lee, et al., "Fiber-to-the-home services based on wavelength-division-multiplexing passive optical network", *IEEE Journal of Lightwave Technology*, vol.22, no.11, pp.2582-2591, Nov 2004.

[PaMa00] A. Patttavina, M. Martinelli, G. Maier and P. Boffi, "Techniques and Technologies Towards All-Optical Switching", *Optical Networks Magazine*, vol.1, no.2, pp.75-93, Apr 2000.

[PaPa03] G.I. Papadimitriou, C. Papazoglou and A.S. Pomportsis, "Optical switching: switch fabrics, techniques, and architectures", *IEEE Journal of Lightwave Technology*, vol.21, no.2, pp.384- 405, Feb 2003.

[PhCh04] M.H. Phung, K.C. Chua, G. Mohan, M. Motani and T.C. Wong, "Absolute QoS Signaling and Reservation in Optical Burst-Switched Networks", *Proc. IEEE GLOBECOM 2004*, Nov 29-Dec 3, 2004, Dallas, Texas, U.S.A., pp.2009-2013.

[PrPr06] B. Praveen, J. Praveen and C. Siva Ram Murthy, "A survey of differentiated QoS schemes in optical burst switched networks", *Optical Switching and Networking, article in press*.

[PuHa02] K. Pun and M. Hamdi, "Distro: A Distributed Static Round-Robin Scheduling Algorithm for Bufferless Clos-Network Switches", *Proc. IEEE GLOBECOM 2002*, Nov 17-21, 2002, Taiwan, China, pp.2298-2302.

[PuSh06] M. Phung, D. Shan, K. Chua and G. Mohan, "Performance Analysis of a Bufferless OBS Node Considering the Streamline Effect", *IEEE Communications Letters*, vol.10, no.4, pp.293-295, Apr 2006.

[Qiao00] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration," *IEEE Communications Magazine*, vol.38, no.9, pp.104-14, Sept 2000.

[QiYo00] C. Qiao and M. Yoo, "Choices, features and issues in optical burst switching", *Optical Networks*, vol.1, no.2, pp.36-44, Apr 2000.

[QiYo99] C. Qiao and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical internet", *Journal of High Speed Networks*, vol.8, no.1, pp.69-84, 1999.

[RaOv04] R. Rajaduray, S. Ovadia and D. J. Blumenthal, "Analysis of an edge router for span-constrained optical burst switched (OBS) networks", *IEEE/OSA Journal of Lightwave Technology*, vol.22, no.11, pp.2693-2705, Nov 2004.

[RaSi95] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks", *IEEE/ACM Transactions on Networking*, vol.3, no.5, pp.489-500, Oct 1995.

[RaTu03] J. Ramamirtham and J. Turner, "Time-sliced Optical Burst Switching," *Proc. IEEE INFOCOM 2003*, San Francisco, CA, U.S.A., Mar 30-Apr 3, 2003, pp.2030-2038.

[RoGo04] M. de Vega Rodrigo and J. Götz, "An Analytical Study of Optical Burst Switching Aggregation Strategies", *Proc. Broadnets 2004 (Workshop on Optical Burst Switching)*, Oct 25, 2004, San Jose, CA, U.S.A..

[RoVu03] Z. Rosberg, H. Vu, M. Zukerman and J. White, "Performance Analysis of Optical Burst-Switching Networks", *IEEE Journal on Selected Areas in Communications*, vol.21, no.7, pp.1187-1197, Sept 2003.

[SaSh03] A. Sahara, K. Shimano, Y. Takigawa and M. Koga, "Optical burst data switching utilizing GMPLS signaling", *IEE Electronics Letters*, vol.39, no.17, pp.1267-1269, August 21st, 2003.

[SeBe96] S. Seo, K. Bergman and P. R. Prucnal, "Transparent optical networks with time-division multiplexing", *IEEE Journal on Selected Areas in Communications*, vol.14, no.5, pp.1039-1051, Jun 1996.

[ShWu06] H. Shin, J. Wu and Y. Hsu, "The study of QoS guarantee in the optical burst switching internet backbone", *Optical Switching and Networking*, vol.3, pp.50-63, 2006.

[StBa00] T. E. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach*, Addison-Wesley, 2000.

[StKl99] K. E. Stubkjaer, A. Kloch, P. Hansen, et al., "Wavelength Converter Technology", *IEICE Trans. Commun.*, vol.E82-B, no.2, pp.390-400, Feb 1999.

[TaAj03] T. Tachibana, T. Ajima and S. Kasahara, "Round-Robin Burst Assembly and Constant Transmission Scheduling for Optical Burst Switching Networks", *Proc. IEEE GLOBECOM 2003*, Dec 1-5, 2003, San Francisco, CA, U.S.A., pp.2772-2776.

[TaGu04] C. Tan, M. Gurusamy and J. Lui, "Achieving Proportional Loss Differentiation Using Probabilistic Preemptive Burst Segmentation in Optical Burst Switching WDM Networks", *Proc. IEEE GLOBECOM 2004*, Nov 29-Dec 3, 2004, Dallas, Texas, U.S.A., pp.1754-1758.

[TaKa06] T. Tachibana and S. Kasahara, "Burst-Cluster Transmission: Service Differentiation Mechanism for Immediate Reservation in Optical Burst Switching Networks", *IEEE Communications Magazine*, pp.s46-s55, May 2006.

[TaOh03] H., TANIDA, K. OHMAE, Y. Choi and H. OKADA, "An Effective BECN /CRN Typed Deflection Routing for QoS Guaranteed Optical Burst Switching", *Proc. GLOBECOM 2003*, Dec 1-5, 2003, San Francisco, CA, U.S.A., pp.2601-2606.

[Tham98] Y. Tham, "Routing in Asymmetrical Multiconnection Three-Stage Clos Networks", *Networks*, vol.32, no.2, pp.77-83, Dec 1998.

[ThVo03] G. P. V. Thodime, V. M. Vokkarane and J. P. Jue, "Dynamic Congestion-Based Load Balanced Routing in Optical Burst-Switched Networks", *Proc. IEEE GLOBECOM 2003*, Dec 1-5, 2003, San Francisco, CA, U.S.A., pp.2628-2632.

[TuMe03] J.S. Turner and R. Melen, "Multirate Clos networks", *IEEE Communications Magazine*, vol.41, no.10, pp.38-44, Oct 2003.

[Turn99] J. S. Turner, "Terabit Burst Switching", *Journal of High Speed Networks*, vol.8, no.1, pp.3-16, Jan 1999.

[UCLP06] UCLP, http://www.uclp.ca/.

[UeEr04] J. Uengsakul and T. Erke, "Performance evaluation of assembly mechanisms for jet-based optical burst switching networks", *Proc. IEEE TENCON 2004*, Nov 21-24, 2004, Chiang Mai, Thailand, pp.101-104.

[VaCh92] A. Varma and S. Chalasani, "Asymmetrical Multiconnection Three-Stage Clos Network", *Proc. IPPS 1992*, Mar 1992, Beverly Hills, CA, U.S.A., pp.411-414.

[VaCh93] A. Varma and S. Chalasani, "Asymmetrical multiconnection three-stage Clos networks", *Networks,* vol.23, pp.427-439, 1993.

[VeCh00] S. Verma, H. Chaskar and R. Ravikanth, "Optical Burst Switching: A Viable Solution for Terabit IP Backbone", *IEEE Network*, vol.14, no.6, pp.48-53, Nov/Dec 2000.

[ViBe00] R. Vickers and M. Beshai, "PetaWeb Architecture," *Proc. Networks 2000 Symposium*, Sept 2000, Toronto, Canada.

[VlMo03] K. Vlachos, I. Monroy, et al., "Stolas: switching technologies for optically labeled signals", *IEEE Communications Magazine,* vol.41, pp.s9-s15, Nov 2003.

[VoTh03] V. M. Vokkarane, G. P. V. Thodime and V. U. B. Challagulla, "Channel Scheduling Algorithms Using Burst Segmentation and FDLs for Optical Burst-Switched Networks", *Proc. IEEE ICC 2003,* May 11-15, 2003, Anchorage, Alaska, U.S.A., pp.1443-1447.

[VoJu02] V. M. Vokkarane, J. P. Jue and S. Sitaraman, "Burst Segmentation: An Approach for Reducing Packet Loss in Optical Burst Switched Networks", *Proc. IEEE ICC 2002*, Apr 28-May 2 2002, New York, NY, U.S.A., pp.2673-2677.

[VoJu03] V. M. Vokkarane and J. P. Jue, "Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks", *IEEE Journal on Selected Areas in Communications*, vol.21, no.7, pp.1198-1209, Sept 2003.

[VoZh02] V.M. Vokkarane, Q. Zhang, J.P. Jue and B. Chen, "Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks", *Proc. IEEE GLOBECOM 2002*, Nov 17-21, 2002, Taiwan, China, pp.2747-2751.

[Wang03] S. Y. Wang, "Using TCP congestion control to improve the performances of optical burst switched networks", *Proc. IEEE ICC 2003,* May 11-15, 2003, Anchorage, Alaska, U.S.A., pp.1438-1442.

[WeMc00] J. Y. Wei and Jr. R. I. McFarland, "Just-In-Time Signaling for WDM Optical Burst Switching Networks", *IEEE/OSA Journal of Lightwave Technology*, vol.18, no.12, pp.2019-2037, Dec 2000.

[WuTs92] T. Wu and R. Lau, "A Class of Self-Healing Ring Architectures for SONET Network Applications", *Proc. IEEE GLOBECOM'90*, Dec 2-5, 1990, San Diego, CA, U.S.A., pp.444-449.

[XiVa00] Y. Xiong, M. Vandenhoute and H. Cankaya, "Control Architecture in Optical

Burst-Switched WDM Networks", *IEEE Journal on Selected Areas in Communications*, vol.18, no.10, pp.1838-1851, Oct 2000.

[YaMa88] O. W. W. Yang and J. W. Mark, "Reliability analysis of a multi-center tree network", *Proc. IEEE CCECE 1988*, Nov 1988, Vancouver, BC, Canada.

[YaMu00] S. Yao, B. Mukherjee and S. Dixit, "Advances in Photonic Packet Switching: An Overview", *IEEE Communications Magazine,* vol.38, no.2, pp.84-94, Feb 2000.

[Yang94] O. W. W. Yang, "Two-Center Tree Topologies for Metropolitan Area Network", *IEE Proceedings-I: Communications, Speech and Vision*, vol.141, no.4, pp.280-288, Aug 1994.

[YeHl87] Y. Yeh, M. Hluchyj and A. Acampora, "The Knockout Switch: A Simple, Modular Architecture for High-Performance Packet Switching", *IEEE Journal on Selected Areas in Communications*, vol.5, no.8, pp.1274-1283, Oct 1987.

[YoQi00] M. Yoo, C. Qiao and S. Dixit, "QoS Performance of Optical Burst Switching in IP-Over-WDM Networks", *IEEE Journal on Selected Areas in Communications*, vol.18, no.10, pp.2062-2071, Oct 2000.

[YuCh02] X. Yu, Y. Chen and C. Qiao, "A Study of Traffic Statistics of Assembled Burst Traffic in Optical Burst Switched Networks", *Proc. SPIE 4874 (Opticomm 2002)*, Jul 30, 2002, Boston, MA, U.S.A., pp.149-159.

[YuLi04] X. Yu, J. Li, X. Cao, Y. Chen and C. Qiao, "Traffic statistics and performance evaluation in optical burst switched networks", *IEEE/OSA Journal of Lightwave Technology*, vol.22, no.12, pp.2722-2738, Feb 2002.

[YuTo01] T. S. Peter Yum, F. Tong and K. T. Tan, "An Architecture for IP Over WDM Using Time-Division Switching", *IEEE/OSA Journal of Lightwave Technology*, vol.19, no.5, pp.589-595, May 2001.

[ZaJu00] H. Zang, J. P. Jue and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks", *Optical Networks Magazine*, pp.47-60, Jan 2000.

[ZaVu04] A. Zalesky, Hai Le Vu, Z. Rosberg, E.W.M. Wong and M. Zukerman, "Modeling and performance evaluation of optical burst switched networks with deflection routing and wavelength reservation", *Proc. IEEE INFOCOM 2004*, Mar 7-11, 2004, Hongkong, China, pp.1864-1871.

[ZhBa04] B. Zhou, M. A. Bassioun and G. Li, "Improving fairness in optical-burst-switching networks", *Journal of Optical Networking*, vol.3, no.4, pp.214-228, Apr 2004.

[ZhMo05] J. Zheng and H. T. Mouftah, "Media access control for Ethernet passive optical networks: an overview", *IEEE Communications Magazine*, vol.43, no.2, pp.145-150, Feb 2005.

[ZhVo04] Q. Zhang, V. M. Vokkarane, J. P. Jue and B. Chen, "Absolute QoS Differentiation in Optical Burst-Switched Networks", *IEEE Journal on Selected Areas in Communications*, vol.22, no.9, pp.1781-1795, Nov 2004.

[ZhVo05a] Q. Zhang, V. M. Vokkarane, Y. Wang and J. P. Jue, "Evaluation of Burst Retransmission in Optical Burst-Switched Networks", *Proc. Broadnets 2005*, Oct 3-7, 2005, Boston, MA, U.S.A., pp.297-303.

[ZhVo05b] Q. Zhang, V. M. Vokkarane, Y. Wang and J. P. Jue, "Analysis of TCP over optical burst-switched networks with burst retransmission", *Proc. IEEE GLOBECOM 2005*, Nov 28- Dec 2, 2005, St. Louis, MO, U.S.A., pp.1978-1982.

APPENDIX A: AN OPNET EXAMPLE

OPNET Modeler is adopted for the simulations in this thesis. A series of network model, node models, and process models are developed in a hierarchical manner. Here we show an OPNET example being used for the simulations in Chapter 5.

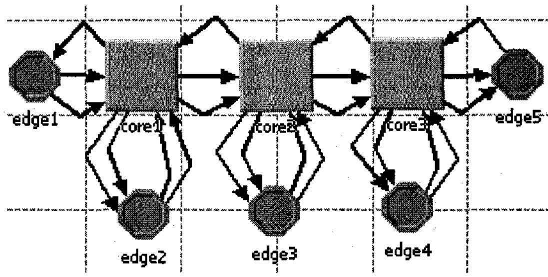
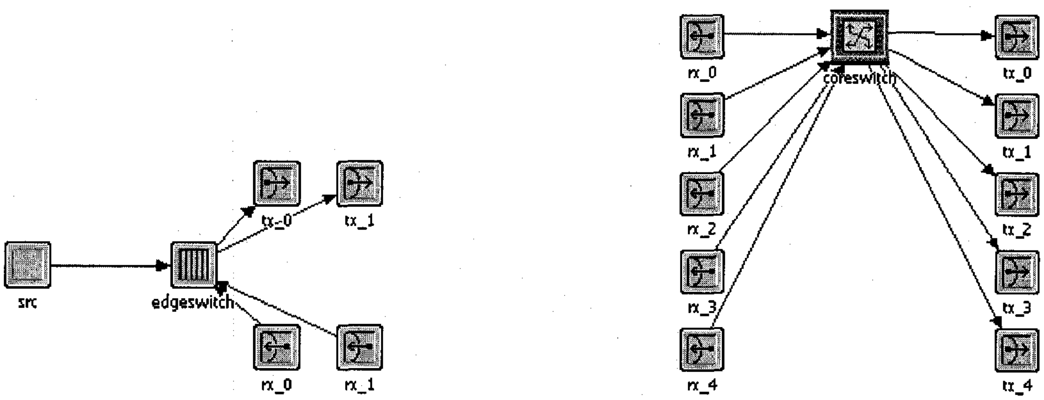


Fig. A.1. An OBS network model

Fig. A.1 shows the OPNET network model for Fig. 5.5. The square blocks represent the core nodes and the octagon blocks represent the edge nodes. These nodes are interconnected by links which include two sets, one for the DBs and another one for the BHPs. The OPNET implementation of node models is as follows.

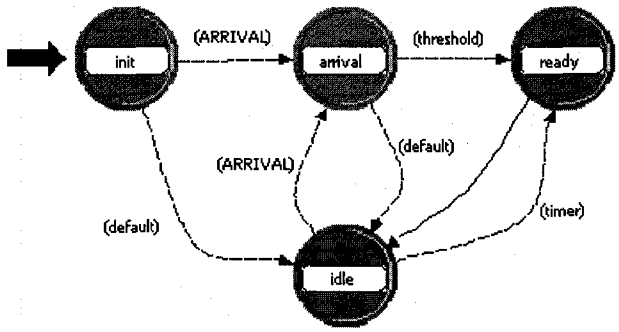


(a) An edge node model

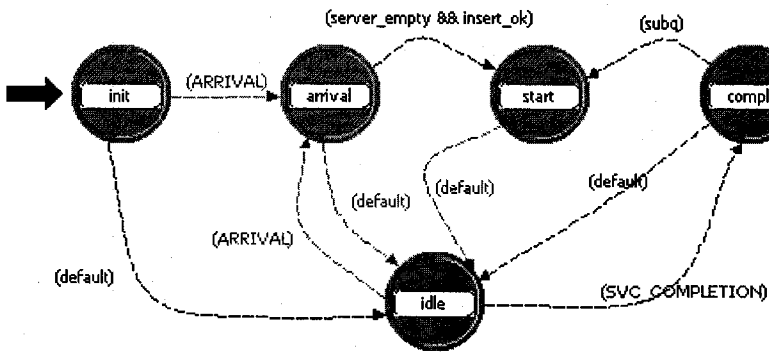
(b) A core node model

Fig. A.2. OBS node models

Fig. A.2 depicts the OPNET node models for the edge and core nodes in Fig. A.1. Fig. A.2a is the edge node model in which the local packets are generated by the "src" process and the "edgeswitch" process implements burst assembly, burst scheduling, burst transmission and burst receiving functions. Fig. A.2b illustrates the core node model which transmits and receives messages to and from the links. The "coreswitch" process implements the scheduling algorithm and the signaling protocol.



(a) An edge process model



(b) A core process model

Fig. A.3. OBS process models

Fig. A.3a is the OPNET process model for the "edgeswitch" in Fig. A.2a. The "init" state machine initializes necessary parameters and settings. If there is no packet arrival or operations, the system stays in the "idle" state. Once a packet arrives, the system enters the "arrival" state to analyze the packet information and implement the burst assembly algorithm. A burst becomes "ready" if (1) the waiting time at the idle state has reached the predefined timer, or (2) the predefined threshold has reached after a packet's arrival. At the "ready" state, the system chooses an appropriate wavelength channel, sets up the offset time, generates a control packet, and sends out the message. Fig. A.3b is the OPNET process model for the "coreswitch" in Fig. A.2b. A newly-arrived burst is processed from the "start" state. After finishing the processing in the "compl" state, it starts the next burst. Otherwise, the system waits in the "idle" state.

APPENDIX B: BURST-BASED ANALYSIS

Here we review some existing works using the burst-based model for OBS analysis. They are used for the comparison for the models we proposed in Section 3.2.

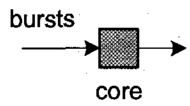


Fig. B.1. A burst-based model

A commonly-adopted burst-based model is given in Fig. B.1, which is used to model the first core node along the routing path in Fig. 3.1. It does not take the burst assembly process into account. Instead, it assumes the offered traffic to be in optical bursts already. Let W_c be the number of output wavelength channels at the core node, and ρ be the total offered load to the system. The loss rate in such a system using the JET protocol is usually approximated as an *M/M/k/k* queue [Klei75]:

$$p_{MMkk} = \frac{\rho^{W_c} / W_c!}{\sum_{n=0}^{W_c} \rho^n / n!} \quad . \tag{B.1}$$

When the burst-based system adopts the JETseg protocol, it is called an OCBS (Optical Composite Burst Switching) network [DeEr02]. An analytical model has been proposed for the estimation of the loss performance [NeRo02]:

$$p_{OCBS} = \sum_{i=1}^{\infty} \frac{i}{W_c + i} \cdot \frac{\rho^{W_c + i} \cdot e^{-\rho}}{(W_c + i)!} \quad . \tag{B.2}$$

Both burst-based equations above are based on the following assumptions: the number of input channels is infinite; the burst arrival is a Poisson process; and all bursts passing through a core node have a same loss rate. However, these equations neither capture the effects of the burst-assembly process on the traffic characteristics nor reflect the real connection pattern in a network. For example, some properties of the assembled bursts, e.g., self-similarity, are still controversial [GeCa00, HuDo03]. Therefore, the above methods often cause a significant difference from the real performance of an OBS network as demonstrated in Section 3.2.4.

APPENDIX C: DELAY ANALYSIS

In Section 4.2.1, we have given an approximated relationship of the delay and the assembly timer. Because we have not specified the packet arrival process and packet length distribution, we have resorted to simulations to verify it. Here we investigate this relationship further under Poisson packet arrival. To find out the delay performance under the controlling operation, we shall study the assembly procedure in one buffer which stores bursts of *Flow s-d*. If the packet arrival is a Poisson process, given a burst interval *t*, the *assembly delay* of packets, D_a, has a *probability* of $\Pr[D_a \le x \mid I_{s,d} = t] = \begin{cases} 1, x \ge t \\ \frac{x}{t}, x < t \end{cases}$.

The *cdf* of the assembly delay is $F_{D_a}(x) = \Pr[D_a \le x] = \int_{t=0}^{\infty} \Pr[D_a \le x \mid I_{s,d} = t]] \cdot f_{I_{s,d}}(t) \cdot dt$ $= \int_{t=0}^{x} \lambda_{s,d} e^{-\lambda_{s,d} t} \cdot dt + \int_{t=x}^{\infty} \frac{x}{t} \cdot \lambda_{s,d} e^{-\lambda_{s,d} t} \cdot dt$. The average assembly delay becomes

$$\overline{D_a} = E[D_a] = \int_{x=0}^{\infty} x \cdot f_{D_a}(x) \cdot dx = \int_{x=0}^{\infty} (1 - F_{D_a}(x) \cdot dx = \frac{1}{\lambda_{s,d}} = T_{s,d} \tag{C.1}$$

The number of arrived packets during a period *t* is denoted by $A_{s,d}^P(t)$. The probability that *n* packets have arrived during the period $[0,t]$ is equal to $\Pr[A_{s,d}^P(t) = n] = \frac{(\lambda_{s,d}^P t)^n}{n!} e^{-\lambda_{s,d}^P t}$, $t \ge 0$, $n = 0,1,2,...,\infty$.

The number of packets in a burst, $P_{s,d}$, is equal to the total number of arrived packets during the burst interarrival period (using conditional probability), i.e., $\Pr[P_{s,d} = n] = \int_{t=0}^{\infty} \Pr[A_{s,d}^P(t) = n \mid I_{s,d} = t] \cdot f_{I_{s,d}}(t) \cdot dt$ $= \int_{t=0}^{\infty} \frac{(\lambda_{s,d}^P t)^n}{n!} e^{-\lambda_{s,d}^P t} \cdot \lambda_{s,d} e^{-\lambda_{s,d} t} \cdot dt$ $= \frac{\lambda_{s,d}^{P\ n} \cdot \lambda_{s,d}}{(\lambda_{s,d}^P + \lambda_{s,d})^{n+1}}$, $n = 0,1,2,...,\infty$. The *burst length* $L_{s,d}$ (measured in second) is the sum of many IP packets. Since the packet length has an *i.i.d.* distribution, $L_{s,d}$ can be approximated by a Gaussian distribution according to the Central Limited Theorem [Leon94]. For a burst containing *n* packets,

$\overline{L_{s,d}} = n \cdot \overline{L_{s,d}^P}$. We then have $\overline{L_{s,d}} = \sum_{n=0}^{\infty} \Pr[P_{s,d} = n] \cdot n \cdot \overline{L_{s,d}^P}$ [5]. The average burst length becomes

$$\overline{L_{s,d}} = \sum_{n=0}^{\infty} \Pr[P_{s,d} = n] \cdot n \cdot \overline{L_{s,d}^P} = \overline{L_{s,d}^P} \cdot \sum_{n=0}^{\infty} n \cdot \frac{{\lambda_{s,d}^P}^n \cdot \lambda_{s,d}}{(\lambda_{s,d}^P + \lambda_{s,d})^{n+1}} = \overline{L_{s,d}^P} \cdot \frac{\lambda_{s,d}^P}{\lambda_{s,d}} = \overline{L_{s,d}^P} \cdot \lambda_{s,d}^P \cdot T_{s,d} .$$

Let $L^{(N)}$ to represent the length of all bursts generated at this source with a mean value $\overline{L^{(N)}} = \overline{L_{s,d}}$. When we examine the generated bursts going to each output channel, it can be approximated by an *M/G/1* queueing model. In this *M/G/1* system, the burst arrival rate is $\lambda^{(N)} = \frac{N \cdot \lambda_{s,d}}{W}$ bursts/sec and the traffic load is $\rho^{(N)} = \lambda^{(N)} \cdot \overline{L^{(N)}} = \frac{N \cdot \lambda_{s,d} \cdot \overline{L_{s,d}}}{W} = \frac{N \cdot \lambda_{s,d}^P}{W} \cdot \overline{L_{s,d}^P}$. Each burst imposes two types of delays on the packets: the *queueing delay* D_q and the *transmission delay* D_t, which have been widely studied in many literatures (e.g., [Klei75]). The average values of these two delays, $\overline{D_q}$ and $\overline{D_t}$, are respectively

$$\overline{D_q} = \frac{\lambda^{(N)} \overline{(L^{(N)})^2}}{2(1-\rho^{(N)})} + T_{off} = \frac{N \cdot \lambda_{s,d} \cdot \overline{(L^{(N)})^2}}{2(W - N \cdot \lambda_{s,d}^P \cdot \overline{L_{s,d}^P})} + T_{off} \text{, and} \tag{C.2}$$

$$\overline{D_t} = \overline{L^{(N)}} = \overline{L_{s,d}} = \overline{L_{s,d}^P} \cdot \lambda_{s,d}^P \cdot T_{s,d} . \tag{C.3}$$

The total system delay, D_{sys}, has a mean

$$\overline{D_{sys}} = \overline{D_a} + \overline{D_q} + \overline{D_t} = \frac{N \cdot \lambda_{s,d} \cdot \overline{(L^{(N)})^2}}{2(W - N \cdot \lambda_{s,d}^P \cdot \overline{L_{s,d}^P})} + T_{off} + T_{s,d} + \overline{L_{s,d}^P} \cdot \lambda_{s,d}^P \cdot T_{s,d} , \tag{C.4}$$

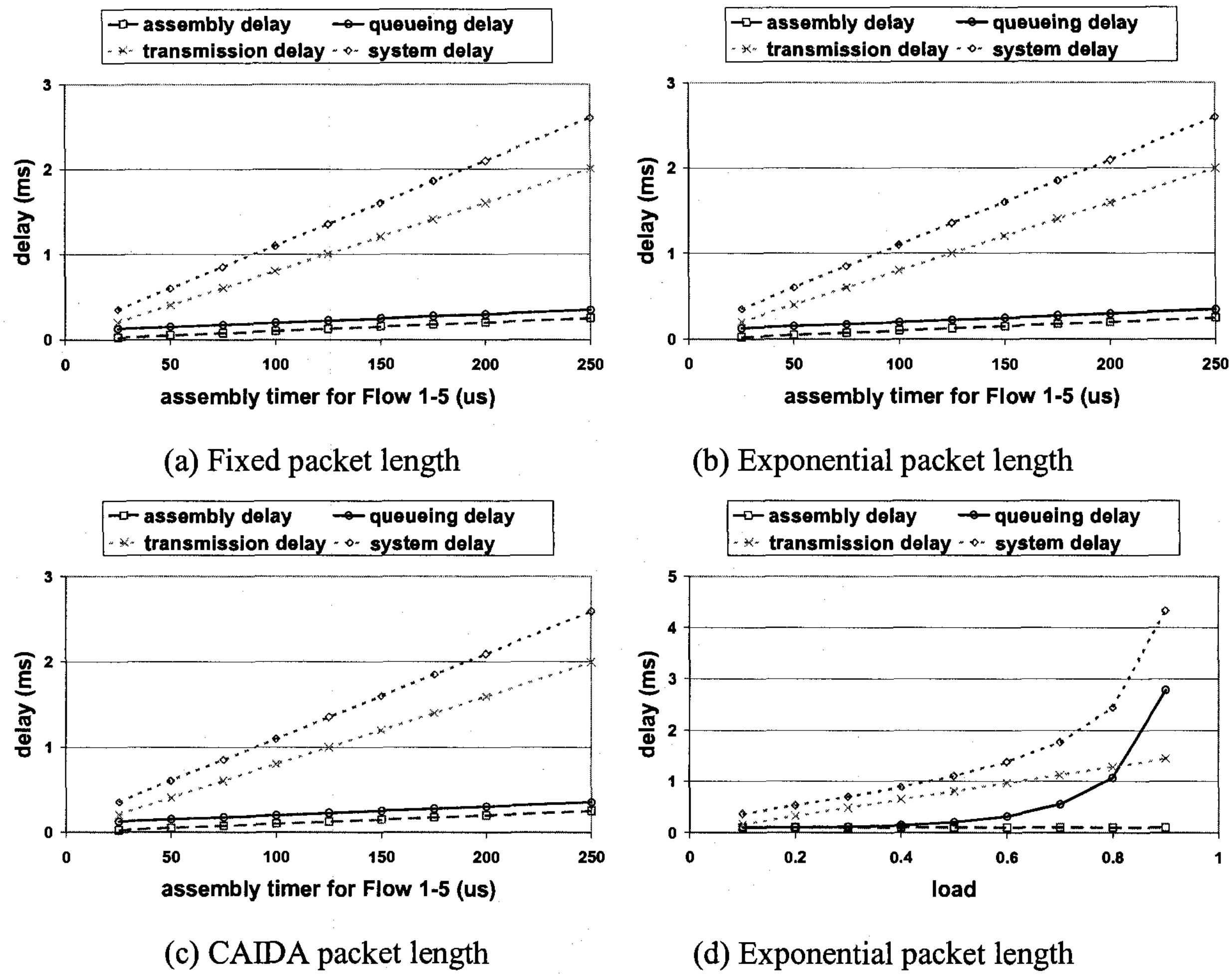
where T_{off} is the fixed offset time.

From the above equations, it can be seen that the assembly timer $T_{s,d}$ plays a critical role on the system delay at the edge node (it decides $\lambda_{s,d}$ and then affects $L_{s,d}$, $\lambda^{(N)}$, $L^{(N)}$ and $\rho^{(N)}$, which determines the delay in (C.1) to (C.4)). We can see the average system delay has an almost linear relationship with the assembly timer. Other issues affecting the delay include the traffic characteristics of incoming IP packets and the propagation delay. We can only adjust the assembly timer to meet the delay requirement since we cannot modify the properties of incoming IP traffic or the propagation delay from the source to the destination. This observation substantiates the principle of our flow controller again.

[5] In a strict sense, the central limit theory is only valid for a large n. But our simulation results show that we can use it to approximate the burst length closely when it is used here.

APPENDIX D: DELAY PERFORMANCE

In Section 4.4.1, we have examined the delay performance using the CAIDA packet length distribution and Poisson packet arrival. Here more simulation results under different traffic models are shown.



(a) Fixed packet length

(b) Exponential packet length

(c) CAIDA packet length

(d) Exponential packet length

Fig. D.1. Delay performance: Poisson packet arrival

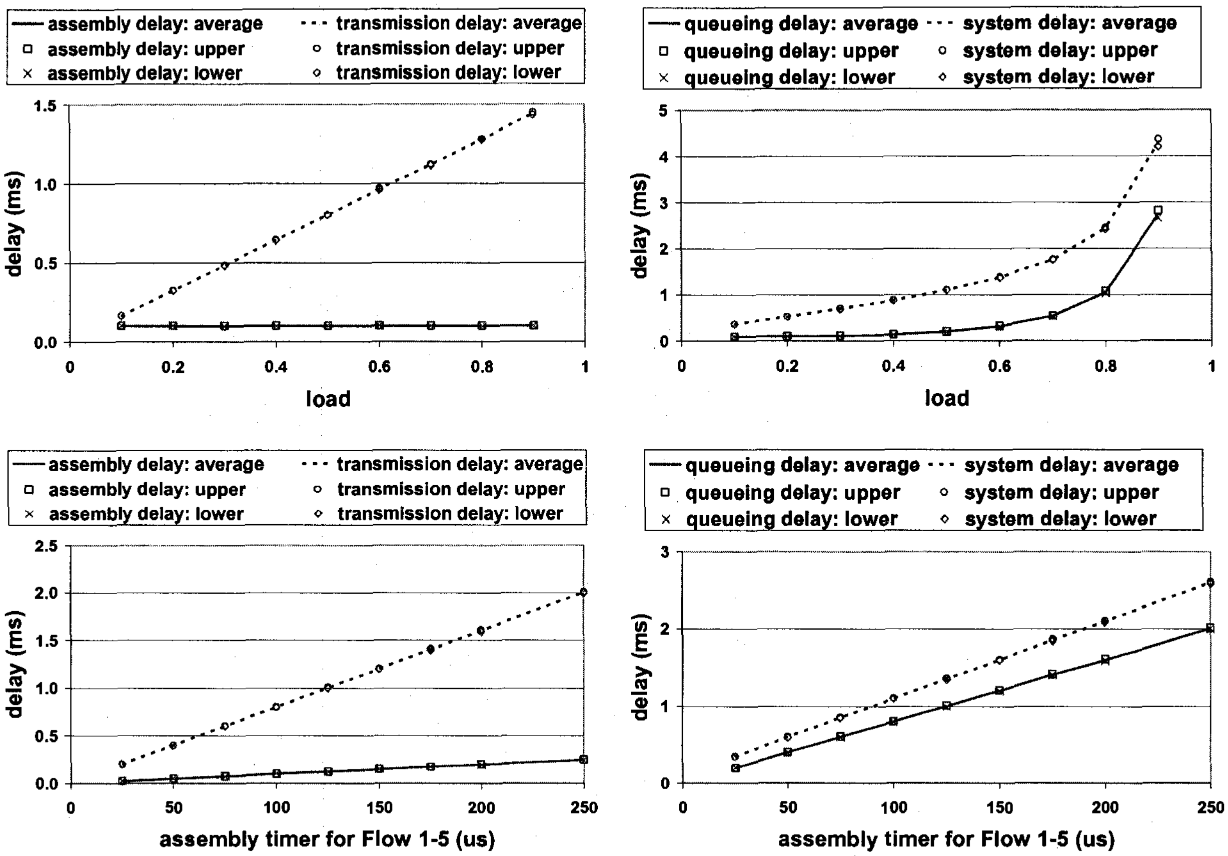
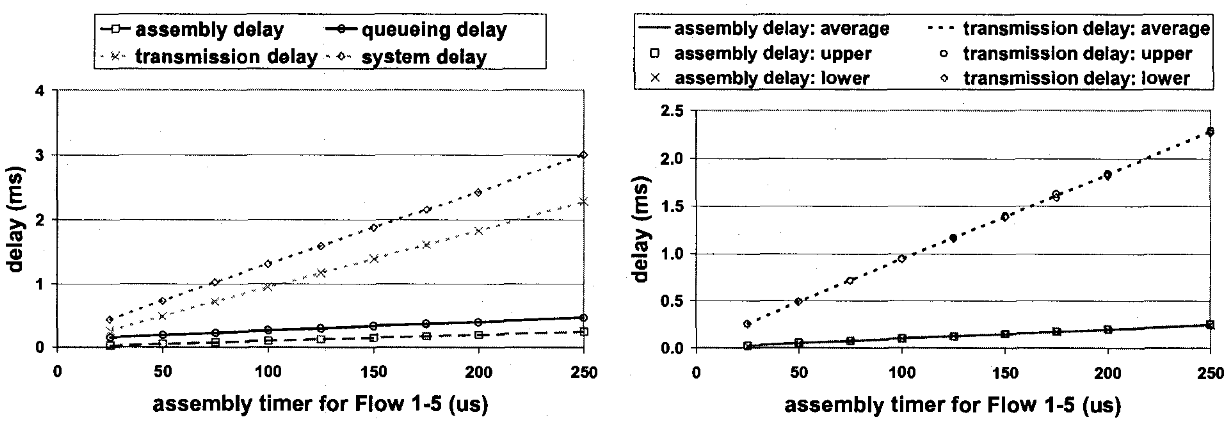


Fig. D.2. 95% confidence interval of delay: Poisson packet arrival and fixed packet length



(a) Average delay (b) Delay performance: 95% confidence interval

Fig. D.3. Delay performance: Pareto packet arrival and CAIDA packet length

From Figs. D.1-D.3, we can see that the delay is always a linear function of the assembly timer under different packet models, which agrees with our approximation in Section 4.2.1.

APPENDIX E: LOSS RATE AND THROUGHPUT

In Section 4.4.1, we have examined the loss and throughput performance using the CAIDA packet length distribution and Poisson packet arrival. Here more simulation results under different traffic models are shown.

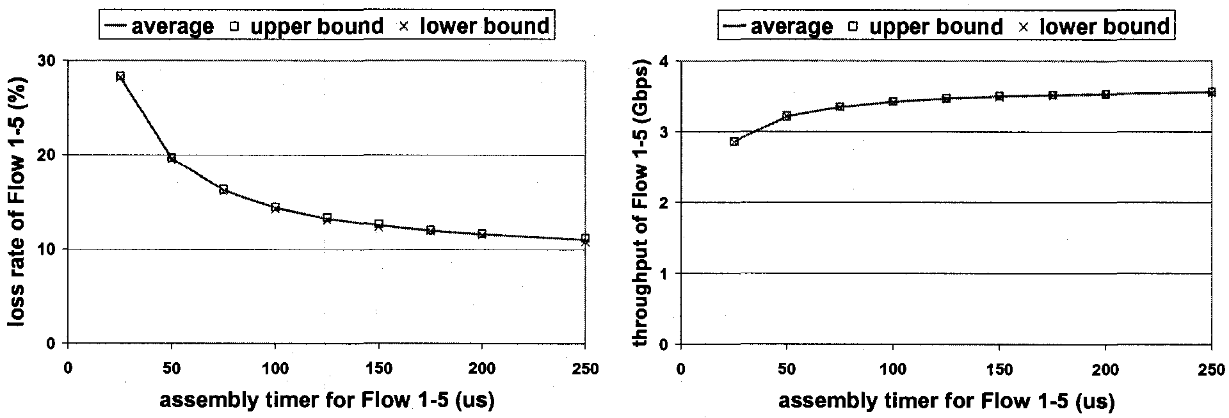


Fig. E.1. Poisson packet arrival and fixed packet length using JIT

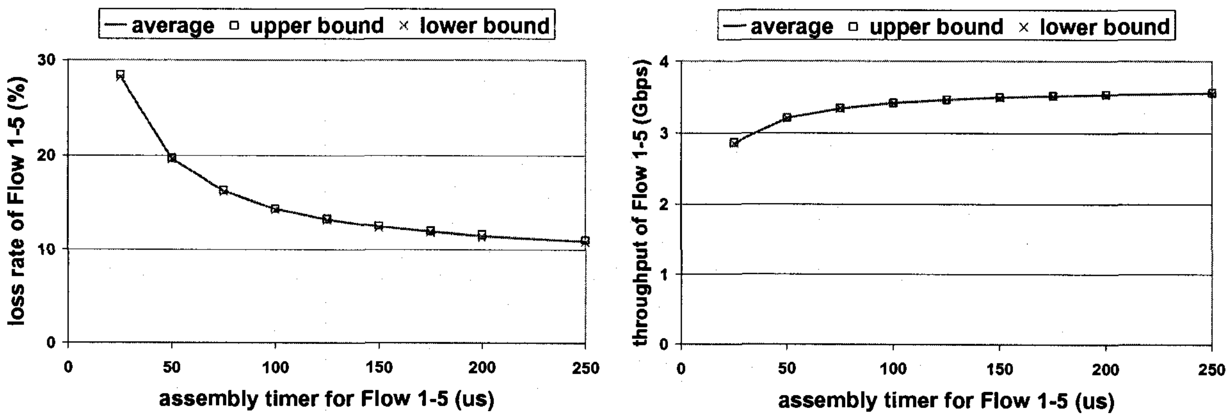


Fig. E.2. Poisson packet arrival and exponential packet length using JIT

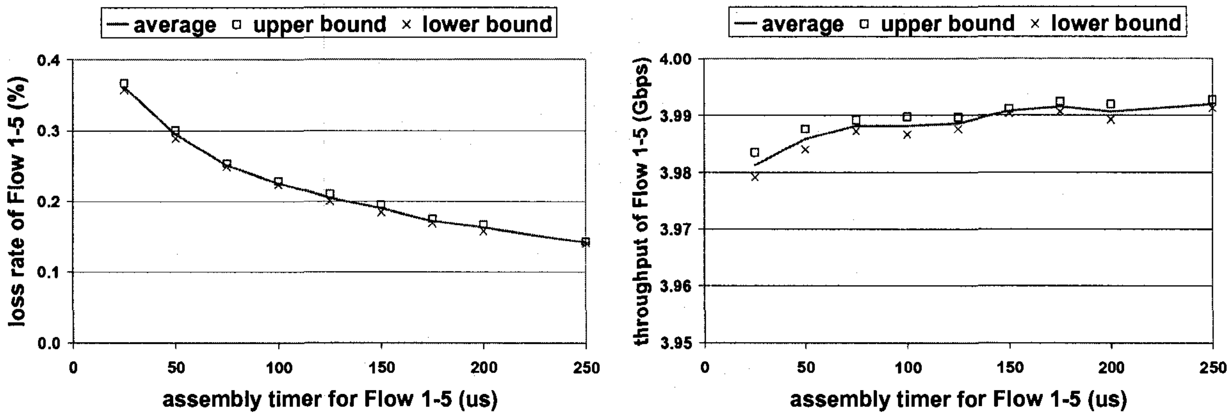


Fig. E.3. Poisson packet arrival and exponential packet length using JETtailSeg

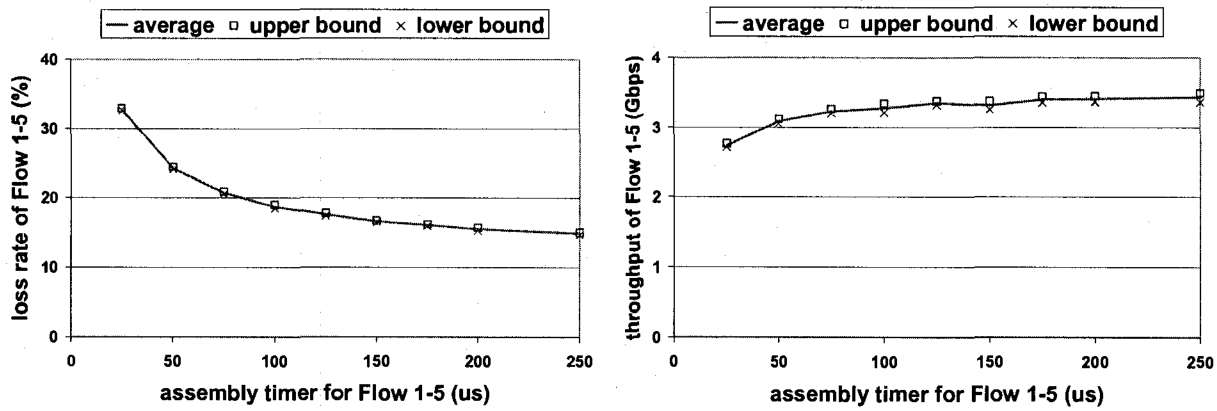
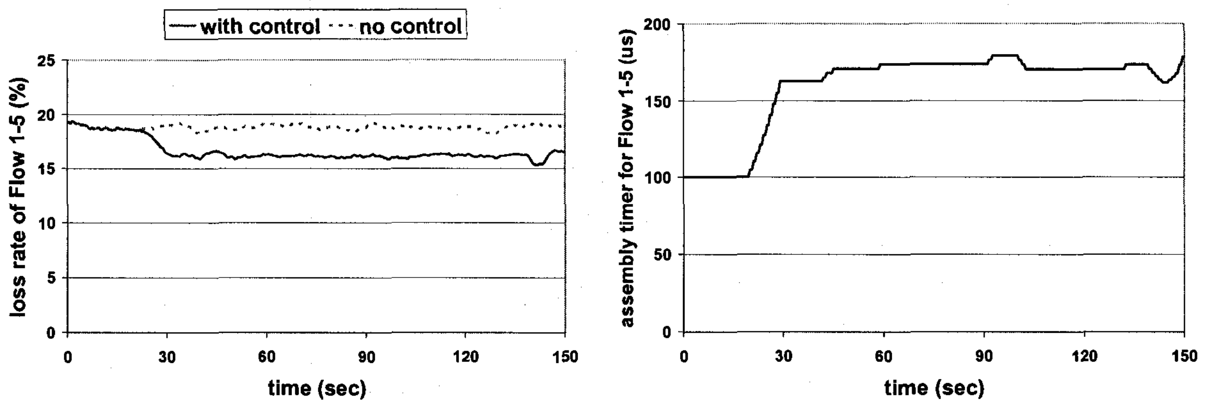


Fig. E.4. Pareto packet arrival and fixed packet length using JIT

From Figs. E.1 to E.4, we can see that the loss rate of *Flow* 1-5 is always a decreasing function of its assembly timer under different packet models. At the same time, the throughput of *Flow* 1-5 will increase as the assembly timer increases. All observations agree with our analysis in Section 4.2.2.

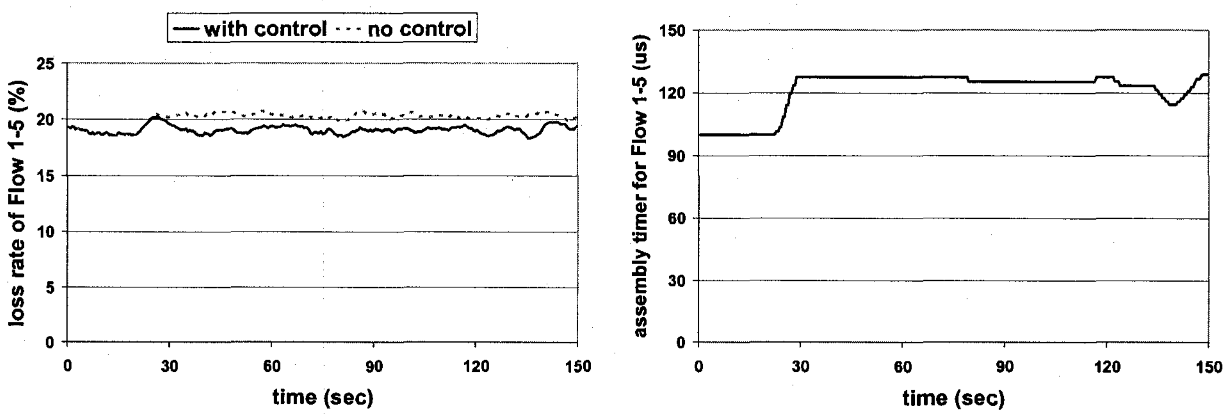
APPENDIX F: PERFORMANCE OF THE SIMPLE CONTROLLER: PARETO TRAFFIC

In Sections 4.4.2 and 4.4.3, we have examined the flow controller proposed in Section 4.3 using Poisson packet arrival. Here we evaluate the controller under Pareto packet arrival.



(a) Loss rate (JIT) (b) Assembly timer for *Flow* 1-5 (JIT)

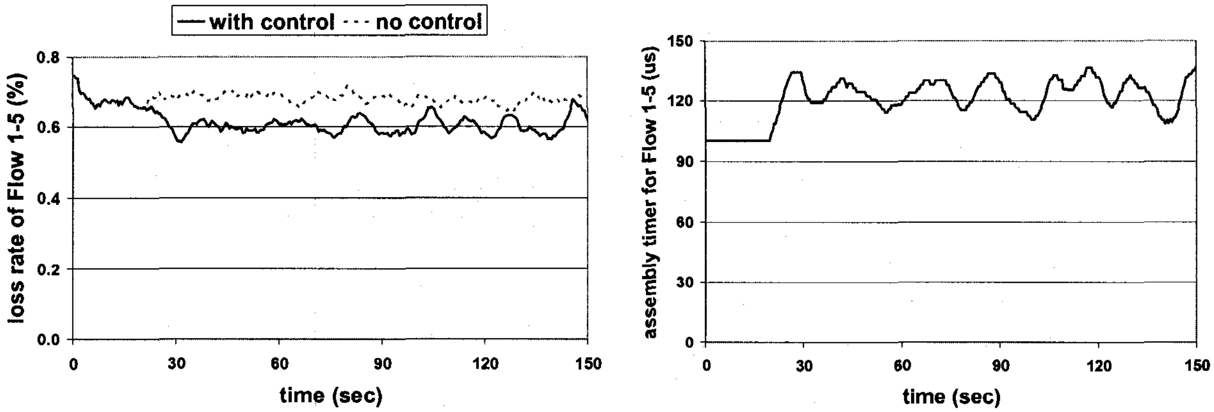
Fig. F.1. Reducing loss rate: Pareto packet arrival and CAIDA packet length



(a) Loss rate (JIT) (b) Assembly timer for *Flow* 1-5 (JIT)

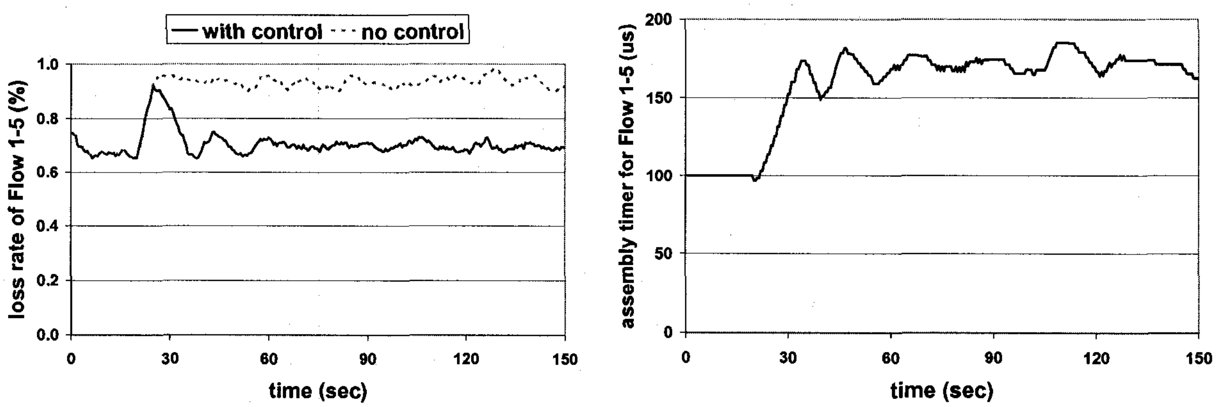
Fig. F.2. Avoiding loss increasing: Pareto packet arrival and CAIDA packet length.

Fig. F.1 demonstrates how our control scheme in Section 4.3 successfully reduces the loss rate of *Flow* 1-5 from 19% to 16% by adjusting the assembly timer from 100us to 170us. Fig. F.2 shows how our control scheme in Section 4.3 decreases the loss rate of *Flow* 1-5 from 20.5% to 18% in a dynamic network by adjusting the assembly timer from 100us to 130us. These two cases adopt the JIT protocol.



(a) Loss rate (JETtailSeg) (b) Assembly timer for *Flow* 1-5 (JETtailSeg)

Fig. F.3. Reducing loss rate: Pareto packet arrival and CAIDA packet length



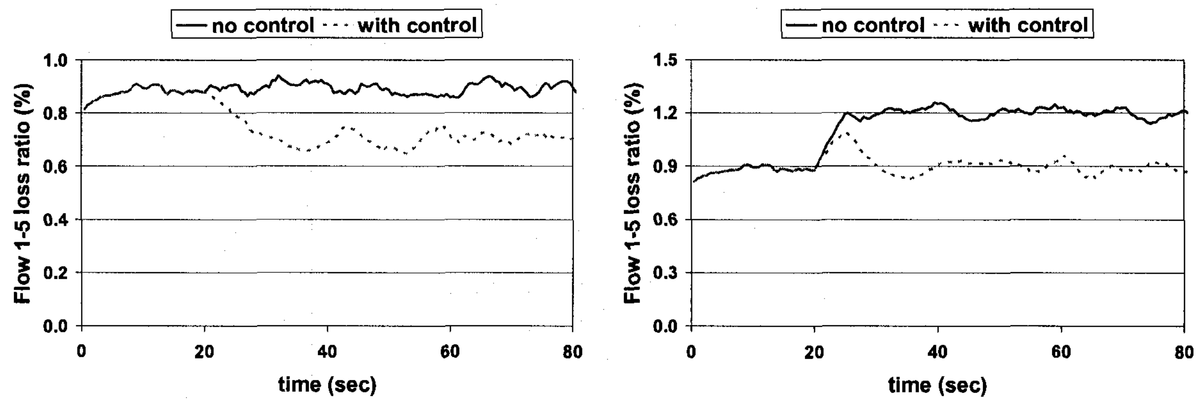
(a) Loss rate (JETtailSeg) (b) Assembly timer for *Flow* 1-5 (JETtailSeg)

Fig. F.4. Avoiding loss increasing: Pareto packet arrival and CAIDA packet length

In Fig. F.3, the loss rate of *Flow* 1-5 is reduced from 0.68% to 0.6%, the assembly timer from 100us to 125us in a static network. In Fig. F.4, the loss rate decreases from 0.95% to 0.68% in a dynamic network, while the assembly timer changes from 100us to 170us. These two cases adopt the JETtailSeg protocol.

APPENDIX G: PERFORMANCE OF THE SIMPLE CONTROLLER: EFFECT OF THE PROCESSING TIME

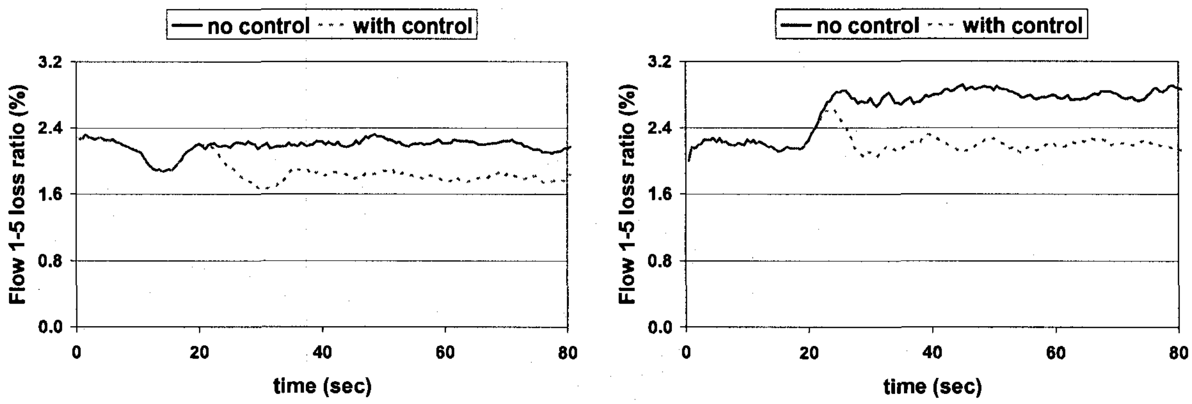
In Sections 4.4.2 and 4.4.3, we have examined the flow controller proposed in Section 4.3 by assuming negligible switching time and header processing time. Here we evaluate the controller when these two values are non-zero.



(a) In a static network (b) In a dynamic network

Fig. G.1. Controller performance: Poisson packet arrival

Fig. G.1 depicts loss performance using the controller in Section 4.3 when the switching time of the core switch is 1us and the header processing time is 100ns per control packet. Other parameters are the same as in Section 4.4. The packet arrival is Poisson and the signaling protocol is JETtailSeg. It is shown that in both a static network and a dynamic network, the controller can successfully steer the loss rate of *Flow* 1-5 to a desired value. The controller's behavior is the same as the case in Section 4.4 when the switching time and the header processing time are neglected. This observation reveals that our controller is a realistic design.



(a) In a static network (b) In a dynamic network

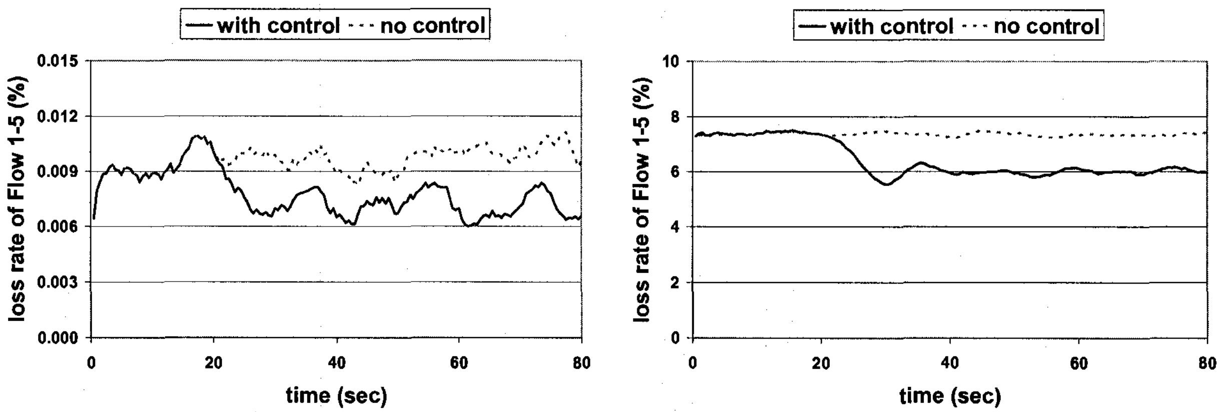
Fig. G.2. Controller performance: Pareto packet arrival

Fig. G.2 depicts loss performance using the controller in Section 4.3 when the switching time of the core switch is 1us and the header processing time is 100ns per control packet. Other parameters are the same as in Section 4.4. The packet arrival is Pareto and the signaling protocol is JETtailSeg. It is shown that in both a static network and a dynamic network, the controller can successfully steer the loss rate of *Flow* 1-5 to a desired value. The controller's behavior is the same as the case in Section 4.4 when the switching time and the header processing time are neglected. This observation reveals that our controller is a realistic design.

We have examined the controller performance under more scenarios when the switching time and the header process time are not negligible. All results are similar as here, so they are not shown.

APPENDIX H: PERFORMANCE OF THE SIMPLE CONTROLLER: EFFECT OF THE INTERVAL DISTRIBUTIONS

In Chapter 4, we have used the exponentially-distributed intervals for the flow control. Here we evaluate the controller when the intervals follow other distributions.



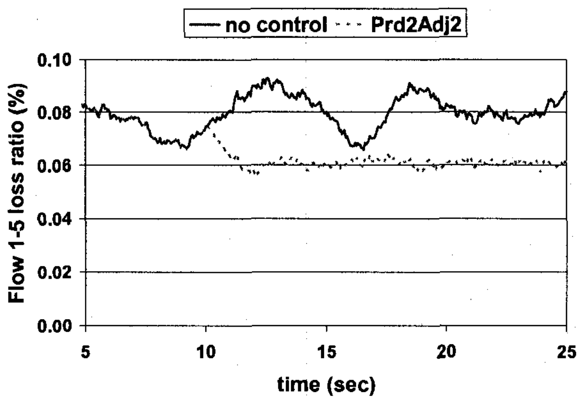
(a) Using uniform distribution (b) Using lognormal distribution

Fig. I.1. Controller performance

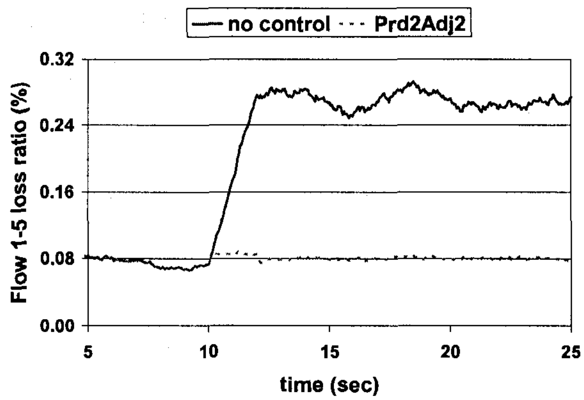
Fig. H.1 depicts loss performance of the simple controller in Chapter 4, whereas the intervals for burst assembly follow either a uniform distribution or a lognormal distribution. All other parameters are the same as in Section 4.4. It can be seen our controller is still able to reduce the loss rate of *Flow* 1-5 to a target level. Please note that we just want to demonstrate the effectiveness of the controller, while it is possible to achieve better convergence or stability than Fig. H.1 through carefully choosing the parameters.

APPENDIX I: PERFORMANCE OF THE GENERIC CONTROLLER: EFFECT OF THE PROCESSING TIME

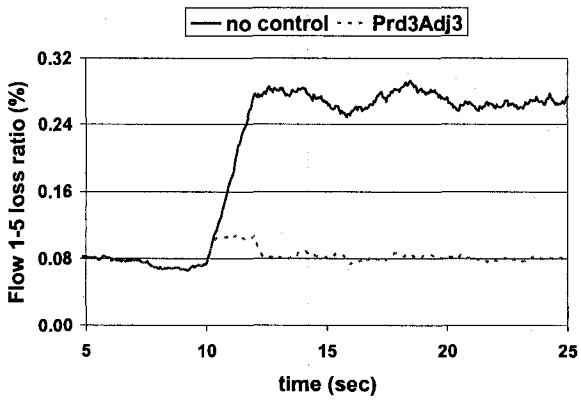
In Section 5.4, we have examined the flow controller proposed in Section 5.3 assuming negligible switching time and header processing time. Here we evaluate the controller when these two values are non-zero.



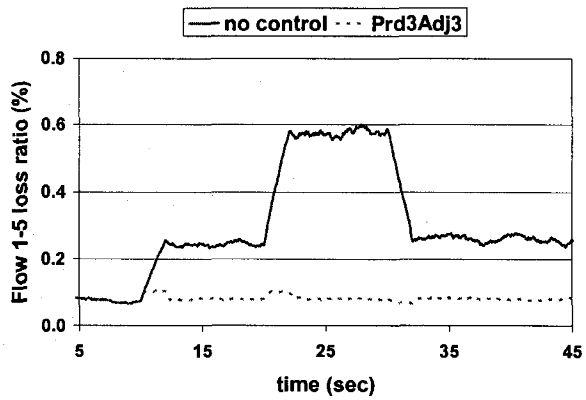
(a) Static network



(b) Load change



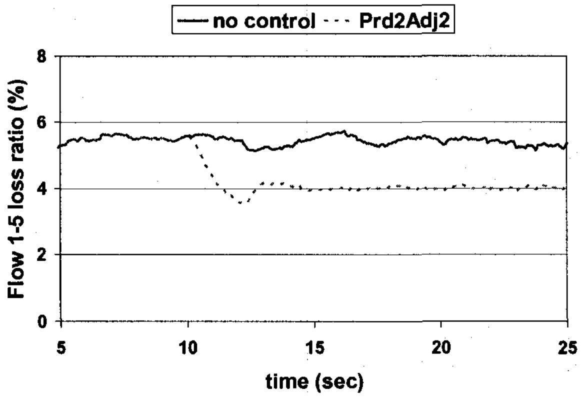
(c) Channel broken



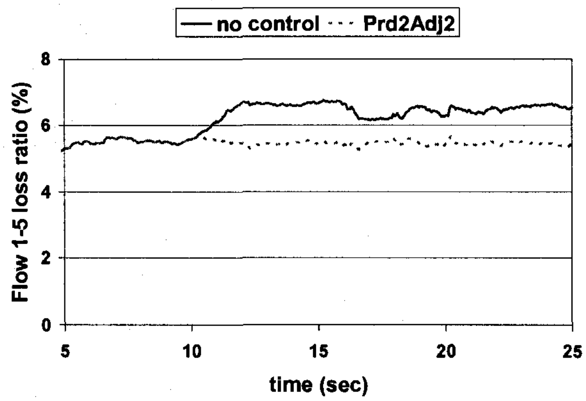
(d) Load change + channel broken

Fig. I.1. Controller performance in a P1-A2 network: Poisson packet arrival

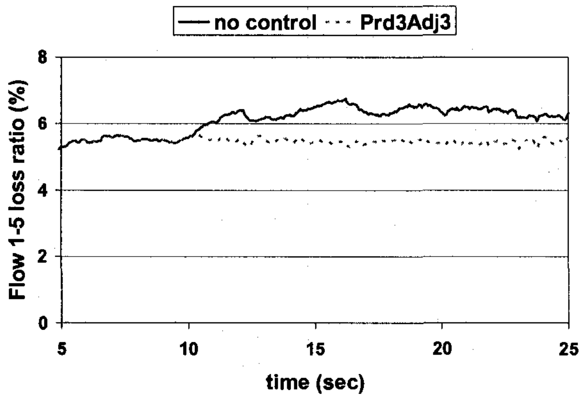
Fig. I.1 depicts loss performance using the controllers in Section 5.3 when the switching time of the core switch is 1us and the header processing time is 100ns per control packet. Fig. I.1a is a static network. Fig. I.1b is a dynamic network in which the load of *Flow* 1-5 is increased from 0.7 to 0.72 at t=10 sec. Fig. I.1c is a dynamic network in which one channel between C1 and C2 becomes broken at t=10 sec. Fig. I.1d is a dynamic network in which the load of *Flow* 1-5 is increased from 0.7 to 0.72 at t=10 sec, dropped back to 0.7 at t=30 sec, and one channel between C1 and C2 becomes broken at t=20 sec. The packet arrival is Poisson.



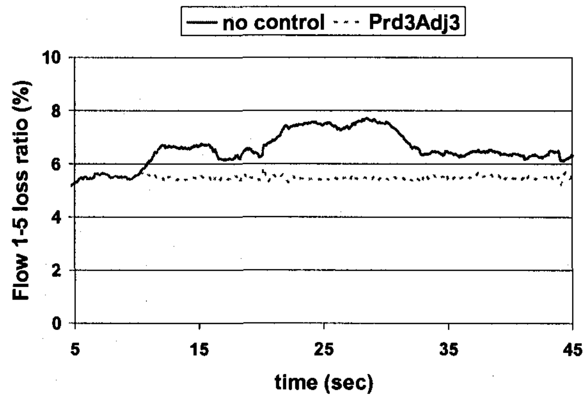
(a) Static network



(b) Load change



(c) Channel broken



(d) Load change + channel broken

Fig. I.2. Controller performance in a P1-A2 network: Pareto packet arrival

Fig. I.2 depicts loss performance using the controllers in Section 5.3 when the switching time of the core switch is 1us and the header processing time is 100ns per control packet. Fig. I.2a is a static network. Fig. I.2b is a dynamic network in which the load of *Flow* 1-5 is increased from 0.7 to 0.72 at t=10 sec. Fig. I.2c is a dynamic network in which one channel between C1 and C2 becomes broken at t=10 sec. Fig. I.2d is a dynamic network in which the load of *Flow* 1-5 is increased from 0.7 to 0.72 at t=10 sec, dropped back to 0.7 at t=30 sec, and one channel between C1 and C2 becomes broken at t=20 sec. The packet arrival is Pareto.

From Figs. I.1-2, it is shown that in both a static network and a dynamic network, the generic controller of our design can successfully steer the loss rate of *Flow* 1-5 to a desired range. The controller's behavior is the same as the case in Section 5.4 when the switching time and the header processing time are neglected. This observation confirms that our controller is a realistic design. More simulation results on other scenarios are omitted for clarity reason.