uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Akbar Ghaffar Pour Rahbar
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Bandwidth Management in Slotted All-Optical Packet-Switched Networks under the DiffServ Domain

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

Abdulmotaleb El Saddik (absent)

Tet Yeap

Nasir Ghani (teleconference)

Jérôme Talim

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

Bandwidth Management in Slotted All-Optical Packet-Switched Networks under the DiffServ Domain

by

Akbar Ghaffar Pour Rahbar

A dissertation submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirement for the degree of

Ph.D. in Computer Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Ontario, Canada

November 2006

Your file Votre référence
ISBN: 978-0-494-25872-9
Our file Notre référence
ISBN: 978-0-494-25872-9

Abstract

Due to its finer granularity, optical packet switching can efficiently use the bandwidth provided by all-optical networking. This research develops a new framework to manage the bandwidth in buffer-less slotted all-optical packet-switched networks suitable for the next generation IP networks where the quality of service must be addressed.

We first present an ingress switch architecture along with a new control and signaling structure for the slotted-OPS network. Then we design different components of our ingress switch architecture including packet scheduling, retransmission management, and bandwidth access units.

A class-based packet scheduling is designed to reduce the inter-transmission time from different source routers, and to provide packet differentiation so that fairness issues for DiffServ support are addressed and resolved.

The contention problem at the core switch is studied. Inexpensive contention avoidance and resolution schemes in the optical domain are considered. For contention avoidance, we have used the software approach from the ingress switch, while for contention resolution we have analyzed the prioritized retransmission technique to limit the number of retransmissions, and to improve network throughput. We have also designed new algorithms in core switches to resolve contention for class-based traffic.

A new contention-based DTDM (Distributed TDM) bandwidth access technique is designed in which the ingress switches can evenly distribute traffic among available wavelengths and fibers. It can shape the traffic which plays an important role in reducing loss rate when accessing a slotted all-optical OPS network. The DTDM technique is further improved in an integrated scheme in a slotted all-optical single-hop OPS network that can benefit from the positive aspects of the centralized reservation-based scheme during high traffic load. The AAPN (Agile All-Photonic Network) network using the integrated technique is illustrated as an example.

In summary, all the above methods are utilized to build our OPS network architecture. We have studied both design and analysis issues in a cost-effective OPS network including packet scheduling, contention avoidance, retransmission, optical network access, and packet assembling, all in the optical domain.

Acknowledgements

I am deeply grateful to my supervisor Prof. Oliver W. W. Yang for his consistent knowledgeable guidance, persistent but helpful comments, and suggestions on the improvement of this work. My achievements are deeply rooted on these comments and feedbacks.

I would also like to thank the Ministry of Science, Research and Technology of Iran; Sahand University of Technology; and the Canadian Natural Sciences and Engineering Research Council (NSERC) and industrial and government partners, through the Agile All-Photonic Networks (AAPN) Research Network for partially supporting my study. I also appreciate the CCNR lab students for their help.

Finally, I am thankful to my wife for her patience and support. I give my great thanks to my family for their understanding and sacrifices. They are just wonderful. Without their endless support both physically and spiritually, I never could have finished my study.

In the Name of Allah,

the Most Gracious,

the Most Merciful

To my wife, my parents and my daughter Hanieh.

Table of Contents

List of Figures

List of Tables

Table of Acronyms and Abbreviations

		Subsection of 1st Appearance
AAPN	Agile All-Photonics Network	1.2
AAR	Average Aggregate Rate	3.2
ACK	Acknowledge	2.6
AF	Assured Forwarding	1
Arch1	ingress switch Architecture 1	1.1.3
Arch2	ingress switch Architecture 2	2.4
BE	Best Effort	1
BMPS	Buffer Management and Packet Scheduler	2.4
BvN	Birkhoff and von Neumann	1.1.2.1
CSA	Composite Slot Assembly	2.3
CST	Coordinated Slot Transmission	4.3.3
CTT	Combined Threshold-based and Timer-based	1.1.3
DiffServ	Differentiated Services	1
DMUX	De-Multiplexer	4.5
DRR	Deficit Round Robin	1.1.4
CTDM	Centralized TDM	1.5
DTDM	Distributed TDM	1.5
ITDM	Integrated TDM	1.5
DWDM	Dense Wavelength Division Multiplexing	1
EF	Expedited Forwarding	1
EPON	Ethernet Passive Optical Network	1.1.1.2
FCFS	First Come First Serve	1.1.2.3
ID	IDentification	2.5
IETF	Internet Engineering Task Force	1
ILP	Integer Linear Program	4.5.1
IP	Internet Protocol	1
M-Z	Mach-Zehnder	4.5.1
MRR	Multiple Round Robin	3.2.1
MUX	Multiplexer	4.5
NACK	Negative Acknowledge	2.6
OBM	Optical Bandwidth Manager	1.1.3
OBS	Optical Burst Switching	1.1.1.1
OCGRR	Output Controlled Grant based Round Robin	1.5
OCS	Optical Circuit Switching	1.1.1.1
OFS	Optical Flow Switching	1.1.1.1
OLT	Optical Line Terminal	1.1.1.2
ONU	Optical Network Unit	1.1.1.2
OPS	Optical Packet Switching	1.1.1.1
OXC	Optical Cross Connect	1.1.2.2
p.d.f	Probability Density Function	3.4
PQ	Priority Queuing	1.1.4

PQWRR	Priority Queuing Weighted Round Robin	1.1.4
PR	Prioritized Retransmission	1.4
PSR	Photonic Slot Routing	1.1.1.2
QoS	Quality of Service	1
RR	Random Retransmission	1.1.2.2
SSH	Slot-Set Header	2.5
TDM	Time Division Multiplexing	1.1.2.1
UST	Uncoordinated Slot Transmission	4.3.3
WAN	Wide Area Network	1.1.2.1
WC	Wavelength Converter	1.1.2.2
WRR	Weighted Round Robin	1.1.4

Table of Notations and Symbols

		Subsection of 1st Appearance
B_C	Wavelength channel bandwidth in bps	2.1
B_a	Available bandwidth in each ingress switch in bps	2.2
C	Bandwidth for the OCGRR scheduler in bps	3.2.2
C_{MZ}	Cost of a Mach-Zehnder filter	4.5
C_N	Network segment cost	4.5
C_{OXC}	Optical switch cost	4.5
C_{SE}	A 2x2 switching element cost	4.5
$C_{U,f}$	Average channel utilization of time-slots in the network within a frame	7.4
$C_{U,T}$	Threshold value for the maximum channel utilization of time-slots set by the network manager	7.4
$C_f(f,W)$	Fiber cost for a link with f fibers and W wavelengths inside each fiber	4.5
C_i	Index of class i streams, $i=1(EF),2(AF1),...,\ \psi\ (BE)$	3.1
C_{lp}	Link provisioning cost	4.5
C_p	Coherence parameter	3.2.1
C_λ	Wavelength cost	4.5
D_P	One-way propagation delay from an edge switch to the optical switch as an integer number of slots	5.1.4
$\overline{D_T}$	Average transmission delay	5.1.4
$\boldsymbol{D} = [d_{ij}]_{nxn}$	Traffic demand matrix from n ingress switches to n egress switches	7.1
E	Empty slots in an *nf-slot-set*	4.1.1
$\overline{E}$	Average number of empty slots in an *nf-slot-set*	4.2.1.3
F	Frame length in slots	2.4
$G_{i,j}$	Available grant for stream j in class i , i= *EF,AF1,AF2, BE.*	3.2
H	Maximum number of retransmissions needed to transmit a slot successfully	5.1.1
K	The grant intensity allocation parameter used in OCGRR scheduling	3.2.2
L	Normalized traffic load on wavelength channels	4.2.1.3
$\overline{L_d}$	The average number of delivered slots to a tagged output link	4.1.1
L_k	The number of slots arrived to a tagged output link at retransmission level k and higher	5.1.2
L_{max}	Largest packet size in system	3.3
L_p	Normalized traffic arrival load in an ingress switch/scheduler	3.5
$L_{p,min}$	Minimum normalized traffic arrival load in an ingress switch	7.3
$L_{p\ max}$	Maximum normalized traffic arrival load in an ingress switch	7.3
N	Average number of non-empty slots in an *nf-slot-set*	5.1.2

$N_{B,ij}$	In the ITDM system, the number of slots required to carry the traffic that is left in the buffers relevant to ingress switch i and egress switch j (under DTDM), as a non-granted traffic under CTDM	7.4
$N_{NG,ij}$	Number of non-granted slots during the previous frame between ingress switch i and egress switch j	7.1
$\overline{N_R}$	Average number of retransmissions	5.1.2
$N_{R,j}$	The number of slots in the retransmission buffer ready to be retransmitted to egress switch j	6.3.1
$N_{R,ij}$	Number of slots that must be retransmitted from ingress switch i to egress switch j	7.4
N_T	Total number of slot combinations in an *nf-slot-set*	4.1.1
$N_{T,i}$	Total number of non-empty slots transmitted from ingress switch i within a frame	7.4
N_c	Total possible combinations of *c-slot* collisions	4.1.1
$N_{c,d}$	Total possible combination of *c-slot* collisions to output link d	4.1.3
$\mathbf{P}$	Permutation matrix	7.2
$P_{D,i}$	Propagation delay between ingress switch i and the core switch	2.2
$P_{D,max}$	Propagation delay from the furthest ingress switch to the core switch	7.1
$Pa\{L_k = a\}$	Arrival probability of α slots to a tagged output link at retransmission level k and higher	5.1.2
P_c	Probability of *c-slot* collisions in an *nf-slot-set*	4.1.1
$P_{i,drop}$	Probability of slot drop at i-th retransmission level	5.1.2
$Q_{i,j}$	Quantum for stream j in class i, i= *EF,AF1,AF2, BE.*	3.2.2
R	Number of immediate upstream input source routers connected to an edge switch	2.1
$R_L\{n_r, k\}$	Slot loss rate for more than k slots among n_r tagged-priority slots	5.1.1
R_{SL}	Average slot loss rate at the core switch when the ITDM system is running under DTDM	7.4
$R_{SL}(n,E)$	Slot loss rate on a wavelength where n is number of ingress switches and E is number of empty slots	4.1.1
$R_{SL,f}$	Average slot loss rate within a frame at the core switch when the ITDM system is running under DTDM	7.4
$R_{SL,SW}(n,E)$	Slot loss rate of the switch where n is number of ingress switches and E is number of empty slots	D.2
$R_{SL,T}$	Slot loss rate when $\Lambda_s = C_{U,T}$	7.4
$\boldsymbol{R_d}$	A doubly stochastic matrix	7.2
S_O	Slot-offset	2.2
S_T	Slot time	2.2

S_p	Stability period to avoid IDTM from false switching between CTDM and DTDM	7.2.1
T_f	Expected logical frame time period (in OCGRR scheduler)	3.2.2
$T_{f,i}$	Actual logical frame time period for logical frame i (in OCGRR packet scheduler)	B.1
$T_{f,max}$	Maximum logical frame time period (in OCGRR packet scheduler)	3.3
T_v	An index for traffic value in a slot	6.3.4
$U_{i,j}$	Used-grant for stream j in class i , i= *EF,AF1,AF2, BE.*	3.2
V_{EF}, V_{AF}, V_{BE}	"Importance" parameters for classes EF, AF and BE respectively	2.3
W	Number of data wavelengths on each fiber	2.1
W_{max}	Maximum possible number of wavelength channels on a fiber	4.5
W_E	Number of extra channels on each fiber introduced to see the effect of carrying the same traffic	4.2.2.1
$\bar{d}$	Average number of lost slots	4.1.1
d_{ij}	Traffic demand entry between ingress switch i and egress switch j	7.1
f	Number of fibers on each connection link	2.1
n	Number of input/output connections in an optical switch	2.1
$\hat{n}$	Average number of input/output connections in an optical switch in a segment	4.5
n_e	Total number of egress switches in the network	2.1
$\hat{n}_l$	Average number of local edge switches connected to an optical switch in a segment	4.5
n_r	Average number of tagged-priority slots in an *nf-slot-set*	5.1.1
$n_{r,i}$	Average number of tagged-priority slots in an *nf-slot-set* at retransmission level i	5.1.2
p_d	Probability of traffic transmission to output link d	4.1.3
p_e	Probability of having an empty slot on a wavelength channel	4.1.2
r	Drop reduction ratio	4.1.2
t	Virtual time that is equal for all streams and classes in the OCGRR scheduler	3.2.2
w_i	Wavelength channel i	4.3.3
Γ	Expected frame length	3.1
ΔC	Percentage of network segment cost increase when considering optimal configuration	4.5.2
ΔC_N	Objective function of network segment cost	4.5.1
Λ	Set of assigned slots to n_e torrents in each ingress switch	6.1
Λ_i	Number of assigned slots to torrent i within a frame	6.1
Λ_s	Average normalized load of slot arrival from all ingress switches to the core switch	7.4
Υ	Fiber cost for a single fiber with only one wavelength inside	4.5.2

ξ	Desired slot loss rate related to contention avoidance	4.5.1
η	Coefficient to calculate the cost of a fiber with a number of wavelengths.	4.5
λ_T	Normalized traffic load of newly transmitted slots at $A_s = C_{U,T}$	7.4
λ_c	Normalized traffic load at the core switch	7.4
λ_i	Average arrival rate of class i traffic , i= *EF,AF1,AF2, BE.*	3.2.2
$\lambda_{j,i}$	Average arrival rate of stream i in class j, j= *EF,AF1,AF2, BE.*	3.2
λ_{min}	The smallest AAR among all streams in all classes	3.3
λ_s	Average normalized traffic load of newly generated slots from all ingress switches	7.4
π_i	Probability that a slot is retransmitted for the i-th time	5.1.2
ρ	Normalized slot arrival rate to the core switch	6.3
σ	Smoothing factor in running average method to provide bandwidth	6.2.1
σ_1	Smoothing factors in running average process	7.2.1
τ	Measurement frame in slot-sets to measure the distribution parameters	6.1
φ_i	Phase representing the time zone of ingress switch i	7.3
ψ	Maximum number of traffic classes in the DiffServ domain	2.3
ω	Number of required wavelengths on each connection link to carry full traffic load in network	4.5.1
ℓ	Average packet length among all classes	3.1
$\Im_{DE,i}$	Index for EDF-density-distribution for torrent i	6.1.1.2
$\Im^*_{DE,i}$	Index for optimum EDF-density-distribution for torrent i	6.1.1.2
$\Im_{DI,i}$	Index for EDF-distance-distribution for torrent i	6.1.1.1
$\Im^*_{DI,i}$	Index for optimum EDF-distance-distribution for torrent i	6.1.1.1
$\Im_{FD}$	Index for fair distribution	6.1.3
$\Im^*_{FD}$	Index for optimum fair distribution	6.1.3
$\Im_{LB}$	Index for EDW-load balancing distribution	6.1.2
$\Im^*_{LB}$	Index for optimum EDW-load balancing distribution	6.1.2

Chapter One: Introduction

The Internet demand is increasing 70% - 150% each year [Odly03]. New real-time applications such as video-on demand, emergency services, online gaming, and video-conferencing continue to grow and will consume more and more network bandwidth [XiNi99, CaCo02]. This can be a problem when the network bandwidth is limited, the network supports only the best effort traffic, and the traffic does not have a uniform characteristic. For example, studies show that the Internet traffic has two non-uniform characteristics [PaFl95, FlPa01]: 1) Diurnal pattern in which the pattern of network traffic follows the daily patterns of human activity and almost the same every day, except weekends [ThMi97, FlPa01, BaKl02]. Dynamic traffic for diurnal pattern is usually represented through a time-dependent stationary process that follows a sinusoidal traffic pattern, e.g., [Medh02, HuKa03, GuZh05]; and 2) Burstiness.

Since different applications need different levels of Quality of Service (QoS), service differentiation must also be considered in future networks [XiNi99, Dasi00, DhTa01, CaCo02]. Under the best-effort service in which no guarantees can be given to any packet regarding loss rate, delay and delay jitter [XiNi99], all traffic in the network is equally treated. This will in turn degrade the QoS requirements for the real-time traffic. Thus, having a QoS-capable optical backbone network will be a requirement in the near future [DhTa01]. There are two common service differentiation mechanisms in Internet: IntServ [BrCl94] and Differentiated Services (DiffServ) [BlBl98]. IntServ achieves QoS guarantees through end-to-end bandwidth reservation for IP flows and performing per-flow scheduling in all intermediate routers or switches in network [KaKh02]. On the other hand, DiffServ provides QoS differentiation for different classes of traffic aggregates [KaKh02]. Due to the scalability problem of IntServ [StMa02], DiffServ is used in optical networks, e.g., [LoTu03, ØvSt04] as well as in this dissertation.

Under DiffServ, edge routers are in charge of classifying, marking, dropping, or shaping of the IP packets based on the service level agreement and preventing the DiffServ network from malicious attacks [HaFa05]. While core routers perform high speed routing of classified packets. The DiffServ model provides a relative per-class QoS

differentiation, such as higher bandwidth, lower delay, or lower loss at an aggregate level, by allocating more bandwidth to one aggregate than another [SiBa03]. Three services are defined for DiffServ: Expedited Forwarding (EF) [JaNi99], Assured Forwarding (AF) [HeBa99] and Best Effort (BE). The EF service facilitates the applications that demand lower loss rate, lower latency, lower jitter, and bandwidth guarantees, while AF offers different levels of forwarding assurances to IP packets. The remaining traffic is treated as best effort (BE) with no QoS guarantees.

All-optical networking with widespread deployment of the Dense Wavelength Division Multiplexing (DWDM) technology appears to be the sole approach to transport the huge network traffic in future backbone networks [DiDe03]. The DWDM technology provides the multiplexing of many wavelength channels in a single optical fiber, resulting in several Tbps bandwidth. An all-optical network is interconnected with a number of transparent all-optical switches (each called a core switch) and a number of edge switches connected to the all-optical switches where each edge switch is operating in the electronic domain. By buffering packets electronically and forwarding them to the optical network, the edge switches provide the interfacing between the electronic networks and the all-optical backbone. An all-optical network uses a transparent optical signal transmission without any conversion to the electronic domain in the core switches [StBa00, DhTa01] for data, while may process the header of data traffic in the electronic domain. Under transparent optical networking, transparent core switches are not only cheap but also the bandwidth offered by DWDM can be fully utilized [StBa00, DwSm02, HeEl04].

The optical and electronic networks have essential differences in switching speed, buffer architecture, and bandwidth granularity. In the optical domain, switching speed is slower [Kart03] and building optical buffers is a complex and an expensive issue than the electronic network. On the other hand, optical networks provide a higher bandwidth, a better signal quality, and a better security over electronic networks [PaOb02]. By considering these differences, different architectures and bandwidth management protocols should be used in order to employ the huge bandwidth that is offered by all-optical networks.

1.1 Literature Review

We shall review the existing work related to our research in all-optical networking,

including bandwidth management, packet aggregation techniques in edge switches, and packet scheduling algorithms in order to comprehend the amount of work done, outstanding or deficient.

1.1.1 All-Optical Switching Schemes

We shall briefly study a number of switching schemes proposed in time-slotted and asynchronous all-optical networks. A detailed survey of optical switching schemes can be found in [PaOb02, Pota02, PaPa04].

1.1.1.1 Asynchronous Optical Switching

In Optical Circuit Switching (OCS), a logical connection is established between a source-destination pair [StBa00] by setting up a permanent light-path using routing and wavelength assignment algorithms [ZaJu00]. However, bandwidth wastage and scalability issues are two drawbacks [Pota02]. To improve the bandwidth usage, the light-path can be dynamically set up and turned down, e.g., in Optical Flow Switching (OFS) [MoNa00, GaCh02] and the PetaWeb network [BeFi03]. However, OCS still suffers from lower channel utilization because a connection cannot employ the whole bandwidth of a light-path, and scalability issue.

A packet switching technique can obtain a higher channel utilization due to its finer granularity and can yield a better bandwidth efficiency [BrCh05] by different approaches: 1) Optical Packet Switching (OPS): each packet is individually switched like its electronic counterpart, while keeping payload in the optical domain and processing the packet header electronically [ChCh99a, YaMu00, HuAn00, OmSi01, Chia04]. This network is necessary for both metro and backbone networks [Chia04, PaPa04]; 2) Optical Burst Switching (OBS) [YoQi99]: This is a switching protocol with a finer granularity comparing to OCS and a coarser granularity than OPS. OBS assembles packets in bursts, sends burst header on a control channel, waits for an offset time, and then transmits the burst over a data channel. Each OBS core switch processes the burst header and opens an output channel to forward the arriving burst while resolving the burst contention at the core switch.

Among the switching schemes, OPS is not only scalable but also is flexible and can also dynamically allocate network resources with a finer granularity [PaPa04]. It can

efficiently use the network bandwidth, which enables it to support diverse services [Deve02, FaHe04]. However, there are three limitations for OPS: 1) Its inability to save optical data indefinitely with random access capabilities [PaPa04]; 2) The lack of sophisticated processing in the optical domain [PaPa04], while higher processing is a requirement in the optical switches due to the larger amount of overhead [Deve02]; and 3) Its requirement of a fast switching speed [Deve02] (e.g., nano-seconds), while switching in this range seems to be far from being commercialized [ElSh02, MaKu03, PaPa04].

1.1.1.2 Slotted Optical Switching

To provide a finer granularity and improve bandwidth usage, the Optical Time Division Multiplexing (OTDM) concept can be deployed in all-optical networks. Under OTDM, many source-destination pairs can share the network bandwidth. However, synchronization of traffic arrival in core switches [RaTu04] is the limitation of this switching. In slotted networks, fiber delay lines are required for synchronization issue at the input ports of core switches [SeBe96, ChCo01, YaMu00, RaTu04].

In Slotted Optical Circuit Switching (Slotted-OCS), e.g., [HuLi00, WeSi02, ChWo04], the bandwidth of a wavelength is divided into frames of fixed time-slots and traffic for a given source-destination pair is periodically carried in pre-allocated time-slot(s) (when connection is up) in each frame. The Routing and Wavelength Assignment problem, e.g., [ZaJu00] in the wavelength-routed networks is changed to the Routing, Wavelength and Time-slot Assignment problem in Slotted-OCS.

There are different approaches to deploy the packet switching concept in slotted networks:

1. In slotted-OPS [YaMu00, YaYo01b, ElSh02], a fixed-length packet together with a header is placed inside a fixed time-slot and transmitted to OPS network. Larger packets must be fragmented and transmitted in a number of time-slots [HuAn00]. Slotted-OPS network has a lower complexity in terms of switch control than asynchronous OPS [PaPa04]. In addition, it has a higher throughput than asynchronous OPS due to a lower contention in slotted-OPS [YaYo01a, PaPa04]. However, different packet-sizes may cause some bandwidth degradations for slotted-OPS. Of the variable length IP packets, almost 46% have the length of 40 bytes, 18%

have the length of 1500 bytes and the remaining packets have the average size of almost 560 bytes [CAID06]. Hence, the variance of the IP packet sizes is very high and choosing a slot-size[1] of 1500 bytes will result in higher bandwidth wastage. On the other hand, using a slot-size of 40 bytes requires not only a much faster optical switching speed but also a faster processing at core switches to process a large number of packet headers in a very short time [Chia04, PaPa04]. Clearly, this requirement will be even higher when 40Gb/s transmission rate is used in the network. It also leads to packet fragmentation issue, which imposes more header complexity and induces cost due to the reconstruction of received packets at egress switches [Tane02, CaPa03]. The other problem with fragmentation is that when a part of a packet is dropped, the whole packet will be useless.

2. In slotted-OBS [FaVo03, RaTu03], each burst is subdivided into multiple time-slots and transmitted at fixed positions in a periodic frame structure. The control wavelengths carry Burst Header Cells, so that core switches along the path can setup a connection. This architecture may need optical buffers in the core switches in order to interchange time-slots. The problem with the burst division is that the whole burst can be blocked due to the blocking of a small part of the burst, thus leading to an inefficient resource utilization.
3. In Photonic Slot Routing (PSR) [ChEl97, ChFu99], simultaneous slots (each slot may carry a number of packets) transmitted on distinct wavelengths are aggregated in one photonic slot and routed through a core switch. Since this technique does not require a wavelength-selective optical switch, it is cost-effective. However, this network is not too flexible and the network bandwidth may not be fully utilized because the whole traffic on a fiber can only be switched to a specific output port of the core switch. Consequently, this switching usually finds application in ring networks.
4. In Ethernet Passive Optical Network (EPON) [KrPe02, YuAn05], a central Optical Line Terminal (OLT) provides bandwidth for Optical Network Units (ONUs) in a tree topology. This architecture is only suitable for a local area network.

---

[1] In slotted-OPS, consider a slot as a container to carry an integer number of IP packets within a time-slot. Slot-size is the capacity of a slot in bits. For example, for a 1μsec time-slot and channel bandwidth of 10Gbs/sec, the slot can carry traffic up to 10kbits.

1.1.2 Bandwidth Management in All-Optical Networks

There are two common frameworks that would allow the sharing of bandwidth in an all-optical network: reservation-based schemes and contention-based schemes. Reservation-based schemes are usually used in slotted networks to reserve bandwidth for a connection pair. They work well under (semi) static traffic [WiSa04]. On the other hand, contention-based schemes, which can be used in both slotted and asynchronous networks, can cope with the dynamic traffic fluctuations as well. Contention-based schemes may suffer from collision, while reservation-based schemes may experience connection blocking.

1.1.2.1 Reservation-based Bandwidth Management Schemes

The decision made in reservation-based schemes can be either centralized or distributed. In the former, only one central module is in charge of all bandwidth provisioning procedure while in the latter a number of modules cooperate with each other to reserve the bandwidth. Both techniques are usually executed periodically to re-compute the loss-free slot schedule in response to the new traffic demands of edges switches.

A) Centralized Bandwidth Reservation: In centralized techniques, e.g., [MaBi00, MoPa03, MiIn04, YuAn05, SaPa06], a central controller dynamically allocates bandwidth based on the traffic demands from ingress switches. When the number of ingress switches is high, the scheduling complexity increases both in terms of memory and time requirements. Then, the edge switches send their traffic following a predetermined schedule. This technique is usually used in single-hop all-optical networks. A single-hop WDM network could be either based on a central passive switch [MaBi00, SaPa06, MaRe02, FaAd05], or a wavelength-selective all-optical cross connect switch (denoted as active switch), e.g., [BeFi03, LiVi05, MaVi05, JiYa06]. By using the active switches, a more efficient WDM network can be realized than using the passive switches due to the spatial wavelength reuse and splitting loss problems in passive components [StBa00]. Note that apart from the design simplicity, the synchronization required in a TDM network can be easily achieved in an all-optical star network [BlLe02, MaVi05].

A centralized scheme is also natural for a star network in which there is no optical or electronic buffer inside the core switch, while the edge switches have electronic buffers.

However, a star network suffers from the central node failure. So the overlaid star topology is usually considered instead along with the traffic protection problem, e.g., [FaMa04, ShYa04]. In particular, the AAPN (Agile All-Photonic Network) network [AAPN06] uses this overlaid star topology as well as the central scheduling technique, e.g., [Mcke99, ChCh00, LiAn01, BiGu03, KeKo05] for its core switches to schedule traffic for the edge switches with different propagation delays, e.g.,[LiVi05, MaVi05, JiYa06]. Note their schedulers have been originally used for the scheduling in electronic switches with input buffers (i.e. input queued switches).

The scheduling in input queued switches can be mapped to the bipartite matching problem. Building a perfect matching, e.g., [ChCh99b, ChCh00] has a complexity of $O(n^{2.5})$ [PaSt82]. To reduce this complexity, a number of heuristic matching algorithms, e.g., [AnOw93, Mcke99] have been proposed, but at the expense of bandwidth wastage, because they cannot guarantee a perfect matching.

The scheduling in input queued switches can be either slot-based or frame-based. In the former, the scheduling is done in a time-slot level. For example, the slot-based scheduling is used to perform the scheduling in AAPN network, e.g., [LiVi05, MaVi05, JiYa06]. Here, for each traffic demand requested from edge switches, only the slots within one time-slot period are reserved by the core switch (i.e. a centralized scheduling). These algorithms account for the propagation delay because electronic buffers are remotely located at the edge switches. The slot-based design is not suitable for a WAN topology because of the performance degradation when an edge switch and the core switch are far apart [MaVi05]. A multi-processor architecture [JiYa06] can also be used to overcome the complexity of the scheduling algorithm, but such architecture is usually more expensive.

In frame-based scheduling, the scheduling algorithm is run once in a time frame (consisting of a number of slots) to obtain the switch configuration in each time slot inside the frame. Frame-based scheduling can decrease the frequency of matching computation [KaLa00, KeKo05] comparing to the slot-based algorithms. In addition, frame-based scheduling can reduce the communication overhead during scheduling [ChLe04], and can accommodate more scheduled slots within a frame period of F slots than F times slot-by-slot scheduling [CaRa03]. Furthermore, the frame length can be

tailored by the network manager based on the running time of the scheduling algorithm [BiGi04]. Let us consider an example for frame-based scheduling. A traffic matrix sent from edge switches to the core switch must first be converted to a doubly stochastic at the core switch [LiAn01]. Then, the demanded traffic can be scheduled within a frame by using the Birkhoff and von Neumann (BvN) decomposition (see Appendix G for details) algorithm [ChCh99b, ChCh00, ChCh01]. Since BvN uses a perfect matching algorithm, it guarantees to schedule the demanded traffic. However, since both of its computational and memory complexities are high, it is not scalable.

B) Distributed Bandwidth Reservation: In distributed techniques, the edge switches or core switches can use one of the two approaches to reserve bandwidth: 1) Each ingress switch first sends its traffic request to a desired egress switch (i.e. destination edge switch), e.g., [StBa00, WiSa04]. The egress switch performs a loss-less slot assignment for all ingress switches. Then, it informs each ingress switch about its schedule to send its traffic. This technique experiences more communication overhead, and longer packet delays at the ingress switches; and 2) Bandwidth is reserved for the ingress switches by intermediate core switches in the optical network, e.g., in slotted-OCS [WeSi02, ChWo04].

1.1.2.2 Contention-based Bandwidth Management

The contention-based schemes rely on a random access to the network. They are used in switching schemes such as OBS/OPS, PSR and slotted-OBS/OPS. Due to a lower complexity, they are easily scaleable, and can respond more quickly to bursty traffic [NoBj03] than the reservation-based schemes. When traffic fluctuates, the reservation-based schemes would reserve the bandwidth throughout the network in order to handle the traffic. However, the contention-based schemes can handle the traffic fluctuations better. For example, as soon as a client packet arrives, an ingress switch creates an optical packet and sends it directly to the network in un-slotted OPS. The optical packet can only be sent to the network at the time-slot boundary (the waiting time for the slot boundary is negligible comparing to the bandwidth reservation period in a bandwidth-reservation scheme) after a possible fragmentation in slotted-OPS [YaMu00, YaYo01b, CaPa03, RaTu03]. In other words, OPS can handle traffic fluctuations as soon as possible.

In a contention-based all-optical network, there is no collaboration among ingress switches. Their uncoordinated traffic transmitted on the same-wavelength and the same time going to the same output link of an optical switch may collide with one another at the optical switch. Some of the contending traffic will go through while others are dropped. Comparing to asynchronous optical networks, slotted operation can reduce the vulnerable period of information contention and reduce traffic loss, e.g., [YaYo01a, PaPa04]. However, one must still resolve any contention that may occur. The contention can be reduced by resolution and avoidance schemes.

A) Contention Resolution Schemes: Contention resolution schemes can be divided into hardware-based and software-based techniques. The following are the major hardware–based techniques:

1) Optical buffers [ChFu96, HuCh98, DiCh99, TaYe00, ZhYa03, BaDe04] are used to resolve contention at the time domain. Optical buffering technology is immature and so far, it relies on bulky optical fiber delay lines. A core switch needs a large number of hardware and complex scheduling algorithms in order to implement optical buffers [YaMu00]. In addition, optical signals degrade in fiber delay lines [YaMu00]. To compensate the signal degradation, optical amplifiers are often used. However, the cascaded amplifiers accumulate noise that can severely limit the network size at very high bit rates, unless expensive signal regeneration units are employed [YaMu00]. Shared optical buffers are also studied in OPS [YaLi05, YaZe05, YaZe06]. For a very low loss rate, we require many optical buffers [HuCh98].
2) Wavelength Converters (WC) [DaHa98, ErLi03a, ErLi03b, ErLi05] can reduce traffic loss at the wavelength domain, but they are expensive. Wavelength converters can be used in a shared architecture in order to reduce the conversion cost [LeLi95]. Although optical buffers are more effective to resolve contention than wavelength converters, wavelength conversion can provide noise suppression and signal reshaping [YaMu00]. Combination of shared WCs and optical buffers are studied in [Gaug04, LuHu05a, YaZe06]. When a very low loss rate is required, the required number of wavelength converters [ErLi03a, ErLi03b] will drastically increase.
3) Local drop port dimensioning [XuPa03] can also resolve contention because of a non-blocking receiver. In this technique, an OPS core switch has enough number of drop

ports to its local edge switch so that no packet is dropped due to a collision to the local edge switch.

The following are the software-based resolution techniques:

1) Deflection routing [WaMo00, ChCo01, WaMo02, HsLi02, LeSr03] is a cheap and simple technique that reroutes a contending optical packet (in a slotted-OPS network, an optical packet is referred to as slot) toward its egress switch through other paths in the network. It can only be used in a multi-hop network. Moreover, optical packets may deflect in multi-hop networks for a long time, and deflection causes an additional burst or slot delay. This technique does not resolve the contention. It only deflects the contention problem to another core switch in the network.
2) Retransmission at the optical domain is another technique to recover lost traffic. The retransmission issue is even useful for real-time media [PeSc96]. In OBS, the retransmission of dropped bursts/packets by higher layers is preferred due to the complexity of keeping huge data in the optical layer [NoBj03]. However, this issue may cause a false TCP congestion detection problem because loss may happen even in lower traffic loads [YuQi04]. This in turn triggers the TCP slow-start congestion control mechanism, thus reducing the TCP throughput. The retransmission issue could be suitable for single-hop metro networks or mesh networks with a short diameter, but not as good in multi-hop networks with a longer diameter due to higher propagation delays. To implement retransmission, each edge switch keeps a copy of the transmitted traffic in its electronic buffer and retransmits whenever required. The conventional retransmission technique, denoted by the Random Retransmission (RR) technique from now on, retransmits slots until a success. RR has been used even in optical networks [ChFu94, Modi99, SaBr00, YoQi00a, WaMa03, LuZh04, MaBo04a, ZhVo05]. Since the number of retransmissions is not limited, the multiple retransmissions may lead to further retransmission at the TCP level, which may in turn reduce the TCP throughput in the network. Although simple prioritization of the retransmissions has been used in the wireless and TCP domain [ReRe01, LaEd03, LiFa03], to the best of our knowledge, this idea has not been used or analyzed in all-optical networks.

B) Contention Avoidance Schemes: Contention avoidance schemes reduce the number of collision events. The following are some common techniques found in the literature:

1) Using load balanced routing to minimize loss, e.g.,[ThVo03, DuPu06, LiYe06].
2) Employing a feedback-based mechanism, e.g., [MaBo04b, UnØv04, FaZh04, LuHu05b, KiMu05, LuHu06] to minimize traffic loss. The contention avoidance is achieved by dynamically varying traffic flows at an edge switch to match the latest status of the available network resources. Edge switches learn from the status of contending slots reported by core switches, and change slot positions at frame boundaries to avoid contention [LuHu05b]. This protocol is proposed for an OPS network using wavelength insensitive core switches, and may be suitable for semi-static traffic where traffic arrival status changes at longer intervals.
3) Using a number of fibers to connect any edge-to-core or core-to-core [LiXi04].
4) Duplicating transmission of the same traffic on the network to increase the chance of traffic delivery to egress switches [Øver04, HuVo05, LiYe06].
5) Transmitting optical packets to OPS network with even spacing [SiMo04].
6) Extending the traffic shaping schemes used in electronic networks to shape bursty traffic at edge switches, e.g.,[VeCh00, ElCh03, SiMo04, SiMo05, LuHu05b, LuHu05c].
7) Aggregating traffic at edge switches and entering the shaped/regulated traffic to the network, e.g.,[XuYa02, YaXu02, ElMe03, XuPa03, XuPa04]. Another approach to reduce the burstiness could be packet aggregation [YaXu02, XuYo02]. Aggregation of IP packets in optical packets reduces the coefficient of variation and the first-lag autocorrelation function for both inter-arrival time and lengths of optical packets to the network [XuPa03]. This is a contention avoidance technique that appears to be quite popular, e.g., [YaXu02, XuPa03, ElMe03]. As the aggregation-size goes up, the aggregated traffic tends to be smoother. However, packet assembly does not reduce burstiness in general [HuDo03].

By all counts, the contention avoidance schemes appear to be much cheaper than the resolution schemes since most of the avoidance schemes are the software level tools. In addition, a multi-fiber architecture without using full wavelength converters inside any optical switch is also cheaper than single-fiber architecture when using full wavelength

converters [SoMi04]. In addition, devices such as dispersion compensators, used for a fiber with a higher number of wavelengths is much more expensive than the devices used for a fiber with a lower number of wavelengths [GeRa03, SoMi04]. In other words, a multi-fiber architecture with a fewer number of wavelength channels per fiber is cheaper than a single-fiber architecture with a large number of wavelength channels per fiber. Multi-fiber architecture has been widely used in OCS networks, but not much in OPS networks.

Contention analysis has been studied in slotted OPS, e.g., [ZhYa03, ErLi03a, ErLi03b, UnØv04], under symmetric traffic (equal traffic between any source-destination pair) in which an upper-bound is obtained for the loss rate. There appears to be no work on the lower-bound analysis.

1.1.2.3 QoS Support in All-Optical Packet-Switched Networks

All the current proposals to provide differentiation in store-and-forward electronic routers/switches all rely on buffers and are not suitable for all-optical core switches due to the lack of optical buffers (i.e. either having a very little or no optical buffers since this option is several years ahead), e.g., [KaKh02, UnØv04, ØvSt04, CaSo05]. Therefore, there are interests in providing new approaches to provide service differentiation in all-optical networks without the use of optical buffers.

QoS can be provided in an OBS network by a number of techniques such as: 1) Exploiting an extra offset-time for higher priority traffic in ingress switches, e.g., [YoQi00b]; 2) Assembling composite bursts at an ingress switch and segmenting the burst at any core switch downstream, e.g., [VoZh02]; 3) Intentional dropping of lower priority bursts at any core switch with a certain probability (i.e. an early drop technique), e.g., [Gaug03, ZhVo03]; 4) Scheduling control packets at any core switch rather than using a FCFS (First-Come-First-Served), e.g., [KaKh02, Gaug03]; 5) Reserving network resources (e.g., wavelength channels and optical buffers) for higher priority traffic and limiting them for lower priority traffic at any core switch, e.g., [Gaug03, NoBj03]; 6) Group scheduling of burst headers at any core switch within an interval period in order to provide a better service for higher priority bursts, e.g., [TaLu04, ChEl06]; 7) Wavelength grouping and adaptive adjusting the number of reserved wavelengths for higher priority traffic, e.g., [ZhVo03]; and 8) Preempting lower priority bursts (i.e. disrupting an

ongoing transmission) by higher priority bursts at any core switch, e.g., [Gaug03]. The preempting can also be performed with a certain probability, e.g., [YaYj03].

To provide QoS in an asynchronous OPS network, a number of techniques have been proposed such as: 1) Preempting lower priority traffic by higher priority traffic at any core switch, e.g., [BjØv05]. The preempting can also be performed with a certain probability, e.g., [ØvSt04]; 2) Managing optical buffers at any core switch to provide a better differentiation for higher priority traffic, e.g., dropping lower priority traffic from optical buffers when the amount of lower priority traffic exceeds a threshold, e.g., [CaCo02, NoBj03]; 3) Intentional dropping of lower priority packets at any core switch with a certain probability, e.g., [ØvSt06], where core switches are buffer-less; 4) Reserving a set of wavelengths or wavelength converters exclusively for high priority traffic at any core switch, e.g., [CaCo02, ElAt01]; 5) Redundancy of higher priority packets at ingress switches, e.g., [BjØv05]; 6) Traffic shaping at ingress switches, e.g., [GrGo01]; and 7) Adaptively adjusting the number of wavelength converters reserved for high priority traffic at any core switch, e.g., [Øver03].

Due to the fundamental difference in the operation of slotted and asynchronous OPS networks as discussed before, only the reservation-based and the redundancy techniques are applicable to buffer-less slotted-OPS. For slotted-OPS, the differentiation technique proposed in [Øver05] isolates the service classes at a core switch by ensuring that a certain number of higher priority packets can be transmitted at a given output port in a time-slot when contention happens.

1.1.3 Bandwidth Access in Ingress Switches of OPS/OBS Networks

Apart from the switching techniques reviewed in Section 1.1.1, an important issue is to provide network access for traffic streams in an ingress switch, where packet differentiation must be considered as well. The common approach to support the DiffServ traffic in an ingress switch is to save all same-class packets from different source routers going to the same egress switch in a shared FCFS electronic buffer, e.g., in OBS [PaYo02, LoTu03, PuPe05], and, e.g., in OPS [RaZa03, YoXu03, RaZa04]. However, a bursty source in a class may cause both a higher delay and even loss for well-behaved sources within that class. Fig.A.1 (see Appendix A) shows this architecture in an ingress switch.

As discussed in Section 1.1.2.2, an OPS ingress switch can immediately send an arriving IP packet to the OPS network. However, this may increase traffic loss at the network due to the burstiness of the IP traffic. As reviewed in Section 1.1.2.2.B, there are two groups of traffic shaping schemes proposed at the ingress switches to cope with the bursty traffic and avoid contention in an OPS network. The first group uses the traffic shaping schemes borrowed from the electronic networks, e.g., [VeCh00, ElCh03, SiMo05]. The second group aggregates a number of IP packets in an optical packet or burst to reduce the traffic burstiness at the optical domain. The second group has two more advantages: 1) It allows us to use relatively larger optical packet sizes, and larger time-gaps between time-slots in slotted-OPS. This alleviates the requirement to use a very fast optical switch in an OPS core switch; and 2) It reduces the complexity of an OPS core switch because the number of entities per unit time to be processed by the core switch is decreased [OmSi01]. All these advantages results in a cheaper OPS core switch implementation.

Packet aggregation is a technique to form bursts or optical packets in ingress switches of all-optical networks in order to take the advantages of its enormous bandwidth and avoid contention as discussed before. Ingress switches are responsible for assembling the packets from source hosts to a particular egress switch. Once an egress switch receives a burst or an optical packet, it is unpacked and the client packets are routed individually to the relevant destination host.

Packet aggregation can be either a timer-based, e.g., [Gaug03], or a threshold-based, e.g., [Gaug03, ZhLu03], or a combination of them, e.g., [XiVa00, XuYa02, XuPa03, RaZa03, RaZa04]. In the timer-based packet aggregation technique, a timer is started whenever an IP packet arrives in egress switch i queue and the queue is empty. The time-out mechanism is used to limit the waiting time of packets. The aggregation algorithm waits for an aggregation period and during this period, collects all arriving packets to egress switch i. At the end, a bursts or an optical packet is created. In the threshold-based technique, when traffic in a queue reaches a threshold value, a fixed-size burst or an optical packet is formed.

In the Combined Timer-based and Threshold-based (CTT) technique, a burst or an optical packet is created whenever there is enough traffic to make a full optical packet or

burst, or whenever a time-out event has happened. Then, the aggregated packets will be sent to a FCFS transmission buffer (an infinity buffer), e.g., in OBS [VeCh00, LiAn04, HuKo06] and, e.g., in OPS [KhMo02, RaZa03, RaZa04, ZhYo05, RaZa06], even if there are not enough IP packets in the buffer to make a full burst or optical packet [XiVa00, KhMo02, XuYa02, XuYo02, YaXu02, RaZa03, XuPa03, YoXu03, RaZa04, Kotu04, RaZa06]. Without using the transmission buffer, optical packets/bursts may be discarded at an ingress switch. For example, bursts generated by different sources in OBS network are discarded in the ingress switches [RaTu03] because of not using the transmission buffer. An Optical Bandwidth Manager (OBM) unit then transmits the optical packets/bursts from the transmission buffer to the optical network on the available wavelengths and fibers that can be randomly chosen. Note that some works only study CTT for a single-class traffic, e.g.[YaXu02, KhMo02, XuPa03, RaZa04].

A common architecture is to use a shared buffer for the same-class packets from all sources and the CTT technique to access the bandwidth in a slotted-OPS network. Referred to as *Arch1* in this dissertation and depicted in Fig.A.1 in Appendix A, an integer number of the same-class IP packets are assembled in an optical packet[2] where each optical packet is transmitted within a time-slot [Kotu04, RaZa06, GoMa06]. Time-out mechanism is used to provide packet differentiation in which smaller time-outs are assigned for higher-priority traffic and larger time-outs are assigned for lower priority traffic. However, the choice of time-out value is a challenging problem. To choose a time-out, one must consider different network parameters such as the traffic load, the number of QoS classes, the slot-size, the wavelength channel rate, and the number of egress switches, e.g., [XiVa00, CaPa03, LoTu03, RaZa03, RaZa04]. Since the volume of higher priority traffic is usually smaller than lower priority traffic, the average assembling time for the higher priority traffic is longer than the lower priority traffic [CaLu04, Kotu04] and this is the reason that smaller time-outs must be chosen for higher priority traffic than lower priority traffic. However, this may increase the load on OPS network, and the loss as well. Moreover, smaller time-outs or even larger time-outs at higher traffic loads may result in an unstable edge switch operation in which the slot

[2] An optical packet in slotted-OPS is referred to as a slot in this dissertation.

(recall in a slotted-OPS network, an optical packet is referred to as a slot) generation rate to the OPS network is higher than the slot service rate. Therefore, the waiting time in the slot transmission buffer may increase the packet waiting time in an edge switch unbounded

For class-based traffic, current packet aggregation techniques assemble packets of the same class in a burst or optical packet [PaYo02, Dolz02, VoHa02, LoTu03, RaZa03, RaZa04]. However, [VoZh02, VoJu03, ZhLu03] have introduced composite packet aggregation in burst in an OBS network in which packets from different classes can be aggregated in a burst. It is showed that the end-to-end packet delay and burst loss rate of the composite assembly is less than the non-composite case because bursts are assembled sooner in the composite case [VoZh02, VoJu03]. Note that all the work above is for an OBS network, and the idea of composite packet aggregation has not been studied in a slotted-OPS network so far.

1.1.4 Packet Scheduling

Since we are planning to use packet scheduling in the edge switches of an all-optical network to provide packet differentiation, we study the packet scheduling algorithms in this section. Although by using a shared output-queued buffer for all traffic streams (see Fig.D.1) buffer management would be easier [KaPa02], it is difficult to control the service order of packets from different sources because a malicious source in a class may cause both a higher delay and even loss for well-behaved traffic streams within that class. Consequently, fair queuing algorithms such as Deficit Round Robin (DRR) [ShVa96], DRR++ [MaSh00, ZhMa02] and weighted fair queuing [BeZh96] are proposed for output-queued switches so that a dedicated buffer is assigned to each flow to isolate a malicious flow from well-behaved traffic streams.

Two types of scheduling algorithms in terms of operation are timer-based (sorted priority), e.g., [Gole94, BeZh96, Goya97, DoSt02] and credit/frame-based, e.g., [KaSi91, ShVa96, SaMu98, ZhYa99, MaSh00, KaPa02]. Time-stamps are used in the timer-based techniques to determine the departure time of the packets. Apart from their benefits, e.g., [LeMi04], these types of algorithms have real-time restrictions in their implementation [BeTs01, KaPa02] due to the calculation of the time-stamps and the sorting process to choose eligible packets for transmission, especially at the edge switches of optical

networks with a huge number of packets.

The credit-based schedulers divide time into logical frames and process traffic streams in a round robin manner. Credit-based scheduling algorithms can have different capabilities such as handling different packet lengths and traffic types. For instance, algorithms like [KaSi91, SaMu98, KwLe98, ZhYa99, WaSh01, WoHa03] are suitable for fixed-length packets. Deficit Round Robin (DRR) [ShVa96], Smooth Round Robin (SRR)[Guo01] and DRR++ can well handle variable-length packets while DRR/SRR can support smooth traffic and DRR++ can support both smooth and bursty traffic for a latency critical flow. DRR has a tendency to generate bursty output when serving a traffic stream, thus leading to a higher startup latency and jitter. To remove this problem, some techniques are proposed in [KaSe01, TsLi01]. All DRR-based algorithms, e.g., [KaSe01, LeMi04], must know the maximum packet length to achieve the work complexity [ShVa96] of *O(1)*. When scheduling a packet from a traffic stream, unlike [KaPa02], DRR/DRR++ must know the packet size of the head of the traffic stream in order to decide whether to schedule the packet or not. *DRR/DRR++* usually generate bursty output, while Nested DRR [KaSe01], SRR, and modified WRR [WaLi94] can generate a smooth output by having each stream to send traffic up to the maximum-length packet.

Scheduling techniques have also been used in the multiple-class domain such as the Priority Queuing (PQ) [BeGa87], Token Bank Fair Queuing (TBFQ) [WoHa03, WoTa04], Weighted Round Robin (WRR) [KaSi91], Dynamic WRR [KwLe98, WaSh01, JiAr03], PQWRR [MaMo01], DRR+ [ShVa96] and DRR++. They all support variable-length packets, except TBFQ and algorithms based on WRR that are designed well for fixed-size packets. On the other hand, only DRR++ and the Dynamic WRR techniques can support bursty traffic. DRR++ suffers from head of line blocking when scheduling more than one higher priority traffic stream. This problem can be resolved by increasing the service time complexity of DRR++ to *O(n)*. PQ is unfair to the lower priority traffic. PQWRR is unfair to AF and BE by using PQ for the EF traffic and WRR for the AF and BE traffic. Finally, a number of class-based algorithms like DRR+ and DRR++ are originally designed for two classes only.

1.2 Motivations

Of the several switching schemes in the literature, the OCS network has been very

popular in simplifying packet forwarding. However, this switching is neither efficient on the bandwidth usage nor scalable as discussed. Contention-based schemes appear to be more suitable in responding quickly to Internet traffic dynamics. On the other hand, OPS as a contention-based scheme appears to have the finest granularity comparing to OCS and OBS. An all-optical OPS network combining the higher bandwidth of optical technology with the flexibility of packet switching can utilize the network bandwidth more efficiently. Therefore, it is a good candidate solution for future all-optical networks, especially for both metro and backbone networks. Unfortunately, many OPS architectures resolve network contention by using immature optical buffering technology, which is expensive, complex, and bulky (using optical fiber delay lines) as discussed before. To reduce the cost and complexity of network, the core switches should avoid the expensive optical buffer for contention resolution. In addition, slotted networks can provide a much finer granularity in bandwidth sharing and can improve the bandwidth usage as reviewed. Finally, next generation networks must be QoS-capable in order to support the applications that require different levels of QoS as discussed. Based on all the above discussion, the slotted buffer-less all-optical packet-switched networking under the DiffServ domain appears to be a good candidate that needs more in depth investigation. Although this network can provide a very efficient bandwidth utilization for the next generation networks, there are still a few issues related to design and bandwidth management that must be addressed:

Reconfiguring an optical switch in slotted-OPS requires a significant time under the current technology [Kart03], which is not suitable for an OPS network with nano-seconds switching requirement. By handling each packet header, a burden is forced on the optical core switch design and operation. In addition, choosing an improper slot size would lead to either bandwidth wastage or fragmentation due to large variation of packet sizes (e.g., in Internet as reviewed). Therefore, we would be interested in using a larger slot-size to aggregate a number of IP packets.

When a very low loss rate is required, the required number of wavelength converters and optical fiber buffers will drastically increase, and clearly, this will result in a very high network cost. The deflection routing technique has its own problems. With electronic buffers decreasing in price, it would be worthwhile to use retransmission in the

optical layer. We would be interested in contention avoidance issue to see how it affects the loss rate and how it can help to reduce the burden and cost of the contention resolution hardware.

Currently, the bandwidth access technique in an edge switch of an OPS network has a number of drawbacks: 1) It is difficult to control the service order of packets from different sources in a shared buffer because a bursty source in a class may cause a higher delay and even loss for well-behaved sources within that class; 2) Since the volume of higher priority traffic is usually smaller than lower priority traffic, the average assembling time for higher priority traffic is longer than lower priority traffic. The solution may lie in the choice of time-outs. However, there are some tradeoff issues that need to be addressed; 3) The management issue of load balancing on the transmission channels is difficult because the OBM unit would not know the number of slots to be generated from the fluctuating traffic arrival. Therefore, an obvious approach is to aggressively transmit slots. Consequently, transmission channels may be fully utilized at some times and they may be almost empty at some other times; 4) There may be unfair bandwidth allocation for well-behaved traffic sources. For example, consider a burst of traffic arrives at an ingress switch going to a particular egress switch, in the worst case it could be BE traffic. Then, the threshold-based aggregation nature of CTT generates a train of (malicious) slots going to that egress switch. These malicious slots are sent to OPS network, while well-behaved slots going to other egress switches are delayed. This issue becomes even worse when the malicious slots carry the BE traffic which would be experiencing even a lower queuing delay than higher priority traffic; and 5) CTT may not provide enough delay differentiation among traffic classes whenever the time-out event is inactivated due to a higher traffic arrival. In view of these, we would like to isolate traffic streams from different source routers, use a packet scheduler to provide packet differentiation, and design a new access technique to OPS network in order to resolve the time-out related problems.

Since packet differentiation using the time-out mechanism has the aforementioned problems, we would like to use packet scheduling in each OPS edge switch in order to provide packet differentiation, to schedule client packets from the source routers connected to an edge switch, and then to aggregate slots with the scheduled traffic.

However, the bandwidth given to a traffic going to a particular egress switch is a fraction of the network bandwidth that may not always be available because of the predetermined slot schedule in our access technique. Here, smoothness of packet scheduling has an important role in providing fairness among traffic sources[3]. Moreover, scheduling algorithms suffer from some network performance drawbacks in supporting only fixed packet lengths, supporting only single or double class traffic streams, higher service times, and a higher jitter and startup latency. This is why we would be interested in designing a new fair and smooth scheduler to deal with the aforementioned problems.

Finally, we are also interested in evaluating the suitability of our bandwidth management system in the AAPN network as a candidate for a metro network and compare it with the BvN centralized reservation-based protocol, which provides the best guarantee for traffic scheduling.

1.3 Objectives

We have a general interest in the bandwidth management of slotted all-optical packet-switched networks. Specifically, we are interested in

1. Bandwidth management in edge switches.
2. Bandwidth management in all-optical core switches.
3. Addressing the performance issues affecting the slotted all-optical packet-switched networks.
4. Developing algorithms to (1) and (2).

1.4 Approaches and Methodology

We consider a slotted all-optical buffer-less packet-switched network within the DiffServ domain. In this network, transmission is done in the optical domain while buffering at edge switches and logical operations are performed in the electronic domain. We shall not use any optical buffers for contention resolution in OPS core switches for cost

[3] For clarification, let us choose a non-smooth scheduler like DRR to schedule single-class packets from R source routers to a given egress switch. Under DRR, a bulk of packets is scheduled from each source router and filled in a slot. Hence, packets from Source-R may have access to the network bandwidth after several time-slots comparing to Source-1. This is unfair in bandwidth allocation among different source routers. Thus, a smooth scheduler must be used so that the same opportunity can be given to all source routers to send their packets in the provided bandwidth.

reasons. However, since our network is slotted, some fiber delay lines will be required for synchronization issue at the input ports of each core switch. To design the bandwidth management system, we would like first to provide a proper system model to the system under study from which we can carry out analysis and evaluate all important issues as the following.

We would like to use a relatively larger slot-size in our network. In general, by increasing the slot-size and the slot-offset interval proportionally, we expect an optical switch with a relatively higher switching speed (e.g., micro-seconds) can be used in an OPS core switch. Then, we shall rely on packet aggregation which will in turn reduce the number of entities that should be processed at the core switch. A larger slot-size will also resolve the fragmentation, slot dependency and bandwidth wastage problems.

In order to design a simple signaling protocol, an out-of-band (common channel) signaling technique would be adopted instead of the in-band signaling in slotted-OPS. By decoupling the data channels from the control channel, routing and forwarding functions are performed in the electronic domain (after converting header of slots from the optical to the electronic domain), while the slots are switched transparently in the optical domain.

We would like to design a new edge switch architecture with the following features/capabilities: 1) To provide fairness among the upstream source routers connected to the edge switch, an individual buffer would be allocated to the traffic of each class from any source router; 2) To provide differentiation among different DiffServ packets, packet scheduling would be used; 3) To provide access to OPS network, we would like to design a distributed TDM protocol in the ingress switch; and 4) To have a loss-free OPS network, the ingress switch would be able to manage the retransmission of the dropped traffic in the optical domain. In other words, our ingress switch design requires three protocols: packet scheduling, retransmission management, and bandwidth provisioning.

In order to support QoS in our network, we would like to design two mechanisms: 1) A new fair and smooth packet scheduler in ingress switches to provide differentiation among IP packets in the DiffServ domain; and 2) An "importance" parameter to each class so that a rank can be assigned to each slot. Based on this rank, a slot can pass through a core switch.

In order to obtain some ideas in designing our bandwidth access protocol, we would like to investigate and analyze various software-based and hardware-based contention avoidance schemes, and then evaluate the hardware-based schemes in terms of their cost and performance tradeoff. We would like to obtain a lower-bound for the loss rate, and show a new aspect in contention avoidance that is much suitable for single-hop networks. Note that the contention avoidance schemes may not solely be able to provide a desired loss rate in OPS network. Therefore, contention resolution schemes must also be used at the OPS core switches as well in order to reduce the loss rate to a desired value, but then at a lower volume and therefore a cheaper network design. In our network, we would like to rely on the inexpensive contention avoidance schemes more than the contention resolution schemes.

We would like to investigate the use of retransmission in the optical domain where less contention hardware in core switches can be used. We would provide a model to describe retransmission and analyze the performance of two retransmission techniques: Prioritized Retransmission (PR), and conventional retransmission. The PR technique is expected to limit the number of retransmissions.

We would like to design a distributed TDM protocol in order to resolve the aforementioned problems for the bandwidth access provisioning technique in all-optical OPS networks. Unlike CTT, we shall use a packet aggregation method that is neither threshold-based nor timer-based, instead we would want to distribute allocated slots to different egress switches through a frame period and among wavelengths and fibers. In doing so, we hope to achieve a smooth access to OPS network in.

We would also like to study the TDM resource-sharing schemes without any traffic loss in the AAPN network as an application. First, we would like to use the BvN scheduling as our reservation-based scheme. Second, we would like to use our distributed TDM protocol combined with our PR technique (both protocols together provides a loss-free OPS network). Third, we would design an integrated TDM resource-sharing scheme that is based on the good attributes of the aforementioned two schemes.

To carry out our performance evaluation and their tradeoff study, we would employ counting techniques, numerical methods, mean analysis, bound analysis and probabilistic techniques. We would study how different network parameters affect the performance

issues such as loss rate, jitter, startup latency, throughput and delay. Since most analysis is approximation, we would verify them by simulation. Of the few simulation packages, we would use OPNET [OPNE06] because of its modularity in designing simulation models and its capability in supporting *C* language to implement simulation codes (see Appendix H). We also use *C* programming language to calculate mathematical formulas, and to implement optimization problems. For all our simulations 95% confidence intervals are found to be within less than 5% of the mean values shown. Since these intervals are very small to show, we do not display them in our diagrams.

1.5 Contributions

The following are the contributions of this research to all-optical OPS networking:

1. An ingress switch model with new buffering and transmission protocols.
2. Design and analysis of a class-based fair packet scheduler in the DiffServ domain, called the OCGRR (Output Controlled Grant-based Round Robin), to provide packet differentiation in our edge switches.
3. Present a new contention avoidance technique called Coordinated Slot Transmission (CST) based on the lower-bound analysis.
4. An optimization design to achieve a cost-effective contention avoidance architecture by the proper choice of network parameters.
5. The exploration of the Prioritized Retransmission (PR) technique in slotted all-optical OPS networks and its performance analysis in order to limit the number of retransmissions and in order to increase the network performance.
6. A presentation of the even slot-distribution concept through the frame and among the output wavelengths/fibers of an ingress switch; A development of mathematical formulas, for the first time, to determine how even a distribution is; Design of the Distributed TDM protocol as a bandwidth access provisioning technique.
7. A design of an Integrated TDM (ITDM) protocol in a slotted-OPS network that combines the best features of DTDM and CTDM; A comparison of the DTDM, CTDM and ITDM protocols in the AAPN network.

To the best of our knowledge, the lower-bound analysis, the PR analysis, the formulation of the optimization problem in contention avoidance, and the DTDM

technique are the first time they are carried out.

1.6 Organization of Dissertation

The remainder of dissertation is organized as follows. Chapter 2 discusses the network modeling, definitions and general assumptions used in the rest of this dissertation. The ingress switch model there allows more specific components to be developed in the following chapters. Chapter 3 develops the OCGRR packet scheduling algorithm used for packet differentiation in our ingress switch model. In order to obtain some basic ideas in the design of the DTDM protocol, we discuss and analyze the contention avoidance issue in Chapter 4. Chapter 5 analyses the Prioritized Retransmission technique in a multi-fiber slotted all-optical OPS network. The prioritized retransmission is used in our ingress switch architecture in order to manage the retransmission of dropped slots at the OPS network. Chapter 6 designs the DTDM protocol that provides a smooth access to the optical network for traffic streams in our ingress switch model. Chapter 7 details the CTDM algorithm as well as our designed ITDM technique, and then compares the DTDM, CTDM and ITDM protocols in AAPN network. Chapter 8 provides some design guidelines related to our algorithms. Chapter 9 concludes the dissertation with a summary of its results.

1.7 Publications

The following is a list of publications related to this research:

Chapter 3:

[1] Akbar Ghaffar Pour Rahbar and Oliver Yang, "The Output-Controlled Round Robin Scheduling in Differentiated Services Edge Switches," *Proc. IEEE BROADNETS 2005*, Boston, USA, Oct. 2005, pp.237-245.

[2] Akbar Ghaffar Pour Rahbar and Oliver Yang, "OCGRR: A New Scheduling Algorithm for Differentiated Services Networks," accepted for publication in *IEEE Transactions on Parallel and Distributed Systems,* Aug. 2006.

Chapter 4:

[3] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Reducing Loss Rate in Slotted Optical Networks: A Lower Bound Analysis" *Proc. IEEE International Conf. on Communications (ICC)*, Istanbul, Turkey, Jun. 2006.

[4] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Contention Avoidance By Composite Slot Assembling," *Proc. IEEE/OSA Optical Fiber Comm. (OFC)*, Anaheim, CA, Mar. 2006.

[5] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Contention Avoidance in Slotted Optical Networks," *Proc. International conference on Optical Communication Systems and Networks, SPIE Photonics North*, vol.5970, Toronto, Canada, Sep.2005.
[6] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Slot Contention Resolution for Distributed TDM Scheduling in an Optical Star Network," *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE 2005)*, Saskatoon, Canada, May 2005, pp.286-289.
[7] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Contention Avoidance in Slotted All-Optical Packet Switching Networks", submitted to *Elsevier Computer Networks*, 2006.

Chapter 5:

[8] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Retransmission in Slotted Optical Networks," *Proc. IEEE High Performance Switching and Routing (HPSR),* Poznan, Poland, Jun.2006.
[9] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Prioritized Retransmission in Slotted All-Optical Packet Switching Networks," to appear in *OSA Journal of Optical Networking*, vol.5, no.12, Dec.2006.

Chapter 6:

[10] Akbar Ghaffar Pour Rahbar and Oliver Yang, "A New Bandwidth Access Framework in Slotted-OPS Networks," to appear in *Proc. IEEE Conference on Local Computer Networks (LCN)*, Tampa, Florida, USA, Nov.2006.
[11] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Ingress Switch Architectures in Slotted AAPN Network," *AAPN Annual Research Review*, Ottawa, Canada, June 2006.
[12] Mushi Jin, Akbar Ghaffar Pour Rahbar, and Oliver Yang, "Comparison Analysis of Two Scheduling Schemes in AAPN Networks," *AAPN Annual Research Review*, Ottawa, Canada, June 2006.
[13] M.Jin, O.Yang, Y.Zhang, A.G.P.Rahbar and W.Yang, "Time-Division-Multiplexing in Star-Based Optical Networks," *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE 2005)*, Saskatoon, Saskatchewan Canada, May 2005.
[14] Akbar Ghaffar Pour Rahbar, Mushi Jin, Choudhury A. Al Sayeed, and Oliver Yang, "APOSN under the MAN and WAN environments" *Proc. IEEE Canadian Conf. on Electrical and Computer Engineering (CCECE)*, Ottawa, Canada, May 2006, pp.1211-1214.
[15] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Distributed TDM Protocol in Slotted All-Optical Networks," submitted to *IEEE Journal on Selected Areas in Communications (JSAC) - Optical Communications and Networking (OCN) series*, 2006.

Chapter 7:

[16] Akbar Ghaffar Pour Rahbar and Oliver Yang, "An Integrated TDM Architecture for AAPN Networks," *Proc. International conference on Optical Communication Systems and Networks, SPIE Photonics North*, vol.5970, Toronto, Canada, Sep. 2005.

[17] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Distributed TDM vs. Centralized TDM Scheduling Algorithms," *2005 AAPN Annual Research Review*, Ottawa, Canada, Jun.2005.

[18] Akbar Ghaffar Pour Rahbar and Oliver Yang, "How Important is Scheduling in a Single-Hop Optical Network," *2nd Workshop on Optimization of Optical Networks (OON 2005),* Montreal, Canada, Apr. 2005.

[19] Akbar Ghaffar Pour Rahbar and Oliver Yang, "Agile Bandwidth Management Techniques in Slotted All-Optical Packet Interconnection Networks," *submitted to IEEE Transactions on Computers*

Chapter Two: Network Model, Definitions and Assumptions

We shall provide the general network model and operations of our all-optical packet-switched network with a slotted architecture for the design, analysis and performance evaluation carried out in later chapters. A new ingress switch architecture, packet handling mechanism, and traffic transmission protocol are also introduced for the slotted OPS architecture. These models are implemented in the OPNET simulator [OPNE06] for performance evaluation. Appendix H provides a summary of the simulator and the level of simulation of each chapter this dissertation.

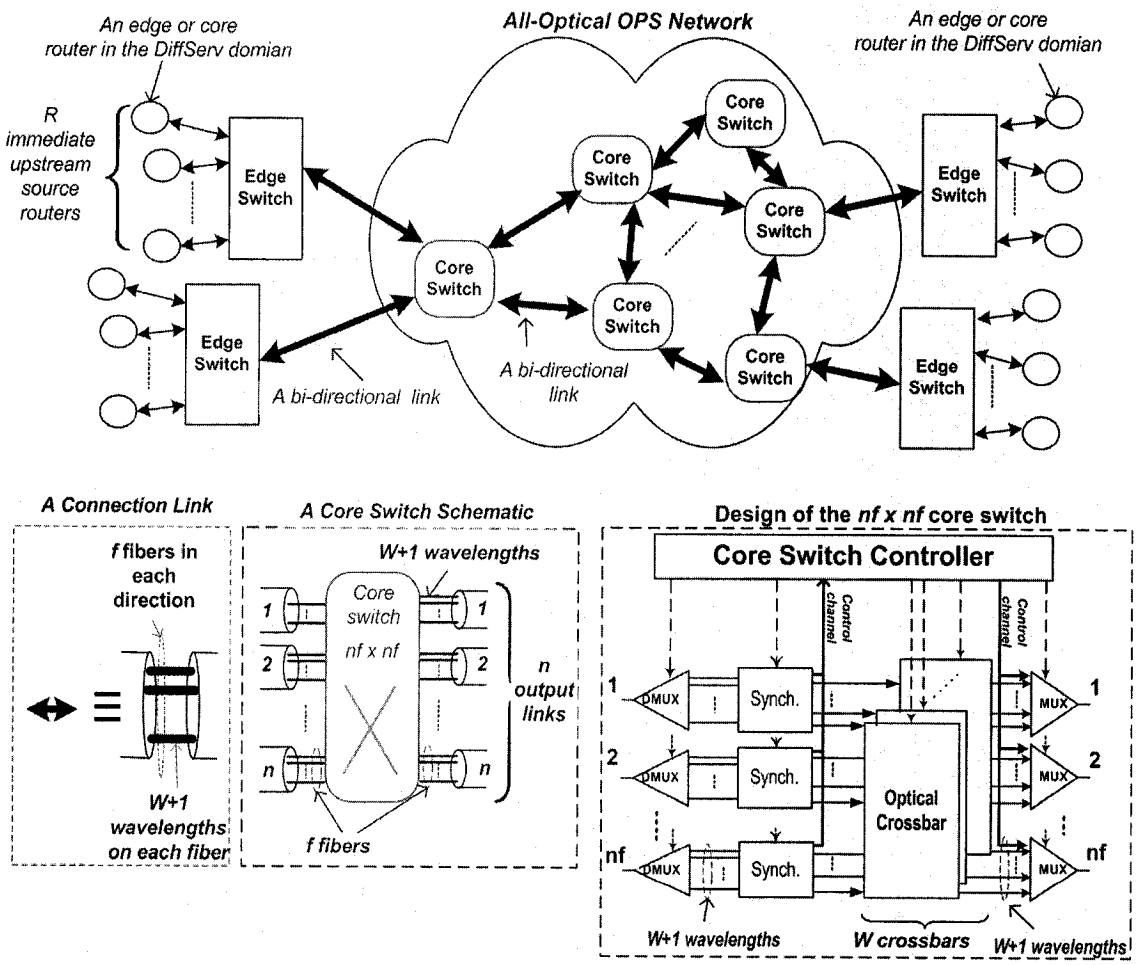


Fig.2.1: General Network Topology

2.1 Generic Network Layout

We consider a slotted buffer-less all-optical packet-switched backbone network (Fig.2.1)

in the DiffServ domain with a general mesh topology inter-connected with a number of non-blocking all-optical wavelength-selective cross-connect switches, where each all-optical switch is called a ***core switch***. The switching speed of a core switch should not be necessarily very fast, e.g., in the range of few nano-seconds. We can use acoustic-optic switch [Kart03, Wagn06] with the switching time in the range of a microsecond or even faster. Each core switch may be connected to a number of edge switches and a number of core switches. Fig.2.1 also shows a core switch schematic (the lower center diagram) and its design (the lower right diagram), connected to n input switches (either edge or core) and to n output switches (either edge or core) where there are f input (output) ports on each input (output) link in the core switch. The demultiplexer (DMUX) and multiplexer (MUX) components shown in the switch design are used to separate and aggregate wavelength channels on an input fiber and output fiber respectively.

The connection links shown in Fig.2.1 (detailed in the lower left diagram) are bi-directional and there are f fibers in each direction between any core switch-core switch pair or a core switch-edge switch pair. There are $W+1$ channels on each fiber. One of the channels is used for controlling purposes and W channels for data transmission/receiving purposes. Each channel has a bandwidth of B_C bps (all channels have the same bandwidth).

There are n_e+1 edge switches in the network. Each edge switch can transmit traffic to all other edge switches. A transmitting (receiving) edge switches is called an ***ingress*** (***egress***) switch respectively. Each edge switch uses $W+1$ fixed optical transmitters on each output fiber and $W+1$ fixed optical receivers on each input fiber. Each edge switch is physically connected to R immediate routers, where each router could be either an edge router or a core router in the DiffServ domain. An edge switch either aggregates traffic from R upstream routers (hereafter each is referred to as a ***source router***) or delivers received traffic to R downstream routers.

2.2 Traffic Model and Definitions

Our network follows the DiffServ model with ψ classes of traffic (e.g., ψ=3 for EF, AF and BE in the DiffServ model). Through this dissertation, we denote EF for the highest priority (i.e. class 1) traffic and BE for the lowest priority (i.e. class ψ) traffic. We define

an "importance" parameter V_i for class i that measures how valuable the traffic is in class i. We require V_i to have the property that $0<V_i<1$ such that $\sum_{i=1}^{\psi} V_i = 1$.

We define a ***stream*** to be the traffic of the same-class packets in the DiffServ domain from one particular source router and going to the same egress switch. Each one of the R source routers connected to an edge switch generates up to ψ classes of traffic to each one of the egress switches. Therefore, there are R streams in each class of traffic arriving at an ingress switch going to the same egress switch. Borrowing from [BrCl02], we define a ***torrent*** in an ingress switch to be the whole traffic going to the same egress switch. Therefore, the traffic of torrent i goes to egress switch i, and we may use torrent and egress switch interchangeably throughout this dissertation later. The bandwidth of fW wavelength channels in each ingress switch can be shared among n_e torrents.

We shall use the Poisson traffic to model smooth traffic arrivals. For the bursty traffic [PaFl95], we use a self-similarity model such as the Pareto distribution with the probability density function of $p(t) = ab^a t^{-a-1}$ to model the inter-arrival times of packets for each class.

2.3 Slotted Operation

Our backbone network uses the slotted operation in which time on each wavelength channel is divided into fixed-interval optical time-slots. Each time-slot with the duration of S_T time units is separated by a small time-gap (called slot-offset interval) of duration S_O. This gap includes guard time (for timing uncertainties), processing time at the core switch and switching time. Within each time-slot, an integer number of packets can be carried.

Each ingress switch can transmit traffic within a time-slot on any wavelength channel. For simplicity, the traffic carried within a time-slot is referred to as a ***slot*** from now on. One may also think a ***slot*** as a container to carry traffic within a time-slot. An ***empty slot*** is a slot with no traffic to any egress switch.

Considering that our slots are big enough, we define Composite Slot Assembly (CSA) to be the operation that would allow an integer number of IP packets from different DiffServ classes to be carried in one slot. There is no limit on the number of

packets from a particular class within a time-slot. Note that in the current CTT technique, e.g., [RaZa06, GoMa06], each slot must carry packets from the same traffic class[4] (referred to as non-CSA). Note that the variable length packets are aggregated in each slot as far as there is enough space. This issue avoids packet fragmentation, but there may be some packet aggregation overhead in each slot.

There are two types of bandwidth overhead in our network. The bandwidth overhead fraction due to the time-gap is $\frac{S_O}{S_O + S_T}$. By considering the average packet length of ℓ, the bandwidth overhead due to the packet aggregation in a slot is $\frac{(B_C S_T) \ mod \ \ell}{B_C(S_T + S_O)}$, where $B_C S_T$ is the maximum traffic in bits to be carried in a slot and $X \ mod \ Y$ means the remainder of X divided by Y. Then, the bandwidth wastage in a slotted network is $\frac{S_O}{S_O + S_T} + \frac{(B_C S_T) \ mod \ \ell}{B_C(S_T + S_O)}$. Therefore, the normalized input traffic load in an ingress switch cannot exceed $1 - \frac{S_O}{S_O + S_T} - \frac{(B_C S_T) \ mod \ \ell}{B_C(S_T + S_O)}$, and the available bandwidth B_a in each ingress switch is $B_a = fWB_c(1 - \frac{S_O}{S_O + S_T} - \frac{(B_C S_T) \ mod \ \ell}{B_C(S_T + S_O)})$.

In a slotted network, a core switch must receive all control information and slots at the same time. However, edge switches may have various propagation delays to the core switch. Thus, the slot boundary is provided for the synchronization purposes and edge switch i must start its first transmission (after initialization) at time $T_{s,i}$ given by $T_{s,i} = P_{D,i} - (S_O + S_T)\left\lceil \frac{P_{D,i}}{S_O + S_T} \right\rceil$, where $P_{D,i}$ is the one-way propagation delay from edge switch i to the core switch. In a multi-hop network, the fiber length between any pair of core switches must be cut in a way that the propagation delay between them becomes almost equal to an integer numbers of time-slot durations. In addition, the arrival of slots at a core switch may still be misaligned with one another due to chromatic dispersion and temperature variation that affect the propagation delay. To provide proper alignment, the synchronization hardware [SeBe96, TuZh99, YaMu00,

[4] We will show in Section 4.2.1.2 that CSA could have a better performance results than non-CSA in slotted OPS.

ChCo01, RaTu04] must be used at each input port of any core switch in the network (see the core switch design in Fig.2.1). The synchronizers are built with finely calibrated set of optical delay lines.

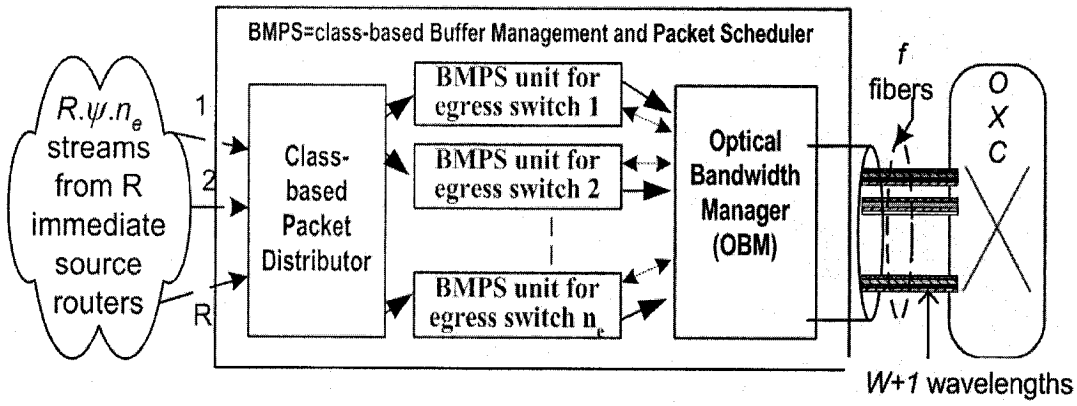


Fig.2.2: Ingress Switch Architecture

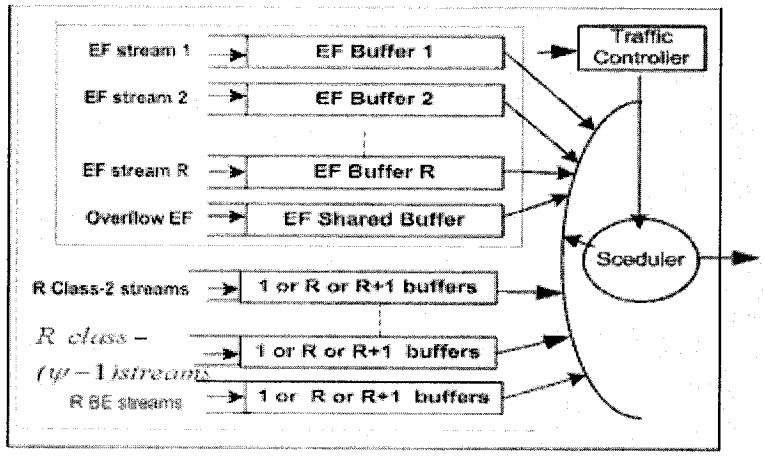


Fig.2.3: The BMPS Unit *j* Architecture for Egress Switch *j*

2.4 Ingress Switch Architecture

Fig.2.2 shows a general view of our designed ingress switch architecture (referred to as ***Arch2*** in this dissertation) in which $R\psi n_e$ streams are arriving at an ingress switch and are going to n_e egress switches (the egress switches are not shown in the figure). There are a class-based packet distributor unit, n_e class-based Buffer Management and Packet Scheduler (BMPS) units, and an Optical Bandwidth Manager (OBM) unit in the ingress switch. In this architecture, the functionality of the blocks BMPS and OBM and the communication between them are novel because of a new packet handling mechanism

and traffic transmission protocol used in these blocks respectively.

The class-based packet distributor unit includes the interfacing cards to the source routers and a switching module. The switching module receives streams from all source routers, classifies them, and then routes all streams going to egress switch j (j=1,2,...,n_e) to BMPS j, while saving the traffic belonging to a particular stream and class in its relevant buffer in BMPS j. There is a packet scheduler unit inside each BMPS unit for the sharing of bandwidth C and maintaining the transmission order of packets from R streams in each class to the relevant egress switch when required. Note that in *Arch2,* we adopt the CSA packet aggregation.

Figure 2.3 displays the BMPS unit j (showed in Fig.2.2), dedicated to egress switch j. The BMPS unit consists of three components: an OCGRR scheduler, a traffic controller, and sorting buffers. The OCGRR scheduler, to be designed in Chapter 3, determines the service order of packets from the streams and implements packet differentiation. The scheduler also cooperates, to be discussed in Section 6.3.3, with the OBM unit to fill the slots going to egress switch j. The traffic controller unit provides the information on the arrival rates of all streams and the average packet length among all streams to the OCGRR scheduler.

To fairly process streams, a dedicated buffer may be allocated to each stream within each class inside a BMPS unit. Therefore, R buffers are used for R streams in each class. Hence, the total number of buffers used in an ingress switch is $R\psi n_e$. A shared buffer may also be used for all traffic streams within each class so that the packet drop rate due to the burstiness of the traffic is minimized if desired. Here, overflow packets from buffer i dedicated to stream i within class J will be redirected to the shared buffer of class J. Consequently, the packets for stream i inside the shared buffer may be processed before the packets waiting in buffer i. This leads to packet mis-sequencing for the bursty stream i, and consequently a penalty. As a result, the network manager may selectively allocate 1, R, or $R+1$ buffers to other classes, depending on the QoS requirements. For example, we have used $R+1$ buffers for R EF streams and one shared EF buffer in a BMPS unit in Fig.2.3.

Note that in practice the number of source routers R connected to an edge switch and the number of egress switches n_e in an optical network are usually small. In addition, ψ is

a small number as well. Thus, the system is scalable in terms of buffer requirements. In our buffering, the same memory originally shared by the streams within a class in *Arch1* is partitioned into at most $R+1$ buffers, one for each stream plus the shared buffer. This buffering has no extra burden on the edge switch memory and can be managed by buffer management techniques, e.g., [KaGu98, SuLa99].

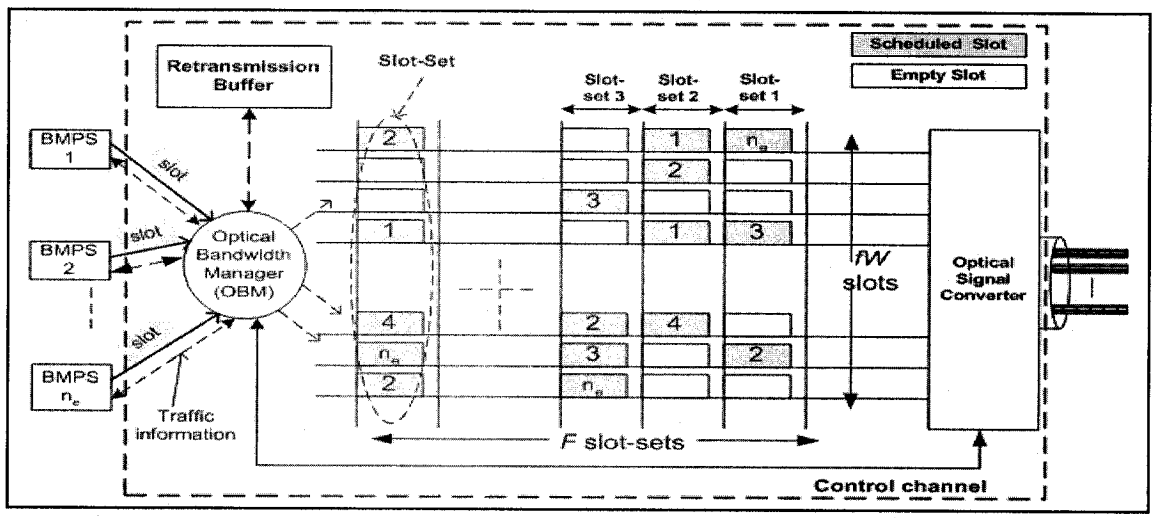


Fig.2.4: The OBM Unit in an Ingress Switch

Fig.2.4 shows the OBM unit in an ingress switch. OBM can transmit up to fW slots (called a ***slot-set***) at the same time to OPS network based on the schedule determined within a frame, where a ***frame*** is defined to be a collection of slots from all wavelengths on all fibers over a fixed duration of F slots-sets (see Fig.2.4). The slot transmission schedule is provided by the DTDM protocol to be designed in Chapter 6. The OBM unit can also manage the retransmission of dropped slots at the OPS network, to be detailed in Section 2.6 and Chapter 5.

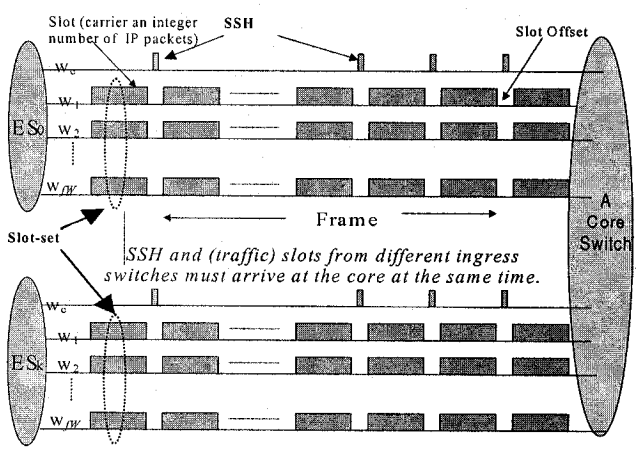


Fig.2.5: Control and Slot Transmission Protocol

2.5 Control Structure

For control purposes, signaling between each edge switch and a core switch must be provided by the OBM unit. Fig.2.5 shows the control and the slot transmission architecture for two sample edge switches (ES_0 and ES_k). Before transmitting the slot-set and during the slot-offset interval, each ingress switch makes a Slot-Set Header (SSH) for its *fW* slots and then sends it to the network over the control channel. Each SSH addresses traffic information for each slot in the slot-set, i.e., the ingress switch address, the egress switch address, the slot ID, the slot priority, and the information about the traffic that is carried in the slot (e.g., the number of EF and BE packets carried in the slot).

The slot ID and the slot priority generated by an ingress switch are used for retransmission. The ID has two parts: 1) A constant bit pattern that is unique in the network, e.g., the IP address of the edge switch; and 2) A variable part that is the slot number generated in the ingress switch. The ID is used for sequencing at the egress switches and retransmission at the ingress switches. The slot ID set for a slot is fixed and never changed during the transmission and all subsequent retransmissions. The number of the bits in the ID should be long enough, e.g., 32 bits, to avoid the problem of slot numbers wrapping around. The slot priority is used in Chapter 5 to implement the prioritization of dropped slots in the network.

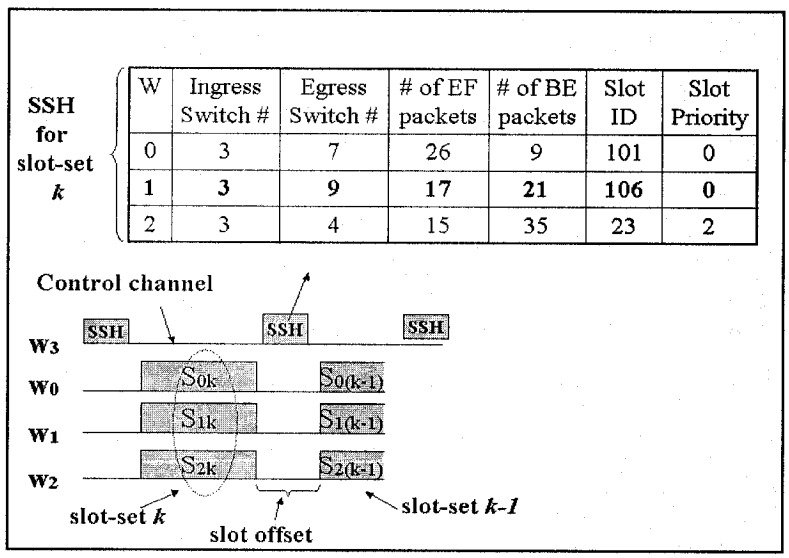


Fig.2.6: The SSH Structure Example

Fig.2.6 shows an example of the SSH structure accommodating just two classes of traffic: the EF and the BE traffic. The SSH for *slot-set k* includes the information related

to slots S_{0k}, S_{1k}, and S_{2k}. For example, the information in the second row (shown with bold font) shows that the slot to be arriving on wavelength channel w_1 is coming from ingress switch #3, is going to egress switch #9, and is carrying 17 EF packets and 21 BE packets. This slot has the ID of 106 and is a newly transmitted slot (because its priority is 0). As another example, the slot to be arriving on wavelength channel w_2 has been transmitted three times (because its priority is 2).

Each core switches receives an SSH from each one of its input ports. Then, it determines the switching scheme for the slots to be arriving based on the information in SSH. The set of nf slots on the same-wavelength from n ingress switches that simultaneously arrive at a core switch is referred to as ***nf-slot-set*** for $f \geq 1$. When f=1, we refer it to as ***n-slot-set.*** In a multi-hop network, a new SSH is rebuilt based on the slots departing from an output port and then transmitted on the control channel to the next hop core or egress switch.

2.6 Retransmission Model

As discussed, the OBM unit in our network can support retransmission of the dropped slots in the optical domain. To manage retransmission, the OBM unit in each ingress switch must save each transmitted slot for further retransmission in a retransmission (electronic) buffer (see Fig.2.4) implemented with link-lists. Whenever, a slot is dropped at a core switch in OPS network, a Negative Acknowledge (NACK) command is returned to the relevant ingress switch. Then the ingress switch retransmits the backup of the dropped slot to the network. The ingress switch will remove the backup slot from the retransmission buffer if it does not receive any slot drop information within a slot lifetime (say twice the propagation delay to the furthest core switch plus some processing time).

A core switch in the network first receives the information of slots at the same time in its input ports. After receiving a SSH and during the slot-offset interval, it evaluates the potential contention, then resolves the contention, and finally makes the core switch ready to switch the incoming slots toward their desired egress switches. For a successful slot transmission, no Acknowledge (ACK) command is required to be sent back to the related ingress switch. However, when a slot is dropped, the related slot ID is encapsulated in a Negative ACK (NACK) command and sent back to the relevant ingress switch over the control channel to identify the blocked slot. If several slots from ingress

switch i are dropped at a core switch, the core switch sends back the information of all dropped slots in a single NACK command to ingress switch i.

2.7 Assumptions

Unless specified otherwise, the following general assumptions pertain to the remainder of this dissertation.

1. Each edge switch can measure its propagation delay to the core switch at start up time.
2. The egress switch addresses carried for the slots on the same-wavelength and at the same time-slot are independent of one another, for reasons of tractable analysis.
3. Any core switch in the network receives all SSH headers from its input ports at the same time. This is also true for the slots.
4. There is no error in the transmission and receiving channels (both data and control) because of the very low bit error rate (less than 10^{-9} [Kart03]) generally expected in optical communication networks. Therefore, traffic data carried in time-slots as well as the NACK commands arrival in ingress switches are assumed to have no error.
5. No optical buffer is used in any core switch for the contention resolution purposes.
6. The wavelength conversion time is less than the slot-offset interval to guarantee on time wavelength conversion for contended slots. This is practical as we are using large slots and slot-offsets at the range of microseconds.
7. Average arrival rates of all streams and the average packet length among them are all known to the OCGRR packet scheduler in a BMPS unit.

Chapter Three: The OCGRR Packet Scheduling Algorithm

As discussed in Section 2.4, packet scheduling is required to schedule class-based traffic of streams in each BMPS unit of an ingress switch in order to make a slot ready for transmission to a given egress switch. In this chapter, we design a new fair scheduling technique, called OCGRR (Output Controlled Grant based Round Robin), for the support of DiffServ traffic. We first detail its general usage in any switch/router with output-queued buffering architecture before tailing it to our slotted network model. Performance evaluation is also provided.

3.1 The Logical Frame

We shall use the logical frame concept to summarize the operation of our OCGRR scheduler that would take advantages of the class-buffers in order to schedule traffic from streams inside a class. Recall that streams within each class are isolated from each other and saved in different buffers. Note that the ***logical frame*** definition is different from the ***frame*** definition mentioned in Section 2.4. In this chapter, we always use the "logical frame" term to avoid confusion.

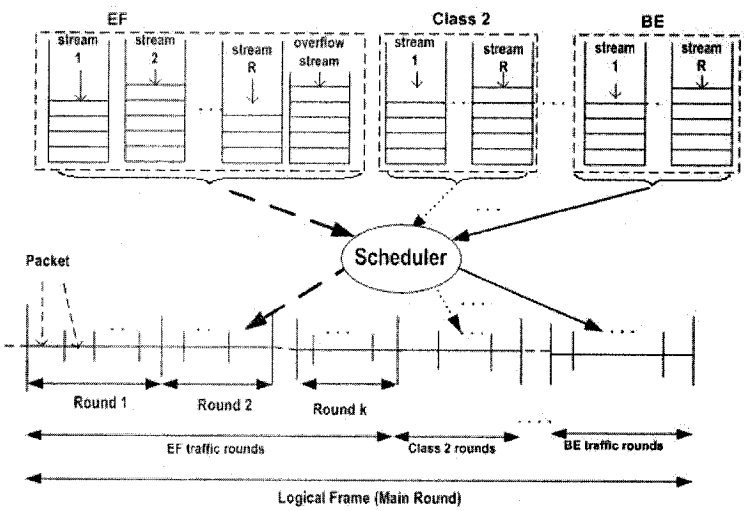


Fig.3.1: General Logical Frame Structure

The logical frame is the vehicle through which the OCGRR schedules the traffic streams. In each logical frame, the sequence of scheduling starts from the EF traffic (the

highest priority), then after processing some EF traffic it continues to Class-2 traffic, ..., and finally to the BE traffic. Each logical frame is divided into EF rounds, Class-2 rounds, .., and BE rounds (Fig.3.1).

Since we would like to limit the number of bits to be transmitted within a frame in order to provide a better service for higher priority traffic, we propose to use a variable Γ based on R, the average packet size[5] ℓ, and the class indices C_i, i.e.,

$$\Gamma = \ell(R+1)\sum_{i=1}^{\psi} C_i \quad , i = 1,2..,\psi \quad . \tag{3.1}$$

The term $R+1$ accounts for the possibility of allocating up to $R+1$ buffers in each class. Alternatively, the term R can also be used instead. The idea behind Γ while limiting the logical frame length is to allow $(R+1)\sum_{i=1}^{\psi} C_i$ worth of average-length packets to be scheduled from different classes within each logical frame. The class index of the highest priority class, EF, is 1 (i.e. $C_1 = 1$). For a lower priority class i, index C_i is a number less than 1. Each index can be increased up to 1, so that $1 = C_1 \geq C_2 \geq ... \geq C_\psi$.

Define a backlogged stream to be a stream with at least one packet in its relevant buffer and a positive grant (to be computed later). Otherwise, it shall be called a non-backlogged stream from now on[6].

Fig.3.1 shows the EF class with k rounds, where k depends on the grant of the backlogged stream and Γ. Within one particular round of each class, only the backlogged streams within that class can transmit, but each backlogged stream can only send one packet. A backlogged stream can transmit its next packets in the next rounds in that class. Each logical frame might include only the EF traffic if there is enough EF traffic qualified to fill the logical frame. Clearly, this may have a temporal negative impact on the loss and delay performances of lower priority traffic. Likewise, if there is no higher priority traffic, the logical frame can be filled with lower priority traffic only.

Note that a logical frame may end in two ways: 1) Whenever the transmitted traffic

---

[5] Average packet size in bits is measured by the edge switch as assumed, and therefore, it can be known a *priori* to OCGRR. The edge switch can measure it from packet arrival rate at intervals. The interval choice for the time sliding-window rate measurement technique is discussed in RFC2859. The interval can also be dynamically updated depending on the traffic arrival status e.g,[AgLe03].

[6] The remaining three combinations are: 1) empty buffer and negative/zero grant; 2) empty buffer and positive grant ; 3) non-empty buffer and negative/zero grant.

within the logical frame exceeds Γ. Here, the last packet of the logical frame may belong to any class (i.e. not necessarily the last class), and the actual logical frame length may be greater than Γ. Note that when a lower priority stream cannot transmit traffic in a frame due to the presence of higher priority traffic, its right to access to bandwidth is saved for future logical frames; and 2) Whenever there is no backlogged stream in any class, the actual logical frame length becomes smaller than Γ.

3.2 Output-Controlled Grant-Based Round Robin Scheduling

We shall detail OCGRR for class *J*, *J=EF,AF,..,BE*. Each major operation is further elaborated in the following subsections. Let us first define first a few parameters required in the algorithm.

1. **Grant Parameters:** To control each stream in accessing its bandwidth share, OCGRR assigns a grant (in bits) to each stream, and then based on the grant it schedules stream's traffic within a logical frame. There are two grant parameters for each stream in class *J*. The parameter $G_{J,i}(t)$ is the available grants for stream *i* at time *t*, and the parameter $U_{J,i}(t)$ is the total used-grants for stream *i* until time *t*. Positive $G_{J,i}$ means that stream *i* has underused its allocated grant while negative $G_{J,i}$ means that stream *i* has overused its allocation.
2. **Rate Parameters:** Define $\lambda_{J,i}$ to be the Average Arrival Rate (AAR) of stream-*i* packets in class *J*. OCGRR can also handle $\lambda_{J,i}$ if it is defined as a fair share or weight. The shared buffer has an AAR of λ_s to be updated based on the packets overflowed to the shared buffer. Note that the parameter $\lambda_{J,i}$ is a priori known to OCGRR independent of the type of interpretation.
3. **ActiveList:** To keep track of all backlogged streams within class *J*, we use a dedicated link-list for class *J*. Called *ActiveList J*, OCGRR uses it to schedule traffic within class *J*. When a stream not in *ActiveList J* becomes backlogged, its reference is appended to *ActiveList J*. Conversely, when the status of a stream is changed to non-backlogged, its reference is removed from *ActiveList J*. When a shared buffer is used for all *R* streams within class *J*, there is no need to maintain an *ActiveList* for class *J*.

3.2.1 The OCGRR Algorithm

The OCGRR algorithm is used in an output-queued switch architecture to schedule traffic

from different classes in a logical frame structure. For each class, traffic from different streams can be saved in a single buffer or can be separated in different buffers. OCGRR can be used for both buffering architectures. Traffic from different traffic streams can be transmitted whenever qualified that is determined by grant parameter. At the beginning of each logical frame, the grant parameter for each stream is updated. In addition, higher priority traffic is always first evaluated for scheduling, then the next priority streams.

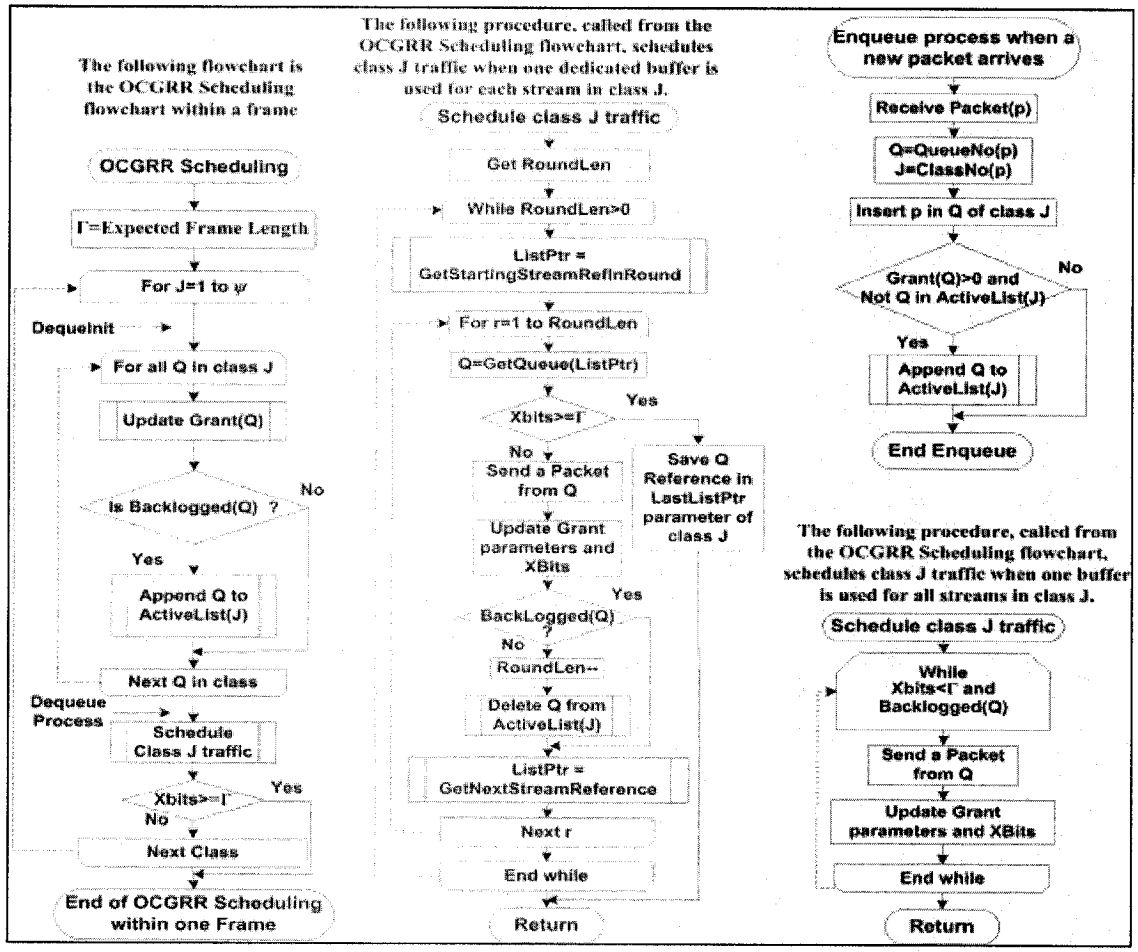


Fig.3.2: The Flowchart of the OCGRR Scheduling Algorithm

Fig.3.2 presents the OCGRR algorithm. This is expressed in flowcharts using the symbols defined in Microsoft Visio. We have determined the function of each flowchart in Fig.3.2. The parameter Q represents a stream, and *Xbits* is the total transmitted bits within a logical frame. The Enqueue process inserts each new packet of a stream in its relevant buffer, and then appends the stream to its class *ActiveList* (if the class has more than one buffer) provided that the stream not in the *ActiveList* becomes backlogged (had a

positive grant but was empty).

The scheduling for each class is divided into two parts: 1) In the DequeueInit process, the grant of each stream inside class J is incremented by some quantum computed based on the frame beginning time (see Section 3.2.2). Then, if a non-backlogged stream becomes backlogged, its reference is appended to *ActiveList J*; and 2) In the DequeueProcess, packet scheduling is performed. There are two scheduling processes: one for the classes that use a dedicated buffer per stream in a class, and the other for the classes that use one shared buffer for all streams in a class. The former case has the following steps:

Schedule Domain Determination: Define a schedule domain J to be the backlogged streams within class J before processing this class. This domain is controlled by the *RoundLen* parameter in Fig.3.2. The backlogged streams from the beginning of *ActiveList J* and from this domain can only transmit traffic during the current logical frame period, and the serving of newly backlogged streams during this period will be postponed to the next logical frame.

Traffic Transmission: When scheduling class J traffic within the logical frame, only the backlogged streams in class J can transmit traffic. Whenever a stream in this class becomes non-backlogged (defined earlier), the stream is removed from *ActiveList J*, and the schedule domain J becomes smaller. Thus, in the next round, a small number of streams will participate. In OCGRR, when a stream becomes non-backlogged, its grant parameters remain unchanged.

The scheduler visits the backlogged streams one by one. In each visit, only one packet is transmitted from a stream and then the next stream is visited. This continues until the last backlogged stream in the schedule domain J is visited. Then the scheduler starts visiting the backlogged streams from the head of *ActiveList J*. We call this scheme Multiple Round Robin (MRR) transmission because each stream may transmit its packets in multiple rounds, but only one packet in each round. Scheduling for class J is stopped whenever there is no backlogged stream left in *ActiveList J* or the total transmitted traffic exceeds Γ. The latter condition is evaluated at the end of each packet transmission.

OCGRR first schedules the packet with any size from the head of the stream. Then,

it updates both grant parameters of that stream with the length of the transmitted packet. This update process will lead to a negative grant if the size of the transmitted packet is greater than the stream's grant. When the grant of the stream becomes negative, the stream becomes non-backlogged.

If the logical frame ends right before processing stream *i* in class *J*, OCGRR flags this stream in the *LastListPtr J* parameter dedicated to class *J*. At a future logical frame when it is the turn of class *J* to be processed, scheduler would start processing class *J* from the stream referenced by *LastListPtr J*, but not from the beginning of the streams referenced by *ActiveList J*. This is handled by "*ListPtr=GetStartingStreamRefInRound*" in Fig.3.2. This process ensures fairness for stream *i* and reduces the inter-transmission time from the same stream.

In OCGRR, packets from the head of streams have almost the same chance to be transmitted under MRR because coherent transmission of packets from the same stream is reduced. Moreover, a packet in a newly backlogged stream encounters a lower latency in OCGRR. In addition, jitter among packets of the same stream is reduced in OCGRR. Note that by using MRR, the generated burst from the same stream is smoothened in that packets from the same stream are evenly distributed in the output because consecutive transmission of packets from the same stream is reduced.

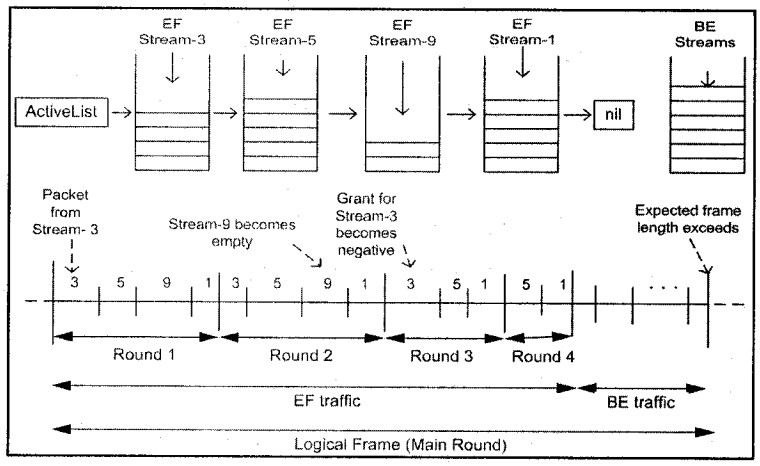
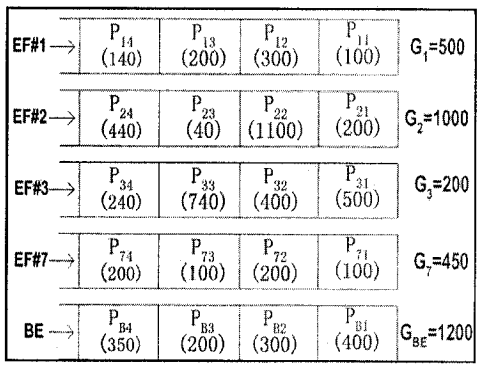


Fig.3.3: The MRR Example

Example 3.1: We shall provide a simple example of two classes to illustrate our algorithm. Fig.3.3 shows four EF backlogged streams in the *ActiveList* and a BE traffic that compete to access to the output bandwidth. In the first and the second rounds, each backlogged stream transmits one packet. Stream-9 becomes empty at the second round

and leaves the *ActiveList*. Assume the grant of Stream-3 becomes negative at the third round and does not participate in the fourth round. Suppose Streams-1 and 5 lose their grants at the fourth round. Then, the rest of the logical frame is filled with the BE traffic until the logical frame length exceeds Γ.



Round #	G_1	G_2	G_3	G_7	*Output sequence in OCGRR*	*Total bits*
1	400	800	-300	350	P_{11} , P_{21} , P_{31}, P_{71}	900
2	100	-300	-----	150	P_{12} , P_{22} , P_{72}	2500
3	-100	-----	-----	50	P_{13} , P_{73}	2800
4	-----	-----	-----	-150	P_{74}	3000
BE	G_{BE}=-50				P_{B1}, P_{B2}, P_{B3}, P_{B4}	4250

Fig.3.4: The Status of the Streams and Packet Transmission Order

Example 3.2: Fig.3.4 shows a two-class system buffer outline in which $P_{ij}(k)$ denotes packet j from stream i with size k bits and G_i is the available grant of stream i. The fifth buffer displays packets in the shared BE buffer. Let Γ be 4400 bits at the beginning of the frame. The table in this figure shows the grant values, the output sequences and the total transmitted bits at the end of each EF round. Stream-3 is removed from the *ActiveList* EF at the first round when G_3 becomes negative. Streams 2, 1 and 7 are removed from this list at the 2nd, 3rd and 4th rounds respectively. The transmission of the EF packets is finished at the 4th round with the total transmission of 3000 bits. The transmitted EF sequence is: $P_{11},P_{21},P_{31},P_{71},P_{12},P_{22},P_{72},P_{13},P_{73},P_{74}$. This sequence shows that packets from different streams have a fair access to the bandwidth via a smooth schedule. In comparison, the DRR [ShVa96] output sequence with the same grants would be: $P_{11},P_{12},P_{21},P_{71},P_{72},P_{73}$, which is not a smooth packet transmission because there are burst of packet transmission from the same stream (Stream-2 and Stream-7). After finishing the EF packets, four packets from the BE buffer are sent until the BE grant becomes negative, and the frame ends before exceeding the frame length than Γ.

3.2.2 Grant Calculation

Before processing the streams within class J, each stream in class J obtains some quantum proportional to its AAR. Let t denote the starting time of a new logical frame.

The time variable t is a virtual time that can be measured from any convenient reference point like when the edge switch powers up and everything is reset. This virtual time is the same for all streams of all classes. Hence, only one virtual time is enough to manage the bandwidth of the streams. At time t, the total given grant to stream i, *i.e.*, $TotalGrant_i(t)$, must be proportional to its AAR,

$$\frac{TotalGrant_i(t)}{t} = \frac{U_{J,i}(t) + G_{J,i}(t)}{t} = K\lambda_{J,i} \quad , \tag{3.2}$$

where the numerator is the total given grant to stream i until time t. Although the parameter K could be set as a constant value without considering the traffic arrival rate and available bandwidth, say K=1, we determine K by the following heuristic parameter

$$K = \frac{C}{\sum_{j=1}^{\psi} C_j \sum_{i=1}^{R} \lambda_{ji}} \quad , \tag{3.3}$$

The parameter K is used to adjust the allocated quantum according to the output bandwidth C, the class indices, and the AAR values. If the average arrival rates remain unchanged, K is constant. This parameter can therefore control the transmission policy from different classes. For example, in an edge switch with $\psi=3$, C=100Mb/s, $C_1=1$, $C_2=0.8$, $C_3=0.5$, $\lambda_1=10Mb/s$, $\lambda_2=30Mb/s$ and $\lambda_3=60Mb/s$, we have K=1.56. Consider a second case with $C_1=1$, $C_2=0.9$, and $C_3=0.7$, one obtains K=1.26. Since R and ℓ are the same for both cases, the parameter Γ (the parameter K) in the first case is smaller (higher) than the second case. Hence, the service of higher priority streams is much better in the first case. Lower traffic loads also lead to a higher K, and streams can use the unused bandwidth flexibly. Now, let us increase class arrivals to $\lambda_1=50Mb/s$, $\lambda_2=70Mb/s$ and $\lambda_3=80Mb/s$ and keep the other parameters as before. Then, we obtain K=0.68 and K=0.59 in the first and second cases respectively. This gives a chance to all classes to have access to the bandwidth. If we keep K=1, class 3 will unlikely be served at all. Thus, by controlling K, the starvation of lower priority traffic can be avoided in a congested network.

At time t, stream i should obtain some quantum of $Q_{J,i}(t)$ for the new logical frame until the end of the logical frame, $t+T_f$, so that the total given grant to stream i in class J at time $t+T_f$ still satisfies Eq.(3.2), i.e.,

$$\frac{TotalGrant_i(t+T_f)}{t+T_f} = \frac{U_{J,i}(t)+G_{J,i}(t)+Q_{J,i}(t)}{t+T_f} = K\lambda_{J,i} \quad , \tag{3.4}$$

where T_f is the expected logical frame transmission time obtained by

$$T_f = \frac{\Gamma}{C} = \frac{\sum_{i=1}^{w} C_i(R+1)\ell}{C} \quad . \tag{3.5}$$

Since the exact logical frame period is unknown at the beginning of the logical frame, T_f is taken as an estimate for the duration of the logical frame period. Rearranging Eq.(3.4), we obtain

$$Q_{J,i}(t) = \lambda_{J,i} K (t+T_f) - (G_{J,i}(t) + U_{J,i}(t)) \quad . \tag{3.6}$$

Then, the available grant for stream i can be calculated from $G_{J,i}(t)=G_{J,i}(t)+Q_{J,i}(t)$. Now if $G_{J,i}(t)$ becomes positive and the stream is non-empty, the stream reference is appended to *ActiveList J*.

3.2.3 Various OCGRR Operation Issues

Some issues must be pointed out here for the OCGRR algorithm:

1. To avoid the wraparound problem in the virtual time and used-grant parameters, the virtual time t must be reset to zero at $t=t_l$. Since all calculations are done in a similar fashion independent of class i, we drop the subscripts (J,i) in Eq.(3.6) so that $Q(t_l)=\lambda K(t_l+T_f)-(G(t_l)+U(t_l))$ at $t=t_l=0^-$ before resetting. Similarly, we have $Q(0^+)=\lambda K(0^++T_f)-(G(0^+)+U(0^+))$ at time $t=0^+=0$. Since all calculations are dependent on the quantum value $Q(t)$, the process of the quantum allocation to any stream must remain the same before and after $t=t_l$. Due to the continuity of $Q(t)$ and $G(t)$ at $t=0$ we must have $Q(t_l)=Q(0^+)$ and $G(t_l)=G(0^+)$. Then, by equating the above mentioned equations, we obtain $U(0^+)=U(t_l)-K\lambda t_l$. In other words, in order to keep the quantum unchanged, the used-grant parameter of any stream must change to $U(t)=U(t)-\lambda K t_1$ according to the relationships from $Q(t_l)$ and $Q(0^+)$. All other parameters will remain unchanged.
2. Assume at the beginning of a logical frame at $t=t_2$, the parameter K has already changed due to the change in the class indices or in the AAR of a stream. Then, the

grant parameters of each stream in any class can be reset to $U(t)= K\lambda t$ and $G(t)=0$ in order to start quantum allocation with the new K parameter.

3. OCGRR needs at most $(R+1)\psi$ buffers, $2R\psi$ grant counters, ψ *ActiveLists,* and ψ *LastListPtrs*; all can be static memory rather than dynamic because R and ψ are small in practice.
4. When scheduling a stream, OCGRR only sends one packet from the stream. Then, a small packet from another stream may wait for the transmission of the current packet, which probably may be a large-size packet. If this is a problem, instead of sending one packet, each stream can send a number of packets until the scheduled bits exceed ℓ. This approach, however, will increase jitter and startup latency.
5. When the scheduler is used in a slotted system to fill fixed-size slots, the logical frame length determined in Eq.(3.1) is set to $\Gamma = S$ where S is the slot size in bits. Moreover, the scheduling algorithm always keeps the scheduled traffic size to be less than or equal to Γ.

3.3 Analysis

We have carried out the mathematical analysis of OCGRR fairness and latency bound. Lemmas 3.1 to 3.4 state some basic lemmas that will be used in Theorems 3.1 and 3.2 in order to determine fairness and latency bound of the algorithm respectively. Theorem 3.3 provides the per-packet work complexity of the algorithm. The smallest traffic rate among all classes is denoted by λ_{min}. Since lemmas and theorems are true for each class, we remove the class index from the parameters for simplicity. In the analysis, we assume that packet lengths are i.i.d exponentially distributed with mean ℓ. The maximum packet length is also assumed to be $L_{max}=30\ell$ and equal for all classes. This gives a very small probability (less than $e^{-30\ell/\ell}\approx 9.4\text{x}10^{-14}$) for a packet length to be longer than L_{max}. Let $T_{f,max}$ denote the maximum logical frame period bound.

Lemma 3.1: The total amount of the acquired quantum Q_T during the period $[t_s , t_e]$, where t_s and t_e are the beginning of the two logical frames, for a given stream with rate λ, is in the range of $Q_T \in [K\lambda(t_e - t_s) - K\lambda T_{f,max}, K\lambda(t_e - t_s) + K\lambda T_{f,max}]$. This lemma will be used in Theorem 3.1.

Proof: See Appendix B.1.

Lemma 3.2: The Maximum logical frame period is upper-bound by $T_{f,max} \le \frac{L_{max}}{K\lambda_{min}}(0.1+\frac{1}{R+1})$ when $1+\frac{1}{R} \le \sum_{j=1}^{\psi} C_j \le 3$. This lemma will be used in Theorem 3.1 and Theorem 3.2.

Proof: See Appendix B.2.

Lemma 3.3: The minimum grant, G_{min}, at the end of a logical frame is *1-*L_{max}. This lemma will be used in Theorem 3.1.

Proof: See Appendix B.3.

Lemma 3.4: The maximum grant G_{max} for a well-behaved stream is $\frac{K}{C}\lambda L_{max}+2K\lambda T_{f,max}$ if $K \le 1$. This lemma will be used in Theorem 3.1.

Proof: See Appendix B.4.

Theorem 3.1: The OCGRR scheduling algorithm is fair.

Proof: See Appendix B.5.

Theorem 3.2: The startup latency bound $\hat{L}$ in a single-class system in OCGRR is $\hat{L} \le \frac{L_{max}}{C}(\frac{\sum_{j=1}^{R}\lambda_j}{\lambda_{min}}+\frac{31R+1}{30})$.

Proof: See Appendix B.6.

Corollary 3.1: By increasing R, the latency is increased, and bounds of DRR and OCGRR become closer to each other.

Theorem 3.3: The per-packet work complexity[7] of OCGRR is *O(1)*.

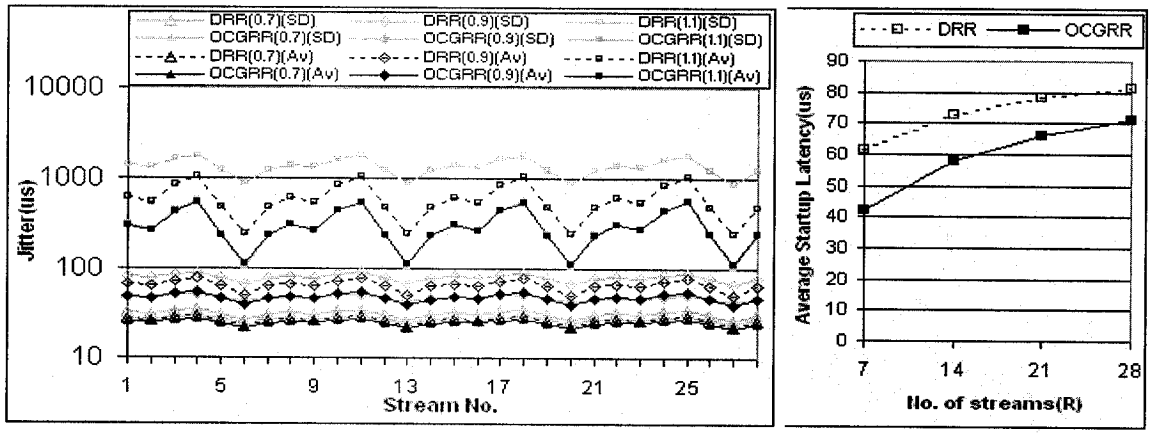
Proof: See Appendix B.7.

3.4 Performance Evaluation

We compare the performance of OCGRR with DRR (see Appendix C.1), DRR+ (see Appendix C.2), DRR++ (see Appendix C.3), and PQWRR (see Appendix C.4) under different classes in terms of queuing delay, traffic loss, jitter, and startup latency in the first three sub-sections. The operation of these algorithms has been summarized in

[7] The *work* is defined as the maximum of the time complexities to en-queue or de-queue a packet [ShVa96].

Appendices C.1 to C.4 respectively. Our simulation model for this section is shown in Fig.2.3. In the following, L_p is the normalized traffic arrival load (arrival rate of all classes normalized with respect to the service rate) in the scheduler. Unless otherwise mentioned, we have C=100Mb/s, L_{max}=10240 bits, and packet lengths are exponentially distributed with mean 1024bits. We give AAR (defined in Section 3.2) for each stream based on traffic load L_p =1.0, and AAR in other loads is adjusted accordingly. For example, if AAR for a stream is given as 5Mbits/sec (i.e. at load L_p =1.0), the AAR for this stream is 3Mbit/ses at L_p =0.6. To model bursty traffic [PaFl95], packet inter-arrival times for each class are i.i.d distributed according to a Pareto distribution with a p.d.f (probability density function) $p(t) = \alpha b^{\alpha} t^{-\alpha-1}$. We achieve a Hurst parameter of (3–α)/2=0.85 by using α=1.3. We have used OPNET [OPNE06] to develop our simulation models; and 95% confidence intervals are found to be within 1%(5%) of the mean values shown under Poisson (Pareto) traffic. More than ten million packets are simulated in each simulation replication. We use a Pentium IV, 2.4Ghz computer with 1GB memory to run our simulations.



a) Average Jitter and Standard Deviation Jitter at R=28

b) Startup Latency vs. R at L=0.9

Fig.3.5: Average Jitter and Latency in DRR and OCGRR in a Single Class System

3.4.1 Single-Class Performance Evaluation

We compare the jitter and startup latency performance of OCGRR and DRR under Poisson traffic in a single class network. Note that DRR+ and DRR++ reduce to DRR in a single class network. We use a dedicated buffer with size 100kbits to each of the R streams in this section.

To study how the schedulers treat similar-rate streams, we divide the streams into four groups of seven rates. The rates within each group are 2.92, 3.33, 2.08, 1.67, 3.75, 7.5 and 3.75 Mb/s respectively. For example, the rate of streams 1, 8, 15 and 22 are all equal to 2.92Mb/s. Fig.3.5a compares the average jitter (black curves) and the standard deviation of jitter (gray curves) among R=28 streams at L_p=0.7, L_p=0.9, and L_p=1.1. The jitter for two consecutive scheduled packets from the same stream is defined as $|d\text{-}d'|$ where d and d' are the queuing delays of the packets. OCGRR has always a lower average jitter and a lower standard deviation of jitter than DRR. At L_p=0.7, the difference between the jitter characteristics of the two algorithms is smaller. However, there is a significant difference between OCGRR and DRR at L_p =1.1. One can see a periodic behavior for the jitter. This is obviously due to the four groups of similar-rates used. Streams 4, 11, 18 and 25 have the smallest AAR of 1.67Mb/s and all experience almost the same and the maximum jitter. Conversely, streams 6, 13, 20 and 27 with the highest AAR all encounter the smallest jitter.

We also study the startup latency versus the number of streams. The AAR of streams are in the ranges [6.67Mb/s, 30Mb/s], [3.33 Mb/s, 15 Mb/s], [2.22Mb/s, 10Mb/s], and [1.67Mb/s, 7.5Mb/s] for R=7, R=14, R=21, and R=28 respectively. Fig.3.5b compares the startup latency at L_p=0.9. By increasing R, the latency is also increased. This confirms the analytical results obtained in Theorem 3.2. The latency is smaller for OCGRR than DRR, and the difference between them is higher at R=7 streams than R=28 streams. Note that the number of source routers (R) connected to an edge switch is small, and OCGRR is better for smaller R.

3.4.2 Two-Class Performance Evaluation

We compare next the performance of OCGRR with DRR+ and DRR++ in a two-class network that DRR+ and DRR++ are mainly designed for. However, to have a meaningful comparison between OCGRR and DRR+/DRR++, we have extended both DRR+ and DRR++ to include class index (to provide a better performance for the EF traffic). We modify DRR+ to MDRR+ (Modified DRR+) in which 1) One EF quantum [ShVa96] is defined as $Q_{EF,i} = \lambda_{EF,i} L_{max} / \lambda_{min}$, where λ_{min} is the smallest rate among all streams in all classes. However, one BE quantum is given by $Q_{BE} = C_{BE} \lambda_{BE} L_{max} / \lambda_{min}$, thus leading to a

better service for EF class; 2) The period T (see Appendix C.2) is given by $T = \sum_{i=1}^{R} Q_{EF,i} / C$ enough to service one quantum from each EF stream; and 3) Since it is difficult for a stream to conform its contract [MaSh00], MDRR+ is allowed to transmit traffic from EF stream i up to $Q_{EF,i} + L_{max}$ bits within T and stream i still is treated as an EF stream.

We also modify DRR++ to MDRR++ (Modified DRR++) in which 1) To resolve the head of line blocking in the priority transmission queue *(PTQ) (*see Appendix *C.3),* we increase the service time complexity of DRR++ to *O(n),* where *n* is the *PTQ* size, so that *PTQ* is linearly searched for an eligible EF packet for transmission; and 2) Quantum calculation is the same as MDRR+.

In all simulation scenarios in this section, we save all R BE streams in only one buffer with size 1.6Mbits. In both OCGRR and MDRR+, we use a dedicated buffer of 100kbits to each EF stream and also a shared buffer of 800kbits. For MDRR++, the *PTQ* size is set to 3.6Mbits (equivalent buffer size for the EF traffic in OCGRR and MDRR+). We set C_2 =0.5 for all algorithms.

A) Studying Poisson Traffic: Consider an edge switch with R=28 sources with the following AARs: $\sum_{i=1}^{R} \lambda_{EF,i} = 60\text{Mb/s}$, $\sum_{i=1}^{R} \lambda_{BE,i} = 40\text{Mb/s}$, where $\lambda_{EF,i}, \lambda_{BE,i} \in [1\text{Mb/s}, 4.5\text{Mb/s}]$.

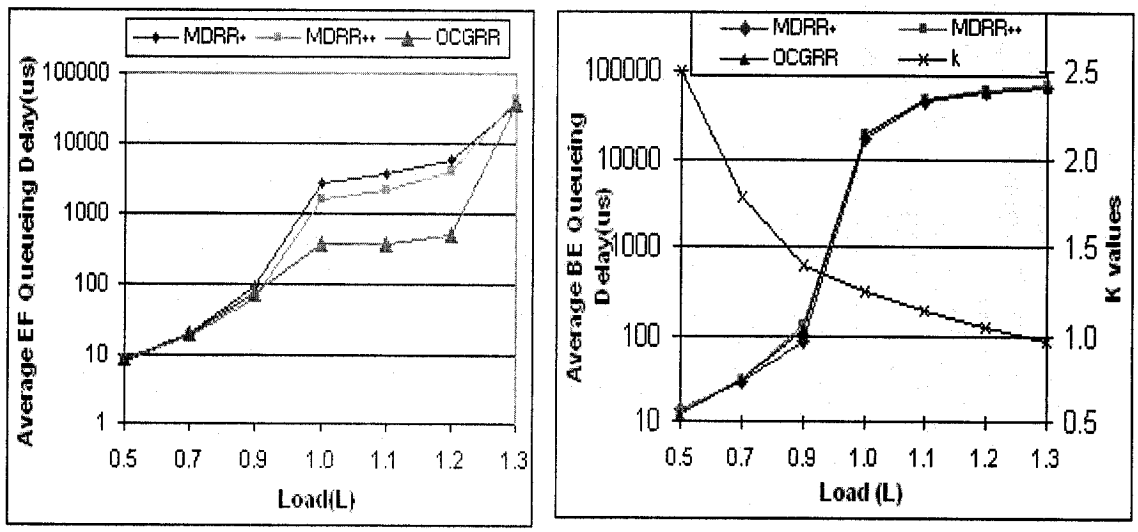


Fig.3.6: OCGRR / MDRR+ and MDRR++ Delay under Poisson Traffic

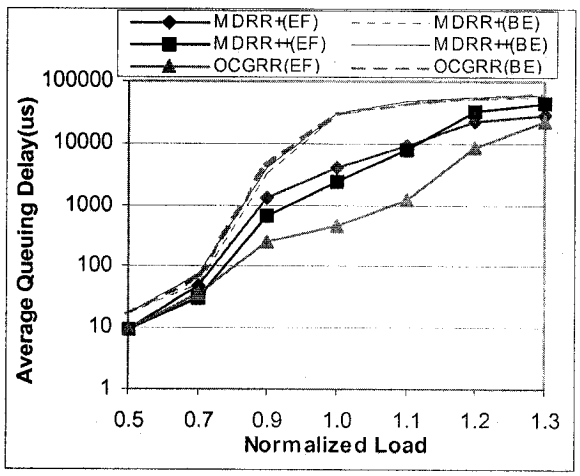
Fig.3.6 shows the EF and BE traffic average queuing delays (in a logarithmic scale) and K under Poisson traffic. For all loads, MDRR+ has the smallest BE delay followed by OCGRR, and MDRR++. For L_p<1, MDRR++ has the smallest EF delay and MDRR+ has the highest, and OCGRR lies in between. Note the higher service complexity ($O(n)$) of MDRR++.

For $1 \leq L_p \leq 1.2$, the input traffic exceeds the link bandwidth, however, we still have K >1 and the EF traffic can be served with a higher intensity than the BE traffic. Therefore, a sharp increase can be observed for the BE delay. Since one quantum worth of traffic from each EF stream must be served in MDRR+ and MDRR++, the round robin length increases for these algorithms at higher loads. However, Γ in OCGRR is independent of load and smaller Γ provides more chances to serve EF streams by the MRR manner, which is more significant at higher loads. The EF delay of OCGRR for L_p=1.0, L_p=1.1 and L_p=1.2 are 360μs, 380μs and 491 μs respectively. Since K becomes very close to 1.0 at L_p=1.2, there is a noticeable increase in EF delay from L_p=1.1. The EF delay for MDRR++ and MDRR+ vary in ranges [1.5ms, 4ms] and [2.7ms, 5.7ms] respectively. Since K<1 for L_p=1.3, the EF traffic is served with a lower intensity than the previous cases. Thus, a rapid increase is observed for the EF delay in OCGRR, and other algorithms. EF Delay for MDRR+, MDRR++ and OCGRR are 36.1ms, 44.3ms, and 35.2ms respectively. Due to the decrease in serving of EF traffic, BE delay has a smoother jump comparing to L_p=1.2.

The difference between the delay of DRR+/DRR++ and OCGRR is not just a few msec improvement, but the percentage change is quite big. For example, EF delay in DRR+ is almost 8 times of EF delay in OCGRR at L=1.0. By reducing C, the level of EF delays and the difference will increase. Our further evaluations show that the difference between EF delay of DRR+ and OCGRR is almost 0.3sec at L=1.0 and C=1.0Mbits/s, and the level of importance is obvious.

B) Studying Bursty Traffic: We use the same simulation setup as before, except all 28 EF and 28 BE streams are now bursty. Fig.3.7a shows the average delay of both EF and BE traffic for the 3 algorithms. Notice that the delays are higher than the Poisson case. This is because many packets may arrive suddenly at a stream, but they will only be processed gradually. At L_p=0.5, the EF delay is almost the same for the three algorithms. At L_p=0.7,

MDRR++ has the smallest EF delay followed by OCGRR, and MDRR+. However, the EF delay under OCGRR is significantly lower than the other algorithms at $L_p \geq 0.9$. The performance of BE traffic is about the same among the 3 algorithms although MDRR+ has almost the smallest BE delay and OCGRR lies between MDRR+ and MDRR++.



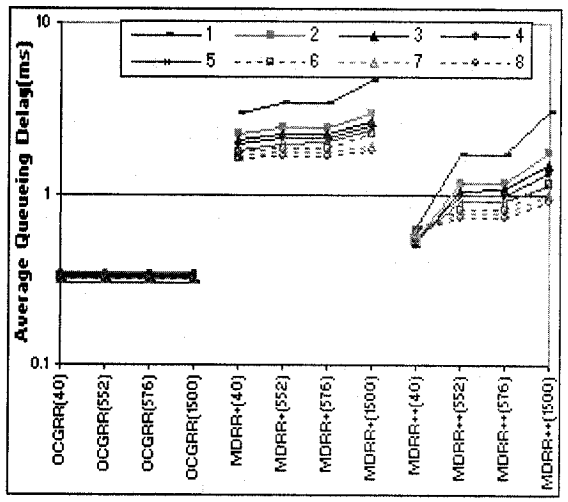
a) Average Queuing Delay

L	MDRR+		MDRR++		OCGRR	
	EF	BE	EF	BE	EF	BE
0.5	0.00	0.00	0.00	0.00	0.00	0.00
0.7	0.00	0.00	0.00	0.00	0.00	0.00
0.9	0.00	0.05	0.00	0.05	0.00	0.07
1.0	0.00	9.83	0.00	9.50	0.00	9.72
1.1	0.01	29.00	0.00	29.46	0.00	28.57
1.2	2.38	43.19	2.05	42.50	0.40	45.19
1.3	7.86	49.20	8.52	48.73	6.52	53.51

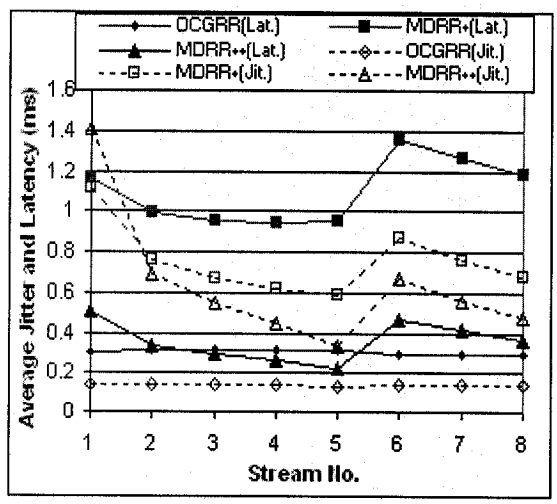
b) Average Traffic Loss Percentage

Fig.3.7: Comparison of OCGRR / MDRR+ and MDRR++ in Bursty Traffic

Fig.3.7b shows the corresponding average traffic loss percentage. The EF traffic loss becomes significant for $L_p \geq 1.0$, while loss for the BE traffic is observed as early as L_p=0.9. Overall, OCGRR has a significant improvement in EF percentage loss (e.g., from 2.38% to 0.4% at L_p=1.2) while sacrificing for a small increased in BE loss rate.



a) Average queuing delay



b) Average jitter delay and startup latency

Fig.3.8: Performance under Different Packet Sizes and Arrival Processes at L_p=0.9.

C) Studying Different Packet Sizes: We want to study how the algorithms perform under different packet sizes. Instead of the Poison distribution, each stream follows the packet size distribution as measured in [CAID06]: 40B (Bytes) (46%), 552B(18%), 576B(18%), and 1500B(18%), with a maximum size of L_{max}=1500B and a mean length $\ell \approx$ 3931bits. The rates of EF streams 1 to 8 are 2, 4, 5, 6, 8, 4, 5, 6 Mbit/s respectively. Streams 1 to 5 and 6 to 8 follow bursty and Poisson processes respectively. All BE streams are bursty and $\sum_{i=1}^{R} \lambda_{BE,i} = 60$Mb/s. Fig.3.8a displays the average EF queuing delay performance in each of the 8 streams at L_p=0.9. Each cluster represents the performance of different packet sizes of one algorithm. The observations are:

1) Delays for all packet sizes are almost the same in OCGRR. However, MDRR+ and MDRR++ perform differently according to different packet sizes. The 40byte packets experience the smallest delay while the 1500-bytes has the longest. On the other hand, for any packet size, OCGRR has the smallest delay followed by MDRR++ and MDRR+.
2) Packet delay of OCGRR is an increasing function of AAR so that stream-5 and stream-1 has the largest and smallest delays respectively. However, this increase is not significant. Thus, each stream has almost the same chance to transmit traffic independent of its AAR. This is not the case in MDRR++ and MDRR+ where the delay is significantly reduced by increasing AAR.
3) For the streams with the same-rate, e.g., Stream–2 and Stream–6, delay is higher for the bursty ones, e.g., Stream–2, except for the 40byte packets in MDRR++.

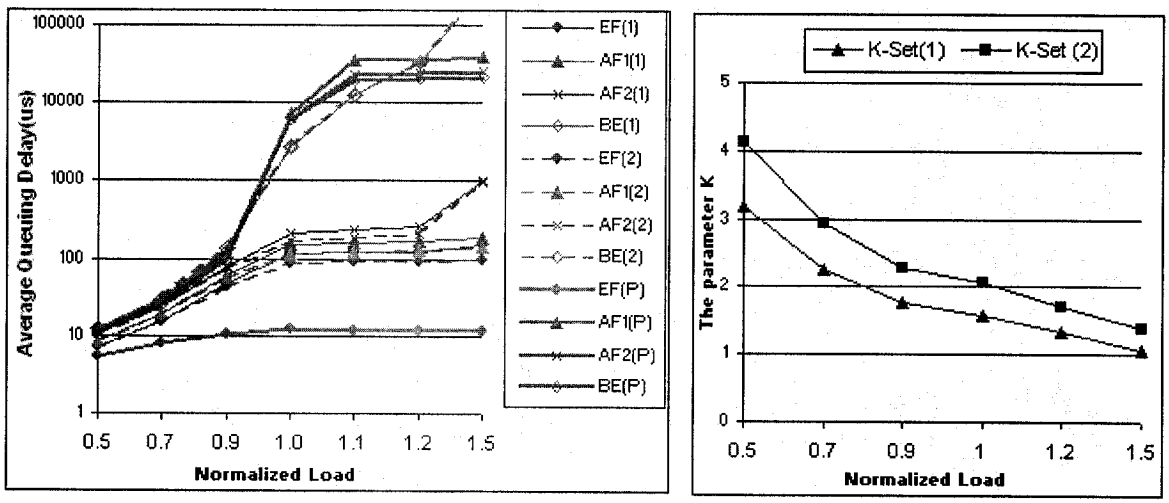
Fig.3.8b shows average EF jitter and startup latency at L_p=0.9 for eight streams. In OCGRR, jitter is almost the same for both bursty (Streams 1-5) and non-bursty streams (Streams 6-8). For other schedulers, jitter (the dotted lines) is reduced by increasing AAR for both bursty and non-bursty streams. In addition, comparing the two streams with the same-rate (e.g., Stream-2 vs. Stream-6) shows that the bursty stream has a better jitter than the non-bursty one.

In OCGRR, the latency is almost the same for the bursty streams and also for the non-bursty streams, but it is a bit higher for the bursty streams than the non-bursty ones. Latency, however, is reduced by increasing AAR in MDRR+ and MDRR++. Moreover,

for two same-rate streams, the bursty stream has a lower latency than the non-bursty one. Latency for the bursty streams is mostly lower under MDRR++ than OCGRR because packets of higher-rate bursty streams may be mostly at the head of *PTQ*, and MDRR++ will serve them sooner than the non-bursty ones.

3.4.3 Four-Class Performance Evaluation

We compare OCGRR and PQWRR in a network with four classes EF, AF1, AF2, and BE. The edge switch is connected to R=7 sources and traffic in any class follows the Poisson process. We use one 100kbit buffer for each stream in each class in OCGRR. Since no shared buffer is used in any class in this scenario, the term R is used instead of $R+1$ in Eq.(3.1). In PQWRR, traffic of each class is saved in a shared buffer, and thus, we use one 700kbit shared buffer for each class.



a) Average Queuing Delay b) K Values for the two Class Index sets

Fig.3.9: Performance of OCGRR and PQWRR under a Four-Traffic Class System

A) Studying PQWRR and the Impact of Class Indices in OCGRR: Let traffic comprise 15%, 20%, 30% and 35% of EF, AF1, AF2, and BE respectively, where for stream i, we have $\lambda_{EF,i} \in [1\text{Mb/s}, 4.5\text{Mb/s}]$, $\lambda_{AF1,i} \in [2\text{Mb/s}, 4.5\text{Mb/s}]$, $\lambda_{AF2,i} \in [4\text{Mb/s}, 5\text{Mb/s}]$, $\lambda_{BE,i} \in [4\text{Mb/s}, 8\text{Mb/s}]$. Two sets of class index parameters are investigated for OCGRR: Set 1 with C_1=*1*, C_2=*0.8*, C_3=*0.6*, C_4=*0.4*; and Set 2 with C_1=*1*, C_2=*0.6*, C_3=*0.3*, C_4=0.2. The notations (1), (2) and (P) in Fig.3.9a indicate OCGRR under Set1, OCGRR under Set 2 and PQWRR respectively.

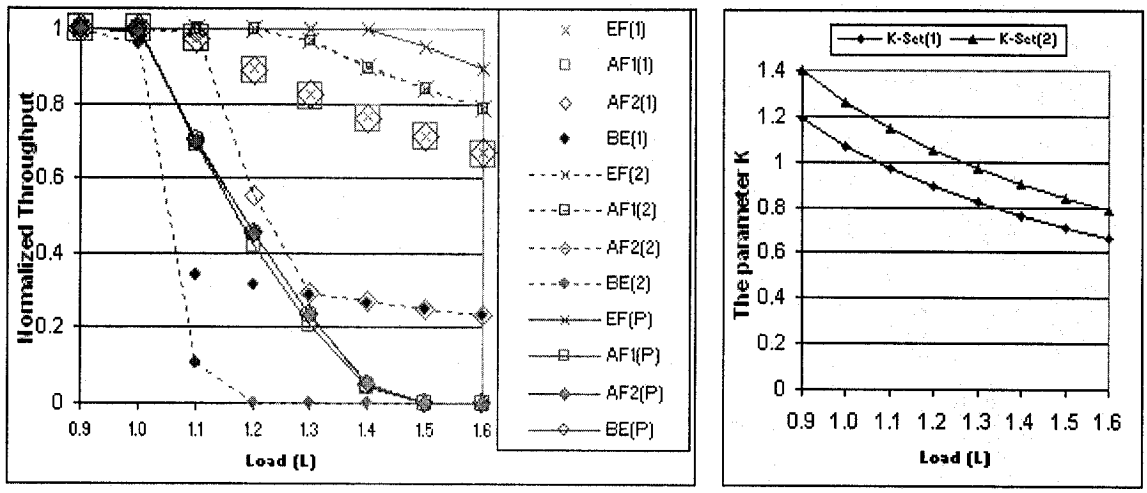
The queuing delay (Fig.3.9a) of each traffic class within a set is an increasing function of the traffic load. The delay for any class in Set 2, except BE traffic, is less than the delay for the same traffic class in Set 1 because of the smaller logical frame lengths (Γ_1=20070bits and Γ_2=15052bits), and higher K (Fig.3.9b) values in Set 2. For the smaller traffic loads, the average delay values are closer to each other. However, this difference becomes more obvious when the traffic load goes up.

The EF traffic in Set 1, EF(1), experiences a delay from 7.3μs to 150.3μs when the traffic load varies between 0.5 and 1.5. For this range, the delay for EF(2) changes from 7.3μs to 98.5μs. For $L_p \geq 1$, the EF traffic delay has a very small slope than the other classes. Due to a higher traffic load, the AF1 slope is between EF and AF2 slopes, and the slope for the BE traffic is very high.

In each set, the EF and BE traffic have the smallest and the highest delays respectively. The BE traffic delay is closer to the other classes for $L_p < 1.0$. However, for $L_p \geq 1$, the BE traffic both faces drop and higher delay than the others. The BE packet drop rate and delay is higher in Set 2 than Set 1. In short, by choosing smaller class indices, lower priority traffic will starve.

We have also included PQWRR results in Fig.3.9a, in which PQ is used for EF traffic and WRR is used for AF1+AF2+BE. The EF traffic in PQWRR, EF(P), has the smallest delay as expected. Since AF1, AF2 and BE have the same priority and WRR weights are proportional to AAR of the classes, the BE traffic with a higher AAR has a lower delay than AF1 and AF2. We observe almost the same packet loss rates for AF1, AF2 and BE in PQWRR for L_p>1.0. However, AF means assured forwarding. In OCGRR, BE traffic is only starved with higher delay and loss.

B) Studying Starvation of Lower Priority Classes: We study the behavior of OCGRR and PQWRR for lower priority classes under a high volume of EF traffic arrival. Let traffic comprise 70%, 5%, 15% and 10% of EF, AF1, AF2, and BE respectively, where the AAR of streams varies between 0.1 Mb/s and 15Mb/s in different classes. We study OCGRR under the two class index sets: Set 1 with $C_1=1$, $C_2=0.9$, $C_3=0.8$, $C_4=0.7$; and Set 2 the same as before. Note that at L_p=1.5, the EF traffic rate is 105Mb/s, which is larger than the available bandwidth C.



a) Normalized Throughput b) K Values for the two Class Index Sets

Fig.3.10: Performance of OCGRR and PQWRR under a High Volume of EF Traffic

Fig.3.10a depicts the normalized throughput for each class (ratio between the departure traffic rate and the arrival traffic rate of the traffic class). The notations (1), (2) and (P) in the legend refer to OCGRR with Set1, OCGRR with Set 2 and PQWRR respectively. By increasing the load, the throughputs of AF1, AF2 and BE reduces in PQWRR and reach to zero at L_p=1.5. In OCGRR, the throughput can be adjusted by class indices so that smaller class indices lead to a higher starvation for lower priority classes as the figure shows. Note that the parameter K (see Fig.3.10b) becomes less than 1 at L_p=1.1 and L_p=1.3 for Set 1 and Set 2 respectively. Whenever K is less than 1, the higher priority traffic is served with a lower intensity and more lower priority traffic can be serviced instead.

3.5 Conclusion

We have designed and studied OCGRR in the DiffServ domain. Our performance evaluation indicates that OCGRR has the features/capabilities of using smaller logical frame lengths and rounds; sending traffic packet by packet in smaller rounds; producing less packet burst from the same stream; reducing jitter and startup latency; controlling the starvation of lower priority classes; and beginning the transmission in each class from a delayed stream in the previous logical frame to ensure low latency and fairness. It also keeps the fairness for the streams at an acceptable level. We also showed that a desired performance can be obtained by adjusting class indices.

Chapter Four: Contention Avoidance

As discussed in Section 2.4, ingress switches are responsible for providing access to the OPS network. The objective in this chapter is to investigate the inexpensive hardware and cheap software schemes that can reduce traffic collision in a slotted-OPS network in order to use in the design of our DTDM protocol. We show that the ingress switches can have an important role in reducing the loss rate by using various techniques of reducing traffic load, coordinating traffic transmission, symmetric traffic transmission, balancing traffic load on the wavelength channels, assembling a slot with different classes, using more fibers, and using more wavelengths to carry the same traffic. We also develop a cost model in order to optimize the number of fibers and additional wavelength channels required to achieve a given loss rate.

4.1 Contention Performance Analysis

We shall develop an analytical model for the lower-bound on the slot loss rate, defined as the percentage of lost slots in a core switch. Since the collision happens on the same-wavelength, we analyze the loss rate over only one wavelength at f=1. Then, we can calculate the loss rate for the core switch. The analysis focuses on calculating the slot loss rate in both symmetric and asymmetric traffic. Since we study contention avoidance, we do not consider any contention resolution scheme at the core switch in our analysis. We also prove some lemmas based on the symmetric traffic analysis in order to better understand the contention avoidance issue.

4.1.1 Slot Loss Rate Analysis: Symmetric Traffic

Assume that each ingress switch transmits traffic to each output link of the core switch with an equal probability under any traffic load (called symmetric traffic). Let E indicate the number of empty slots out of n slots in an *n-slot-set*. The n slots may all be full (E=0) or there may be some empty slots (E >0). Define *a c-slot* collision as the event that there is a collision at the core switch from any c slots (out of the n-E slots) going to the same output link (called tagged output link) on the same wavelength in such a way that only one is allowed to be transmitted (and therefore *c-1* slots are dropped). Then, the total possible number of combinations for each event (for *c>1)* is

$$N_c = \binom{n}{E}\binom{n-E}{c}(n-1)^{n-c-E} \quad ,$$

where the first term is the combination of the empty slots in the *n-slot-set,* the second term indicates the combination of the *c-slot* collisions among *n-E* non-empty slots for the tagged output link, and the third term denotes the combinations for the remaining *n-c-E* slots where each slot can take one of the *n-1* possible output links (notice the tagged output link is excluded). Note that at $n=E$ we have $N_c=0$. In a slightly simpler manner, the total number of the slot combinations in the *n-slot-set,* N_T, is

$$N_T = \binom{n}{E} n^{n-E} \quad , \tag{4.1}$$

where the first term is the combination of the empty slots and the second term denotes the other *n-E* slots each taking one of *n* different output link combinations. Let P_c = *Prob.{Having a c-slot collision in the n-slot-set*, $1 < c \le n\text{-}E$ } at the tagged output link. By considering the uniform slot arrivals, P_c can be obtained as

$$P_c = \frac{N_c}{N_T} = \frac{\binom{n}{E}\binom{n-E}{c}(n-1)^{n-c-E}}{\binom{n}{E}n^{n-E}} \quad , 1 < c \le n-E \ .$$

Hence, the average number of the lost slots $\overline{d}$ is given by

$$\overline{d} = \sum_{c=2}^{n-E}(c-1)P_c \quad ,$$

where *n-E* denotes the upper limit on the total number of non-empty slots in the *n-slot-set*. The average number of the delivered slots to the tagged output link is $\overline{L_d} = \frac{n-E}{n}$. The slot loss rate at the tagged output link over one wavelength is given by

$$R_{SL}(n,E) = \frac{\overline{d}}{\overline{L_d}} = \sum_{c=2}^{n-E}\frac{(c-1)P_c}{\frac{n-E}{n}} \ .$$

This can be rewritten as

$$R_{SL}(n,E) = \frac{(n-E-1)!(n-1)^{n-E}}{n^{n-E-1}}\sum_{c=2}^{n-E}\frac{c-1}{(n-1)^c c!(n-c-E)!} \quad , \tag{4.2}$$

For simplicity, we show the summation term in Eq.(4.2) by $\Psi(n,c,E)$, i.e.,

$$\Psi(n,c,E)=\sum_{c=2}^{n-E}\frac{c-1}{(n-1)^c c!(n-c-E)!} \quad . \tag{4.3}$$

Then, Eq.(4.3) can be expanded as

$$\Psi(n,c,E)=\frac{1}{(n-1)(n-E-1)!}\sum_{c'=1}^{n-E-1}\binom{n-E-1}{c'}(\frac{1}{n-1})^{c'}-\frac{1}{(n-E)!}\sum_{c=2}^{n-E}\binom{n-E}{c}(\frac{1}{n-1})^{c} \quad .$$

By considering the binomial expansion formula and multiplying the terms inside the first and the second summations by $1^{n-E-c'-1}$ and 1^{n-E-c} respectively, we have

$$\Psi(n,c,E)=\frac{1}{(n-1)(n-E-1)!}(\alpha^{n-E-1}-1)-\frac{1}{(n-E)!}(\alpha^{n-E}-1-\frac{n-E}{n-1}) \quad ,$$

$$=\frac{1}{(n-E-1)!}(\frac{\alpha^{n-E-1}}{n-1}-\frac{\alpha^{n-E}-1}{n-E})$$

where $\alpha=1+\frac{1}{n-1}=\frac{n}{n-1}$. By inserting this equation in Eq.(4.2), we obtain

$$R_{SL}(n,E)=1-\frac{n}{n-E}(1-(\frac{n}{n-1})^{E-n}) \quad . \tag{4.4}$$

Since the traffic is symmetric to all output links, the same loss rate is obtained for the switch on the same-wavelength.

4.1.2 Lemmas

We shall provide some lemmas in the contention avoidance issue. The first lemma deals with the issue that by increasing the number of empty slots, the loss rate is reduced. In the second lemma, we relate the loss rate at the core switch to the individual loss rate of a channel. We shall prove in the third lemma that the balanced load on the wavelengths results in the lowest slot loss rate. The fourth lemma shows that the most part of the loss rate is due to c-slot collisions where c is small. The last lemma obtains a drop reduction ratio, suitable for design purposes. Results from these lemma will be used in the contention avoidance discussion in Sections 4.1.3 and 4.2.

Lemma 4.1: Increasing the number of the empty slots on each wavelength reduces the slot loss rate. In other words, $\forall n,E,n>E\geq 0,\ R_{SL}(n,E+1)<R_{SL}(n,E)$. By this lemma we want to show that under the lower-bound analysis decreasing traffic load, which results in increasing the number of the empty slots on each wavelength, reduces slot loss rate in the network.

Proof: See Appendix D.1.

Corollary 4.1: The maximum loss rate happens when $E=0$. Therefore, we have

$$R_{SL}(n,E) \leq R_{SL}(n,0) < Lim_{n\to\infty}(\frac{n-1}{n})^n = e^{-1} \approx 36.7\% ,$$

which is the same bound as in the upper-bound analysis.

Lemma 4.2: In a balanced slot transmission over the wavelengths, average slot loss rate of the core switch equals to the average loss rate of the individual channels. Here, we relate the slot loss rate at the core switch to the individual loss rate of a wavelength channel.

Proof: See Appendix D.2.

Lemma 4.3: The transmission of balanced load on the wavelength channels results in the lowest slot loss rate. We want to prove that the balanced load transmission on the wavelengths leads to in the lowest slot loss rate. This is an important contention avoidance aspect to be considered.

Proof: See Appendix D.3.

Lemma 4.4: Slot loss rate due to k-slot ($k \geq 2$) collisions is larger than the summation of all c-slot collisions when $k < c \leq n\text{-}E$. Here, we show that the most part of the slot loss rate is due to k-slot collisions where k is small. Therefore, by using devices such as more fibers on connection links (see Section 4.2.2.2 for the application of this lemma) one can reduce significantly the slot loss rate.

Proof: See Appendix D.4.

Lemma 4.5: For n ingress switches and p_e the probability of having an empty slot on a wavelength, the slot loss rate reduces by 100 $(1\text{-}r)\%$ in which r, defined as the drop reduction ratio, is given by $r = \frac{(\alpha^n)^{Pe} - p_e\alpha^n}{1-p_e}$. The parameter r relates the slot loss rate when there is no empty slot (i.e. $R_{SL}(n,0)$) and the slot loss rate when there are E empty slots (i.e. $R_{SL}(n,E)$) to each other. In other words, we can easily calculate $R_{SL}(n,E)$ when having r and $R_{SL}(n,0)$. Also note that, we have $R_{SL}(n,0) \approx 36\%$ (see Corollary 4.1).

Proof: See Appendix D.5.

4.1.3 Slot Loss Rate Analysis: Asymmetric Traffic

We now carry out the slot loss rate analysis for the asymmetric traffic case. Let $\mathbf{\Pi_i}=\{\pi_{i,1},$

$\pi_{i,2}, \pi_{i,3},.. \pi_{i,n}\}$ where $\pi_{i,k}$ *is* the probability that ingress switch *i* transmits traffic to output link *k*. Then, the probability distribution of the total traffic to each output link is given by $\mathbf{P_d}=\{p_1, p_2, p_3,.. p_n\}$ where $p_k = \frac{1}{n}\sum_{m=1}^{n}\pi_{m,k}$ and $\sum_{d=1}^{n} p_d = 1$. Since each output link has its own distribution probability, the parameter $N_{c,d}$ is defined as the total number of the *c*-slot collisions at output link *d* given by

$$N_{c,d} = \binom{n}{E}\binom{n-E}{c}(np_d)^c(n-np_d)^{n-c-E} = \binom{n}{E}\binom{n-E}{c}n^{n-E}(p_d)^c(1-p_d)^{n-c-E} \text{ , where } c>1 \quad (4.5)$$

The first term in Eq.(4.5) is the combination of the empty slots in the *n-slot-set*. The second term indicates the combination of the *c-slot* collisions among *n-E* non-empty slots to output link *d*. The third term calculates the possible number of the appearances of output link *d*, i.e., np_d in the *c*-slots. The fourth term denotes the combination for the remaining *n-c-E* slots each taking one of the other possible output links, except *d*, i.e., $n-np_d$ possible output links. Note that there are *n* total possible output links in an *n-slot-set* and of those just np_d slots belong to output link *d*. The parameter N_T is calculated from Eq.(4.2), and therefore, the slot loss rate among *n-E* non-empty slots in the core switch is given by

$$R_{SL,SW}(n,E) = \frac{1}{n-E}\sum_{c=2}^{n-E}(c-1)\sum_{d=1}^{n}\frac{N_{c,d}}{N_T} = \frac{1}{n-E}\sum_{c=2}^{n-E}(c-1)\binom{n-E}{c}\sum_{d=1}^{n}(p_d)^c(1-p_d)^{n-c-E} \quad . \quad (4.6)$$

Exchanging the two summations in Eq.(4.6), we obtain $R_{SL,SW}(n,E) = \frac{1}{n-E}\sum_{d=1}^{n}\sum_{c=2}^{n-E}(c-1)\binom{n-E}{c}(p_d)^c(1-p_d)^{n-c-E}$. The inner summation in this equation is simplified to

$$\begin{aligned}
&(n-E)p_d\sum_{c=2}^{n-E}\binom{n-E-1}{c-1}(p_d)^{c-1}(1-p_d)^{n-c-E} - \sum_{c=2}^{n-E}\binom{n-E}{c}(p_d)^c(1-p_d)^{n-c-E} \\
&= \{(n-E)p_d(1-(1-p_d)^{n-E-1})\} - \{(1-(1-p_d)^{n-E} - (n-E)p_d(1-p_d)^{n-E-1})\} \\
&= (n-E)p_d - 1 + (1-p_d)^{n-E}
\end{aligned}$$

By considering $\sum_{d=1}^{n} p_d = 1$, we have

$$R_{SL,SW}(n,E) = \frac{-E + \sum_{d=1}^{n}(1-p_d)^{n-E}}{n-E} \quad . \quad (4.7)$$

When all traffic is going to a particular output link d, i.e., $p_d \to 1$, and for other output links $p_{d'} \to 0$, where $d' = 1,2,..,n,$ and $d' \neq d$, we obtain the maximum loss rate of $R_{SL,SW}(n,E) \to \frac{n-E-1}{n-E}$.

We are interested in minimizing $R_{SL,SW}(n,E)$ in Eq.(4.7). For this purpose, the summation of $\sum_{d=1}^{n}(1-p_d)^{n-E}$ must be minimized. Considering the constraint of $\sum_{d=1}^{n} p_d = 1$, we need to find p_d values to minimize $\sum_{d=1}^{n}(1-p_d)^{n-E}$. According to the Lagrangian multiplier method [Bert99], we introduce a new function $f(p_1, p_2, ...p_n, \lambda) = \sum_{d=1}^{n}(1-p_d)^{n-E} - \lambda(\sum_{d=1}^{n} p_d - 1)$. For each p_i, we obtain $\frac{\partial f}{\partial p_i} = -(n-E)(1-p_i)^{(n-E-1)} - \lambda = 0$, $i = 1,2,...,n$. Then, we can determine that

$$p_i = 1 - (\frac{-\lambda}{n-E})^{\frac{1}{(n-E-1)}}, \; i = 1,2,...,n. \tag{4.8}$$

By taking summation at both sides of Eq.(4.8), we have $\sum_{i=1}^{n} p_i = n - n(\frac{-\lambda}{n-E})^{\frac{1}{(n-E-1)}}$. We can the determine λ from this equation by invoking $\sum_{d=1}^{n} p_d = 1$, and by inserting λ in Eq.(4.8), we should obtain $p_i = \frac{1}{n}$, $i = 1,2,...,n$ that shows a symmetric traffic transmission leads to the smallest loss rate. Now by inserting $p_d = 1/n$, $d = 1,2,...,n$ in Eq.(4.6), we obtain $R_{SL,SW}(n,E) = \frac{\sum_{c=2}^{n-E}(c-1)\binom{n-E}{c}(n-1)^{n-c-E}}{\frac{n-E}{n}n^{n-E}}$, which is the same slot loss rate obtained in Lemma 4.2 for the symmetric traffic transmission.

Recall that in our network model each slot can carry a number of packets. It is easy to show that packet loss rate (defined as the percentage of lost packets in the core switch) is the same as slot loss rate. Let m be the average number of packets to be carried in a slot. Assume on average of M_1 slots (carrying mM_1 packets), M_2 slots (carrying mM_2

packets) are dropped. Therefore, the slot loss rate and the packet loss rate are the same and equal to M_2 / M_1.

4.2 Contention Avoidance

Contention avoidance schemes can decrease the collision event at the core switch. Here, we shall investigate the software and hardware contention avoidance schemes.

4.2.1 Software Techniques

There are three potential techniques we consider in the following.

4.2.1.1 Balancing Load among Edge Switches

In this technique, the controller in an ingress switch (both edge switch and ingress OXC) balances the traffic load by distributing slots uniformly among the wavelengths so that the traffic at the input ports of the core switch is balanced. We expect the same loss rate on all channels for the symmetry reason, but less than the unbalanced load distribution case as proved in Lemma 4.3.

Symmetry of traffic to output links of the core switch must be considered at ingress switches. Otherwise, the slot loss rate is high (when traffic distribution to the output links is asymmetric). Note that to have a symmetric traffic transmission, an optical ingress switch must have optical buffer (however, note that our core switches do not have any optical buffers inside). Although it may be difficult for an optical ingress switch to always have a symmetric traffic transmission to the output link ports of the core switch due to the limited size of optical buffers, the electronic ingress switches can still attempt to send a symmetric traffic to the core switch using traffic shaping. For instance, the ingress switches may send almost equal number of slots to each output link in any say τ-slots intervals. If the asymmetry is expected to be permanent in the network design process, the number of ports/paths must be proportionally designed based on the traffic distributions.

4.2.1.2 Using Composite Assembling in *Arch1*

As discussed in Section 1.1.3, *Arch1* uses the CTT technique to assemble an integer number of packets inside a slot. In *Arch1*, packets of the same class can be only carried within a slot (we had already referred to it as non-CSA). In CTT, different timers are

assigned to each traffic class. Since a higher priority traffic has a smaller timer, even before having enough packets to fill a slot, a time-out may happen to send traffic from the higher priority class. Thus, the number of transmitted slots to the network will increase due to the non-CSA nature, which leads to a higher collision rate at OPS core switch. This greedy nature can be compensated with CSA[8] where two advantages are obtained: reducing delay and avoiding collision. Under CSA, a slot is generated in two cases: 1) Whenever the summation of the traffic in all classes exceeds the slot-size; 2) Whenever a time-out happens for a class. In this case, the traffic from other classes are also aggregated in a slot and sent to the network.

```
1. On arrival of packet p for class i of egress switch k:
2. If  L'(k,i)=0  then set_timer (k, i)  // set a new timer for class i  if empty
3. Insert packet p in Q'(k,i)
4. q_len(k) = Σ_{i=1}^{ψ} L'(k,i)
5. if(q_len>=S') //if combined traffic length is higher than slot bit size
6.  {
7.    if(q_len=S')
8.    {
9.       gen_composit_slot (k, ALL)  //assemble all packets of egress switch k
10.      clear_timer (k) ///clear timer for all egress switch k buffers
11.   }
12.   else
13.   {
14.      gen_composit_slot(k, i) //assemble all packets, except packet p
15.      clear_timer(k)     // clear timer for all egress switch k buffers
16.      set_timer (k, i)   // set a new timer for class i
17.   }
18.  }

19. On timer interrupt from class i of egress switch k:
20. gen_composit_slot (k, ALL)  //assemble all packets of egress switch k
21. clear_timer (k) //clear timer for all egress switch k buffers

22. gen_composit_slot (egress switch k, class i):
23. Empty a slot
24. for(j=1; j<=ψ; j++) ///count up to ψ  classes
25.  {
26.    c=0;
27.   if( j = i ) c=1;  // not to assemble the newly arrived packet
28.   while(S'(k,j)>c)
29.     {
30.        P=remove a packet from head of Q'(k,j)
31.        slot = slot + P // append the packet to slot
32.     }
33.  }
```

Fig.4.1: Composite Slot Assembly Pseudo Code

Fig.4.1 shows the CSA algorithm. In this pseudo code, we have the following definitions: $Q'(k,i)$ is the queue dedicated for class i of egress switch k, $L'(k,i)$ is the

[8] Note that the idea of composite packet aggregation has been so far studied in OBS network only as reviewed in Section 1.1.3.

length of packets waiting in *Q'(k,i)* in bits, *q_len* is the total size of packets waiting in all classes of a given egress switch, *S'* is the size of a in bits, *S'(k,i)* is the size of *Q'(k,i)* in packets, and *slot is* a data structure to contain a number of packets up to *S'*. On the arrival of packet *p* of class *i* going to egress switch *k*, Lines2-18 create a composite slot whenever *q_len* is equal or larger than *S'*. When *q_len>S'*, there will not be enough room in the slot to accommodate the packet *p*. When an interrupt happens from the timer dedicated to class *i* of egress switch *k*, a composite slot is created (Line-20 and Line-21). The timers are set or cleared appropriately in each case by CSA. Lines 23-33 display the composite slot assembling procedure where packets from all classes may be assembled in a slot.

4.2.1.3 Lowering Traffic Load

An obvious approach to decrease slot loss rate is to reduce the channel utilization (see Lemma 4.1), which is the normalized traffic load on the wavelength. Related to this traffic load, define *L* as the probability of having a slot arriving in the core switch on each wavelength from each ingress switch, i.e., the normalized traffic load on the wavelength channels. Let each ingress switch reduce the load to $L=1-p_e$ where p_e is the probability of having an empty slot on a wavelength. Then, $Prob.\{E=k\}=\binom{n}{k}p_e^k(1-p_e)^{n-k}$, and the average number of the empty slots on each wavelength would be $\overline{E}=np_e=n(1-L)$. Based on the average amount of $\overline{E}$, the loss rate can be calculated by inserting $E=\overline{E}$ in Eq.(4.2).

4.2.2 Hardware Techniques

There are two hardware techniques to avoid contention, but at a higher cost than the software techniques.

4.2.2.1 Using More Wavelengths

An equivalent effect to lowering traffic load is to use more wavelengths to transmit the same traffic so that the number of empty slots is increased. Note that to use more wavelengths, more transceivers in the ingress edge switches and egress edge switches must be employed. This will increase the cost and the complexity of the edge switches.

Assume there is a full traffic that can occupy all W slots in a slot-set. By using W_E extra wavelengths, there are $W+W_E$ slots for the transmission of W slots in each slot-set. Let the slots be uniformly distributed over $W+W_E$ wavelengths. Therefore, one slot on a wavelength channel is empty with the probability of $p_e = \frac{W_E}{W+W_E}$. Without loss of generality, we take f=1. Then, for n slots traveling from n ingress switches on the same wavelength channel, the number of the empty slots at the core switch is

$$E = np_e = \frac{nW_E}{W+W_E} \quad . \tag{4.9}$$

The loss rate can be obtained by inserting E in Eq.(4.2). Conversely, given a defined E value to achieve a certain loss rate, the number of the extra channels can be obtained by $W_E = \frac{WE}{n-E}$. One can see that W_E is a rapidly increasing function with an asymptotic at n=E. For example, we need W_E=2 extra channels for n=10, W=3, and E=4; but W_E =12 for n=10, W=3, and E=8.

Example 4.1: Consider a scenario of a 10x10 switch, f=1 fiber, W=3 wavelengths, and W_E=2 extra wavelengths. Then, the average number of empty slots on each wavelength is E=4 (see Eq.(4.9)). By using Eq.(4.4), the loss rate is $R_{SL}(10,4) = 21.9\%$ using only one wavelength. Comparing this case with the result of the case with no empty slot (when using only W=3 wavelengths), i.e., $R_{SL}(10,0) = 34.8\%$, shows a 13% reduction in the slot loss rate.

4.2.2.2 Using More Fibers

By using k fibers on each connection link, it is possible to have k logical wavelength channels of the same wavelength in each output link and therefore up to k slots (on the same-wavelength) can be transmitted simultaneously. By this scheme, the loss rate due to *2,3,...,k-slot* collisions are all removed. Now let there be f fibers from each ingress switch to the core. Then, similar to Section 4.1.1, we have

$$N_c = \binom{nf}{E}\binom{nf-E}{c}(n-1)^{nf-c-E} \quad (c>1) \ ,$$

$$N_T = \binom{nf}{E} n^{nf-E} \quad ,$$

$$\overline{L_d} = \frac{nf-E}{n} \quad .$$

Since from the c-slot collisions, c-f slots are dropped (when $c>f$), the slot loss rate is obtained by

$$R_{SL}(n,f,E) = \sum_{c=f+1}^{n-E} \frac{(c-f)N_c / N_T}{\overline{L_d}} = \frac{(nf-E-1)!(n-1)^{nf-E}}{n^{nf-E-1}} \sum_{c=f+1}^{nf-E} \frac{c-f}{(n-1)^c c!(nf-c-E)!} \quad , \tag{4.10}$$

Note that E is the average number of the empty slots among nf slots, i.e., *nf-slot-set.*

4.2.2.3 Combined Hardware Approach

Consider the combined hardware approach in which we use a multi-fiber architecture with f fibers under traffic load of L. Each ingress switch needs at most W wavelengths on each fiber to carry its whole traffic at load L=1.0. However, the ingress switch may be designed to use W_E additional channels on each fiber. The probability of having an empty slot on a wavelength channel due to the traffic load L and W_E additional channels are p_{e1}=*1-L* and $p_{e2} = \frac{W_E}{W+W_E}$, respectively. Hence, the combined probability of having an empty slot on a wavelength channel is $p_e = 1-(1-p_{e1})(1-p_{e2}) = 1-L\frac{W}{W+W_E}$. By inserting $E=nfp_e$ in Eq.(4.10), the slot loss rate for the combined hardware approach is given by $R_{L,N}(n,f,W,W_E,L) = R_{SL}(n,f,nf(1-L\frac{W}{W+W_E}))$.

4.3 Lower and Upper-Bounds on Slot Loss Rate

We are interested in the upper-bound and the lower-bound analysis of slot loss rate because it provides a valuable information on the difference between the two bounds, and allows us to understand the significance and impact of this difference. Note that the lower-bound performance has been obtained in the previous analysis because we consider that there are E empty slots at each slot set. Here, we shall provide an upper-bound analysis. We will also show how the lower-bound can be achieved in practice.

4.3.1 Upper-Bound on the Slot Loss Rate

Consider the same switch model discussed in the previous section. The worst scenario (upper-bound) in the loss rate occurs when all ingress switches connected to the core switch transmit slot traffic with a probability L on each wavelength. For a given L, the probability that c slots going to the same output link on the same-wavelength follows a binomial distribution, $P_c = \binom{n}{c}(\frac{L}{n})^c(1-\frac{L}{n})^{n-c}$. Similar to the previous analysis, slot collision happens when $c>1$. Therefore, we obtain the upper-bound of the slot loss rate as

$$R_{SL,UP}(n,L) = \frac{1}{L}\sum_{c=1}^{n}(c-1)P_c = \frac{1}{L}\sum_{c=1}^{n}(c-1)\binom{n}{c}(\frac{L}{n})^c(1-\frac{L}{n})^{n-c} = \frac{1}{L}(L+(1-\frac{L}{n})^n-1) \quad . \qquad (4.11)$$

4.3.2 Difference between Lower and Upper-bounds

The difference between the two bounds allows us to understand the significance of achieving the lower-bound analysis to reduce the slot loss rate. Using $E = n(1-L)$, the difference between the upper-bound (Eq.(4.11)) and the lower-bound (Eq.(4.4)) in the slot loss rate is

$$\Delta = R_{SL,UP}(n,L) - R_{SL}(n,E) = \begin{cases} \frac{1}{L}((1-\frac{L}{n})^n - (1-\frac{1}{n})^{nL}), & if \quad n-E \geq 1 \\ \frac{1}{L}(L+(1-\frac{L}{n})^n - 1), & otherwise \end{cases} .$$

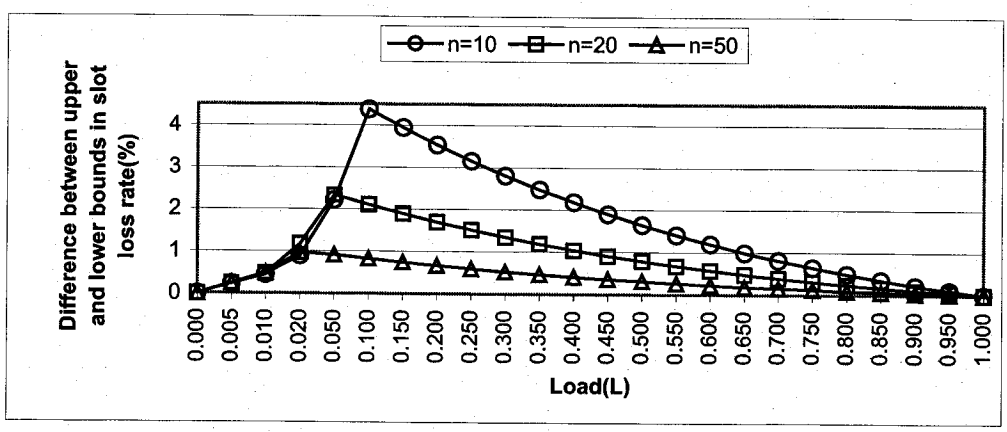


Fig.4.2: Difference between Upper and Lower-bounds in the Slot Loss Rate

The parameter Δ is maximized when the lower-bound slot loss rate becomes 0 at $L=1/n$ with the maximum difference value of $\Delta_{max} = 1-n+n(1-\frac{1}{n^2})^n$. Obviously, Δ_{max} is

a decreasing function of n and in the worst case the maximum difference is 12.5% when n=2.

Fig.4.2 shows the difference Δ between the upper and lower-bounds for different number of input/output connections n=10, n=20 and n=50 under different loads. The maximum difference happens at L=1/10, L=1/20 and L=1/50 respectively. Note that we have expanded the region $L \in [0,0.02]$ to see the performance more clearly. For n=10, the parameter Δ starts from 0 at L=0 and increases up to 4.38% at L=0.1. Then, it reduces by increasing L so that to Δ=0 at L=1.0. The same situation is observed for n=20 and n=50. Note that at the same load the difference becomes larger when n is reduced. Recall that the parameter n is usually small in optical networks and therefore the parameter Δ must be considered carefully at smaller traffic loads.

4.3.3 Achieving the Upper and Lower-bound Performance in Practice

If we consider L as the average probability of slot arrival in the core switch on wavelength w_i from each ingress switch, then two cases may arise:

1. **Achieving the Upper-bound Performance:** To compute the slot loss rate related to this upper-bound, the analysis studied in Section 4.3.1 can be used. To achieve this upper-bound, we use a technique called Uncoordinated Slot Transmission (UST) where each ingress switch sends traffic on w_i with a probability of L. Then, different number of non-empty slots may appear in the core switch at each *n-slot-set*. For example, for n=10 and L=0.7, there may be a different number of non-empty slots between 0 and 10 at each *n-slot-set*. However, the long-term average number of non-empty slots at each *n-slot-set* is seven.
2. **Achieving the Lower-bound Performance:** To compute the slot loss rate related to this lower-bound, the lower-bound analysis studied in this chapter can be used. We use a technique called Coordinated Slot Transmission (CST) to achieve the lower-bound performance. Here, ingress switches are weakly coordinated so that they send traffic on w_i in a way that the variance of the number of non-empty slots at each time-slot is minimized. For example, consider the same scenario mentioned for UST. Under CST, the ingress switches must be coordinated (by the core switch) so that there are only seven non-empty slots at each *n-slot-set*. Another aspect in the

contention avoidance issue is coordination among ingress switches. As another example, considering n=10 and L=0.65, the ingress switches must send slots in a way to have six or seven non-empty slots at each *n-slot-set*.

To implement both CST and UST, the ingress switches must use buffers (electronic buffers in edge switches and optical buffers in ingress OPS core switches) to save slots and then forward them to the core switch at a desired time. In a single-hop network, they can be easily implemented by the edge switches. On the other hand, in a multi-hop network, ingress OPS core switches can use optical buffers for both contention resolution and implementing CST/UST. When the propagation delays are low, a fast coordination can be performed between the core switch and its ingress switches under CST.

4.4 Performance Evaluation

To verify the correctness of our analytical results, we carry out some simulations to study the slot loss rate in an nf x nf core switch. Unless otherwise mentioned, fixed-length packets are generated according to a Poisson process, each slot carries one packet, all traffic loads on wavelength channels are normalized (see Section 4.2.1.3), the analysis results are all based on the lower-bound analysis, the simulation results are obtained under the UST scheme, and we have used a symmetric traffic transmission in ingress switches. The analysis results will show the loss rate that can be obtained if the ingress switches follow CST. For each scenario, enough number of replications is run, and 95% level of confidence interval is obtained.

Since we are only studying the contention avoidance aspect of the techniques discussed in this chapter, we do not employ any contention resolution scheme at the core switch. Therefore, the higher loss rates from the worst scenario are expected. They are just used as benchmarks here. However, other methods such as using wavelength converters (as a contention resolution scheme) at the core switches in the following chapters will achieve a low loss rate.

4.4.1 Effect of Empty Slots

Fig.4.3 shows the performance for the drop reduction ratio r (defined in Lemma 4.5) for n=10, n=30 and n>>1 (e.g., n=50). One can see that increasing n would raise the r value

to a limit of r_{max} for n>>1. By increasing the probability of the empty slots, lower reduction ratios can be obtained. This in turn will lead to a lower loss rate. As expected, the ratio is r=1 when there is no empty slot, and r = 0.0 when 100% of the slots are empty. For n=10, r starts from 1.0 at p_e=0.0 and reduces to 0.0 at p_e=0.9. A similar trend can be observed for other values of n.

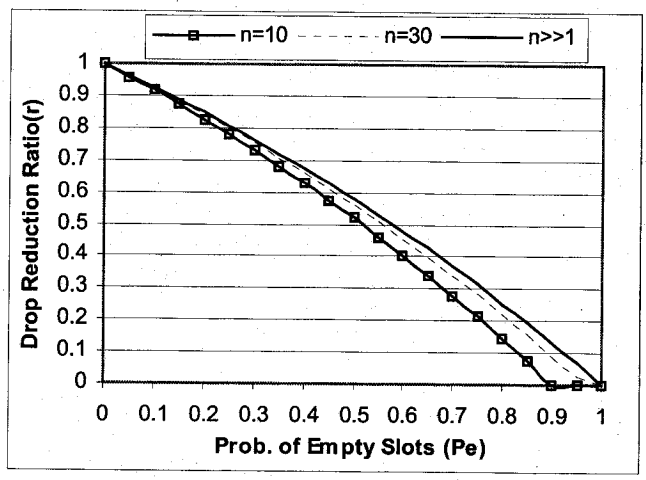


Fig.4.3: Analysis Results for the Drop Reduction Ratio (See Lemma 4.5)

Let us see how we can use the parameter r in order to obtain slot loss rate. For an example, 36.4% of the slots will drop at L=1.0 when n=50, W=3, f=1. Now consider that the fifty ingress switches are sending traffic at L=0.8 on five wavelengths instead of three (W=3, W_E=2). According to the discussion in Section 4.2.2.3, the value of p_{e2} for the extra channel usage is 2/5=0.4. Also, the traffic load L=0.8 leads to p_{e1}=*1-0.8*=0.2. Thus, the total probability of having empty slots is p_e=1-(1-0.2)(1-0.4)=0.52, as. According to the reduction ratio diagram, we have r = 0.559 and therefore slot loss rate is 0.559*36.4%=20.36% (see Lemma 4.5).

Table 4.1: Distribution of Unequal Output Link Probabilities

n	Distribution of Output Link Probabilities
10	P_d={ 0.43, 0.01, 0.04, 0.09, 0.01, 0.15, 0.15, 0.01, 0.08, 0.03}
20	P_d={ 0.01, 0.03, 0.07, 0.01, 0.01, 0.12, 0.01, 0.02, 0.05, 0.01, 0.14, 0.09, 0.01, 0.12, 0.02, 0.01, 0.15, 0.01, 0.08, 0.03}
30	P_d={ 0.01, 0.01, 0.01, 0.01, 0.07, 0.01, 0.01, 0.01, 0.10, 0.01, 0.01, 0.02, 0.05, 0.01, 0.11, 0.01, 0.01, 0.01, 0.09, 0.01, 0.03, 0.04, 0.04, 0.01, 0.02, 0.01, 0.15, 0.01, 0.08, 0.03}

Table 4.2: Slot Loss Rate Under Assymetric Traffic

n	Slot Loss Rate Percentage	
	Analysis	Simulation
10	53.3658	53.3614
20	50.7099	50.7164
30	52.0167	52.0263

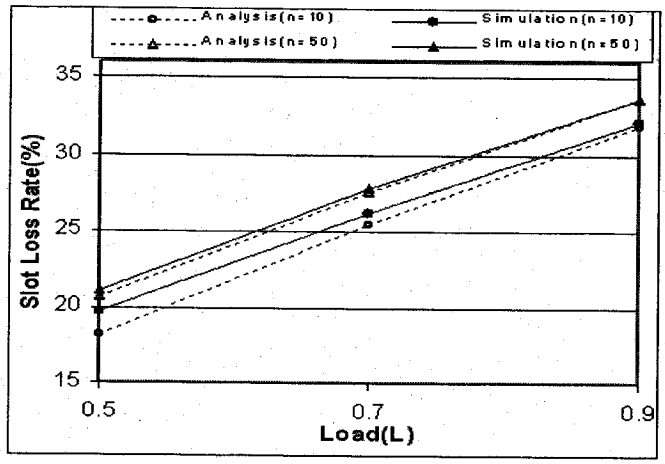


Fig.4.4: Slot Loss Rate when Traffic Loads of the Ingress Switches are all Equal

4.4.2 Software Contention Avoidance Techniques

A) Effect of Asymmetric Traffic: Here, the analysis and simulation results for the case that ingress switches send traffic in an asymmetric situation are presented. Table 4.2 compares the analysis and simulation results for the slot loss rate under traffic load L=1.0, f=1, W=1 and for n=10, n=20 and n=30. The output link probabilities (see Section 4.1.3) are taken from Table 4.1. Both the analysis and simulations results show 52.01%, 50.70% and 53.35% of slot loss rate for n=10, n=20, and n=30 respectively. It is clear to see that the asymmetry of the distribution of the output links is very high and this is why higher loss rates are expected. It is observed that the asymmetry rate is higher for n=10 than n=20 and this is why there is a lower loss rate for n=20 than n=10. One can see that loss rate is higher under the asymmetric load. However, the maximum loss rate under symmetric traffic load is bounded by 36.7% (see Corollary 4.1). In the following, we will study how to further reduce slot loss rate.

B) Effect of Load Reduction: Fig.4.4 compares slot loss rate from the analysis and the simulation results when the traffic loads on the wavelength channels sent from ingress

switches are all equal (see Section 4.2.1.3). Assume all ingress switches send slots with the same load. Different loads of L=0.5,0.7 and 0.9 are investigated. The average number of the empty slots is analytically calculated from $\overline{E} = n*(1-L)$. At lower traffic loads, we expect more empty slots and therefore the slot loss rate should be smaller than the case with higher loads. At n=50 and L=0.5, the observed and the calculated loss rate are almost 21% and this amount increases to 33.7% at L=0.9. The same situation is observed for n=10.

Table 4.3: Effect of Load-Balancing in Slot Loss Rate

	Average Loss Rate (%) on				
	w_0	w_1	w_2	w_3	Switch
Non-balanced	36.42	17.31	24.48	30.8	29.53
Balanced	27.73	27.73	27.73	27.73	27.73

C) Balanced Load Traffic: Consider two scenarios in traffic transmission at L=0.7 and n=50 and W=4. First, assume that each edge switch sends traffic on the wavelengths with unequal loads of say 1.0, 0.4, 0.6 and 0.8 on wavelengths w_0, w_1, w_2 and w_3, respectively. Then, we expect 50, 20, 30 and 40 non-empty slots on wavelengths w_0, w_1, w_2 and w_3 respectively. The simulation results in Table 4.3 show 36.42%, 17.31%, 24.48% and 30.8% drop percentage on the four wavelengths respectively. The average switch loss rate is measured to be 29.53%, which compares well with the theoretical value of the average loss rate of the switch (i.e. 29.5% computed from Eq.(D.3) in Appendix D). Now if each edge switch distributes the load on the wavelengths in a balanced manner (each wavelength with load 0.7), then the average loss rate on all channels and the switch is 27.73%. Thus, the balanced traffic load leads to a lower loss rate.

D) Effect of CSA vs. Non-CSA: We now compare the performance of CSA with non-CSA packet aggregation schemes under Poisson traffic and input load of 0.5 when n=8, f=2 and W=4. Our simulation model for this scenario is depicted in Fig.A.1. For CSA, we have used our protocol discussed in Section 4.2.1.2. Each edge switch generates variable-length packets with IP length distribution mentioned in [CAID06]. Two traffic classes are considered for each ingress switch: EF (high priority) and BE (low priority). Fifty percent of traffic is EF and 50% belongs to BE traffic. Traffic generation at each ingress switch to each destination is uniform. Time-out value for the BE traffic is kept

constant at 30 μs and EF time-out changes from 10μs to 30 μs. At the channel rate of 10Gb/s, data transmission time and the slot-offset interval between two slots for switching and guard time purposes are set to 9 μs and 1μs respectively. The size of slot transmission buffer in each ingress switch (see *Arch1* ingress switch model in Fig.A.1) is infinity.

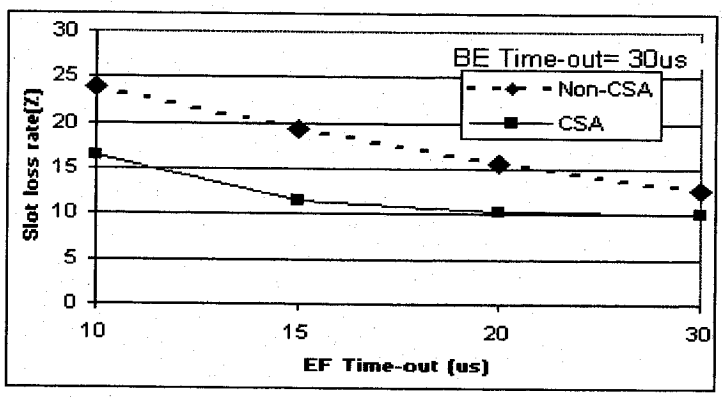


Fig.4.5: Slot Loss Rate for the CSA and non-CSA Packet Aggregation Schemes

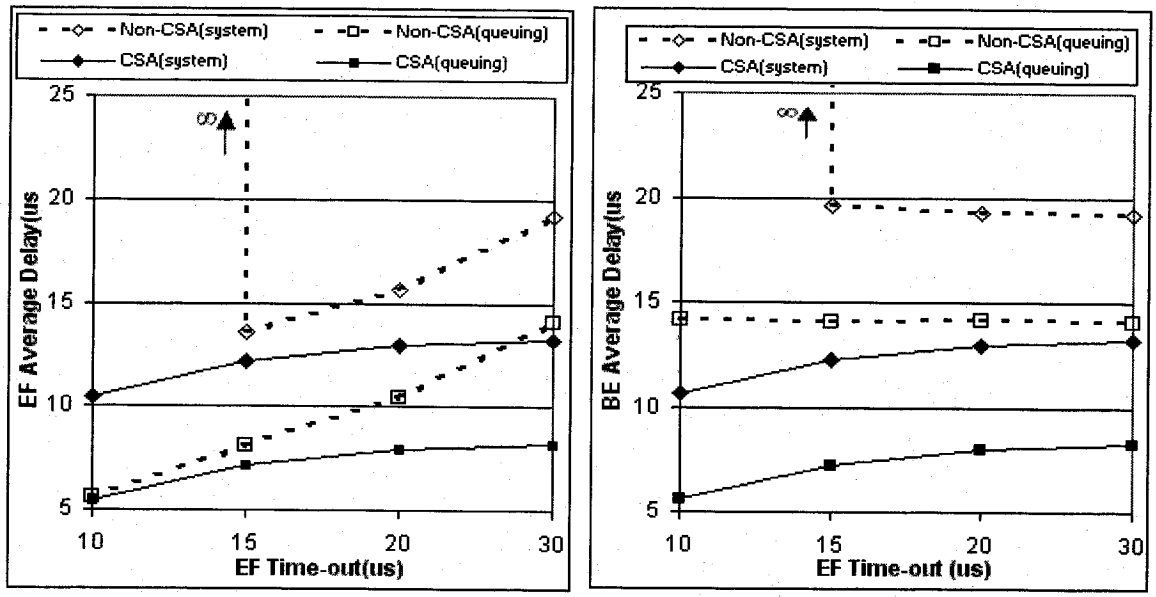


Fig.4.6: Queuing and System Delay for EF and BE under CSA and Non-CSA.

As shown in Fig.4.5, slot loss rate is lower for CSA than non-CSA. Increasing the time-out value reduces the loss rate. Fig.4.6 displays the average delay for queuing (defined as the waiting time in an ingress switch buffer from the time a packet arrives until it is sent to the slot transmission buffer shown in *Arch1*) and system (defined as the

whole waiting time in an ingress switch from the time a packet arrives until transmitted from the slot transmission buffer of the ingress switch to the OPS network). By increasing the EF time-out value, both delay values increase. Note that in lower time-out values, the non-CSA scheme becomes unstable and delay goes to infinity due to a higher slot rate generation than slot service rate. One can see a considerable difference between the EF and BE delays under non-CSA. Although the EF delay is always smaller than the BE delay under the CSA scheme, however, there is not much difference between them. This is because when a time-out happens for the EF traffic, the BE packets may also be served if there is a room left in a slot. In other words, time-out event for each class helps to serve the traffic of the other class. Therefore, BE delays are also reduced.

Although by using a smaller time-out value a lower delay is expected, however, it may lead to a higher loss rate at the core switch. Moreover, time-out cannot be reduced to very small numbers because an unstable case is created and transmission delay increases to infinity.

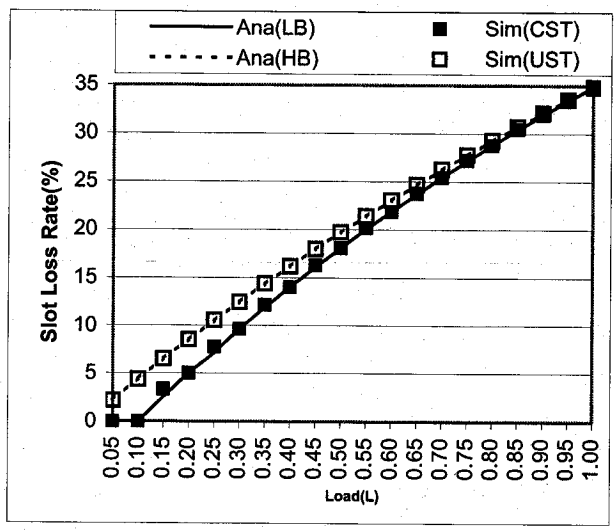


Fig.4.7: Effect of CST vs. UST in Slot Loss Rate

E) Effect of CST vs. UST: Fig.4.7 illustrates CST and UST as well as the lower-bound (LB) and upper-bound (HB) analysis at n=10, f=1 and W=1. One can see that simulation results for CST and UST match well with the lower-bound and the upper-bound analysis respectively. Slot loss rate is in general an increasing function of traffic load for both the CST and UST schemes (except at the lower load for CST). The difference between CST and UST reduces to 0 at L=1.0, but is maximum at L=0.1 (as proved in Section 4.3.2 the

difference between the lower-bound and upper-bound of slot loss rate becomes maximum at $L=1/n$). One can see that CST can provide a loss-less transmission at L=0.1, but the loss rate is 4.4% under UST.

CST has always a smaller loss rate than UST. However, the UST technique can be implemented easily. Note that CST requires some kind of cooperation between a core switch and the ingress switches connected to that core switch.

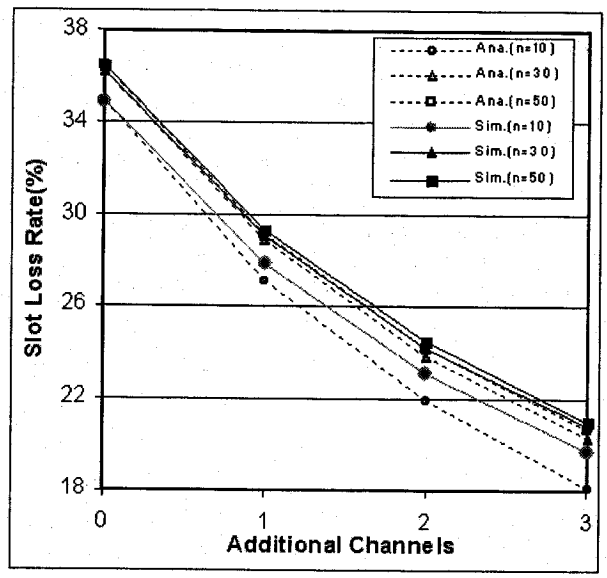


Fig.4.8: Slot Loss Rate Performance for the Extra Channel Usage

4.4.3 Hardware Contention Avoidance Techniques

We now study the hardware contention avoidance techniques that can be combined with some software contention avoidance techniques such as symmetric and load balanced traffic transmission in ingress switches.

A)Effect of Additional Wavelengths: Fig.4.8 compares the analysis (Eq.(4.2)) and the simulation results of the slot loss rate as a function of the extra channel usage (Eq.(4.9)) under a system with W=3 wavelengths at L=1.0. The comparison also examines the effect of different number n of ingress switches. For n=10, using one, two and three extra channels lead to the loss rates of 27.1%, 21.9% and 18.1% respectively. A similar trend can be observed for other values of n. If the ingress switches follow CST, the loss rate will follow the analysis results and the difference between CST and UST becomes

significant for n=10.

As the diagram shows, using one extra channel reduces the loss rate by almost 20% for all cases. The second and the third extra channels reduce the slot loss rate by almost 35% and 45%, respectively. These results for large n, compares well the analysis results found in Fig.4.2. For one, two and three extra channels, the value of p_e is 0.25, 0.4 and 0.5 respectively. By inserting these values in the diagram showed in Fig.4.2, we obtain the reduction ratio of 0.805, 0.674 and 0.579, which results in the loss rate reduction of 19.5%, 32.6% and 42.1% respectively (as opposed to 20%, 35%, 45%). It can also be observed that using the extra channel is almost independent of the network size (for n>>1) and we expect almost the same loss rate reduction. As expected from the analysis of Section 4.2.2.1, the advantages of using more extra channels decreases when W_E tends to W.

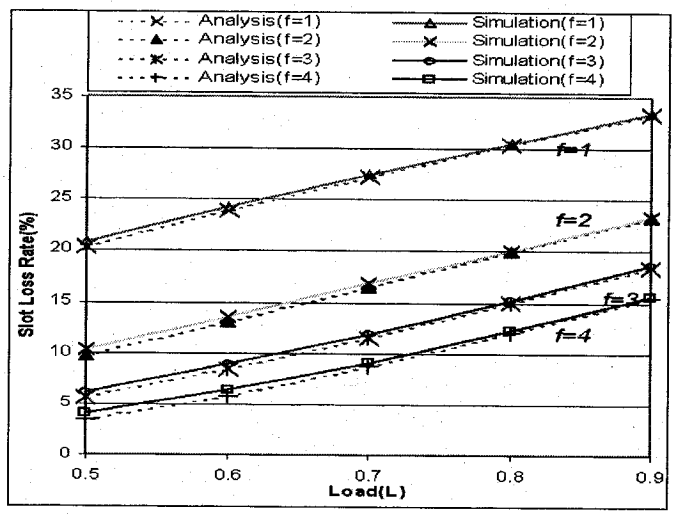


Fig.4.9: Effect of Multi-fiber in Loss Rate Reduction at n=30

B) Effect of More Fibers: Fig.4.9 shows the slot loss rate as a function of traffic load L. The system has n=30 input/output connections where each fiber includes four wavelengths and each wavelength carries the same traffic load. The upper diagram shows the loss rate for the case that all connections are supported with one fiber. The slot loss rate increases from 21% to 33.5% when load changes from 0.5 to 0.9. The diagram for f=2 shows a better gain in the slot loss rate reduction. As mentioned, most part of the collisions is due to two-slot collisions. A two-fiber architecture leads to the loss rate of almost 10% to 23% for different loads in range 0.5 and 0.9 (11% less drop comparing to

f=1). A network with three fibers results in loss rate of 6% to 18.5% at L=0.5 to 0.9. As it can be observed, the loss rate reduction (almost 5%) has reduced comparing to f=2 (11%). The four-fiber network has also reduced the loss rate (almost 3%) comparing to f=3. The loss rate is 15.5% at f=4 and a higher load L=0.9.

As discussed a number of times before, one should also use contention resolution schemes at the core switch to reduce the loss rate to a desired value of say 10^{-6}, but at a lower volume. For instance, in a four-fiber network and at a higher load of 0.9, the contention resolution schemes at the core switch must resolve contention of only 15.5% instead of 33.5% in a single-fiber network. This in turn requires less number of contention resolution schemes at the core switch.

In summary, neither the contention avoidance schemes nor contention resolution schemes can solely reduce effectively the slot loss rate. In a contention avoidance phase, the inexpensive software techniques can be used followed by using the multi-fiber architecture and additional channels. In a contention resolution phase, wavelength converters can reduce the loss rate to a desired value. To reduce the wavelength conversion cost, shared-per-node wavelength converters can be used. Finally, the inexpensive resolution schemes such as retransmission in the optical domain and deflection techniques can reduce the drop rate to almost zero.

4.5 Cost Model for the Combined Hardware Approach

There are different ways to combine the number of fibers and the number of additional wavelength channels in order to achieve a desired slot loss rate. However, not all of them are cost-effective. We are now interested in finding out the combination that would give the best performance, in terms of the least hardware cost. We approach this by optimizing the hardware cost, but first we need to analyze the cost of each component.

4.5.1 Cost Components

We partition an optical network into a number of segments to evaluate its cost. We define a segment to include only one core switch, its input links, as well as edge switches connected to the optical switch, and their input/output links. Suppose, in average, a segment contains $\hat{n}_l$ edge switches, and the number of input ports of the core switch is $\hat{n}$. We then calculate the average segment cost as follows based on the average number of $\hat{n}$

input connection links, $\hat{f}$ fibers in each link, $\hat{W}$ wavelengths on each fiber, and $\hat{n}_l$ input/output links (each link connected to an edge switch).

1. **Switch cost (C_{OXC}):** As suggested in [SrSo02], the optical switch cost consists of the switching elements, the wavelength de-multiplexers (DMUX), and the multiplexers (MUX). We assume that the optical switch in each segment is a non-blocking Batcher Banyan switch with $\hat{n}\hat{f}$ input ports and $\hat{n}\hat{f}$ output ports, and is made up of $N_{SE} = \frac{1}{4}(\hat{n}\hat{f}\hat{W}\log_2^{\hat{n}\hat{f}}(3+\log_2^{\hat{n}\hat{f}}))$ 2x2 switching elements [JeAy96]. To calculate N_{SE}, the number of switch ports $\hat{n}\hat{f}$ must first be rounded up to the nearest power of two[9]. The core switch requires $\hat{n}\hat{f}$ units of $1x\hat{W}$ wavelength demultiplexing modules and $\hat{n}\hat{f}$ units of $\hat{W}x1$ wavelength multiplexing modules. Then, total number of the DMUX and MUX modules required in the core switch is $N_{MD} = 2\hat{n}\hat{f}$. Let $C_{MD}(\hat{W})$ be the MUX/DMUX cost for $\hat{W}$ channels. Then, by assuming the same cost for MUX and DMUX, the switch cost is $C_{OXC}(\hat{n},\hat{f},\hat{W}) = C_{SE}N_{SE} + C_{MD}(\hat{W})N_{MD}$, where C_{SE} is the 2x2 switching element cost. Let us now assume that a de-multiplexer module is constructed with cascaded Mach-Zehnder (M-Z) filters [Kart03]. In this architecture, the required number of filters for demultiplexing $\hat{W}$ channels is $N_{MZ}(\hat{W}) = \hat{W} - 1$ (see Appendix E). Therefore, we have $C_{MD}(\hat{W}) = (\hat{W}-1)C_{MZ}$, where C_{MZ} is a cost of one filter.
2. **Link provisioning cost (C_{lp}):** Since different link configurations (different number of fibers and wavelengths) do not affect the link provisioning cost significantly [SoMi04], we assume this cost, C_{lp}, is the same for any link configuration.
3. **Fiber cost (C_f):** This is the combination of the cost of optical amplifiers, physical fiber, dispersion compensators, slot synchronizer [RaTu04] at the core switch input, channel spacing, desired quality of signal arrived at the core switch or egress switch, and maintenance associated with a fiber in a link, where the number of wavelengths carried in the fiber has a direct impact on this cost [SoMi04]. Let $C_f(\hat{f},\hat{W})$ denote the fiber cost of a link with $\hat{f}$ fibers and each fiber with $\hat{W}$ wavelengths. Based on the

---

[9]This is required because the number of input/output ports in an optical switch is a power of 2. When rounding up $\hat{n}\hat{f}$ to u, there will be $\frac{1}{4}(u\hat{W}\log_2^u(3+\log_2^u)) - \frac{1}{4}(\hat{n}\hat{f}\hat{W}\log_2^{\hat{n}\hat{f}}(3+\log_2^{\hat{n}\hat{f}}))$ switching items unused in the switch.

fiber cost definition, we have $C_f(\hat{f},\hat{W})=\hat{f}C_f(1,\hat{W})$. Since the devices, such as dispersion compensators, used for a fiber with a higher number of wavelengths is much more expensive than the devices used for a fiber with a lower number of wavelengths, we model the cost of a fiber with $\hat{W}$ channels based on the cost of a fiber with one wavelength, by $C_f(1,\hat{W})=(1+\eta)^{(\hat{W}-1)}C_f(1,1)$, where $\eta>0$ and $(1+\eta)^{(\hat{W}-1)}$ is the additional cost factor. The number of wavelengths on a fiber may also be limited at most to W_{max} due to the practical limit on present technology.

4. **Wavelength cost (C_λ):** The cost for each wavelength of each edge switch is defined as $C_\lambda=C_T+C_R$, where C_T and C_R are the cost of optical transmitter and receiver respectively. The cost C_T (cost C_R) also includes the cost related to the multiplexing (demultiplexing) at the transmitter (receiver) side of each edge switch.

Note that $\hat{n}$ input and $\hat{n}_l$ output links must be considered to calculate the total fiber cost in a segment. Also, the total wavelength cost in a segment is relevant to $\hat{n}_l$ input/output links (i.e. totally $n\hat{f}\hat{W}$ wavelength channels). The link provisioning cost in a segment has two parts. First, $\hat{n}_l$ links must be provided from the core switch to $\hat{n}_l$ edge switches on average. Second, we must provide $(\hat{n}-\hat{n}_l)$ links between two optical switches (i.e. between two segments) on average. However, we must only account half of the provisioning cost of $(\hat{n}-\hat{n}_l)$ links for each segment. Therefore, to calculate the total link provisioning cost in a segment, we must consider $\frac{1}{2}(\hat{n}-\hat{n}_l)+\hat{n}_l=\frac{1}{2}(\hat{n}+\hat{n}_l)$ links for each segment on average. The total facility cost in a segment is given by

$$C_N(n,\hat{f},\hat{W})=C_{OXC}(\hat{n},\hat{f},\hat{W})+\frac{\hat{n}+\hat{n}_l}{2}C_{lp}+(\hat{n}+\hat{n}_l)C_f(\hat{f},\hat{W})+\hat{n}_l\hat{f}\hat{W}C_\lambda .$$

The average cost of a segment times the number of segments results in the average optical network cost.

4.5.2 Optimization Problem Formulation

We shall develop an ILP formulation to find the cost-effective choice of f, W and W_E parameters in a switch for the desired loss rate. Define ξ to be the maximum slot loss rate at full traffic load (L=1.0) in a segment that the network provider desires to have using

the contention avoidance schemes, while the rest of the contention may be resolved by contention resolution schemes (such as wavelength converters) employed at the core switch.

Let a desired network need at most ω wavelength channels (based on the requirements of the network traffic) on each connection link in a segment to carry the full traffic load (L=1.0) on each wavelength channel. Let the base network configuration parameters be $f=1$, $W=\omega$ and $W_E=0$. New network scenarios can be designed by setting $f=\hat{f}$, $W=\hat{W}$ and $W_E=\hat{W}_E$ where $\hat{f}$, $\hat{W}$ and $\hat{W}_E$ are all variable parameters. Since all new network scenarios are designed for the same network traffic, the total number of wavelengths on each fiber in any scenario is set in a way that $\hat{f}\hat{W}=\omega$. Using $\hat{W}_E$ additional wavelengths, the same traffic for $\hat{W}$ wavelengths can be transmitted over $\hat{W}+\hat{W}_E$ wavelengths. Let $C_N(\hat{n},1,\omega)$ be the base network segment cost, and define $\Delta C_N(\hat{n},\hat{f},\hat{W},\hat{W}_E)=C_N(\hat{n},\hat{f},\hat{W}+\hat{W}_E)-C_N(\hat{n},1,\omega)$ to be the additional cost that the system must sustain for the new configuration. Then, the objective function is given by

$$minimize:\ \Delta C_N(\hat{n},\hat{f},\hat{W},\hat{W}_E)$$

$$Subject\ to:\begin{cases}\hat{f}\hat{W}=\omega\\ R_{L,N}(\hat{n},\hat{f},\hat{W},\hat{W}_E,1.0)\le\xi\\ \hat{W}_E+\hat{W}\le W_{max}\\ \hat{f},\hat{W}=1,2,\ldots,\omega\\ \hat{W}_E=0,1,2,..\end{cases} \tag{4.12}$$

4.5.3 Tradeoff Performance of the Combined Hardware Approach

Since the number of parameters and their values are limited, we use enumeration procedures (we have developed in C++) to solve the above integer problem, and the results are discussed in the following section. Note that we have computed $R_{L,N}(\hat{n},\hat{f},\hat{W},\hat{W}_E,1.0)$ in Section 4.2.2.3. We consider the following parameters in our optimization evaluation: $\hat{n}=\{8,32\}$, $\hat{n}_l=\hat{n}/4$, ω=24, W_{max}=64, and $\eta=\{0.02, 0.04\}$[10].

[10] Using η =0.02, the fiber cost of a fiber with 32 and 64 wavelength channels becomes (see Section 4.5) almost 1.85 and 3.5 times of a fiber cost with a single channel respectively. Conversely, if the fiber cost of a fiber with 32 wavelength channels is twice of a fiber cost with a single channel, we have η =0.0226.

For simplicity, we denote Υ as the cost $C_f(1,1)$ of one fiber with only one wavelength. Different values of Υ represent the fiber cost of different network segments. For example, smaller Υ is for a segment with a shorter average link length, while larger Υ is for a segment with a longer average link length. In the ILP program (Eq.(4.12)), we use the following relative cost (referred to as cost from now on) values: C_{MZ}= 1, C_{SE}= 4, C_λ= 50 based on the cost values used in [SrSo02, EURE05]. We also use ξ = {0.01, 0.05, 0.10, 0.15}, Υ={100, 1000, 10000, 100000}. Clearly, each combination of the cost values results in a particular problem instance. We obtain the optimized combination (f^*, W^*, W_E^*) for each of the $\xi \times \Upsilon$ scenarios. In the optimization, we have considered that software-based contention avoidance schemes such as CST, load balanced and symmetric traffic transmission are used for traffic transmission, in order to minimize the loss at the network. We define ΔC to be the percentage of the network segment cost increase under the optimized parameters when compared to the base network cost. By assuming $C_{lp}=10\Upsilon$, we have $\Delta C = 100\frac{C_N(\hat{n},f^*,W^*+W_E^*)-C_N(\hat{n},1,\omega)}{C_N(\hat{n},1,\omega)}$.

Table 4.4: Optimization Results for $\hat{n}$=8, $\hat{n}_l$ =2, η=0.02

Υ	ξ=0.01				ξ=0.05				ξ=0.10				ξ=0.15			
	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%
10^2	4	6	10	199	4	6	4	112	2	12	10	79	2	12	5	51
10^3	4	6	10	83	2	12	18	47	2	12	10	32	2	12	5	24
10^4	3	8	18	55	2	12	18	32	2	12	10	23	2	12	5	18
10^5	3	8	18	51	2	12	18	30	2	12	10	22	2	12	5	18

Table 4.5: Optimization Results for $\hat{n}$=8, $\hat{n}_l$ =2, η=0.04

Υ	ξ=0.01				ξ=0.05				ξ=0.10				ξ=0.15			
	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%
10^2	4	6	10	193	4	6	4	105	4	6	2	76	2	12	5	48
10^3	4	6	10	85	4	6	4	54	2	12	10	37	2	12	5	22
10^4	4	6	10	66	3	8	8	41	2	12	10	29	2	12	5	18
10^5	4	6	10	64	3	8	8	40	2	12	10	28	2	12	5	17

Tables 4.4 and 4.5 show the optimum results of f^*, W^*, and W_E^* for η=0.02 and η=0.04 respectively, and when $\hat{n}$=8, $\hat{n}_l$=2. For a given cost Υ, the required number of fibers and additional wavelengths is a decreasing function of loss rate ξ. At a higher ξ, the optimum results are almost the same for any fiber cost at both values of η. At a lower ξ, the fiber cost has a direct effect on the number of optimum fibers and additional wavelengths. For example, consider Υ=10^5 and ξ=0.01. The additional wavelength cost is smaller than the fibers cost when η=0.02, and the optimizer has chosen (f^*, W_E^*)=(3,18) (see the gray cells in Table 4.4). However, when the wavelength cost becomes more expensive than fibers at η=0.04, the optimizer chooses (f^*, W_E^*)=(4,10) for a cost-effective design (see the gray cells in Table 4.5).

According to the tables, ΔC is also a decreasing function of ξ for the same Υ at both η=0.02 and η=0.04 because we need a lower number of fibers and additional channels for a higher ξ. Moreover, the base network segment cost is expensive (cheap) at higher (lower) Υ values and the additional hardware cost is less (more) comparable than the base cost. Hence, ΔC is a decreasing function of Υ for the same ξ.

Table 4.6: Optimization Results for $\hat{n}$=32, $\hat{n}_l$=8, η=0.02.

Υ	ξ=0.01				ξ=0.05				ξ=0.10				ξ=0.15			
	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%
10^2	8	3	3	212	4	6	5	132	4	6	3	99	4	6	1	67
10^3	4	6	12	103	3	8	9	65	2	12	12	41	2	12	7	30
10^4	4	6	12	66	2	12	24	41	2	12	12	26	2	12	7	21
10^5	3	8	24	61	2	12	24	37	2	12	12	24	2	12	7	20

Table 4.7: Optimization Results for $\hat{n}$=32, $\hat{n}_l$=8, η=0.04.

Υ	ξ=0.01				ξ=0.05				ξ=0.10				ξ=0.15			
	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%	f^*	W^*	W_E^*	ΔC%
10^2	8	3	3	201	4	6	5	127	4	6	3	94	4	6	1	61
10^3	6	4	5	107	4	6	5	63	3	8	5	46	2	12	7	31
10^4	4	6	12	75	3	8	9	45	3	8	5	33	2	12	7	22
10^5	4	6	12	72	3	8	9	43	3	8	5	31	2	12	7	21

In Tables 4.6 and 4.7, the optimum parameter (f^*, W^*, W_E^*) combinations are obtained for $\hat{n}$=32 and $\hat{n}_l$=8 for the cases η=0.02 and η=0.04 respectively. A similar trend for the choice of number of fibers and additional wavelengths is observed for $\hat{n}$=32. However, since a higher loss rate is expected for a larger $\hat{n}$, the optimum number of hardware is little higher for $\hat{n}$=32 than $\hat{n}$=8. This leads to a higher $\Delta C\%$ in all optimization combinations for $\hat{n}$=32 comparing to $\hat{n}$=8.

In summary, the optimized number of fibers and additional wavelengths depends on the fiber cost. When the fiber cost is expensive (e.g., in wide area networks), the system is cost-effective when using a small number of fibers ($2 \le f^* \le 4$) and a large number of additional wavelengths ($W^*/2 \le W_E^* \le 3W^*$). Otherwise, the network cost will be low when using a bit higher number of fibers ($2 \le f^* \le 8$) and a lower number of additional wavelengths ($1 \le W_E^* \le 2W^*$).

4.6 Concluding Remarks

We have performed the lower-bound analysis on contention of symmetric/asymmetric traffic in single/multi-fiber connections in a slotted-OPS switch. The lowest loss rate is obtained when each ingress switch transmits traffic in a symmetric and load-balanced way to different output links of OPS core switch. In addition, reducing traffic load on the wavelengths, using additional wavelengths to carry the same traffic, and using multi-fiber connections can reduce slot loss rate. Furthermore, composite slot assembly can lead to a lower loss rate at the OPS switch. As another improvement, the coordinated slot transmission is introduced to achieve a noticeable reduction in loss rate at lower loads and in switches with smaller nodal degrees, and achieve the lower bound as predicted. We also provided an optimization model to choose a cost-effective combination of fibers and additional wavelengths in contention avoidance issue. Clearly, when less traffic collides at an OPS switch, network performance is improved. Then, we may even use less contention resolution hardware at the core switch, thus reducing the network cost.

Chapter Five: Prioritized Retransmission

As discussed in Section 2.4, ingress switches are in charge of retransmitting the dropped slots in the OPS core network. The retransmission management is one of the operations of the OBM unit in each ingress switch. Recall that retransmission in the optical domain is one approach to reduce the cost of the expensive contention resolution hardware in slotted OPS network. By developing the retransmission technique in our network, we can achieve a loss-free all-optical OPS network. In this chapter, we analyze a simple but effective prioritized retransmission technique. After detailing the prioritized retransmission technique in a multi-hop network, we analyze this technique in a single-hop network for simplicity purposes. In our performance evaluation, we show the effectiveness of this technique in improving the TCP throughput.

5.1 Prioritized Retransmission (PR) Protocol

Recall that our network is a contention-based network in which traffic may collide and drop due to unavailability of network resources. We can improve the network performance by retransmitting the dropped traffic in the optical domain. As discussed in the retransmission model in Section 2.6, each ingress switch stores a copy of the transmitted traffic in a slot in its electronic buffer for possible retransmission whenever required. Unlike the common RR (Random Retransmission) technique as reviewed, we develop a different but simple and effective retransmission technique to recover the dropped slots and to limit the number of retransmissions. Called the Prioritized Retransmission (PR), dropped traffic is prioritized when retransmitted from an ingress switch under the PR protocol. Each core switch would then process the prioritized traffic with a higher priority. In this way, the number of retransmissions is limited and the network performance is improved. In the following, we detail the PR protocol.

Consider an ingress-egress switch pair. Let the number of hops between the ingress switch and the egress switch be h. In PR, the priority of a newly transmitted slot is set to h-1, i.e., zero in a single hop network (h=1). This initial priority gives a better chance to the slots in longer-hop connections to pass through the network. Then, whenever a slot from the ingress switch is dropped at the core switch, its priority is increased by one at

the ingress switch when re-transmitting. Therefore, for a 5-hop path, new slots have an initial priority 4, and increased every time it has to be retransmitted. So even if a newly transmitted slot has a small chance to pass through the core switch in its first transmission during heavy traffic, the higher priority it would acquire in its future retransmission(s) will help it to pass through the core switch, thus cutting down the number of retransmissions. This issue also improves the fairness among different traffic streams. This is opposite to the RR technique where one slot may pass through the core switch at the first transmission, and another slot may be retransmitted almost forever (theoretically).

The PR protocol is simple since it only requires few bits to keep the priority of each slot. Thus, no complexity is added to the control layer. For each slot a priority field is assigned that can only be updated at its ingress switch. As mentioned, the SSH header is sent in advance with the information for each one of the slots in a slot-set. The SSH header carries the priority of each slot as well. This field is the criteria on which the core switch decides to prioritize the slots and find the eligible slot(s) for switching. On the other hand, each ingress switch sets the slot priority field with the retransmission number. Each time a slot is returned to the edge switch, its priority field is incremented by one and then retransmitted.

An optical core switch in the network first receives the information of slots at the same time in its input ports. After receiving a SSH and during the slot-offset interval, it evaluates and then resolves the potential contention, and makes the wavelength switch ready to switch the incoming slots to desired egress switches. During contention resolution, the retransmitted slots are given a higher priority to pass through the core switch. For example, if there are slots of up to two retransmissions competing to a tagged output link, then the available output ports of the link must be first given to those slots that are retransmitted for the second time, then those for the first time, and finally the newly transmitted slots (for whatever ports remaining). Note that in the optical switch under consideration, up to f slots with the same output link on the same-wavelength and at the same time-slot can be switched without any collision. Recall from Section 4.2.2.2 that using more fibers can lead to a less contention at the core switches.

For a successful slot transmission, no ACK command is required to be sent back to

the related ingress switch. However, when a slot is dropped, the related slot code is encapsulated in a NACK command and sent back to the ingress switch over the control channel to identify the blocked slots. If several slots from the same ingress switch are going to be dropped then the information of all dropped slots are transmitted back in a single NACK command to that ingress switch. The ingress switch is responsible for re-transmitting any blocked slot while increasing its priority number.

In the following, we shall analyze the PR protocol. To simplify the PR analysis, a single-hop OPS network with a multi-fiber architecture and without using any contention resolution scheme is considered. Each ingress switch transmits symmetric traffic to each egress switch in our analysis. The multi-fiber switch architecture would implement our contention avoidance scheme as discussed in Chapter 4.

5.1.1 Slot Loss Rate

At each *nf-slot-set* (recall *nf-slot-set* is the set of *nf* slots on the same-wavelength from n ingress switches that simultaneously arrive at a core switch) in a given time-slot, at most *nf* non-empty slots may arrive at the core switch on the same wavelength channel from n ingress switches on f fiber links. Let L be the normalized traffic load on each wavelength channel of each fiber. Alternatively, L also represents the probability of a non-empty slot arriving on each wavelength. On the average, there are $N=nfL$ non-empty slots in the *nf-slot-set.*

Let H be the maximum number of retransmissions required to transmit a tagged slot successfully so that a slot with priority H is dropped with a probability of less than X where $X<<1$. The parameter X is considered to be a rare event probability. In other words, there may be a slot that is retransmitted for more than H times with the probability of less than X. We start our analysis at level H and then analyze the drop probability in the next level down. We repeat the analysis process one level at a time until we reach level zero in order to determine all the slot retransmission probabilities. At each level j in the analysis, we refer to the level-j retransmitted slots as the tagged-priority slots, and all other non-empty slots in the *nf-slot-set* are referred to as non-tagged-priority slots. Therefore, the non-empty slots in an *nf-slot-set* at any retransmission level can be divided into two groups: the tagged-priority slots and the non-tagged-priority slots from all other levels.

Define n_r to be the average number of non-empty tagged-priority slots at any level. The remaining N-n_r slots are the average number of the non-tagged-priority slots.

We would like to obtain the loss rate for the tagged-priority slots over one wavelength. Similar to Section 3.1.1, define a *c-slot* collision as the event that there is a collision at the core switch from any c slots (from the pool of n_r tagged-priority slots at a given level) going to the same output link on the same wavelength. Let P_c = *Prob.{Having a c-slot collision event in the nf-slot-set at the tagged output link among all tagged-priority slots}*. By assuming uniform slot transmission from the ingress switches and equal slot distribution (with a probability *1/n*) to each output link, the collision probability P_c can then be approximated by a binomial distribution

$$P_c = \binom{n_r}{c}(1-\frac{1}{n})^{n_r-c}(\frac{1}{n})^c \quad , 1 < c \le n_r \quad .$$

Since there are n_r tagged-priority slots and we assume equal slot distribution to each output link, the average number of delivered slots to the tagged output link is $\overline{L_d} = \frac{n_r}{n}$. Let k be the number of available fibers at the tagged output link for the transmission of at most k slots. Recall that at most f slots can be transmitted through each output link. Define slot loss rate $R_L\{n_r, k\}$ to be the drop probability among the n_r tagged-priority slots going to the tagged output link given only k fibers available at the tagged output link. Since the average number of the lost slots due to more than k slots going to the tagged output link port(s) is $\overline{d} = \sum_{c=k+1}^{n_r}(c-k)P_c$, we have

$$R_L\{n_r, k\} = \frac{\overline{d}}{\overline{L_d}} = \frac{\sum_{c=k+1}^{n_r}(c-k)P_c}{\frac{n_r}{n}} = \frac{\sum_{c=k+1}^{n_r}(c-k)\binom{n_r}{c}(n-1)^{n_r-c}}{n^{n_r-1}n_r} \quad \text{for } n_r \ge 1. \tag{5.1}$$

$R_L\{n_r, k\} = 0$, otherwise.

Since the upper limit n_r of the summation accounts for the average number of non-empty tagged-priority slots, it could be a real number. Using the binomial expansion formula, Eq.(5.1) can be approximated as follows where the upper-bound of the summations become integer values:

$$R_L\{n_r,k\}=\frac{(n-1)^{n_r}}{n^{n_r-1}n_r}\{\sum_{c=k+1}^{n_r}c\binom{n_r}{c}(\frac{1}{n-1})^c - k\sum_{c=k+1}^{n_r}\binom{n_r}{c}(\frac{1}{n-1})^c)\}$$

$$=\frac{(n-1)^{n_r}}{n^{n_r-1}n_r}\{\frac{n_r}{n-1}\sum_{c'=k}^{n_r-1}\binom{n_r-1}{c'}(\frac{1}{n-1})^{c'}(1)^{n_r-1-c'} - k\sum_{c=k+1}^{n_r}\binom{n_r}{c}(\frac{1}{n-1})^c(1)^{n_r-c})\}$$

$$=\frac{(n-1)^{n_r}}{n^{n_r-1}n_r}\{\frac{n_r}{n-1}(\alpha^{n_r-1}-\sum_{c'=0}^{k-1}\binom{n_r-1}{c'}(\frac{1}{n-1})^{c'})-k(\alpha^{n_r}-\sum_{c=0}^{k}\binom{n_r}{c}(\frac{1}{n-1})^c)\} \quad ,$$

where $\alpha=1+\frac{1}{n-1}=\frac{n}{n-1}$, and $n_r \geq 1$. Note that when there is some contention resolution hardware at the optical switch, the loss function, $R_L\{n_r, k\}$, must be calculated accordingly.

5.1.2 Prioritized Retransmission Analysis

We would like to analyze the steady state distribution of the prioritized slot retransmission. Note that there may not be enough capacity to transmit all retransmitted slots at level *j*, and some slots must be dropped. Subsequently, the priority of a dropped slot will increase by one at its ingress switch, and it will be retransmitted as a level *j*+1 priority slot.

Let $\mathbf{\Pi_H}=\{\pi_0\ \pi_1\ \pi_2\ \ldots\ \pi_H\ \pi_{H+1}\}$ denote the steady-state probability vector, in which at most *H* levels of slot retransmissions are required in an *nf-slot-set* to the tagged output link, where π_0 is the transmission probability of new slots; and π_i is the probability that a slot is retransmitted for the *i-th* time. For example, of those slots retransmitted with the probability π_1, some of them will be retransmitted in the future with the probability π_2 and the remaining will be switched at the core switch with the probability π_1-π_2.

By considering $N=nfL$ as the average number of non-empty slots over the same wavelength cannel in the *nf-slot-set*, the set $\mathbf{N_R} = N\mathbf{\Pi_H} = \{N\pi_0 \quad N\pi_1 \quad N\pi_2 \quad \ldots \quad N\pi_H\}$ indicates the average number of the slots at each retransmission level. For example, $N\pi_1$ denotes the average number of the non-empty slots in the *nf-slot-set* that are retransmitted for the first time.

Let $P_{i,drop}$ denote the probability of losing a slot at the *i-th* retransmission level. Since the dropped slots at level *i*-1 (with a probability $P_{i-1,drop}$) will be retransmitted as level *i* priority slots with a probability π_i, we have

$$\pi_i = P_{i-1,drop} \quad , 1 \le i \le H \quad . \tag{5.2}$$

We also should have

$$\pi_{H+1} = P_{H,drop} < X \, . \tag{5.3}$$

and $\sum_{j=0}^{H} \pi_j = 1$. Therefore, we can determine the arrival probability of the new slots as

$$\pi_0 = 1 - \sum_{j=1}^{H} \pi_j \quad . \tag{5.4}$$

The set of $H+1$ equations (Eq.(5.2), Eq.(5.3) and Eq.(5.4)) can be solved to obtain the variables in $\mathbf{\Pi_H}$. To solve these equations, we now need to find $P_{i,drop}$. We first consider the probability that the retransmitted slots at levels $i+1$ and higher have already occupied the output fibers of the tagged output link.

Let L_k denote the number of slot arrivals to the tagged output link at retransmission level k and higher. Let $P_a\{L_k=\alpha\}$ be the arrival probability of α slots (α is an integer number between 0 and f) to the tagged output link at retransmission level k and higher so that α ports of the tagged output link are occupied. According to the main assumptions that slot arrivals to the tagged output link are uniform, and that the normalized traffic load of L is equal on all fibers, then we have

$$P_a\{L_k = \alpha\} = \binom{f}{\alpha} (L\pi_k)^{\alpha} (1 - L\pi_k)^{f-\alpha} \, ,$$

where $L\pi_k$ is the probability of one slot arrival (and occupation) to the tagged output link at retransmission level k and higher. Let $P_{i,\alpha,drop}$ be the probability that a tagged-priority slot is dropped at retransmission level i, provided that α slots have already arrived and occupied the tagged-output link ports at retransmission level $i+1$. Then

$$P_{i,\alpha,drop} = \frac{n_{r,i} P_a\{L_{i+1} = \alpha\} R_L\{n_{r,i}, f - \alpha\}}{N} \, ,$$

where the numerator represents the total number of dropped slots among $n_{r,i} = N\pi_i$ tagged-priority slots with priority i in the *nf-slot-set*. The second factor in the numerator indicates the probability of α slot arrivals at the levels $i+1$ and higher, and the third term represents the drop probability (calculated in Eq.(5.1)) among $n_{r,i}$ tagged-priority slots

given that only f-α slots capacity are available at the tagged output link. The denominator is the average number of non-empty slots in the *nf-slot-set*. The marginal probability $P_{i,drop}$ is

$$P_{i,drop} = \sum_{\alpha=0}^{f} P_{i,\alpha,drop} = \sum_{\alpha=0}^{f} \frac{n_{r,i} P_a \{L_{i+1} = \alpha\} R_L \{n_{r,i}, f-\alpha\}}{N} \quad . \tag{5.5}$$

By simplifying Eq.(5.5), we obtain the loss rate for the retransmission level i, $P_{i,drop}$, from

$$\pi_{i+1} = P_{i,drop} = \pi_i \sum_{\alpha=0}^{f} \binom{f}{\alpha} (L\pi_{i+1})^{\alpha} (1-L\pi_{i+1})^{f-\alpha} R_L \{nfL\pi_i, f-\alpha\} \quad . \tag{5.6}$$

Note that the parameter n_r in the function R_L $\{n_r, k\}$ in Eq.(5.6) is not necessarily an integer value because of multiplying N by some probability values. Thus, we must use the Gamma function to calculate the *factorial(n)*, $n! = \Gamma(n+1)$, and the *choose* function $\binom{m}{n} = \frac{m!}{n!(m-n)!}$ in the equations above.

We now compute the average number of retransmissions $\overline{N_R}$. First, recall that $N\pi_0$ is the average number of newly generated slots that enter in the network. Of $N\pi_j$ slots retransmitted at level j, some slots are dropped and will be retransmitted in the future with a probability of π_{j+1}, while the remaining $N\pi_j$ - $N\pi_{j+1}$ slots will be switched at the core switch. In other words, of the $N\pi_0$ slots, $N\pi_j$ - $N\pi_{j+1}$ slots are successfully transmitted at level j. Then, the probability of successful transmission after j retransmissions is $\frac{N\pi_j - N\pi_{j+1}}{N\pi_0} = \frac{\pi_j - \pi_{j+1}}{\pi_0}$. Therefore, $\overline{N_R}$ is given by

$$\overline{N_R} = \sum_{j=1}^{H} j \frac{\pi_j - \pi_{j+1}}{\pi_0} \quad ,$$

where the second fraction term in the summation is the probability of successful transmission after j retransmissions.

5.1.3 Determining the Maximum Level of Retransmission *H*

We can now solve the H+1 equations in Eq.(5.2), Eq.(5.3) and Eq.(5.4) by the iteration method for different values of H, and then determine the worst case H. Note that by increasing the loss rate, the number of slot retransmissions is also increased. Therefore,

the worst case happens when the loss rate in Eq.(5.1) is maximized. Since the function inside the summation in this equation is an increasing function of n_r, the maximum loss rate happens when $n_r = n_{r,i} = nfL\pi_i$ (at any retransmission level) is maximized and the parameter k in Eq.(5.1) is minimized. The minimum value for the parameter k is obtained when f=1. The maximum value for $n_{r,i}$ is obtained when n and L are maximized. By assuming L=1.0, $X=10^{-7}$ and n=1000 as the largest possible optical switch dimension, and solving the $H+1$ equations, we find with eight precision digits that $\pi_0=0.63230406$, $\pi_1=0.30779933$, $\pi_2=0.05816581$, $\pi_3=0.00172963$, $\pi_4 = 0.00000064$ and $\pi_5 << X$. Therefore, we have H=4 that will guarantee us the required loss level.

5.1.4 Transmission Delay Analysis

We would like to study the average delay experienced by a slot after the initial transmission. Assume that each slot takes one unit of time for transmission. Packets of a given torrent are encapsulated in slots, and depart in intervals with a mean of m-slots. Assume each dropped slot from the tagged-stream can be retransmitted at the first available departure slot.

Define $\overline{D_T}$ to be the average transmission delay experienced by a slot from the time leaving an edge switch for the first time until it is successfully switched at the core switch. Further, define D_P to be the one-way propagation delay between the edge switch and the core switch as an integer number of slots. Then the average transmission delay for the slot is

$$\overline{D_T} = 1 + D_P + \overline{N_R}\ (2\,D_P + m/2)\ ,$$

where the first term is the transmission time for each slot, the second term is the propagation time to the core switch and the third term represents the retransmission delay. In the retransmission delay, twice of the propagation delay and average time of $m/2$ waiting time before retransmission are included. Since in optical networks, the propagation delay can be the dominant parameter in the average transmission time, we could have

$$\overline{D_T} = 1 + D_P + \overline{N_R}\ (2D_p + m/2) \approx D_P\,(1 + 2\overline{N_R})\ .$$

Note that the delay analysis is also true for the RR technique.

5.1.5 Scheduling at the Core Switches

We shall detail how a core switch resolves the contention happened at a given output link. Let set $S_{i,l}=\{s_0, s_1, ..., s_{l-1} \mid 0 < l \leq nf\}$ denote the subset of l contending slots on wavelength w_i $(i=0,..., W-1)$ at the output link. Let the vector (Src_j, ID_j, r_j) denote three parameters for slot s_j carried in SSH, where ID_j is the slot ID, Src_j is the ingress switch address, and r_j is the priority of slot j. We have $r_j=0$ under RR. In PR, the slots in set $S_{i,l}$ are sorted in a descending order according to the priority value. Then the top $min(f,l)$ slots from the sorted list are picked for transmission first and the remaining $l-min(f,l)$ slots in $S_{i,l}$ are either dropped or allowed to resolve their conflicts (say on other wavelength channels). When slot s_j is dropped, the core switch announces to Src_j with a NACK command (under both PR and RR). In the NACK command, the ID_j parameter is sent to Src_j. On the other hand, any contention resolution scheme can be used provided that slots with higher priority values are first resolved. In this way, a slot with a higher priority always finds a higher opportunity to pass through the core switch. This contrasts with the RR technique where slots are randomly chosen for transmission. Obviously, the PR technique has a higher complexity than the RR technique due to the extra sorting procedure. At the end, the core switch makes a new SSH for the slots departing from the output link, and then transmits the SSH on the control channel of the output link to the next hop core switch or edge switch.

5.2 Random Retransmission (RR)

To allow comparison to the common RR approach, we also provide an analytical model for the RR technique in a multi-fiber switch where there is no difference between the newly generated slots and the previously retransmitted slots. Using a geometric distribution for the retransmission, the probability of k transmissions until a success is

$$P\{k \ transmissions \ until \ success\} = \pi_{k-1} = \pi_0(1-\pi_0)^{k-1} \ ,$$

where π_0 is the probability of a success as discussed in Section 5.1.2. In the multi-fiber switch, π_0 can be obtained from

$$\pi_0=1- R_L\{N, f\}=1- R_L\{nfL, f\} \ ,$$

where the second term has already been obtained. The average number of retransmissions for the RR technique is obtained from

$$\overline{N_R} = \sum_{k=1}^{\infty} k\pi_0(1-\pi_0)^k - 1 = \frac{1}{\pi_0} - 1 = \frac{1}{1\text{-}R_L\{nfL, f\}} - 1 \ .$$

5.3 Performance Evaluation

We have prepared two simulation scenarios based on our network model in order to compare the performance of PR and RR for slotted traffic transmission in an all-optical network. In our analysis, we use the Gamma function to calculate the function *factorial*, $x! = \Gamma(x+1)$, and the function *choose* to determine the number of combinations $\binom{n}{c}$. The analysis results for the PR technique are obtained using $X=10^{-7}$ and $K=4$. We have used C language and OPNET to implement our computations and simulations.

Table 5.1: PR Retransmission Performance at L=1.0, f=1 and n=10 or 100.

Retrans. Prob. f=1	*n = 10*		*n = 100*	
	Ana.(%)	Sim.(%)	Ana.(%)	Sim.(%)
π_0	65.0851	65.2061	63.4002	63.4901
π_1	30.6857	29.9696	30.7739	30.6612
π_2	4.2293	4.7200	5.6860	5.6825
π_3	0.0000	0.1042	0.1398	0.1662
π_4	0.0000	0.0001	0.0000	0.00014
$\overline{N_R}$	0.536452	0.533599	0.577281	0.575051

5.3.1 Comparison of PR and RR in a Single-Hop Network

We shall compare the performance of PR and RR in a single-hop all-optical network under Poisson and Pareto traffic arrival processes. In this section, we shall use no contention resolution scheme to reduce the slot loss rate at the core switch. For each scenario, enough number of replications is run to achieve 95% level of confidence intervals, at most, within 1% of the mean values shown.

5.3.1.1 Validation of the PR and RR Analysis: Poisson Traffic

To verify the correctness of our analysis, we consider a scenario where each fiber contains W=4 wavelengths and each ingress switch transmits slots to different output links of the core switch with equal probabilities over $fW=4f$ wavelengths. Each ingress switch generates fixed-length packets according to a Poisson arrival process with a mean

rate of L packets per time-slot symmetric to each egress switch, and transmits randomly to the egress switches. In this scenario, each slot carries only one packet. In the simulations, more than 1,000,000 *nf-slot-sets* are generated from all ingress switches.

A) Retransmission by the PR Technique: The retransmission analysis and the simulation results for n=10 and n=100 edge switches under symmetric traffic load of L=1.0 and in a single fiber network are showed in Table 5.1. Both the analysis and the simulation results for n=10 and n=100 indicate that with two retransmissions, most part of the dropped slots can pass through the optical switch and less than 0.2% would require a third retransmission. For n=10, the table shows that 65% of the traffic at the optical switch belongs to the first arrival, and almost 30% of the slots must be retransmitted for the first time. Then, almost 4.7% and 0.1% of the slots require the second and third retransmissions respectively. There is no observed slot drops after the fourth retransmission. A similar behavior is observed for n=100.

Table 5.2: PR Retransmission Performance at L=0.7, n=100 and f=1 to 4

Retrans. Prob. n=100	f=1		f=2		f=3		f=4	
	Ana. (%)	Sim. (%)	Ana. (%)	Sim. (%)	Ana. (%)	Sim. (%)	Ana. (%)	Sim. (%)
π_0	72.1675	72.1385	83.3811	83.1128	88.2615	88.1256	91.1174	91.0430
π_1	25.4043	25.3403	16.5002	16.7485	11.7334	11.8678	8.8825	8.9568
π_2	2.4196	2.4995	0.1188	0.1387	0.0050	0.0066	0.00016	0.00026
π_3	0.0111	0.0218	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
π_4	0.0000	0.00001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
$\overline{N_R}$	0.38570	0.38622	0.199314	0.203184	0.13300	0.134744	0.097486	0.098383

Table 5.2 displays the PR results in a multi-fiber (f=1,2,3,4) architecture for n=100 at L=0.7. One can see that the analysis and the simulation results agree well with each other. The results show that by increasing the number of fibers, the probability of the retransmission at all levels is decreased. Recall from Section 4.2.2.2 that using more fibers can lead to a less contention at the core switches. One can see that for a higher number of fibers, most of the dropped slots can pass through the core switch even with one retransmission. It is interesting to see that for f=1 and f=4, a slots is retransmitted for the second time with a probability of almost 0.025 and 0.000003 respectively. This is

because by increasing the number of fibers, the collision rate is reduced and therefore we expect a lower number of retransmissions.

Table 5.3: RR Retransmission Performance at L=1.0 and f=1, n=10 or 100

Retrans. Prob. f=1	n = 10		n = 100	
	Ana.(%)	Sim.(%)	Ana.(%)	Sim.(%)
π_0	65.1322	65.1986	63.3968	63.4881
π_1	22.7102	22.6582	23.2053	23.1258
π_2	7.9185	7.8927	8.4939	8.4585
π_3	2.7610	2.7588	3.1090	3.1067
π_4	0.9627	0.9658	1.1380	1.1465
π_5	0.3357	0.3416	0.4165	0.4229
π_6	0.1170	0.1206	0.1525	0.1574
$\overline{N_R}$	0.535339	0.534948	0.577367	0.577242

Table 5.4: RR Retransmission Performance at L=0.7, n=100 and f=1 to 4

Retrans. Prob. n=100	f=1		f=2		f=3		f=4	
	Ana. (%)	Sim. (%)	Ana. (%)	Sim. (%)	Ana. (%)	Sim. (%)	Ana. (%)	Sim. (%)
π_0	72.1659	72.1155	83.1425	83.1029	88.1495	88.1296	91.0658	91.0378
π_1	20.0867	20.0887	14.0158	13.8817	10.4462	10.3564	8.1360	8.0587
π_2	5.5910	5.6111	2.3627	2.4522	1.2379	1.3078	0.7269	0.8019
π_3	1.5562	1.5734	0.3983	0.4529	0.1467	0.1766	0.0649	0.0894
π_4	0.4332	0.4409	0.0671	0.0879	0.0174	0.0252	0.0058	0.0108
π_5	0.1206	0.1233	0.0113	0.0179	0.0021	0.0037	0.00052	0.0013
π_6	0.03356	0.0343	0.0019	0.0037	0.00024	0.0006	0.00005	0.0002
$\overline{N_R}$	0.385696	0.387121	0.20275	0.206144	0.134436	0.136258	0.098107	0.099816

B) Retransmission by the RR Technique: Table 5.3 displays the analysis and the simulation results for the RR technique at n=10 and n=100 for L=1.0 and f=1. The retransmission probabilities are smaller for n=10 than n=100 according to the results. The rate of the retransmission reduction at each level is smoother than what we observed for the PR technique (see Table 5.1). For example, the third retransmission occurs with a probability of 0.001 for the PR technique and a probability of 0.02761 for the RR technique.

Table 5.4 depicts the RR results for a variety of fibers, f=1,2,3,4 at L=0.7 and n=100. By increasing the number of fibers, the retransmission level is decreased as we have

already observed for PR. However, the reduction rate is not as fast as the PR technique (shown in Table 5.2), thus showing the efficiency of our PR technique.

5.3.1.2 Validation of the PR and RR Analysis: Bursty Traffic

Here, eight edge switches are sending slots to the optical switch in a single-hop network where edge switches use the DTDM protocol (to be discussed in Chapter 6) to access to the network. Each edge switch is located at a distance of 1000km from the core switch. To make our simulated traffic as close to the real traffic as possible, we make each ingress switch generate variable-length IP packets to each egress switch with the length distribution mentioned in [CAID06]. That is 46% of size 40B(Bytes), 18% of size 552B, 18% of size 576B, and 18% of size 1500B. To model the bursty traffic [PaFl95], packet inter-arrival times are identically and independently distributed according to a Pareto distribution with a p.d.f $p(t)=\alpha b^{\alpha} t^{-\alpha-1}$. We choose the Hurst parameter 0.95 to make our model more realistic than the Poisson model [PaFl95]. In each ingress switch, the distribution of traffic to each egress switch is uniform. There are f fibers on each link between an edge switch and the core switch. Each fiber has one wavelength. The transmission rate on a wavelength is 10 Gbits/s. We set S_T=9μs and S_O=1μs, and therefore, an integer number of packets up to 90Kbits can be aggregated in a slot and transmitted to the core switch. Up to 20 retransmissions are captured in our simulation. For each scenario, enough replications are run to achieve a 95% level of confidence intervals to within 5% of the mean values shown. The average measured traffic load on wavelength channels in this scenario is almost L=0.9.

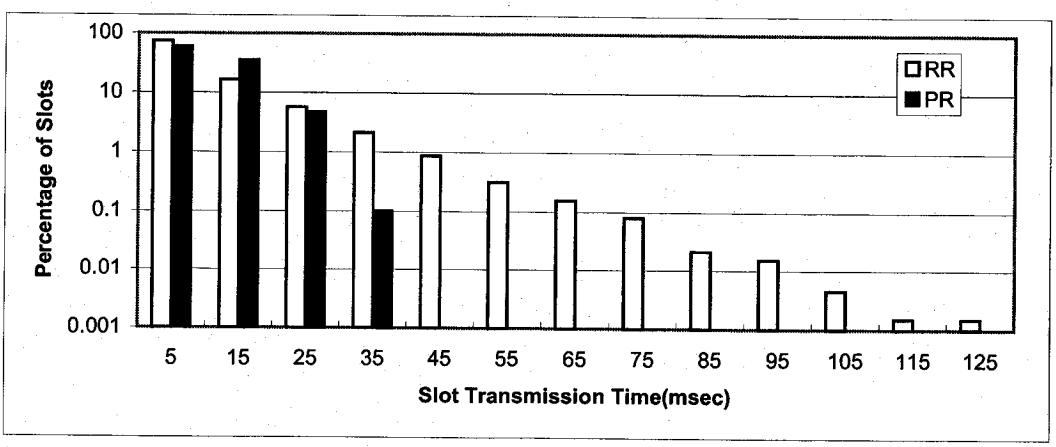


Fig.5.1: Slot Transmission Time Distribution under PR and RR

The transmission time distribution from the time a slot is generated until successful switching at the optical switch is depicted in Fig.5.1 at f=1. PR has reduced the retransmission time to 35msec (no more retransmissions observed after) meaning the maximum delay between two consecutive slots is at most 30ms approximately. However, this delay has gone beyond even 120msec under RR. The amount of newly transmitted traffic under RR is also higher than PR.

Table 5.5: Average Transmission Delay under PR and RR at f=1, f=3

	f=1		f=3	
	Sim.(ms)	**Ana. (ms)**	**Sim. (ms)**	**Ana. (ms)**
RR	9.547126	9.65663	8.20023	7.223445
PR	9.654381	9.663036	8.458367	7.077953

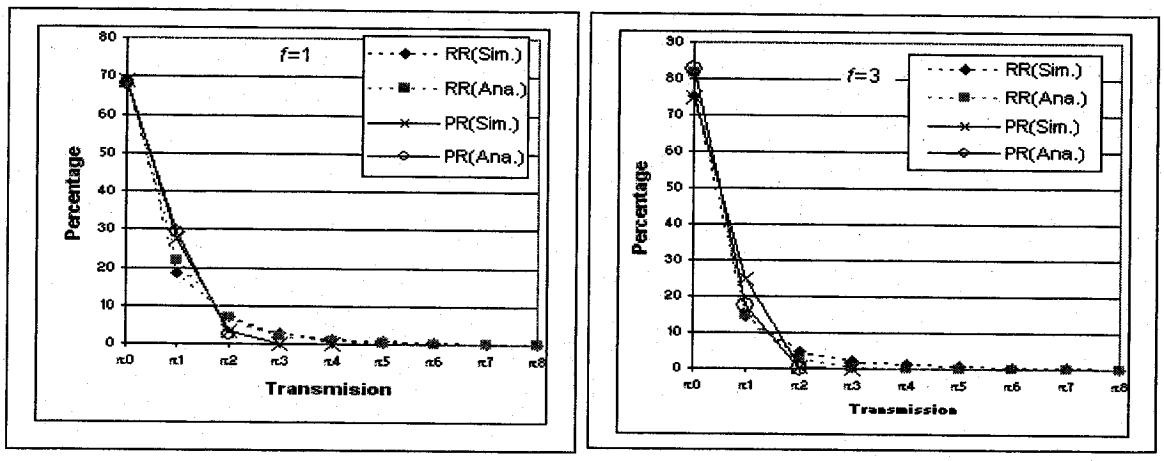


Fig.5.2: Transmission Percentage under PR and RR at f=1 and f=3

Table 5.5 compares the simulation and analysis results for the average transmission time for PR and RR at f=1 and f=3. The average transmission time is almost the same for both PR and RR, although the simulation results show a slightly higher average transmission time for PR. This may be due to a smaller number of retransmissions captured in RR.

Note that the performance is dependent on the traffic arrival at the edges switches. Although the packet generation in each ingress switch is uniform to each egress switch, this uniformity is only true in longer intervals under the bursty traffic. Therefore, the slots going to different egress switches will be non-uniform at the core switch. Moreover, during the simulation in a multi-fiber network, there may be more than one slot transmitted to the same output link, (i.e. collision on the output channels of the edge

switch) over the same channel of different fibers. This issue and the non-uniformity have led to a higher collision rate and therefore a longer transmission delay. This may explain the observation that there is a bigger difference (more than 1ms) between the analysis and simulation results for f=3.

Fig.5.2 illustrates the transmission percentage for up to eight retransmissions at f=1 and f=3. Based on the results, there is no more than four retransmissions observed for PR at f=1. However, we have observed up to 20 retransmissions for RR in our simulation. The observed number of retransmissions is only three when using PR with f=3.

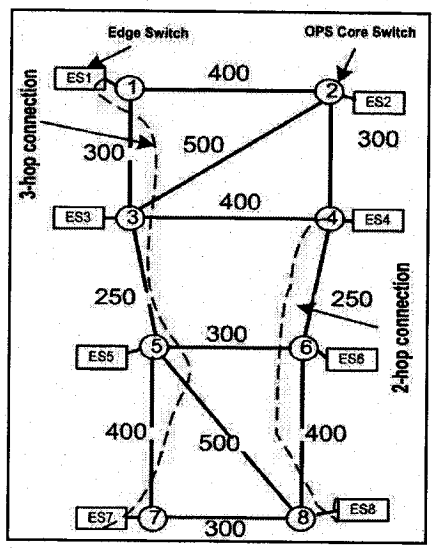


Fig.5.3: A Multi-hop All-Optical OPS Network

5.3.2 Comparison of PR and RR in a Multi-hop Network

We compare the performance of PR and RR as well as the NR (No Retransmission) case in an all-optical OPS network in Fig.5.3 where the distances are in km. This network has eight all-optical core switches. One edge switch is connected to each all-optical core switch. The number of fibers on each link is f=1 and each fiber carries W=4 wavelengths. The transmission rate on each wavelength is 2.5Gbits/s. Each core switch chooses the shortest path (measured in number of hops) to route the slots between an ingress-egress switch pair. Each edge switch uses the DTDM protocol (to be discussed in Chapter 6) to send TCP traffic to the remaining seven edge switches in the network. There are 700 TCP source nodes and 700 TCP destination (sink) nodes connected to each edge switch. In an ingress switch, we divide the 700 TCP streams equally among the seven other egress

switches in the network so that there are 100 TCP connections (referred to as a group connection) between any ingress and egress switch pair in the network. All together, there are 8×7=56 TCP group connections among the eight edge switches in the network. The TCP group connections can be grouped based on their hops as single-hop connections, two-hop connections and three-hop connections (e.g., a TCP group connection between ES1 and ES7 is a three-hop connection). We use TCP Reno [Stev94, Stev01] with an average packet size of 576 bytes in our experiment. In an ingress switch, we use a buffer size of 10,000 packets for each egress switch. Recall that unlike RR, the required number of retransmissions for the PR technique is very small and therefore we expect having to retransmit more than even five times a very rare event. In any case, if a slot is not reached to its destination after 18 retransmissions, it is dropped.

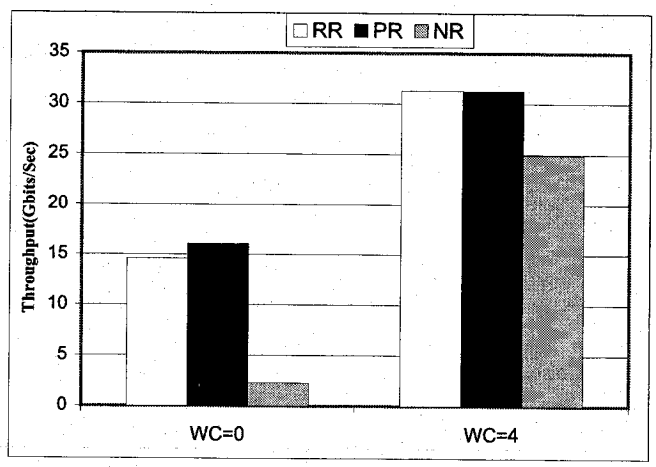


Fig.5.4: TCP Network-wide Throughput

Fig.5.4 compares the TCP throughput within the network under the PR, RR, and NR schemes. Two scenarios are investigated: core switches without any wavelength converter (WC=0) and core switches with four share-per-node wavelength converters (WC=4). One can see that the both retransmission techniques significantly increase the TCP throughput when compared to the NR case. Also, the throughput is significantly increased under all schemes when using wavelength converters, which reduces packet drop at the core switches. The PR has the highest throughput than the other schemes under both scenarios. This is because TCP throughput is dependant on the arrival of ACK commands from receivers. Under RR, an ACK command may take a long time to arrive in the TCP source node. Therefore, the TCP source node would assume a collision in the

network and would reduce its sending rate. This would in turn reduce the TCP throughput in the network.

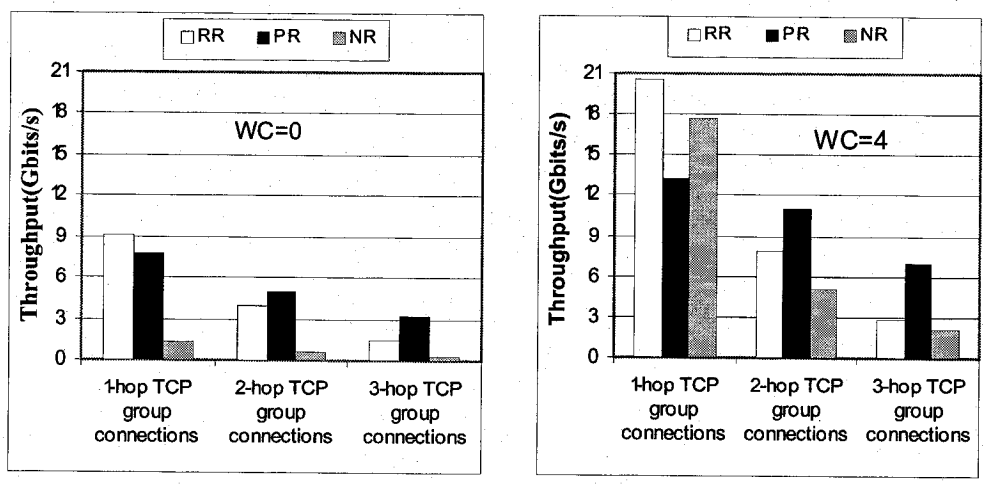


Fig.5.5: TCP Throughput vs. Number of Hops (with and without WCs)

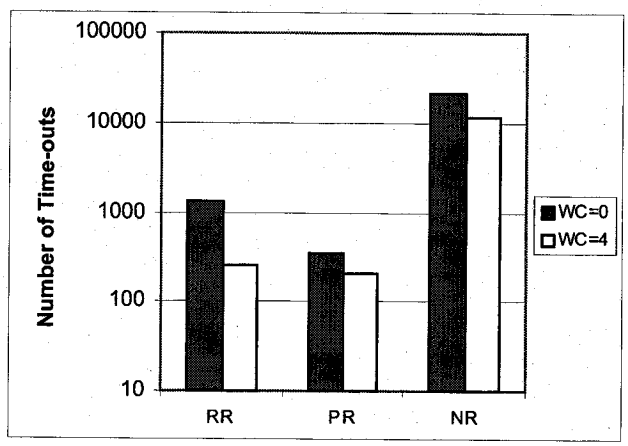


Fig.5.6: Number of Time-out Events in TCP Sources

Fig.5.5 compares the average TCP throughput (in the network shown in Fig.5.3) of all 1-hop TCP group, all 2-hop TCP group, and all 3-hop TCP group connections under the PR, RR, and NR techniques in two scenarios: when using no wavelength converter in the core switches (WC=0) and when using four shared-per-node wavelength converters in each core switch (WC=4). The 1-hop connections have a lower throughput under PR, while the 2-hop and 3-hop TCP group connections all have better throughput in both scenarios under PR than either RR or NR. This is because of the higher initial slot priority (i.e. h-1 as discussed in Section 5.1) that is assigned to the packets belonging to longer-hop connections so that they can find a higher opportunity to pass the network. One can even see a better throughput performance of PR than other techniques for 3-hop

TCP connections. In short, PR provides a better resource allocation for TCP connections than NR and RR.

Fig.5.6 illustrates the number of TCP time-out events (in a logarithmic scale) in TCP sources occurred during the simulation process. Comparing to the NR technique, the retransmission technique has significantly decreased the number of time-out events. The number of time-out events can be further decreased at WC=4. Overall, PR has the smallest number of time-out events.

5.4 Conclusion

The Prioritized Retransmission (PR) technique is shown to be a very simple but efficient protocol for the slotted all-optical OPS networks. Our analysis and simulation results show that this technique can limit the number of retransmissions, and can perform better than the conventional RR under any number of fibers and traffic loads so that in the worst case (i.e. single fiber architecture under full traffic load with no contention resolution scheme used in the core switch) PR can pass most of the dropped slots through the core switch in two retransmissions. Clearly, by limiting the number of retransmissions, the extra load injected into the network due to the higher layer retransmissions can be reduced, which in turn helps to increase the network throughput. We showed that the network throughput has been increased under PR. The TCP throughput can also be distributed much better for longer-hop connections so that for 3-hop TCP connections the PR throughput is more than twice of the RR throughput. In addition, we showed that a multi-fiber architecture can be a good approach to reduce contention in OPS network and to reduce the required number of retransmissions so that in a four-fiber network one retransmission is enough under PR to pass the dropped traffic through the core switch. In summary, the PR technique has a much better performance when the loss rate is high, i.e. when traffic load is high, or network has a single-fiber architecture, or when a smaller number of contention resolution schemes are used at the core switches.

Chapter Six: The Distributed TDM (DTDM) Protocol

Ingress switches are in charge of providing access to OPS network for traffic streams, as discussed in Section 2.4. We complement our ingress switch design here by designing the Distributed TDM (DTDM) protocol that would provide access to a slotted all-optical packet-switched network within the DiffServ domain. In designing DTDM, we use some of the software-based contention avoidance ideas discussed in Chapter 4. Recall that the DTDM and the retransmission management (Chapter 5) protocols both together make the OBM unit of our ingress switch model.

6.1 Protocol Overview

Recall that in an ingress switch, traffic to each egress switch (i.e. a torrent) is kept in a BMPS unit (see Section 2.4). The OBM unit must provide the network access for each torrent. However, the OBM unit can transmit at most fW slots at each time-slot interval to n_e egress switches. Therefore, the DTDM protocol inside the OBM unit must provide a schedule and then based on this schedule transmit traffic from torrents to the OPS network. In general, DTDM gives each torrent a bandwidth measured by a number of slots within a frame (defined in Section 2.4). Then, the slots are distributed through the frame and among the wavelength channels on fibers. The DTDM protocol and the retransmission management protocol cooperate with each other in order to retransmit the dropped traffic, to be discussed in Section 6.3.1 and Section 6.3.3.

Since the DTDM protocol is based on the even slot distribution concept, we must first discuss this concept before detailing the DTDM protocol. Then, we shall detail DTDM by explaining its bandwidth provisioning method, its slot assignment, and its transmission protocol in addition to its overall algorithm.

6.2 Even Slot-Distribution

In this section, we first present the even slot-distribution concept in a slotted OPS network. We measure slot-distribution performances through a measurement frame of length τ slot-sets, where there are τfW slots in this frame. Let $\Lambda = \{\Lambda_0, \Lambda_1, ..., \Lambda_{n_e-1}\}$ denote the set of assigned slots to n_e torrents in an ingress switch, and Λ_i the number of slots (out

of τfW) that torrent-i is allowed to use within the measurement frame. Clearly, we have $\sum_{i=0}^{n_e-1} \Lambda_i \le \tau fW$.

To alleviate the performance loss due to collision, we rely on an intelligent way to distribute the slots of set Λ through the frame and among the wavelengths and fibers. Out of the many combinations to distribute slots, many suffer from longer delays and delay jitter. To improve the network performance, the slots assigned to each torrent can be evenly distributed through both horizontally (through the frame) and vertically (among the fibers and wavelengths) so that a smooth schedule is obtained. We call this an even slot-distribution in which a schedule for traffic transmission is provided to each torrent. Based on this schedule, each torrent knows in advance when and on which fiber and wavelength it will transmit its traffic.

In an even slot-distribution, slots for a desired egress switch are equally spaced from each other. The idea is to reduce the collision of slots going to a desired egress switch by distributing the slots randomly. Note that increasing the randomness will usually decrease the chance of collision at the core switch. Ideally, an even distribution may provide load balancing on wavelength channels so that no wavelength is overwhelmed. Furthermore, the same opportunity can be provided to different torrents to access to the network. This is related to the fairness of slot-distribution to be discussed.

In the following, we shall use $|X|, \lceil X \rceil, \lfloor X \rfloor$, and $X \bmod Y$ to denote respectively the absolute value of X, the ceiling operation of X, the floor operation of X, and the remainder of X when divided by Y.

	0	1	2	3	4	5	6	7	8	9
w_1		3	0		2	6		1	0	
w_2	0	5	1		3			6	4	6
w_3	2	6	4	6	0	4	0	3	2	3

a) Distribution-1

	0	1	2	3	4	5	6	7	8	9
w_1		0	1	3			4	6	6	6
w_2	0	3	0		2	2		4	6	5
w_3	0	0	4	2	3		3	1	6	4

b) Distribution-2

Fig.6.1: Slot-Distributions through a Measurement Frame of τ=10 Slot-sets.

6.2.1 Even Distribution through the Frame (EDF)

We can distribute the slots evenly by their distance through a frame, by their density in slot-sets within the frame, or by both. For the even distance-distribution EDF_{di}, the slots

assigned to the same torrent are separated in equal time-slot distances from each other throughout the frame so that a deterministic slot service rate can be provided to each torrent. For the even density-distribution EDF_{de}, slots assigned to the same torrent are distributed with the same density among all slot-sets through the frame so that an equal number of packets can be served in the available time-slots. Fig.6.1 compares two different distributions within a frame of 30 slots in total where τ =10 slot-sets, fW=3 (i.e. each slot-set can carry at most 3 slots), and the number inside each time-slot indicates a torrent (or alternatively an egress switch) reference. In Distribution-1, slots are evenly distributed in terms of both distance and density. For example, torrent-0 slots (going to egress switch 0) have a distance of 2 time-slots from each other. In addition, under Distribution-1, the density of torrent-0 slots in different slot-sets is even, i.e., always 1. The Distribution-2 has an uneven distribution. For instance, the distance between two torrent-0 slots is not even, i.e., either 1 or 8 time-slots apart. Moreover, the density of slot(s) distributed for torrent-0 in Distribution-2, is 2, 2 and 1 in the first three slot-sets, i.e., an uneven density. The density is the most uneven for torrent-6 slots with 1,3, and 1 at the last three slot-sets. This is not desirable because of a higher latency variation and a lower bandwidth utilization (the following discussion).

By employing either or both of the even distribution schemes, the even slot-distribution has several advantages when implemented at the edge switches.

1. **Reducing latency variation:** Using the even distance-distribution, the slot transmission is smoothed and therefore jitter and queuing delay are reduced due to the deterministic service rate. In a non-deterministic slot service, some packets may experience much queuing time at some time and less queuing time at another time, which results in a higher jitter. In Distribution-2 of Fig.6.1, torrent-0 slots are transmitted at the first part of the frame and torrent-6 slots are transmitted at the end part of the frame. Thus, torrent-0 packets experience less delay whereas torrent-6 packets must queue up at the first part. At the end part, torrent-0 slots will experience a longer delay.
2. **Fairness:** Different torrents find fair opportunities to have access to the output bandwidth when both EDF_{di} and EDF_{de} are respected.
3. **Higher bandwidth utilization:** Here, slots can be transmitted in appropriate time-

slots that would match the volume of traffic arrival for a torrent. The distribution determines when a torrent can send its traffic within the frame. However, not all traffic may have already arrived and therefore get transmitted when they are allowed to. In Distribution-2 of Fig.6.1, five slots of torrent-0 will be transmitted in the first three time-slots of the frame where not all traffic may have arrived in time for torrent-0. This may lead to the transmission of partially filled or even empty slots (bandwidth waste). In Distribution-1 of Fig.6.1, the transmitted traffic is proportional to the traffic in the buffers and therefore the bandwidth utilization will increase.

4. **Avoiding burst generation:** Distribution-2 in Fig.6.1 generates a burst of five slots to egress switch 6, while Distribution-1 sends the same traffic with no burst and this leads to the arrival of regular traffic in egress switch 6. When $fW>1$, the density of the slots through the frame should also be smoothed to avoid bursts, i.e., EDF_{de} distribution.

In the following sub-sections we shall formulate procedures to measure even distributions in distance (EDF_{di}) and in density (EDF_{de}). First, let set $\Phi_{i,M_i}=\{(x_j, y_j) \mid 0 \le j \le M_i-1, 0 \le x_j \le \tau-1, 0<y_j \le \Lambda_i\}$ be the distribution of Λ_i slots allocated to torrent i in M_i slot-set positions where x_j refers the slot-set number within the frame and y_j is the number of the slots going to egress switch i at slot-set x_j. For the remaining τ-M_i slot-sets, we have $y_j=0$. For example, $\Phi_{2,3}=\{(3,2), (5,1), (6,2)\}$ shows that $\Lambda_2=5$ slots are allocated to torrent-2 within the frame and they are distributed in $M_2=3$ slot-sets: $x_0=3$, $x_1=5$ and $x_2=6$ with the density of $y_0=2$, $y_1=1$ and $y_2=2$ slots respectively.

6.2.1.1 EDF Distance (EDF_{di})

We have formulated the following procedure steps to achieve EDF_{di}:

1) Compute the value of an indicator called distance-distribution index $\Im_{DI,i}$ for torrent i, which is defined as

$$\Im_{DI,i}=\sum_{j=0}^{M_i-1}\left|x_j-x_{j-1}-\frac{\tau}{M_i}\right|-\Im^*_{DI,i}, \qquad (6.1)$$

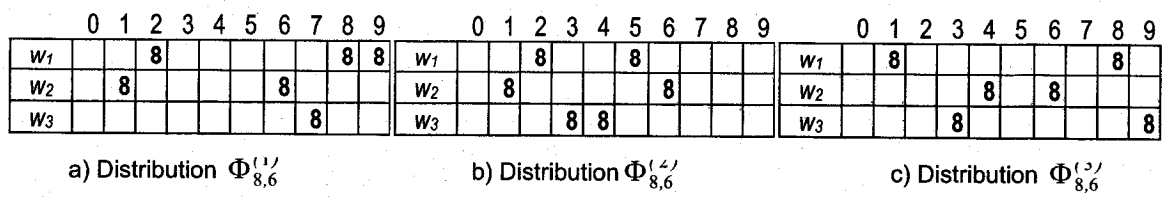
where $x_{-1}=x_{M_i-1}-\tau$ and $\Im^*_{DI,i}$ is the optimum value of the distance-distribution index for torrent i. The summation measures how close are the slots distributed for torrent i to the

ideal distance of $\frac{\tau}{M_i}$. Then, when the EDF distance-distribution is optimum, we have $\Im_{DI,i}=0$. A larger value of $\Im_{DI,i}$ indicates a non-EDF distance-distribution. Note that the index $\Im_{DI,i}$ is sensitive to the difference between the slot positions within the frame not the exact position of the slots.

2) Compute the optimum distance-distribution index $\Im^*_{DI,i}$ used in Eq.(6.1). Let set $\Phi^*_{i,M_i}=\{x^*, y^*\}$ be an optimum EDF distance-distribution for a set of M_i slots. The parameter $\Im^*_{DI,i}$ is calculated from $\Im^*_{DI,i}=\sum_{j=0}^{M_i-1}\left|x^*_j-x^*_{j-1}-\frac{\tau}{M_i}\right|$ where $x^*_{-1}=x^*_{M_i-1}-\tau$. When $\frac{\tau}{M_i}$ is integer, we have $\Im^*_{DI,i}=0$ because the slots can be distributed exactly in equal time-slot distances from one another. In the optimum distribution of M_i slots, n_1 slot-sets must have a distance of $\left\lfloor\frac{\tau}{M_i}\right\rfloor$ time-slots and M_i - n_1 slot-sets should have a distance of $\left\lceil\frac{\tau}{M_i}\right\rceil$ time-slots from each other. In addition, we have $n_1\left\lfloor\frac{\tau}{M_i}\right\rfloor+(M_i-n_1)\left\lceil\frac{\tau}{M_i}\right\rceil=\tau$. Thus, we obtain $n_1=M_i\left(\left\lfloor\frac{\tau}{M_i}\right\rfloor+1\right)-\tau$ and

$$\Im^*_{DI,i}=n_1\left(\frac{\tau}{M_i}-\left\lfloor\frac{\tau}{M_i}\right\rfloor\right)+(M_i-n_1)\left(\left\lceil\frac{\tau}{M_i}\right\rceil-\frac{\tau}{M_i}\right)=2\left(M_i\left(\left\lfloor\frac{\tau}{M_i}\right\rfloor+1\right)-\tau\right)\left(\frac{\tau}{M_i}-\left\lfloor\frac{\tau}{M_i}\right\rfloor\right).$$

3) Compute the distance-distribution index within frame $\Im_{DI}$ defined to be the average of EDF_{di} indices among all torrents, i.e., $\Im_{DI}=\frac{1}{n_e}\sum_i\Im_{DI,i}$. The index $\Im_{DI}$ will be used in our performance evaluation to compare the distance-distribution performance of the CTT and DTDM schemes.



a) Distribution $\Phi^{(1)}_{8,6}$ b) Distribution $\Phi^{(2)}_{8,6}$ c) Distribution $\Phi^{(3)}_{8,6}$

Fig.6.2: Three Slot-Distributions through a Measurement Frame of τ =10 Slot-sets

Example 6.1: Assume Λ_8=6, fW=3 and τ =10. By going through all combinations of (x, y), we obtain one of the best EDF distance-distributions $\Phi^*_{8,6}=\{(1,1),(3,1),(5,1),(7,1),(8,1),(9,1)\}$ with $\Im^*_{DI,8}=\frac{8}{3}$. Now consider three distributions (Fig.6.2): $\Phi^{(1)}_{8,6}=\{(1,1),(2,1),(6,1),(7,1),(8,1),(9,1)\}$, $\Phi^{(2)}_{8,6}=\{(1,1),\ (2,1),\ (3,1),\ (4,1),\ (5,1),\ (6,1)\}$ and $\Phi^{(3)}_{8,6}=\{(1,1),\ (3,1),\ (4,1),\ (6,1),\ (8,1),\ (9,1)\}$. The EDF_{di} parameters for these distributions are 8/3, 4 and 0 respectively. Thus, $\Phi^{(3)}_{8,6}$ is the best and $\Phi^{(2)}_{8,6}$ is the worst under the EDF distance definition.

6.2.1.2 EDF Density (EDF_{de})

We use the same set Φ_{i,M_i} defined before Section 6.2.1.1 to formulate the following steps to achieve EDF_{de}:

1) Compute an indicator called density-distribution index $\Im_{DE,i}$ for torrent i defined as

$$\Im_{DE,i}=\sum_{j=0}^{\tau-1}\left|y_j-\frac{\Lambda_i}{\tau}\right|-\Im^*_{DE,i}\quad , \tag{6.2}$$

where Λ_i is the number of slots assigned to torrent i within the frame and the summation measures how the density of slots distributed at each slot-set is closer to the desired density, and $\Im^*_{DE,i}$ is the optimum density-distribution index. Note that the optimum distribution is obtained when the $\Im_{DE,i}$ index parameter becomes zero.

2) Compute the optimum density-distribution index $\Im^*_{DE,i}$ used in Eq.(6.2) for torrent i. Note that $n_1=\tau\left\lceil\frac{\Lambda_i}{\tau}\right\rceil-\Lambda_i$ slot-sets must carry $\left\lfloor\frac{\Lambda_i}{\tau}\right\rfloor$ slots and τ -n_1 slot-sets have to carry $\left\lceil\frac{\Lambda_i}{\tau}\right\rceil$ slots. When the density-distribution is optimum, we have $\Im_{DE,i}=0$ which results in $\Im^*_{DE,i}=\sum_{j=0}^{\tau-1}\left|y_j-\frac{\Lambda_i}{\tau}\right|$. Therefore, we have

$$\Im^*_{DE,i}=n_1\left(\frac{\Lambda_i}{\tau}-\left\lfloor\frac{\Lambda_i}{\tau}\right\rfloor\right)+(\tau-n_1)\left(\left\lceil\frac{\Lambda_i}{\tau}\right\rceil-\frac{\Lambda_i}{\tau}\right)=2\left(\tau\left\lceil\frac{\Lambda_i}{\tau}\right\rceil-\Lambda_i\right)\left(\frac{\Lambda_i}{\tau}-\left\lfloor\frac{\Lambda_i}{\tau}\right\rfloor\right).$$

3) Compute the density-distribution index within frame $\Im_{DE}$ defined to be the average of EDF_{de} indices among all torrents, i.e., $\Im_{DE}=\frac{1}{n_e}\sum_i\Im_{DE,i}$. The index $\Im_{DE}$ will be used later

in our performance evaluation to compare the density-distribution performance of CTT and DTDM.

Table 6.1: EDF Distance and Density Parameters for Example 6.2

Distribution	$\Phi_{4,6}^{(1)}$	$\Phi_{4,6}^{(2)}$	$\Phi_{4,8}^{(3)}$
$\Im_{DI,4}$	2.67	4	0
$\Im_{DE,4}$	19.2	20.4	9.2

Example 6.2: For Λ_4=26, fW=12, and τ =10, we have three sets of distributions: $\Phi_{4,6}^{(1)}$ ={*(1,3), (2,5), (6,1), (7,4), (8,7), (9,6)*}, $\Phi_{4,6}^{(2)}$ ={*(1,4), (2,8), (3,6), (4,1), (5,2), (6,5)*} *and* $\Phi_{4,8}^{(3)}$ ={*(1,2), (2,3), (3,4), (4,3), (5,2), (6,5), (8,2), (9,5)*}. For all configurations, we have $\Im^*_{DE,4} = 4.8$. For the first two sets, we have $\Im^*_{DI,4} = \frac{8}{3}$ and for the last set, we have $\Im^*_{DI,4} = 3$. Table 6.1 shows the EDF parameters for the three cases. The distribution $\Phi_{4,8}^{(3)}$ has the lowest EDF in terms of both distance and density. On the other hand, $\Phi_{4,6}^{(2)}$ is the worst distribution.

6.2.2 Even Distribution among Wavelengths (EDW)

Unlike EDF_{de}, the Even Distribution among Wavelengths (EDW) tries to distribute evenly the slots among the wavelengths/fibers of a slot-set only so that no wavelength is overwhelmed. In other words, the load becomes balanced among the wavelengths at each slot-set. For example, if traffic load on the wavelengths is 0.6 and fW=20, then 12 wavelengths are occupied at each slot-set on average. Note that EDW may in turn reduce collision at the core switch (Section 4.2.1.1).

To formulate the steps to achieve EDW, let $\Psi_i = \{\omega_j | j \in [0, \tau - 1]\}$ be the distribution of slots on the slot-sets within the frame where ω_k denotes the number of distributed slots on slot-set k. We have $\Lambda_t = \sum_j \Lambda_j$ as the total slots allocated to all egress switches within the frame of τ slot-sets. The EDW parameter, $\Im_{LB}$, can then be calculated in a similar way as EDF_{de} in Section 6.2.1.2, except we use Λ_t and ω_j instead of Λ_i and y_j respectively. We shall use $\Im_{LB}$ to compare the EDW-distribution performance of CTT and DTDM later on. Note that the optimum distribution is obtained when $\Im_{LB}$ becomes zero.

6.2.3 Fair Distribution (FD)

The Fair Distribution deals with fairness issue and measures how uniform the destination of the slots to n_e egress switches are within the frame of τ slots. In a fair distribution, the ingress switches send almost equal number of slots to each egress switch in the τ slot-sets interval. This in turn may lead to an almost symmetric traffic transmission to the network. As discussed in Section 4.2.1.1 the symmetric traffic transmission results in a less contention at the optical switches than an asymmetric traffic transmission. Let $\Omega=\{(j, d_j) \mid j=0,2,..,n_e\text{-}1]\}$ be the distribution of the slots to at most n_e egress switches where d_j denotes the number of the slots going to egress switches j. For example, let n_e=8 and $\Omega=\{(2, 17), (4,19), (5,28)\}$ in an ingress switch. This set shows that 17, 19 and 28 slots are going to egress switches #2, #4, and #5 respectively within the frame and no traffic will be transmitted for the remaining egress switches. The total number of non-empty slots leaving the edge switch within the frame is $n_d = \sum_{j=0}^{n_e-1} d_j$. We shall use the following steps to achieve FD:

1) Compute the value of an indicator called fairness-distribution index $\Im_{FD}$ for all torrents within the frame, which is defined as

$$\Im_{FD} = \sum_{j=0}^{n_e-1} \left| d_j - \frac{n_d}{n_e} \right| - \Im^*_{FD} \quad , \tag{6.3}$$

where the first term measures how the distribution of the slots to each egress switch is closer to the fair distribution $\frac{n_d}{n_e}$, and $\Im^*_{FD}$ is the fair distribution index. Note that the optimum distribution is obtained when the $\Im_{FD}$ index parameter becomes zero.

2) Compute the optimum fairness-distribution index ($\Im^*_{FD}$). In an optimum fairness-distribution, we have $\Im_{FD}$=0 which results in $\Im^*_{FD} = \sum_{j=0}^{n_e-1} \left| d_j - \frac{n_d}{n_e} \right|$. In a fair distribution, $\left\lfloor \frac{n_d}{n_e} \right\rfloor$ slots must be transmitted to $n_1 = n_e \left\lfloor \frac{n_d}{n_e} \right\rfloor - n_d$ egress switches. On the other hand, $\left\lceil \frac{n_d}{n_e} \right\rceil$ slots must be sent to the remaining egress switches. Considering the summation in Eq.(6.3), we have

$$\Im^*_{FD} = n_1\left(\frac{n_d}{n_e} - \left\lfloor\frac{n_d}{n_e}\right\rfloor\right) + (n_e - n_1)\left(\left\lceil\frac{n_d}{n_e}\right\rceil - \frac{n_d}{n_e}\right) = 2\left(n_e\left\lceil\frac{n_d}{n_e}\right\rceil - n_d\right)\left(\frac{n_d}{n_e} - \left\lfloor\frac{n_d}{n_e}\right\rfloor\right) .$$

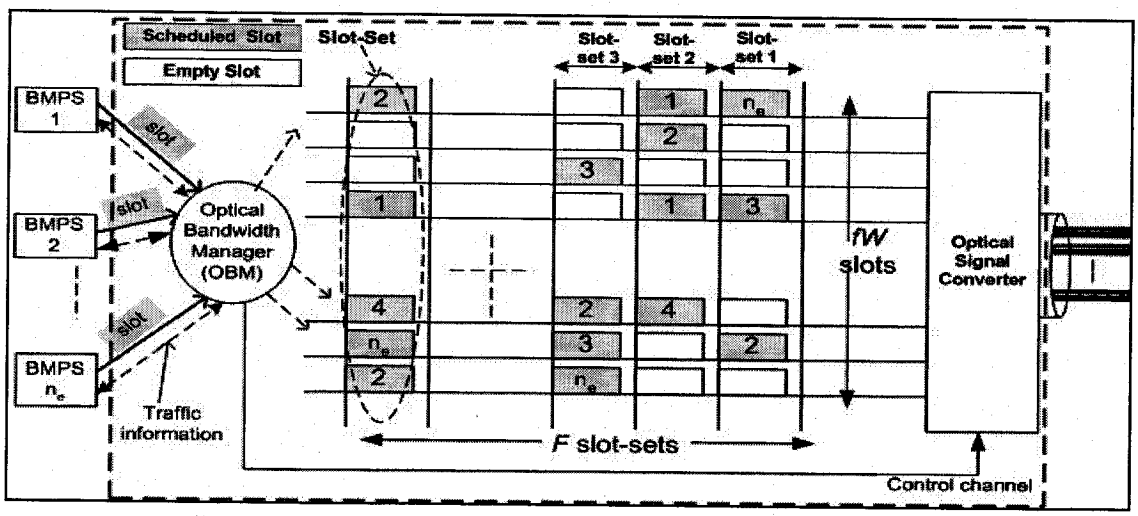


Fig.6.3: The OBM Unit in an Ingress Switch

6.3 The Distributed TDM (DTDM) Protocol

We shall now detail the frame-based DTDM protocol by explaining its bandwidth provisioning method, slot assignment and transmission in addition to its overall algorithm.

Fig.6.3 shows that there are *fW* slots available from the fibers and wavelengths in each slot-set and *FfW* slots in a frame when the frame width is *F* slot-sets. The OBM unit first provides the required bandwidth in slots (in the frame of *FfW* slots) to each torrent. Then it schedules the allocated slots to each torrent among the slot-sets and among the fibers and wavelengths within a slot-set. The gray slots in Fig.6.3 are the scheduled slots and a number inside a slot represents an egress switch (alternatively a torrent) reference. Each torrent is then informed of the time-slot and on which fiber/wavelength will transmit its traffic, i.e., each torrent has a predetermined and a fair guaranteed bandwidth. The following discussion will provide details on how they are distributed.

6.3.1 Bandwidth Provision

In DTDM, the bandwidth is provided at each ingress switch for n_e torrents at frame boundaries or even a little earlier (depending on the processing speed). To provide the bandwidth for a torrent, a traffic estimator algorithm can be used to determine the number

of slots required for each torrent. We use the *Prev+Queue* bandwidth provision scheme [AfCo96] to provide bandwidth for torrent i in which the average number of packet arrivals, $N_{e,i}$, plus the current queue size, Q_i, are the criteria for traffic estimation for the new frame. To calculate the average number of packet arrivals, we use the running average method by $N_{e,i} = \sigma N_{e,i} + (1-\sigma)N_{c,i}$ where $0 \le \sigma \le 1$ is a smoothing factor and $N_{c,i}$ is the number of the packet arrivals during the last frame period. Since the average number of packets that can be carried in a slot is $N_S = \left\lfloor \frac{B_C S_T}{\ell} \right\rfloor$, the average number of slots required to carry torrent i traffic is given by

$$\theta_i = \min\left(\left\lceil \frac{N_{e,i} + Q_i}{N_S} \right\rceil + N_{R,i}, \theta_{\max} \right) , \tag{6.4}$$

where $N_{R,i}$ is the number of slots that must be retransmitted to egress switch i (if the retransmission in optical domain is supported) and θ_{max} is the limit on the number of slots that can be allocated to torrent i in order to avoid over-provisioning of bandwidth to malicious sources. Note that this requires the cooperation between the DTDM protocol and the retransmission management unit inside the OBM unit. In Eq.(6.4), enough number of slots are requested in order to transmit the traffic in torrent i buffers as well as the amount of the traffic that will arrive during the next frame period for torrent i. The number of slots allocated to torrent i is limited by θ_{max}, set by the network manager. For example, we limit θ_i by $\theta_{max}=FW$ in this work. Since the total number of the required slots for all torrents in an ingress switch may exceed the total number of available slots, fFW, the requested slots must be adjusted accordingly. This is done by calculating the slot-scaling coefficient, $\gamma = min(1, \frac{fFW}{\theta})$ where $\theta = \sum_{i=0}^{n_e-1} \theta_i$, and then adjusting the number of slots assigned to torrent i by $\Lambda_i = \lfloor \gamma \theta_i \rfloor$, i.e., the allocated slots are proportionally reduced by the factor γ when $\gamma<1$. Obviously, if the total number of the requested slots do not exceed the total number of the available slots within a frame, $\gamma=1$, then the requested slots to each torrent can be completely allocated to the torrent.

6.3.2 Slot Assignment Algorithm

Recall in Section 6.3.1 we are assigning A_i slots, where $1 \le i \le n_e$, to torrent i (recall that

the torrent i traffic is sent to egress switch i) to use within the frame period. Now the OBM unit in Fig.6.3 must evenly distribute Λ_i slots within the frame and among the fibers and wavelengths for torrent i according to the principles set out in Section 6.2.

```
1. EmptySlotSchedule() //empty SlotSchedules matrix
2. for (S=1; S <=F; S ++) SlotSetCapacity[S]= f*W; // initialize capacity of each slot-set to f*W
3. while (True)
4.   {
5.     i =GetNextTorrent()
6.     if i=Null Then Stop  ///if no more torrent reference is left in the linked list, the process is finished
7.     S =SelectRandomSlotSet
8.     k=0
9.     OptimumDistance= max(1.0, F/ Λi)
10.    Delta=0
11.    while (Λi >0)
12.      {
13.        Delta = Delta + OptimumDistance
14.        step=floor(Delta)
15.        Delta = Delta-step
16.        S =SelectNextEmptySlotSet (S, step) // considering "step" distances, find the next empty,
                                                // slot-set, may circulate to the beginning of the frame
17.        w=FindRandomEmptyChannelOnSlotSet(S)  // choose a random empty channel on slot-set S
18.        SlotSchedules[w][ S]=i  // save torrent reference on wavelength w and slot-set S
19.        SlotSetCapacity[S] = SlotSetCapacity[S]-1 // update the capacity of slot-set S
20.        Λi= Λi – 1  //residual slots for distribution
21.        k=k+1
22.        if k=F Then // if F slots are distributed for torrent i
23.          {
24.            if (Λi >0) AppendToTorrentList(i)  //append the torrent reference to the linked list
25.            break;    // after F distributions for an egress switch, process the next torrent
26.          }
27.    }
28. }
```

Fig.6.4: Pseudo Code of the Slot Assignment Algorithm in DTDM

Fig.6.4 depicts the pseudo code of the DTDM slot assignment protocol. All of the wavelengths are serially numbered in the code (1 to fW). Then, we assign fW slots to each slot-set within the frame (Line 2). The slots assigned to each torrent are distributed with respect to the EDF_{di}, EDF_{de} and EDW definitions. A linked list is used to keep the reference of the torrents with $\Lambda_i > 0$ where $1 \leq i \leq n_e$. Define a distribution round to include all torrents with $\Lambda_i > 0$. In each distribution round, $min(F, \Lambda_i)$ slots are distributed for torrent i through the frame. Then, the reference of torrent i is appended to the next distribution round if $\Lambda_i > F$. This process continues until the linked list becomes empty.

The algorithm obtains the reference of torrent i from the linked list (Line 5) and then removes the reference from the linked list. To distribute the slots assigned to torrent i, a starting slot-set S is first randomly chosen (Line 7). The DTDM algorithm first tries to

make an equal space between any two consecutive slots assigned to torrent *i*. The optimum distance between two slots of torrent *i* with Λ_i allocated slots is $\frac{F}{\Lambda_i}$ slots for the frame of F slot-sets (Line 9). The position of Λ_i slots is found in such a way that the criteria of the equal distance-distributions can be fitted in the most satisfactory way (Lines 11-27). This is because the optimum distance of $\frac{F}{\Lambda_i}$ may not be an integer number and therefore the assigned slots to torrent *i* cannot always be equally distributed. To do this, we use *Delta* ($0 \leq Delta < 1$) as a parameter to provide an optimum distribution. When F is divisible by Λ_i, we always have $Delta=0$. Let the current occupied slot-set be S. Then, the next slot-set for occupation should be $S+step$ where the parameter *step* is obtained by $step = \left\lfloor Delta + \frac{F}{\Lambda_i} \right\rfloor$ (Line 14). If the desired slot-set is full, the next empty slot-set is linearly searched and allocated (Line 16). This in turn may cause uneven slot-distribution within the frame and may degrade the Fairness parameter. The parameter *Delta* is updated by $Delta = Delta + \frac{F}{\Lambda_i} - step$ (Line 13-15).

When an appropriate slot-set is found for a slot of torrent *i*, a wavelength at that slot-set is randomly chosen (Line 17) to make a balance in using wavelengths. In a multi-fiber network, the DTDM avoids collision at the core switch by avoiding the distribution of the slots of the same torrent on the same-wavelength of different fibers. Recall from Section 4.2.1.1 that load balancing can reduce collision at the core switch. Then the parameters involved in the algorithm are updated (Line 18-21). If the residual assigned slots to the torrent is still larger than F, the torrent reference is appended to the linked list to be processed in the next distribution round (Lines 22-26). When there is no torrent left in the linked list, the distribution process is finished.

6.3.3 Traffic Transmission

To effect traffic transmission, SSH must be sent within the slot-offset interval (time-gap between two time-slots) before transmitting the slots in a slot-set. Consider the schedule of a given slot-set. In this slot-set, suppose torrent *j* to have a schedule to send m_j slots to egress switch *j*. On the other hand, suppose there are $N_{R,j}$ outstanding slots in the

retransmission buffer (see the retransmission model in Section 2.6) ready to be retransmitted to egress switch j. In the desired slot-set, egress switch j should first send $min(m_j, N_{R,j})$ slots from the retransmission buffer and then up to m_j - $min(m_j, N_{R,j})$ slots from the new traffic that can be provided by BMPS unit j. In other words, the dropped traffic has a higher priority over the new traffic to be transmitted first. For example, consider Slot-set 2 in Fig.6.3 in which four slots are scheduled: two slots to Egress Switch-1 (i.e. from Torrent-1), one slot to Egress Switch-2, and one slot to Egress Switch-4. Also, suppose we have $N_{R,1}$=0, $N_{R,2}$=1, and $N_{R,4}$=0. Since there is a slot in the retransmission buffer to be transmitted to Egress Switch-2, the OBM should only request BMPS-1 and BMPS-4 to provide two and one slot(s) respectively.

At the beginning of the slot-offset interval or even earlier (depending on the processing speed), the OBM requests the relevant BMPS units to provide their new slots for transmission. Upon request by the OBM, the packet scheduler in BMPS i schedules packets from different streams and from different classes up to $B_C S_T$ bits and makes a slot, and then sends the slot to the OBM. Note that the traffic scheduled in a slot must be an integer number of packets with the total size of less than or equal to $B_C S_T$ bits. As discussed in Section 4.2.1.2, the CSA scheme can provide a better aggregation than non-CSA scheme. This is why we let our slots carry packets from different classes.

During the slot-offset interval or even earlier, the OBM prepares a SSH header based on the slots provided for transmission in the slot-set. Then, the OBM sends SSH over the control channel. At the slot boundary, the OBM sends the provided slots to the relevant egress switches on the predetermined fibers/wavelengths.

6.3.4 Scheduling at a Core Switch

Each core switch in the network first receives the information of slots at the same time in its input ports. After receiving a SSH and during the slot-offset interval, it evaluates the potential contention, then resolves the contention, and finally makes the core switch ready to switch the arriving slots to their desired egress switches. Note that the slot contention happens whenever more than f slots are attempting to reach to the same output link on the same-wavelength and at the same time-slot. The core switch needs to resolve the contention at any output link. We detail the following algorithm that can be used by both DTDM and CTT at a given output link.

Let set $S_{i,l}=\{s_0, s_1, ..., s_{l-1} \mid 0 < l \leq nf\}$ denote the subset of l contending slots on wavelength w_i $(i=0,..., W-1)$ at the output link. Sine each link has f fibers, a total of f slots can be switched on wavelength w_i from the output link. As mentioned in Section 2.5, the traffic parameters for each slot of the subset is recorded in the SSH. Let the vector $N(n_{EF,j}, n_{AF,j}, n_{BE,j})$ denote these parameters for slot s_j, where $n_{EF,j}$, $n_{AF,j}$, and $n_{BE,j}$ denote respectively the number of packets in class EF, class AF and class BE carried in slot j. Since DTDM collects packets from different classes in a slot, we could have $n_{EF,j} \geq 0$, $n_{AF,j} \geq 0$, and $n_{BE,j} \geq 0$, but $n_{EF,j} + n_{AF,j} + n_{BE,j} > 0$. However, only one of the entries in vector N for a slot under CTT is positive. We define the traffic value T_v to be an index for each slot in $S_{i,l}$ that indicates how valuable a slot is to be transmitted through the output link. The core switch resolves the contention by the following procedure:

1. Compute the traffic value parameter $T_{v,j}$ for slot s_j via $T_{v,j} = V_{EF} n_{EF,j} + V_{AF} n_{AF,j} + V_{BE} n_{BE,j}$ where V_{EF}, V_{AF}, and V_{BE} are "importance" parameters for classes EF, AF and BE respectively.
2. Sort the slots in set $S_{i,l}$ in a descending order based on their T_v values.
3. Select the top $min(f,l)$ slots from the sorted list for transmission on wavelength w_i of f fibers. Note that output fibers are randomly chosen to provide load balancing on connection links.
4. Resolve the conflicts for the remaining $l-min(f,l)$ slots in $S_{i,l}$ using a contention resolution scheme. Any scheme can be used provided that the slots with higher T_v values are first resolved. Then the slots that cannot be resolved are dropped.
5. Make a new SSH for the slots departing from the output link, and then transmit the SSH on the control channel of the output link to the next hop switch.

As discussed in Section 4.2.1.1, the load balancing can provide a lower slot loss rate at the network. This is why in Step-3 output fibers are randomly chosen to provide load balancing on connection links. In terms of complexity, note that the only difference between our contention resolution algorithm and the common one that randomly chooses slots for resolution is in sorting. The sorting can be done by the Quick Sort method in $O(mlog(m))$ operations where m is the core switch size. However, m is usually a small number in optical switches. On the other hand, the slot-offset S_O is big enough (in the micro-second range) to perform the sorting in our proposed network framework discussed in Chapter 2. Therefore, we have a scalable system.

Table 6.2: Traffic Value Example

a) Traffic Value under DTDM

Slot	n_{EF}	n_{AF}	n_{BE}	V_{EF}	V_{AF}	V_{BE}	T_v
#1	10	5	11	0.5	0.4	0.1	**8.1**
#2	7	19	11	0.5	0.4	0.1	**12.2**
#1	10	5	11	0.8	0.15	0.05	**9.3**
#2	7	19	11	0.8	0.15	0.05	**9**

b) Traffic Value under CTT

Slot	n_{EF}	n_{AF}	n_{BE}	V_{EF}	V_{AF}	V_{BE}	T_v
#1	10	0	0	0.5	0.4	0.1	**5**
#2	0	19	0	0.5	0.4	0.1	**7.6**
#1	10	0	0	0.8	0.15	0.05	**8**
#2	0	19	0	0.8	0.15	0.05	**2.85**

By selecting a higher "importance" parameter for a class, the parameter T_v can give a higher priority to the traffic of that class to pass through the core switch (Step-3 and Step-4), thus resulting in a higher throughput for that class. For example, Table 6.2a shows the traffic value for two slots, Slot-1 and Slot-2, under the DTDM protocol and two different "importance" parameter sets. Slot-1 has a higher (lower) number of EF (AF) packets than Slot-2. Under the first "importance" set, Slot-2 with $T_v=12.2$ is given a higher priority to pass through the core switch than Slot-1 with $T_v=8.1$. Under the second "importance" set, Slot-1 with $T_v=9.3$ has a higher traffic value to be switched. A similar issue can be observed for CTT in Table 6.2b. In summary, by choosing a higher "importance" parameter, the traffic of a class is given a higher chance to pass through any core switch in the network, and therefore we expect a higher throughput for that class.

6.3.5 Complexity

We now compute the complexity of the DTDM protocol discussed in Fig. 6.4. The complexity of Line-1 and Line 2 is $O(FfW)$ and $O(F)$ respectively. Then, there are two nested loops in the slot-assignment algorithm. The algorithm executes the block of codes inside the second loop for at most $\sum_{i=0}^{n_e-1} A_i \le FfW$ times (i.e. the total number of slots assigned to all torrents). This is FfW runs in the worst case. All codes within both loops can be executed with the complexity of $O(1)$, except Line-16 to search linearly for an empty slot. At the first sight, it takes $O(1)$ in the best case and $O(F)$ in the worst case to

find an empty slot-set. However, since F is a small number in practice (e.g., 100), we can achieve a complexity of almost *O(1)* with the algorithm mentioned in Appendix F. Therefore, the complexity of DTDM will be almost *O(FfW)* in the worst case.

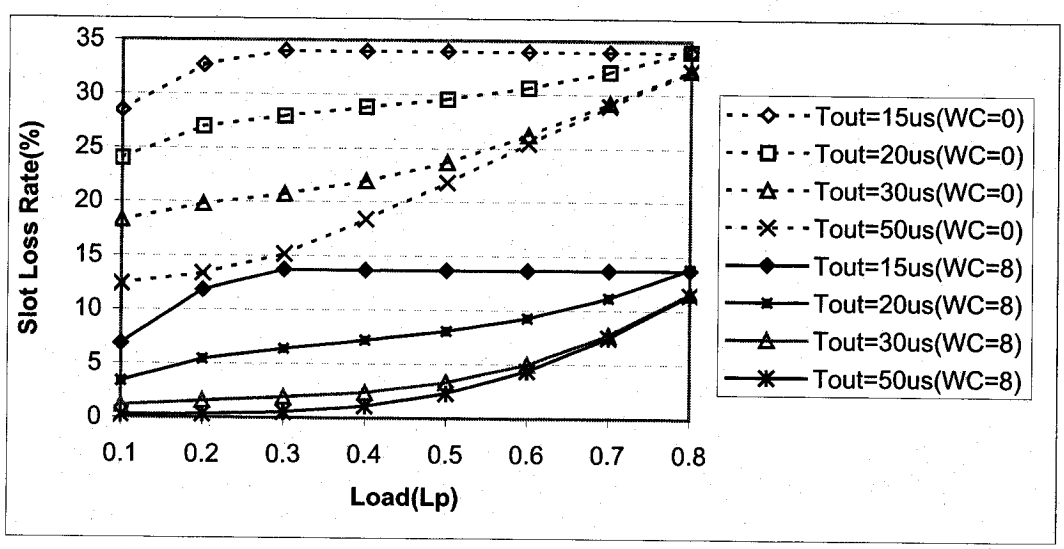
We now compare the complexity of DTDM and CTT. In terms of memory complexity, CTT keeps track of a timer and threshold for each class as well as a slot transmission buffer, while DTDM keeps track of the schedule of all slots within a frame period. In terms of operation complexity, on each packet arrival CTT checks the packet size to see whether a slot can be made or not. This per packet operation may become a problem at higher packet arrival rates in an all-optical network. On the other hand, DTDM predicts future traffic and then distributes packets appropriately within a frame period, but once at a frame interval that is a very big time granularity comparing to a packet inter-arrival. The packet scheduling may also be performed at worst in a time-slot interval. In short, DTDM requires more memory than CTT, however, its operations can be done at higher time granularity and therefore scalability can be assured.

6.4 Performance Evaluation

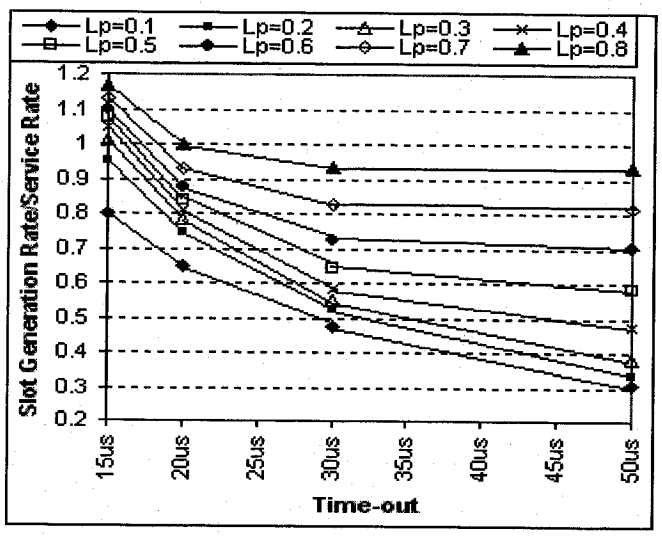
We would like to evaluate and compare the performance of the CTT and DTDM schemes in *Arch1* (see Appendix A) and *Arch2* (see Section 2.4) respectively in order to bring out the virtues of the DTDM scheme. Since our work focuses on the access issue in the edge switches, the underlying network topology has no importance. In all of the following simulation scenarios, each edge switch is connected to R=9 immediate source routers. Other network parameters are as follows: B_C = 10Gb/s, W=4, S_T =9μs, S_O =1μs, F=100 slot-sets, σ = 0.5, V_{EF}=0.6, V_{AF}=0.3, and V_{BE}=0.1 and τ =19 slot-sets. With this parameter setting, the bandwidth is allocated in slots and each slot is transmitted with the rate of 100Kbits/sec. Also, consider the slot transmission buffer size in *Arch1* to be infinite. It should be mentioned that in the following scenarios, L_p denotes the normalized traffic load (from the source routers) arriving at each ingress switch. Since traffic arrivals generate slots, we also define ρ to be the slot generation rate over the slot service rate in an ingress switch as the normalized slot arrival rate to the core switch. This is a more relevant parameter to gauge the operation of the switch and scheduling.

Two traffic arrival characteristics are investigated and compared: Poisson and Pareto traffic. To model a realistic traffic [PaFl95], the Pareto distribution with a p.d.f.

(probability density function) of $p(t) = ab^a t^{-a-1}$ is used to model the inter-arrival time of packets with a=1.2. Moreover, the traffic from each source and in each class is an aggregation of variable-length packets with the distribution mentioned in [CAID06] with a mean 3928 bits. The traffic in each class is uniformly distributed to each one of the egress switches. The traffic load in the following simulations is normalized based on the total bandwidth of each edge switch, i.e., fWB_c. We have used OPNET simulator [OPNE06] to develop our simulation models; and 95% confidence intervals are found to be within 2% of the mean values shown.



a)Slot Loss Rate



b) Slot Generation Rate over Slot Service Rate (ρ)

Fig.6.5: Evaluation of the CTT Scheme under Different Time-Outs

6.4.1 Time-out Experiment in *Arch1*

In this experiment, we would like to show the effect of time-out *Tout* on the slot loss rate and ρ in a single-class single-hop network with f=1, W=4. We study two scenarios: without using any wavelength convert in the core switch (WC=0), and with using eight shared-per-node wavelength converters (WC=8) in the core switch to reduce loss rate. Recall that ρ is interpreted as the normalized slot traffic load on the wavelength channels to the core switch (as opposed to the normalized traffic load L_p to each ingress switch). The core switch is connected to n=8 edge switches (i.e. we consider a core switch of 8x8). The time-outs vary from 15μs to 50μs under Poisson traffic.

Fig.6.5a depicts that for both WC=0 and WC=8 scenarios, the slot loss rate at the core switch is the highest at *Tout*=15 μs because in this case we have $\rho>1$ (see Fig.6.5b) for $L_p \geq 0.3$ and all of the channels are fully utilized. However, the WC=8 scenario has always the lower loss rate than the WC=0 scenario for any time-out and traffic load. By reducing traffic load, the parameter ρ is also reduced and therefore slot loss rate is decreased as a result. A similar behavior can be observed for other time-out values. However, one can see that the slot lost rate is reduced by increasing the time-out value so that the lowest lost rate is obtained at *Tout*=50 μs. Based on Fig.6.5b, it would appear that ρ levels out beyond *Tout*=50 μs indicating no further improvement (e.g., loss rate) can be achieved. Note that the system is still unstable ($\rho>1$) at *Tout*=20 μs and L_p=0.8 according to Fig.6.5b. Obviously when $\rho>1$, waiting delay in the slot transmission queue goes up to infinity. Although by using smaller time-out values a lower delay is expected, this may lead to a higher loss rate at the core switch and an unstable edge switch operation.

6.4.2 Comparison of DTDM and CTT: Single-Hop Network

In the second experiment, we would like to compare the performance of CTT and DTDM under *Arch1* (described in Appendix A) and *Arch2* (described in Section 2.4) respectively in a single-hop network with n=8 edge switches. Each one of the R=9 immediate sources has ψ=3 classes of traffic to each egress switch. The traffic from each source to any egress switch comprises 25%, 35% and 40% of EF, AF and BE traffic respectively under any traffic load. In this experiment, each edge switch is connected with f=2 fibers to a 16x16 core switch. Recall from Section 4.2.2.2 that using more fibers can lead to a less contention at the core switches. We also use eight shared-per-node wavelength converters

at the core switch. In *Arch1*, there are three buffers (each buffer for one class) of size 72Mbits for each egress switch. Time-out (*Tout*) values for the CTT scheme are set to 20μs, 40 μs and 100μs for the EF, AF and BE traffic respectively. For *Arch2*, we use n_e=7 BMPS units in each ingress switch. In each BMPS unit, 27 buffers of size 8Mbits are used to isolate class-based traffic from nine sources. Effectively the total amount of buffers in each scheme is equal. For the OCGRR scheduler used in each BMPS unit, a uniform bandwidth of $C=B_a / n_e$ is assigned to each torrent so that the maximum usable bandwidth for each torrent traffic is C. Note that B_a has already been computed in Section 2.4. We also set class indexes as C_1=1.0, C_2=0.6 and C_3=0.5, and logical frame length (as discussed in Section 3.2.3) to $\Gamma = B_C S_T$ in the OCGRR packet scheduler.

A) Slot Generation Rate:

Fig.6.6 illustrates the analytical results (see Appendix A) for the slot generation rate of three classes (EF, AF and BE), their total generation rates and the slot service rate in *Arch1* using CTT. The analysis results show that the total slot generation rate equals to the slot service rate (ρ=1) at L_p=0.69. Moreover, the time-out event that indiscriminately generates slots becomes ineffective at $L_p \geq 0.18$ and $L_p \geq 0.52$ for the BE and AF traffic respectively. This is because at these ranges there are enough traffic in the class BE and AF to make a full slot before happening the time-out related to the class BE and AF respectively. Therefore, the slot generation becomes smoother later on.

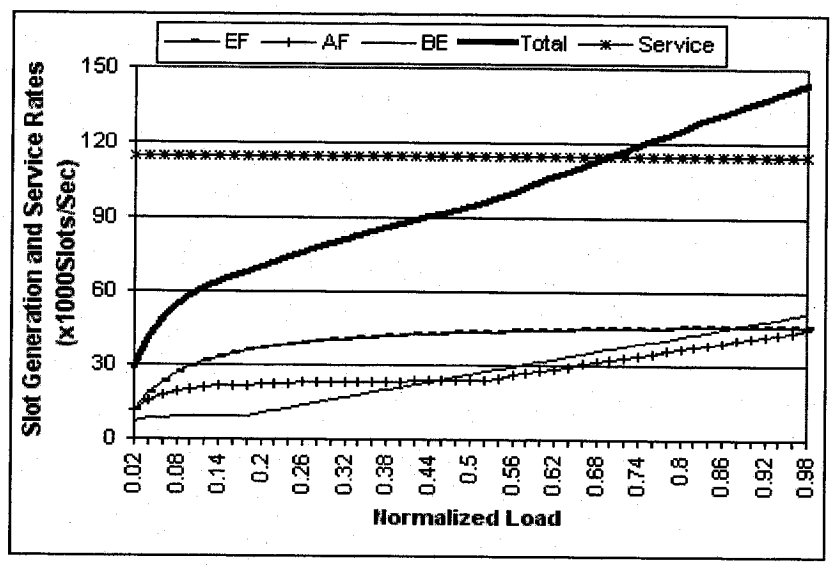
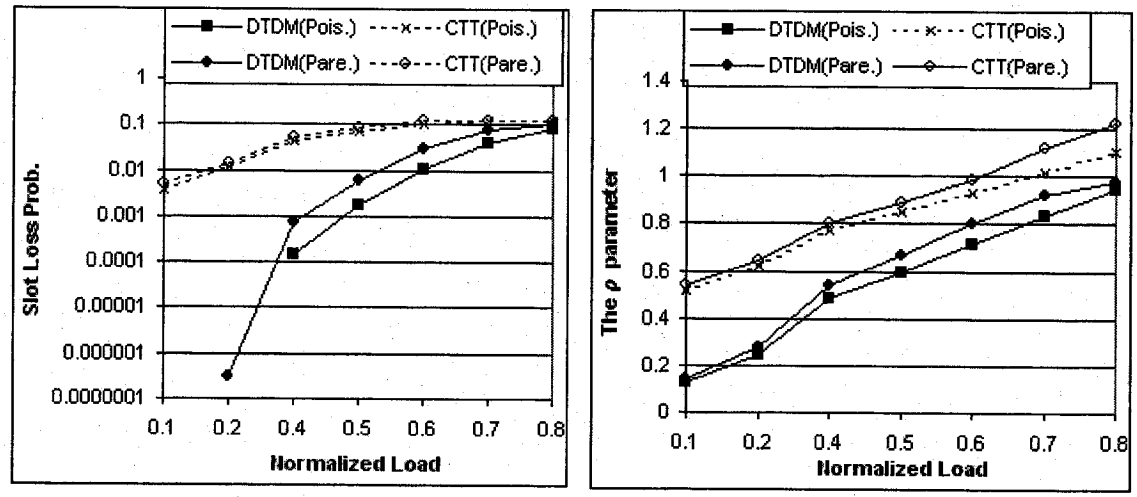


Fig.6.6: Slot Generation Rate under Poisson Arrivals in *Arch1*

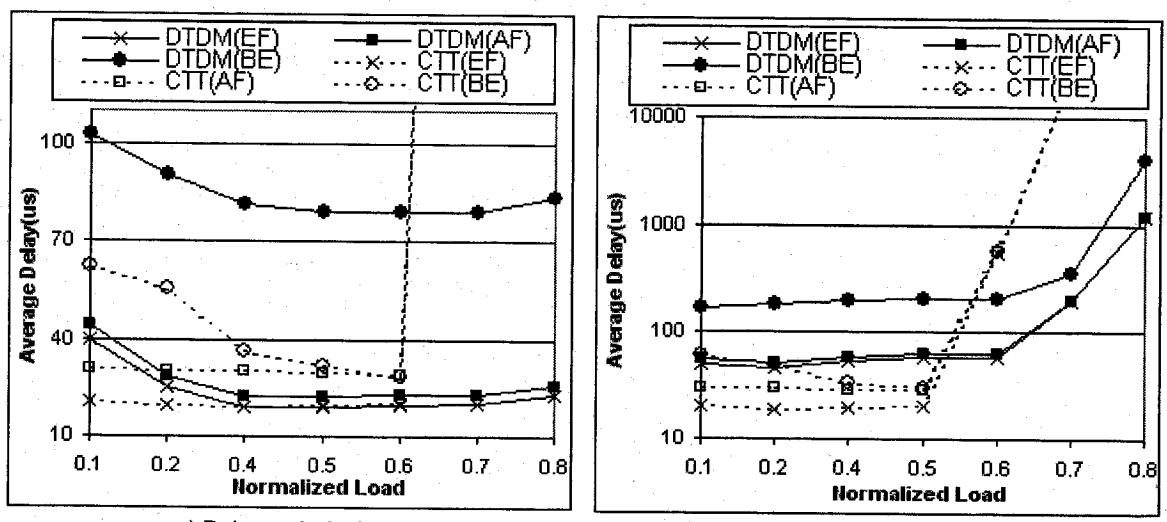


a)Slot Loss Rate at the Core Switch b)Slot Generation Rate over Slot Service Rate(ρ)

Fig.6.7: Slot Loss Rate and Slot Generation Rate over Slot Service Rate (ρ)

Fig.6.7a displays the slot loss rate (in log scale) of both CTT and DTDM under Poisson and Pareto arrivals. Unlike the CTT scheme, there is a very little (not measured) loss for L_p=0.1 and L_p=0.2 in DTDM and under Poisson traffic. The difference between the loss rates under DTDM and CTT is reduced at higher loads. This is because slot assembling at higher loads is performed faster than the lower loads and the frequency of the time-out event is decreased, which in turn reduces the traffic load on the wavelength channels and leads to a lower contention. However, the loss for DTDM is always less than CTT. A similar situation is observed for Pareto arrivals. One may increase time-outs to reduce the slot loss rate in CTT. However, this in turn will degrade the delay of the higher priority traffic. For example, the arrival rate of the EF traffic is less than the BE traffic in practice, and therefore, the BE traffic can be assembled faster than the EF traffic when using higher time-outs, thus leading to a higher delay for EF than BE!

Likewise, the parameter ρ for DTDM is always less than CTT (Fig.6.7b). Observe that $\rho>1$ at L_p=0.7, L_p=0.8 under CTT, and the edge switch becomes unstable in servicing the slots waiting in the slot transmission queue. One can see that the parameter ρ never exceeds 1 in DTDM because even when all fW slots in a slot-set are occupied we have $\rho=1$. A similar situation is observed for Pareto arrivals.



a) Poisson Arrivals

b) Pareto Arrivals

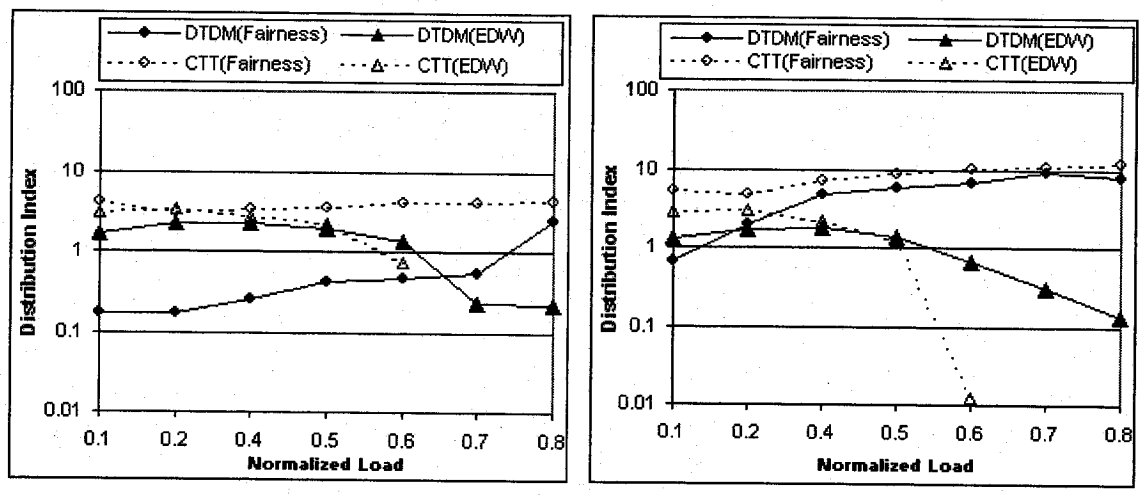
Fig.6.8: System Delay in an Edge Switch under Poisson and Pareto Arrivals

B) Average Delay:

We define system delay to be the duration from the time a packet arrives in an ingress switch until it is transmitted to the optical network. Fig.6.8 shows the average system delay for EF, AF and BE packets in an edge switch. Under the Poisson traffic, the CTT scheme provides a lower delay than DTDM at lower loads, whereas, DTDM has a better (and desirable) EF and AF delay for the mid-range and higher loads. Since the bandwidth provision technique in the DTDM protocol may not be accurate under bursty traffic at the lower and mid-range loads due to traffic fluctuations at shorter intervals, CTT achieves a lower delay than DTDM under the Pareto traffic. However, the delay for all types of the packets under the CTT scheme goes to infinity at higher loads due to its instability problem, but none for DTDM.

For both traffic arrivals, the CTT scheme cannot provide much differentiation among the delay of the packets in different classes at the mid-range and higher loads so that the delay of EF, AF and the BE traffic become almost the same. The reason is that the volume of traffic is high enough to reduce the frequency of the time-out events. Therefore, the traffic of the classes with a higher arrival rate is assembled faster than other classes with lower rates (see Fig.6.6 for L_p>0.9). By choosing higher time-out values, the delay of the BE traffic will be even lower than the EF and AF packets at higher loads. To prevent this problem, one can reduce the time-out value for EF and AF

in CTT. However, this solution increases the slot generation rate, thus leading to a higher loss rate for CTT at the core switch, and an unstable edge switch operation even at the mid-range loads.



a) Poisson Arrivals

b) Pareto Arrivals

Fig.6.9: Average Fairness and EDW Indexes under Poisson and Pareto Arrivals

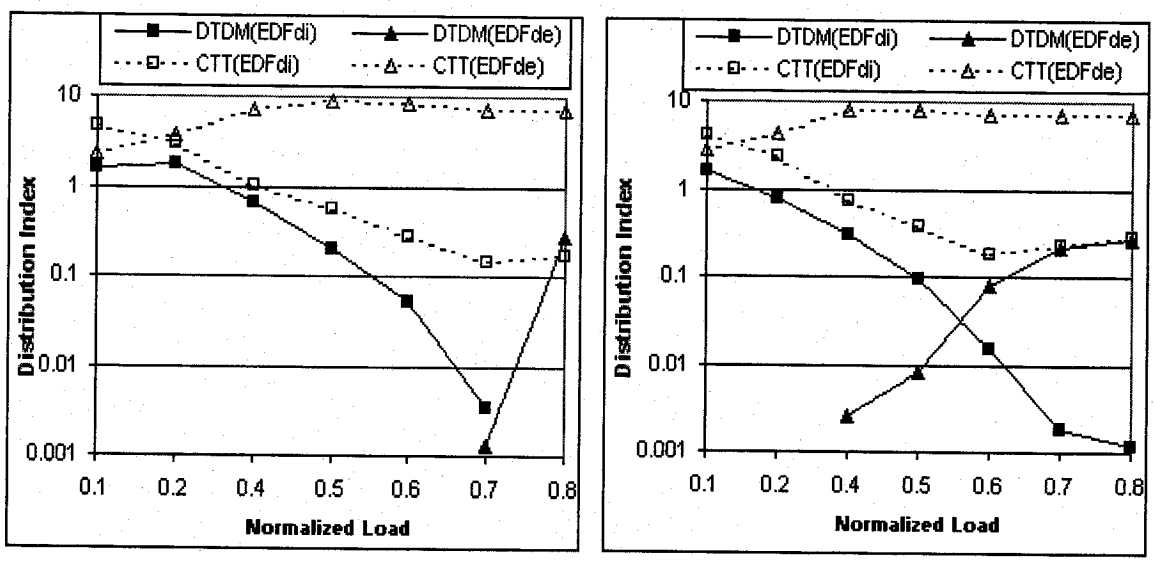
C) Distribution Indexes:

We now compare the average distribution indexes through the simulation period. The average Fairness and EDW distribution indexes are compared in Fig.6.9. First let us recall that the optimum distribution is obtained when an index becomes zero. Then, we see the fairness is always better (lower) for DTDM than CTT under both traffic arrivals. At higher loads, the slot-sets within the frame are gradually becoming filled, and therefore, DTDM cannot find suitable slot-sets to distribute the slots allocated to an egress switch. Then, as expected (see Section 6.3.2), the distribution of the slots within the frame becomes uneven. This is why the Fairness index goes up when load increases. One can see a little increase for the Fairness of CTT at higher loads.

Until L_p=0.5, EDW is better under DTDM than CTT. However, CTT has a very lower EDW at higher loads because almost all wavelength channels in each slot-set become full (see Fig.6.6). By increasing the traffic load, the EDW index of both DTDM and CTT decreases because the channels in each slot-set are gradually becoming full.

The average EDF_{di} and EDF_{de} distribution indexes are compared in Fig.6.10. Both indexes for DTDM are less than CTT under both traffic arrivals. Under DTDM, EDF_{di} decreases when load increases because allocated slots to each egress switch appear with a

higher probability in all slot-sets through the frame. A similar situation is observed for CTT, except there is a small increase at higher loads.



a) Poisson Arrivals

b) Pareto Arrivals

Fig.6.10: The Average EDF_{di} and EDF_{de} Indexes under Poisson and Pareto Arrivals

DTDM has an excellent EDF_{de} = 0 until L_p=0.7 under Poisson traffic and until L_p=0.4 under the Pareto traffic. At higher loads, EDF_{de} is a little higher for DTDM because the slot-sets become almost full, and the probability of finding an empty suitable slot-set for a desired slot is decreased. Since the indiscriminate slot generation nature of CTT causes disorder in slot arrival to the slot transmission queue, the index EDF_{de} for CTT is very high comparing to DTDM. EDF_{de} is small at lower loads because a lower number of slots are generated to each egress switch. When the slot generation rate goes up, EDF_{de} increases until L_p=0.52 (see Fig.6.10a). After that slot generation becomes smoother because AF time-out becomes ineffective (see Fig.6.6), and we observe a little decrease in EDF_{de}.

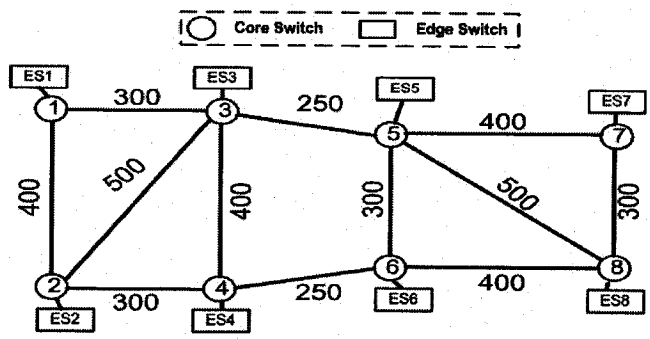


Fig.6.11: A Multi-Hop All-Optical Network

6.4.3 Comparison of DTDM and CTT: Multi-Hop Network

We compare next the performance of CTT in *Arch1* and DTDM in *Arch2* in a multi-hop all-optical network with eight core switches (Fig.6.11). Again *Arch1* has been described in Appendix A and *Arch2* in Section 2.4. An edge switch is connected to each core switch. The distances shown are in km. We have f=2 fibers per link, W=4 channels per fiber, ψ=3 classes, R=9 immediate sources routers connected to each edge switch (not shown in Fig.6.11), and eight shared-per node wavelength converters used in each core switch. To provide another scheme for contention resolution at the core switches, each egress switch has four drop ports (almost a non-blocking receiver) from its relevant core switch.

The source routers connected to each ingress switch generate Poisson traffic where the traffic distribution in each class is the same as before. Each ingress switch transmits traffic to all seven egress switches in the network. Each core switch chooses the shortest path, i.e., the number of hops, to route the slots between an ingress-egress switch pair.

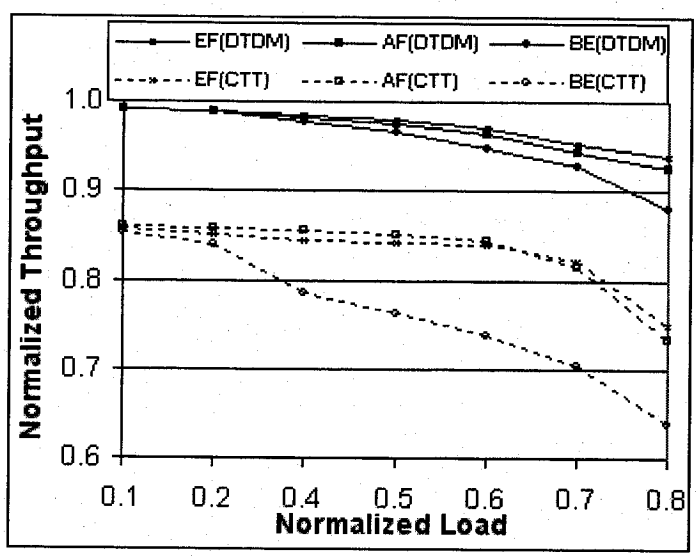


Fig.6.12: Network-wide Average Normalized Throughput of Packets

Fig.6.12 depicts the average normalized throughput of EF, AF and BE packets among all source-destination pairs within the network. It is calculated separately for each class based on the total number of packets received in all egress switches divided by the total number of transmitted packets from all ingress switches. Similar to what shown

before (but omitted here), the slot generation rate is higher under the CTT scheme, which leads to a higher collision in the network. Thus, the DTDM protocol has a better throughput for all classes than CTT.

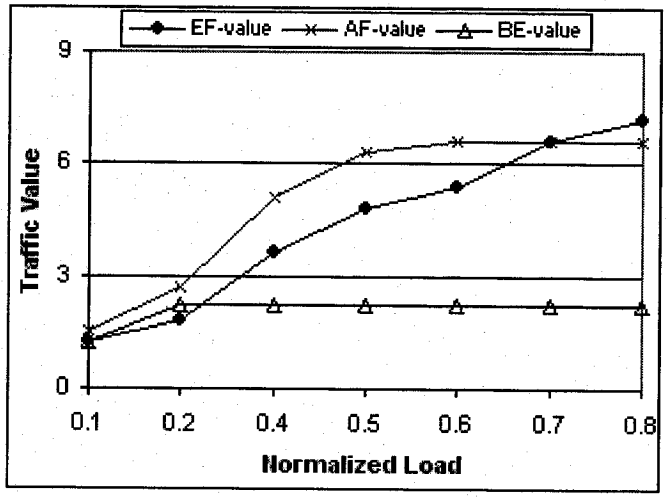


Fig.6.13: The Traffic Value of Slots under CTT

As mentioned in Section 6.3.4, the throughput of a class depends on the traffic value T_v. Under CTT, the traffic value of EF slots shown in Fig.6.13 (calculated based on V_{EF}=0.6, V_{AF}=0.3 and V_{BE}=0.1) is lower than the traffic value of AF slots for L_p<0.7. This is why the AF throughput is higher than the EF throughput for L_p<0.7 under CTT. On the contrary, EF traffic value becomes higher for $L_p \geq 0.7$. Therefore, EF obtains the highest throughput. As mentioned, the traffic value T_v depends on the choice of the "importance" parameters. If we chose V_{EF}=0.8, V_{AF}=0.15 and V_{BE}=0.05, the EF throughput would increase while the AF and BE throughput would decrease.

6.5 Conclusion

The Distributed TDM protocol has been designed to provide access to a slotted all-optical OPS network. The DTDM protocol can provide a guaranteed bandwidth for each torrent within a frame, a stable edge switch operation, and fairness for torrents. It can also reduce collision at the network, thus increasing the network throughput, by smooth traffic transmission, load balancing, and avoiding distribution of the slots going to the same egress switch on the same wavelengths of different fibers. In addition, DTDM has better distribution index parameters than CTT, thus providing a smooth access to the network. Furthermore, DTDM resolves the problem of choosing the time-out value because under

our architecture the service differentiation is provided by class-based packet scheduling. Finally, there is no slot transmission buffer and the slots from the torrents are scheduled directly to the output channels.

Chapter Seven: TDM Architectures for the AAPN Network

Since the next generation Internet is most likely to be based on high-capacity agile all-optical networks, we study and evaluate the performance of the AAPN network using three potential candidates of resource-sharing algorithms. We use this network as an application to our system developed throughout Chapter 2 to Chapter 6. We first detail the AAPN architecture, and then develop the loss-free TDM architectures suitable for AAPN. The DTDM and PR protocols are used together to provide a loss-free contention-based TDM protocol. A centralized TDM protocol is used as a bandwidth-reservation protocol. We then develop an integrated version that combines the good attributes of both TDM protocols. We also provide performance evaluation to show the effectiveness of the integrated TDM in the AAPN network.

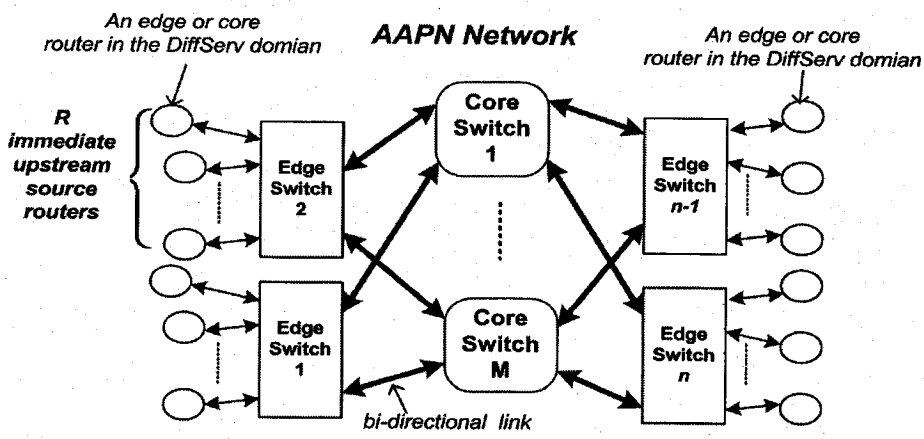


Fig.7.1: The AAPN Network Model

7.1 The AAPN Network Model and Operation

The AAPN network is based on an overlaid all-optical star network with a time-slotted operation as showed in Fig.7.1. AAPN can be considered as a potential candidate for the ultra high-speed next generation all-optical metro networks. The description of the links, the edge switches, the routers/switches (within the DiffServ domain) connected to an edge switch, and the core switches shown in Fig.7.1 are all the same as what described in Section 2.1 for the general network model, except the overlaid part. The overlaid star provides robustness for the AAPN network as discussed in Section 1.1.2.1. In the AAPN network, each core switch at the star is connected to n edge switches where each

connection link has f fibers in each direction. In AAPN, the operation of a core switch in switching traffic is independent of other core switches. This means that the overlaid topology can be divided into independent single-star networks. In the overlaid star topology, network traffic can be equally divided among single-star networks. On the other hand, our general network model can easily support the star topology.

We consider TDM operation of our AAPN network and for the bandwidth management in particular TDM bandwidth management techniques. Two major types are considered: CTDM and DTDM. We use the BvN scheduling (summarized in Appendix G) as our reservation-based protocol that uses the perfect matching idea with the highest performance. Since this is a centralized bandwidth-reservation protocol, we call it CTDM (Centralized TDM). We use our DTDM protocol and the Prioritized Retransmission protocol together in order to have a loss-free all-optical network. Then, based on the pros and cons of the protocols under different network and traffic parameters, we design an ITDM (Integrated TDM) protocol that combines the good attributes of both DTDM and CTDM protocols. We shall characterize AAPN under the mentioned resource-sharing protocols by various measures such as delay and loss probabilities at the edge switches.

The ingress switch architecture is the same for DTDM, CTDM and ITDM, as depicted in Fig.2.2. However, the difference among them lies in the controlling mechanism within the OBM unit. Note the OBM operation for DTDM has already been discussed in Chapter 6, and so only CTDM and ITDM will be detailed in the following sections. All these algorithms use the same traffic transmission protocol as discussed in Section 6.3.3 (i.e. the cooperation between and OBM unit and a BMPS unit to send traffic to OPS network).

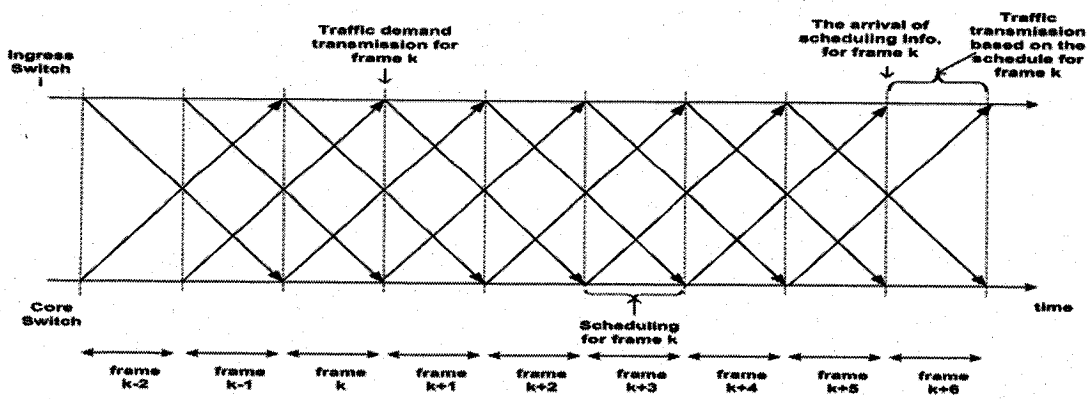


Fig.7.2: The CTDM Scheduling Framework

7.2 The Centralized TDM (CTDM) Protocol

The CTDM framework is similar to a pipeline processing in which ingress switches provide their traffic demand information to the core switch at frame intervals. The core switch schedules within each frame a timetable for each edge switch to transmit its traffic. Fig.7.2 shows the time evolution of the CTDM operation. The propagation delay between the ingress switch and the core switch is two frame periods. The transmission time for either the traffic demand information or the scheduling information between an ingress switch and the core switch is negligible (and therefore not shown) due to the high data rate of the wavelength channels. Various considerations of the CTDM protocol are detailed in the following:

A) Synchronization at the Frame-Level: Since the arrival of slots to the core switch from different edge switches must be synchronized at the frame level in order to respect the collision-free schedule, the edge switch with the longest propagation delay has the most impact on others. Define $T_{s,i}$ to be the time offset that edge switch i must start transmitting its traffic to the core switch after resetting (initializing), $P_{D,i}$ to be the propagation delay from edge switch i to the core switch, and $P_{D,max}$ to be the propagation delay for the furthest edge switch. Assume edge switch m to be the furthest edge switch. Then for a synchronized arrival, one can see that the equality $T_{s,i} + P_{D,i} = T_{s,m} + P_{D,max}$ for $i = 1,2,..n$ must satisfy. For simplicity, we let $T_{s,m}$=0 for the furthest edge switch, thus resulting in $T_{s,i} = P_{D,max} - P_{D,i}$ for any other edge switch that is closer to the core switch.

B) Constructing the Traffic Request in Edge Switches: The OBM unit in an ingress switch calculates the required number of slots to carry the traffic to each one of the egress switches within a frame period. Each ingress switch always performs this procedure at each frame boundary, and we assume this requires only a very small processing time. Define $N_{RS,ij}$ to be the number of required slots to carry the traffic from ingress switch i to egress switch j (where $1 \leq i, j \leq n$ and $i \neq j$) that has arrived during the previous frame, d'_{ij} to be the demand of required slots between ingress switch i and egress switch j, and d_{ij} to be the number of required slots between ingress switch i and egress switch j that OBM reports to the core switch. Now, we say the demand of required slots between ingress switch i and egress switch j is granted if $d_{ij} = d'_{ij}$. Otherwise, $N_{NG,ij} = d'_{ij} - d_{ij}$ is the number

of required slots that are not granted (see the following procedure) between ingress switch i and egress switch j during the previous frame. The operation of ingress switch i can now be described by the following procedure:

1. Measure the traffic demand entry d'_{ij} between ingress switch i and egress switch j in an integral number of slots such that
   $d'_{ij} = N_{NG,ij} + N_{RS,ij}$, (7.1)
2. Calculate the slot-scaling coefficient, $\gamma_i = min(1, \frac{fFW}{\theta_i})$ where $\theta_i = \sum_{j=1}^{n_e} d'_{ij}$.
3. Adjust the number of slots demanded to egress switch j by $d_{ij} = \lfloor \gamma_i d'_{ij} \rfloor$.
4. Update the parameter $N_{NG,ij} = d'_{ij} - d_{ij}$ for use in the next frame.
5. Send the traffic demand to the core switch at the frame boundary so that the core switch will receive all traffic demands at the same time.

Whenever the total number of the slots demanded to all egress switches exceeds the total number of available slots within a frame period, i.e., fFW, we need to reduce the required number of slots to each egress switch proportionally by the factor γ_i when $\gamma_i<1$ (see Step-2 and Step-3). Then, the number of non-granted required slots is updated at Step-4. Otherwise, when $\gamma_i=1$, then the requested slots to each egress switch can be completely demanded to the core switch, i.e., the demand is granted, where we have $N_{NG,ij}=0$ at Step-4.

Example: For $i=1$, $n=4$, $f=1$, $F=50$, $W=8$, $N_{RS,12}=180$, $N_{RS,13}=130$, $N_{RS,14}=150$, $N_{NG,12}=10$, $N_{NG,13}=20$, and $N_{NG,14}=5$, we compute $d'_{12}=190$, $d'_{13}=150$, and $d'_{14}=155$. Based on this demand, we calculate $\gamma_1=0.8081$. We obtain traffic demand entries by $d_{12}=153$, $d_{13}=121$, and $d_{14}=125$. Then, the non-granted required slots for the next frame are $N_{NG,12}=37$, $N_{NG,13}=29$, and $N_{NG,14}=30$.

C) Scheduling Traffic Demand Matrix in the Core: The following details the scheduling process at the core switch:

1. Receive the traffic requests from each ingress switch at frame boundary.
2. Divide each request equally among fW wavelength channels.
3. Repeat the following Steps-4 to 10 for wavelength channel N_w ($1 \le N_w \le W$) on fiber N_f ($1 \le N_f \le f$).
4. Create the traffic demand matrix $\mathbf{D_{n,n}} = [d_{ij}]_{nxn}$, where d_{ij} is the number of requested slots between ingress switch i and egress switch j.
5. Update the traffic demand matrix by $d_{ij} = d_{ij} + max(0, r_{ij})$ where $i,j=1,2,..,n$ and r_{ij} is the difference between the number of the requested and the granted slots for the traffic between ingress switch i and egress switch j at the previous frame.
6. Compute the matrix scale down factor $v = max(r_m, c_m)$, where r_m is the maximum of all row summations and c_m is the maximum of all column summations both in matrix $\mathbf{D_{n,n}}$.
7. If $v > F$, then adjust each entry in $\mathbf{D_{n,n}}$ by $d_{ij} = \frac{F}{v} d_{ij}$, and go to Step-4.

8. Convert the matrix $\left[\frac{d_{ij}}{v}\right]_{n,n}$ to a doubly stochastic[11] matrix $\mathbf{R_d}$.
9. Decompose the doubly stochastic matrix $\mathbf{R_d}$ using the BvN theorem such that $\mathbf{R_d} = \sum_{i=1}^{k} u_i \mathbf{P_i}$, where $\mathbf{P_i}$ is a permutation matrix, k is the number of permutation matrices, and u_i is the permutation weight.
10. Obtain the schedule related to each edge switch from $\mathbf{S} = v\sum_{i=1}^{k} u_i \mathbf{P_i}$, where v is computed in Step-6.
11. Announce each edge switch its new schedule on each wavelength channel and fiber.

In Step-5, two cases may happen for r_{ij}: 1) If $v > F$ (see Step-7), then the core switch scales down the traffic demand entries proportionally and grants less slot-schedule to any ingress switch, thus leading to a positive r_{ij}; and 2) The procedure of converting the demand matrix to a doubly stochastic matrix may grant more slots than the requested for the traffic between ingress switch i and egress switch j. Here, we have $r_{ij} < 0$, but we do not reduce the extra granted slots in the previous frame from the requested demand during the current frame, a bonus to CTDM.

Step-4 to Step-10 in the above procedure is executed once for the same traffic demand matrices. In the best case, these steps are executed once for all fW traffic demand matrices. However, in the worst case, these steps must be executed for each one.

$$\mathbf{D} = \begin{bmatrix} 0 & 3 & 5 & 0 \\ 4 & 0 & 3 & 2 \\ 2 & 0 & 0 & 6 \\ 1 & 4 & 2 & 0 \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 0 & 5 & 5 & 0 \\ 5 & 0 & 3 & 2 \\ 2 & 0 & 0 & 8 \\ 3 & 5 & 2 & 0 \end{bmatrix} = 3\mathbf{P_1} + 5\mathbf{P_2} + 2\mathbf{P_3} = 3\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} + 5\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} + 2\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

a) Traffic Demand Matrix Example

b) Decomposed Matrix

ES_1	2	2	2	3	3	3	3	3	2	2
ES_2	3	3	3	1	1	1	1	1	4	4
ES_3	4	4	4	4	4	4	4	4	1	1
ES_4	1	1	1	2	2	2	2	2	3	3
	3 columns extracted from $\mathbf{P_1}$			5 columns extracted from $\mathbf{P_2}$					2 columns extracted from $\mathbf{P_3}$	

c)Timetable of each Edge Switch within a Frame of F=10 slots on a Single Wavelength Channel

Fig.7.3: A Demand Matrix, its Decomposition, and Edge Switches Schedule

[11] In a doubly stochastic matrix, the summation of each row and the summation of each column is equal to 1.0.

Example: Fig.7.3a gives a 4x4 traffic demand matrix created for a wavelength channel by a core switch connected to n=4 edge switches. For instance, ingress switch#2 needs 4, 3 and 2 slots (see the second row of matrix **D** in Fig.7.3a) to send its traffic to egress switches 1, 3 and 4 respectively within a frame. Fig.7.3b shows decomposed matrix **S** (see Step-10) with three permutation matrices. Fig.7.3c displays the timetable of each edge switch in traffic transmission, where a number shown in the table is and egress switch address. One can see that, the first three (3 is the coefficient of $\mathbf{P}_1$) columns have the same schedule. Then, the next five (5 is the coefficient of $\mathbf{P}_2$) columns have the same schedule. Finally, the last two (2 is the coefficient of $\mathbf{P}_3$) columns have the same schedule. Given a permutation matrix $\mathbf{P}=\{p_{ij}\}$, edge switch i has the schedule to egress switch j if p_{ij}=1. For example, consider the first column, extracted from $\mathbf{P}_1$, in the frame shown in Fig. 7.3c. In matrix $\mathbf{P}_1$, we have $p_{12}=p_{23}=p_{34}=p_{41}=1$, and therefore, ES_1, ES_2, ES_3, and ES_4 each can have a schedule to send one slot to egress switches 2,3,4,1 respectively.

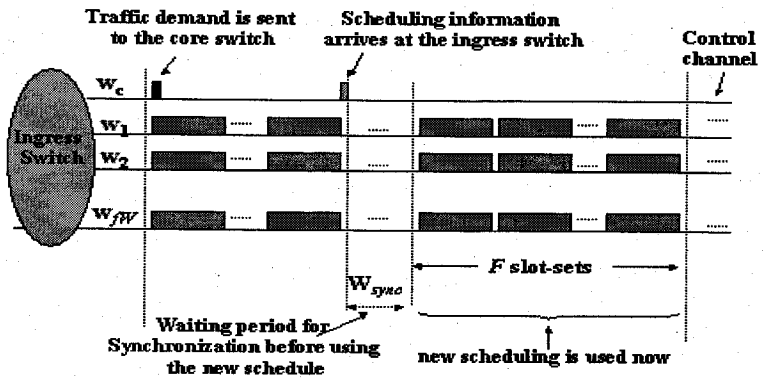
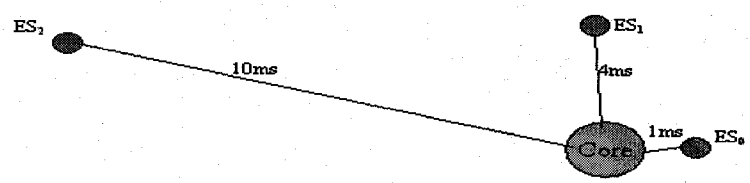


Fig.7.4: Timing Diagram for Waiting Period for Synchronization in CTDM



Edge Switch #	Edge sends traffic demand to core at time	Core receives the demands and sends the scheduled information at time	Edge receives new schedule from core at time	Edge transmits traffic based on the new schedule at time	Core receives traffic based on the new schedule at time	W_{sync} in msec	Minimum waiting time (penalty) in msec
0	t_0	t_0+2	t_0+3	t_0+21	t_0+22	18	18
1	t_0-3	t_0+2	t_0+6	t_0+18	t_0+22	12	12
2	t_0-9	t_0+2	t_0+12	t_0+12	t_0+22	0	0

Fig.7.5: A Sample Scenario and Waiting for Synchronization Issue

D) Transmitting Scheduled Traffic at the Edge Switches: Ingress switch i must always wait for the synchronization period of $W_{sync,i}= 2(P_{D,max}-P_{D,i})$, before transmitting its traffic based on the new schedule (see Fig.7.4). However, there is nothing to stop it from transmitting traffic from the previous schedule while waiting for the synchronization of the new schedule. This overlapping of operation allows us to increase the utilization of channel while reducing the waiting time.

Example: Fig.7.5 shows the waiting period for synchronization among three edge switches located at different distances from the core switch, where all displayed times are in msec. Let us review the traffic transmission procedure for edge switch #1 (ES_1). ES_1 sends traffic demand to the core switch at time t_0-3. The core switch receives the traffic demands from all ingress switches at t_0+1 and after performing the scheduling (assume the scheduling is performed in 1msec), it sends the scheduled information to all ingress switches at time t_0+2. ES_1 receives the new scheduling information from core switch at time t_0+6, and then transmits its traffic based on the new schedule at time t_0+18. One can see that ES_1 must wait for a time penalty of 12msec (i.e. W_{sync}=2(10-4)=12) before using the new schedule. If ES_1 transmits its traffic based on the new schedule as soon as it receives the new scheduling information, its slots may collide with the slots of ES_2 from the previous schedule at the core switch. Finally, the core switch receives traffic based on the new schedule starting at time t_0+22.

7.3 Comparison of CTDM and DTDM

The DTDM and CTDM schemes have a number of pros and cons. First, waiting time for packet transmission to the network may be higher for CTDM than DTDM because packets may need to wait for the scheduling process. In addition, the distributed scheme can quickly handle the Internet bursty traffic, while the centralized scheme cannot because of the rapid fluctuation in traffic arrival. On the other hand, a distributed scheme has the aforementioned advantages, at the expense of some traffic drop at the core switch. To have a loss-free distributed scheme, the dropped traffic has to be retransmitted. However, the retransmission of the dropped traffic may lead to a higher traffic transmission in an ingress switch especially at higher loads. This in turn may result in traffic drop at the ingress switches.

As already shown in [RaYa05a], DTDM has a better performance results than

CTDM at lower traffic loads due to a lower number of collisions and retransmissions. However, CTDM can have a better performance than DTDM at higher traffic loads due to the absence of growing volume of retransmitted traffic found in DTDM. Clearly, by increasing either the number of contention resolution hardware such as wavelength converters at the core switch or the number of contention avoidance hardware such as fibers in connection links [RaYa05a] the collision rate reduces, and DTDM can obtain a better performance than CTDM even at higher loads. Since our switch architecture can support both DTDM and CTDM, it can be easily converted into an integrated architecture, as discussed in the following section.

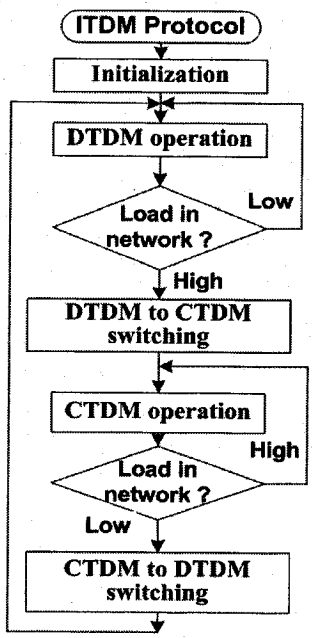


Fig.7.6: The Evolution of the ITDM Protocol

7.4 The Integrated TDM (ITDM) Protocol

We design a hybrid scheme called Integrated TDM (ITDM) that combines the good attributes of DTDM and CTDM. The evolution of the ITDM protocol is depicted in Fig.7.6 in which the network works under DTDM at lower traffic loads, and under CTDM at higher traffic loads. Note that our frame-based mechanism, the slotted network, and the core switch architectures can easily support both DTDM and CTDM. Therefore, the ITDM system can easily switch between the protocols. The core switch decides when to switch between the protocols at frame boundaries and then announces

the switching by sending a command on the control channel to all edge switches. Then, each edge switch will have to switch its operation as requested at a synchronized time (see the synchronization issues in Sections 7.2 and 7.3). For simplicity, we assume the system always starts from DTDM whenever ITDM is reset. Before detailing the switching protocols, let us first define a few parameters for the ITDM protocol:

1. Channel utilization definitions under DTDM: Let $N_{T,i}$ be the total number of non-empty slots transmitted from ingress switch i within a frame. We then define $C_{U,f}$ to be the average channel utilization of time-slots in the network within a frame. We also define $C_{U,T}$ to be the threshold value for the maximum channel utilization of time-slots set by the network manager.
2. Traffic arrival definitions under DTDM: Let Λ_s be the average normalized load of slot arrival from all ingress switches to the core switch, and λ_s be the average normalized traffic load of newly generated slots from all ingress switches. Define λ_T to be the normalized traffic load of the newly transmitted slots, when $\Lambda_s = C_{U,T}$.
3. Traffic loss definitions under DTDM: Let R_{SL} be the average slot loss rate, and $R_{SL,f}$ be the average slot loss rate within a frame, at the core switch when system is running under DTDM. Also, define $R_{SL,T}$ to be the slot loss rate when $\Lambda_s = C_{U,T}$.
4. Traffic schedule definitions under CTDM: Let the traffic scheduled at the core switch be denoted by matrix $[s_{ij}]_{n,n}$ where s_{ij} is the number of slots that ingress switch i is allowed to transmit to egress switch j within a frame period. We then define the normalized traffic load at the core switch by $\lambda_c = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}}{nFfW}$, where the denominator is the total number of slots that can be carried in one frame from all ingress switches within the frame.
5. Miscellaneous definitions: 1) Define S_p to be the stability period in consecutive frames. This parameter avoids system oscillation between CTDM and DTDM; 2) Let $N_{B,ij}$ be the number of slots required to carry the traffic that is left in the buffers relevant to ingress switch i and egress switch j (under DTDM), as a non-granted traffic under CTDM; and 3) Let $N_{R,ij}$ be the number of slots that must be retransmitted from ingress switch i to egress switch j.

7.4.1 Switching from DTDM to CTDM

The following considerations and procedures are required to switch from the DTDM operation to the CTDM operation:

A) Switching Criterion: The core switch performs the following algorithm at each frame boundary for a possible switching:

1. Compute R_{SL} by the running average method $R_{SL} = \sigma R_{SL} + (1-\sigma)R_{SL,f}$ where σ is a smoothing factor.
2. Compute $C_{U,f} = \frac{\sum_{i=1}^{n} N_{T,i}}{nFfW}$, where $0 \le C_{U,f} \le 1$.
3. If $C_{U,f} \ge C_{U,T}$, then $k = k + 1$, else $k = 0$
4. If $k \le S_p$, then goto Step-8
5. Set $R_{SL,T} = R_{SL}$ and compute $\lambda_T = C_{U,T}(1-R_{SL,T})$.
6. $k = 0$
7. Send a command to each edge switch to switch from DTDM to CTDM and follow the CTDM operation
8. Goto the evaluation of DTDM → CTDM criterion in the next frame

When ITDM is running under the DTDM operation, we have $\Lambda_s = \lambda_s + \Lambda_s R_{SL}$ where the second term represents the normalized feedback traffic load of the dropped slots due to the slot loss rate R_{SL} at the core switch. By increasing λ_s, the slot loss rate will also increase until the critical point $\Lambda_s = \lambda_s + \Lambda_s R_{SL} = 1$. Beyond this point, any further increase would cause the queues in the ingress switch to grow. Packet drop would occur if a finite buffer is used. The ITDM system tries to switch from DTDM to CTDM before reaching to the critical point, handled by $C_{U,T}$ in our work. At the critical point, we then have $C_{U,T} = \lambda_T + C_{U,T} R_{SL,T}$. Before switching to CTDM, the core switch calculates λ_T (see Step-5), to be used later on when switching from CTDM to DTDM.

The system will switch to CTDM if the average channel utilization remains greater than or equal to the threshold value $C_{U,T}$, i.e., $C_{U,f} \ge C_{U,T}$, for at least a stability period S_p (handled by counter k in the above algorithm). This avoids false switching from DTDM to CTDM due to traffic fluctuations. For example, S_p=100,000 means that the condition $C_{U,f} \ge C_{U,T}$, must be held for a period of 100,000 consecutive frames.

B) Switching Protocol at an edge switch: Ingress switch i must respect the following algorithm:

1. Receive the switching command at time τ_i.
2. Set $N_{NG,ij} = N_{B,ij}$, for all j.
3. Obtain $N_{R,ij}$ and $N_{RS,ij}$, for all j.
4. Compute d'_{ij} by $d'_{ij} = N_{NG,ij} + N_{RS,ij} + N_{R,ij}$ instead of Eq.(7.1) , for all j.

5. Set a new frame boundary starting at time $\tau_i + 2(P_{D,max} - P_{D,i})$.
6. Follow the Step-2 to Step-5 in the algorithm detailed in Section 7.2.B.
7. Follow the CTDM operation, as discussed before.

When the system has already switched from DTDM to CTDM, there may be a number of slots (left by the DTDM protocol) that must be retransmitted under CTDM. This is why we account $N_{R,ij}$ in d'_{ij} right after switching (see Step-4). Then the traffic demands, computed from Section 7.2.B, are sent to the core switch from the frame boundary given in Step-5. Then, the CTDM protocol is followed by each edge switch and the core switch in the network.

7.4.2 Switching from CTDM to DTDM

The following considerations and procedures are required to switch from the CTDM operation to the DTDM operation:

A) Switching Criterion: The core switch performs the following algorithm at each frame boundary for a possible switching:

1. Compute $\lambda_c = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}}{nFfW}$, where $0 \le \lambda_c \le 1$.
2. If $\lambda_c \le \lambda_T$, then $k = k + 1$, else $k = 0$
3. If $k \le S_p$, then goto Step-6
4. $k = 0$
5. Send a command to each edge switch to switch from CTDM to DTDM and follow the DTDM operation
6. Goto the evaluation of CTDM → DTDM criterion in the next frame

The core switch will request the edge switches to switch to DTDM provided that the average load λ_c at the core switch remains less than λ_T (given in Section 7.4.1), i.e., $\lambda_c < \lambda_T$, for at least a stability period S_p. Therefore, the false switching from CTDM to DTDM due to traffic fluctuations can be avoided.

B) Switching Protocol at an edge switch: After receiving the switching command, each ingress switch will follow the transmission of its traffic based on the DTDM protocol at the first frame boundary after receiving the command.

7.4.3 Complexity

We can now compare the complexity of DTDM, CTDM and ITDM as follows:

1. As discussed in Appendix G, the complexity of CTDM for one channel is $O(n^{4.5})$. This complexity increases to $O(fWn^{4.5})$ for fW wavelength channels in the worst case

(see Section 7.2.C).

2. The complexity of DTDM as discussed in Section 6.3.2 is almost $O(FfW)$ for fW wavelength channels in the worst case. Note that the complexity of DTDM for one channel is increasing linearly. Since in practice F is a small number (say F=100), DTDM has a lower complexity than the complexity of $O(n^{4.5})$ in CTDM, except when n is very small, *e.g.*, $n < 3$. However, $n < 3$ is not a reasonable value in practice.
3. The ITDM may cause some complexity overhead as follows: 1) When switching from CTDM to DTDM, the parameter λ_c is computed with $O(n^2)$ complexity, but this is once within a frame. The switching command has a negligible overhead. An ingress switch starts transmitting based on DTDM as soon as receiving the switching command with no extra overhead on the operation of ingress switch; and 2) When switching from DTDM to CTDM, the core switch must compute channel utilization in each time-slot. This needs to scan all nfW slots in each time-slot to determine the number of non-empty slots. However, this process can be done with $O(1)$ complexity when the core switch controller scans each SSH (for the purpose of switching and contention resolution) to obtain the information of the slots to be arriving. Then, the parameter $N_{T,i}$ can be easily updated. Therefore, the parameter $C_{U,f}$ is computed with $O(1)$ complexity. The other parameters related to switching from DTDM to CTDM are all computed with $O(1)$ complexity. The transmission of the switching command has a negligible complexity. After receiving the switching command from the core switch, each ingress switch follows CTDM. This causes no overhead, however, it takes some time from transmitting the first traffic demand until using the loss-free slot schedule for that demand. During this period, each ingress switch uses the previous DTDM schedule to send its traffic.

In summary, DTDM has a lower complexity than CTDM in practice. The ITDM reduces to the complexity of DTDM in its best-case and increases to the complexity of CTDM in its worst-case scenarios.

7.5 Performance Evaluation

We would like to compare the performance of DTDM, CTDM and ITDM agile resource-sharing schemes in terms of end-to-end packet delay and traffic loss in all ingress switch

buffers throughout the AAPN network. We assume that the CTDM scheduling can be executed in less than a frame length (e.g., 1ms). Define the end-to-end delay to be the interval between the arrival-time of a packet at an ingress switch until its successful arrival at the egress switch. Then the average network end-to-end delay of class *i* packets is the average end-to-end delay of all class *i* packets throughout the network. In the following simulations, we show the traffic loss in ingress switches and the average network end-to-end delay (simply referred to as "delay") in 3-minute intervals.

We model the daily background traffic of the network by a sinusoidal traffic pattern. Define $L_i(t)$ to be the normalized load of packet arrival (normalized with respect to the total available bandwidth) in ingress switch i at given time t, where $L_{p,min} \leq L_i(t) \leq L_{p,max}$. Then, $L_i(t) = \frac{L_{p,max} - L_{p,min}}{2} \sin(2\pi t/T + \varphi_i) + \frac{L_{p,max} + L_{p,min}}{2}$, where T is the 24-hour period, and φ_i is the phase representing the time zone of ingress switch i. The parameter φ_i is randomly chosen between $-\pi/6$ and $+\pi/6$ in ingress switch i. To model the bursty traffic, packet inter-arrival times for each class are i.i.d distributed under any normalized traffic load according to a Pareto distribution with a p.d.f $p(t) = \alpha b^{\alpha} t^{-\alpha-1}$. We use α=1.3 to obtain the Hurst parameter $(3-\alpha)/2$=0.85. The generated traffic follows packet length distributions obtained from [CAID06], where the distribution of packet sizes are: 46% of size 40B(Bytes), 18% of size 552B, 18% of size 576B, and 18% of size 1500B. The traffic from each source router in any ingress switch is symmetric to any egress switch and comprises 25%, 35% and 40% of EF, AF and BE traffic respectively under any traffic load. We also set $L_{p,min}$=0.2 and $L_{p,max}$=0.8 for our traffic generation.

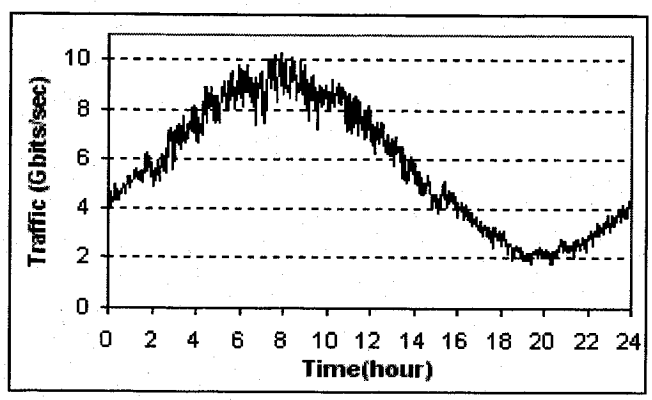
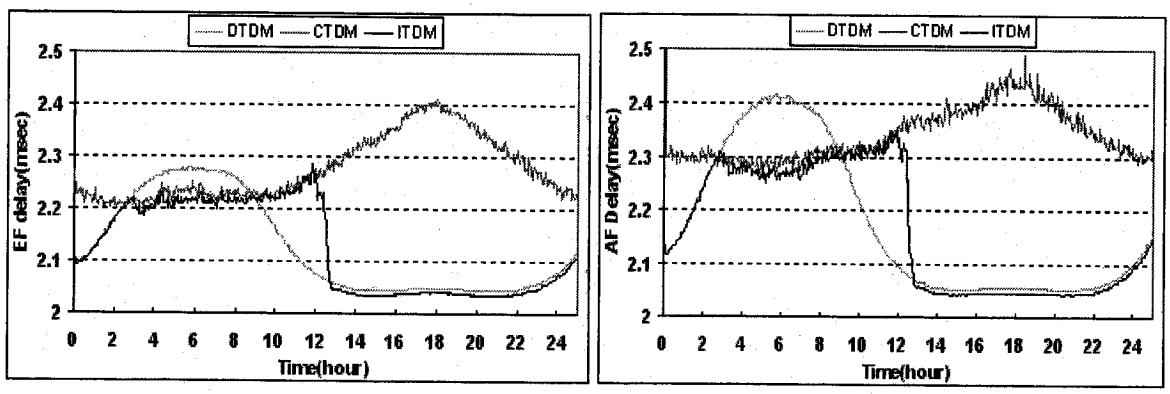


Fig.7.7: A Sample Traffic between an Ingress-Egress Switch Pair with $\varphi_i = -\pi/6$

Fig.7.7 shows a sample traffic rate pattern between an ingress switch to an egress switch with $\varphi=-\pi/6$ which follows the aforementioned sinusoidal pattern under the Pareto packet arrivals. One can think of time t=0 in this figure as 7:00am when the daily work starts. This pattern can be divided into two interval types. In the congested interval, the network utilization is high (say interval 9:00am to 7:00pm), while the network utilization is medium or low in the non-congested interval.

We run simulation on one single-star (Fig.7.1) with n=8 edge switches, f=2 fibers on each connection link, W=2 wavelengths per fiber, B_C=10Gbits/sec, four shared-per-node WCs are used at the core switch, S_T=9μs, S_O=1μs, and buffer size for each stream traffic of any source router at each ingress switch is set to 8Mbits. The protocol parameters are set as: a frame length of F=100 slot-sets (going a frame period of 1ms), S_p=360,000 frames units (i.e. 6 min), σ_1 = 0.5, $C_{U,T}$= 0.9, V_{EF}=0.6, V_{AF}=0.3, V_{BE}=0.1, C_1=1.0, C_2=0.6 and C_3=0.5 We have used OPNET simulator [OPNE06] to develop our simulation models. The results are based on the average from six simulation replications.



a) The EF Traffic Delay in Network

b) The AF Traffic Delay in Network

Fig.7.8: End-to-End Delay Comparison in Scenario-1

7.5.1 Performance Comparison for Equal Distance Edge Switches

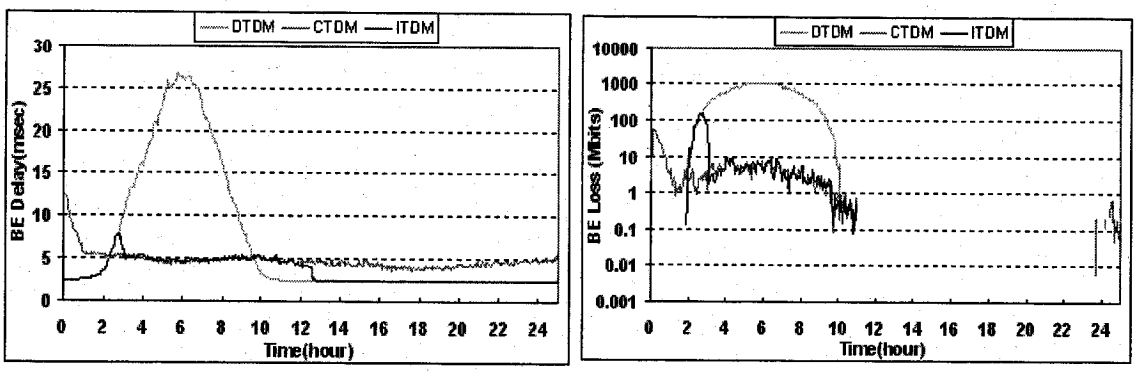
Fig.7.8 compares the three resource-sharing techniques in an equal-distance scenario (referred to as Scenario-1 from now on) in which all edge switches are located at the distance of 200km from the core switch, where the propagation delay between any two edge switches by itself is 2msec. Under DTDM, the EF delay is an increasing function of traffic load (see Fig.7.8a). On the other hand, the CTDM scheme has a higher EF delay at the non-congested interval than the congested interval. This is because the number of

required and also granted slots within the frame to carry the CTDM traffic at the non-congested interval is less than the congested interval. Therefore, the packets must wait a longer time until being transmitted in slots. The EF delay is higher for DTDM than CTDM at the congested interval. This is because DTDM experiences a higher loss rate at the core switch at the congested interval and therefore the number of retransmissions is increased. The retransmission of the lost slots by the ingress switches reduces the available bandwidth for the new traffic. This issue in turn results in a higher waiting time and even the traffic drop in the ingress switch buffers.

Under ITDM, DTDM is running in an interval roughly around [0, 2.89] when the traffic load is low. Then, switching to CTDM occurs at t=2.89hour, using the switching criterion discussed in Section 7.4.1, at the point (λ_T = 0.794, C_U = 0.924, and $R_{SL,T}$ = 0.118) because we have $C_U > C_{U,T}$ for the stability period of S_p=360,000 consecutive frames. Then, CTDM is running in the network until 12.47hour on average when traffic load is high before dropping back. At time 12.47hour, the system switches to DTDM, using the switching criterion discussed in Section 7.4.2, because during all S_p=360,000 consecutive frames we have $\lambda_c < \lambda_T$. At the switching point, we have λ_c = 0.683. One can see a rapid drop in delay after that where DTDM is again running when the traffic load is low. One can see that, ITDM has a better delay performance throughout the traffic cycle, except around the switching interval [9.4, 12.5].

Fig.7.8.b shows that under the same protocol, the EF delay is always less than the AF delay. This differentiation is provided by the operation of the packet schedulers used in each ingress switch. At a lower traffic load, the difference between them is smaller. However, the difference between the EF and AF delays increases up to 0.14, 0.49 and 0.10 msec under DTDM, CTDM and ITDM respectively.

As discussed in Section 7.4.1, the sensitivity of operation switching is controlled by the parameter S_p. To see the effect of this parameter, we have performed another simulation for ITDM with S_p=108,000 (the results are not shown here), and found out that the system changes from DTDM to CTDM at 2.77 hour and from CTDM to DTDM at 11.64 hour. This is compared to 2.89 hour and 12.47 hour when S_p=360,000 is used before. We thus can reduce the interval in which ITDM does not have a better performance.



a) The BE Traffic End-to-End Delay in Network b) The BE Traffic Loss (Mbits) in Edge Switches

Fig.7.9: Performance Comparison of the BE Traffic in Scenario-1

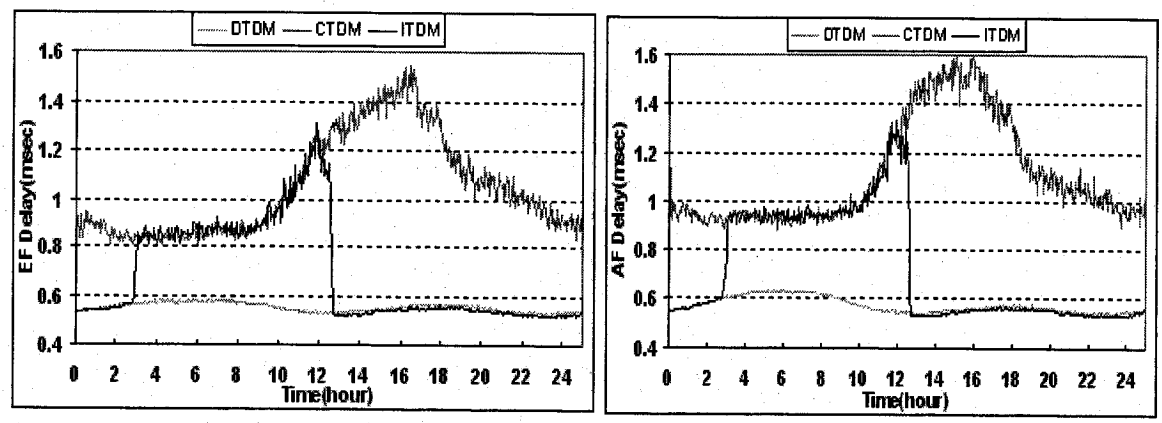
Fig.7.9a compares the delay performance for the BE traffic in Scenario-1. The BE delay is higher for DTDM than CTDM at the congested interval due to the aforementioned retransmission issue. However, DTDM outperforms CTDM in the non-congested interval. ITDM always provides the better delay, except around the switching intervals [2.4, 3.0] and [9.6, 12.7].

Fig.7.9b shows the network-wide BE traffic loss in ingress switches (in a logarithmic scale). We did not expect any loss for the EF and AF traffic because each OCGRR packet scheduler starts from the EF traffic, then AF traffic, and then the BE traffic. Therefore, only the BE traffic may be under a higher risk of traffic drop. Since there is a lower traffic arrival in the non-congested interval, there is no loss under any protocol in the interval [12.25, 23.25]. The BE traffic loss is higher under DTDM than CTDM at the congested interval because of the retransmission of slots that is lost at the core switch. Unlike DTDM, the CTDM scheme experiences the BE traffic drop even in mid-range traffic loads (e.g., normalized traffic loads around 0.5). In general, the BE traffic has a smaller loss rate under ITDM, except around the switching interval [2.0, 3.05].

In summary for Scenario-1, ITDM provides a better performance in terms of delay and loss for the EF, the AF and the BE traffic in all traffic loads comparing to DTDM and CTDM, except around time intervals in which switching happens.

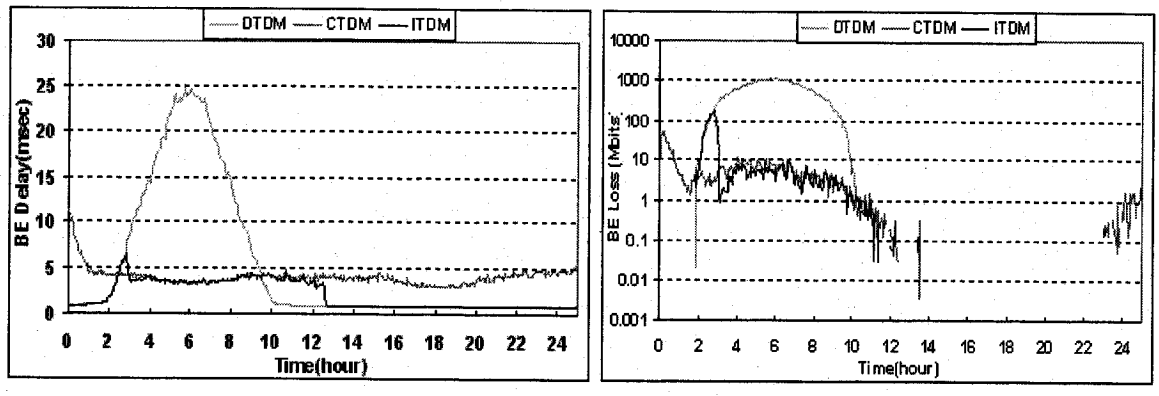
7.5.2 Performance Comparison for Non-Equal Distance Edge Switches

We now compare DTDM, CTDM and ITDM in a non-equal-distance scenario (referred to as Scenario-2 from now on) where the eight edge switches are located at the distance of 2km, 4km, 10km, 16km, 20km, 40km, 100km and 200km from the core switch.



a) The EF Traffic Delay in Network b) The AF Traffic Delay in Network

Fig.7.10: End-to-End Delay Comparison in Scenario-2



a) The BE Traffic End-to-End Delay in Network b) The BE Traffic Loss (Mbits) in Edge Switches

Fig.7.11: Performance Comparison of BE Traffic in Scenario-2

Fig.7.10 illustrates the delay performance comparison for the EF and AF traffic in Scenario-2. The behaviors are similar to Fig.7.8, except for the following: 1) The EF and AF delays are smaller than the Scenario-1 due to the smaller propagation delays. In addition, DTDM has always a smaller EF and AF delay than CTDM. This is because, the overall propagation delays in Scenario-2 are less than Scenario-1 and slots experience a lower retransmission delay; and 2) The ingress switches must now consider the synchronization issue (see Section 7.2) under CTDM. These two factors results in a significant delay difference between DTDM and CTDM in Scenario-2. Under ITDM, the system switches from DTDM to CTDM at 2.91 hour on average and returns to DTDM at 12.54 hour on average. We have the following parameters before switching from DTDM to CTDM: $\lambda_T = 0.790$, $C_U = 0.913$, and $R_{SL,T} = 0.119$. In this scenario, the EF and AF delay becomes higher than DTDM at higher traffic load. On the other hand, the

performance for the BE traffic becomes much better (see Fig.7.11).

Fig.7.11 compares the delay performance and the network traffic loss performance (in a logarithmic scale) for the BE traffic in Scenario-2. The volume of the BE traffic loss is lower in this scenario than Scenario-1. The observations here are similar to Scenario-1. In general, the BE traffic loss in Scenario-2 is higher than the BE traffic loss at Scenario-1 under CTDM. This is because ingress switches under CTDM must wait for the synchronization period (Section 7.2) before using the up-to-date scheduling information. Recall that the ingress switches use the previous scheduling information during the synchronization period. Under the bursty traffic, this in turn leads to a higher waiting time and even loss in the edge switches. As aforementioned, OCGRR shifts the loss from the EF and AF traffic to the BE traffic and therefore BE traffic experiences a higher loss.

In summary for Scenario-2, ITDM again provides a better performance in terms of delay and loss for the EF, the AF and the BE traffic but in lower and middle-range traffic loads when compared to DTDM and CTDM, except around time intervals in which switching happens. In a higher traffic load, the performance is better for the BE traffic, but not for the EF and AF traffic.

Although we used the BvN scheduling as our CTDM scheme in this chapter, its complexity is very high to allow BvN to be implemented easily in hardware. In practice, CTDM scheduling should use a heuristic algorithm with a lower complexity. Since heuristic algorithms do not use perfect matching as BvN, they cannot guarantee to schedule the whole traffic demand. Therefore, CTDM would experience even a higher delay values than what we observed in this chapter.

7.6 Conclusion

We have compared our DTDM scheme with the best performance CTDM scheme from what we have improved the performance by using a hybrid approach in the AAPN network. DTDM has a better performance results under a lower traffic load. However, CTDM has better results at higher traffic loads. Therefore, we have proposed the ITDM scheme to combine these two protocols. Our simulations demonstrate ITDM is more superior than DTDM and CTDM in the AAPN network since it can achieve most of the time a better performance under both lower and higher traffic loads.

Chapter Eight: Design Guidelines and Recommendations

In this chapter, we shall provide the operation ranges of our protocols discussed in this dissertation. The following is a list of important parameters:

1) **Choosing the time-slot intervals S_O and S_T :** The slot-offset interval S_O includes a guard time to allow for timing uncertainties (T_{guard}), a processing time (T_{proc}) at the core switch, and a switching time (T_{sw}). As already discussed in Section 2.3, the processing time during the slot-offset interval consists of the evaluation of the potential contention, the resolution of the contention, the transmission of NACK commands to relevant ingress switches if required, and finally making the core switch ready to switch the arriving slots toward their desired egress switches. Therefore, we should have $S_O > T_{guard} + T_{proc} + T_{sw}$. For example, if we require a guard time of 100nsec, processing time of 500nsec, and switching time of 600nsec, then S_O must be at least 1200nsec. Note that S_T must be chosen in a way that the bandwidth wastage, discussed in Section 2.3, is minimized. For instance, consider the design of a slotted network with a wavelength channel bandwidth B_C =10Gbits/sec and the average packet size ℓ=4000 bits. If we now have S_O =1.2μsec and require at most 10% bandwidth overhead, then we have

$$\frac{S_O}{S_O+S_T}+\frac{(B_C S_T) \bmod \ell}{B_C(S_T+S_O)}=\frac{1.2}{1.2+S_T}+\frac{(10000 S_T) \bmod 4000}{10000(S_T+1.2)} \le 0.10 \Rightarrow 1000 S_T-[(10000 S_T) \bmod 4000] \ge 10800$$

.One can see so many desirable values for S_T such as S_T =10.8μsec, S_T =11.2μsec, S_T=11.21μsec, etc. However, choosing a large value for S_T increases queuing delay in ingress switches because of a higher slot inter-departure time. Therefore, to have both the desirable bandwidth overhead and lower queuing delay, one should select a smaller value for S_T from the list of answers satisfying the above equation.

2) **Class indices in OCGRR:** Choosing these indices in OCGRR is important in determining how to serve different classes. The class indices can be adjusted by the network manager or the edge switch using heuristics. From our many simulation results, we would recommend to start with K = 1.1 for the specification of indices using Eq.(3.3). After the initial index setting, the adjustment continues until the

desired QoS requirements are achieved. Using smaller class indices would make Γ (i.e. the expected logical frame length in OCGRR) smaller and K larger. This adjustment gives more chance to higher priority streams to use bandwidth than lower priority streams because the frequency of scheduling from higher priority traffic increases, but this in turn may cause bandwidth starvation for lower priority streams. Conversely, bandwidth starvation can be prevented by using larger class indices, where Γ becomes larger and K becomes smaller.

3) **Number of fibers and additional channels in contention avoidance:** Contention avoidance schemes should be also respected when designing an OPS network. We determine that choosing f=2 and f=3 fibers per connection link has a much more significant effect in reducing loss rate than using a higher number of fibers. This is also true for using additional wavelengths to reduce loss rate. That is, usually a smaller number of additional wavelengths such as W_E=1, W_E=2 will suffice.
4) **The choice of retransmission protocol:** The prioritized retransmission technique (discussed in Chapter 5) is much more effective than the conventional retransmission technique (i.e. the RR technique) whenever the loss rate in the OPS network is not low. We expect such a loss rate whenever: 1) traffic load is high; 2) a lower number of contention resolution hardware has been used in OPS core switches; 3) a lower number of contention avoidance hardware has been in the network (e.g., a single fiber network); and 4) software-based contention avoidance schemes are not respected. Therefore, we would recommend using PR in networks with a medium or higher loss rate, say more than 0.1%. In an OPS network with a low loss rate (this low loss rate network can be achieved when using so many expensive contention resolution hardware in core switches), there may be no significant difference between the performance of PR and RR, although PR is always better than RR. On the other hand, RR has a lower computational complexity than PR. Thus, we recommend RR for such a network.
5) **Frame length in DTDM:** The frame length in the DTDM protocol should also be chosen in an appropriate manner. Although choosing larger frame lengths reduces the processing work for the ingress switches, it may lead to an inaccurate bandwidth provisioning, especially when traffic arrival in ingress switches is bursty. For example,

if we choose F=500 and the interval of a time-slot is 10μsec, then the bandwidth provisioning process must be computed within 500 x 10μsec = 5msec. This time is relatively higher when compared to traffic arrivals in an ingress switch. On passing F=500 for a time-slot of 1μsec is found to be a suitable design. On the other hand, choosing smaller frame lengths will lead to frequent bandwidth provisioning and slot assignment process, which increases the processing load of an ingress switch. Moreover, it may still lead to an inaccurate bandwidth provisioning due to the lack of traffic arrival. For example, if we have F=10 and the interval of a time-slot is 10μsec, then the bandwidth must be made available at 100μsec intervals to each torrent. This time value may be relatively small to have an accurate bandwidth provisioning. We have obtained better results whenever we opt $30 \leq F \leq 100$ for a time-slot of 10μsec.

6) **The "importance" parameters:** As discussed in Chapter 6, by selecting a higher "importance" parameter for a class, a higher priority can be given to the traffic of that class to pass through the core switch, resulting in a higher throughput and lower loss for that class. The "importance" parameters can be selected based on the relative pricing for different classes. For example, if the price of EF, AF and BE classes are 25, 10, and 5 units, then the importance parameters can be chosen as V_{EF}=25/40=0.625, V_{AF}=10/40=0.25, V_{BE}=5/40=0.125 respectively.

7) **The parameters S_p and $C_{U,T}$ in the ITDM protocol:** The parameter S_p (stability period in a number of consecutive frame intervals) and $C_{U,T}$ (channel utilization threshold) in the ITDM protocol must be chosen carefully. As discussed, we shall set the parameter S_p in a way to avoid the ITDM protocol from oscillating between DTDM and CTDM. A shorter S_p may lead to this oscillation whenever traffic is bursty. On the other hand, a longer S_p may lead to a delay in switching between DTDM and CTDM. We have chosen $100{,}000 \leq S_p \leq 400{,}000$ in our work in order to avoid oscillation and provide an almost on time switching. We should also pick an appropriate value for $C_{U,T}$. Choosing a value very close to 1.0 may prevent switching from DTDM to CTDM at all. For example, consider a network with eight edge switches where we have fW=10 wavelengths. Therefore, there are 80 slots arriving at the core switch at each time-slot. Now consider the slots coming from the first seven edge switches are all fully occupied. However, only half of the slots coming from the

last edge are full. In this scenario, the average channel utilization is $C_{U,f}$ =75/80=0.937. Now, if one choose $C_{U,T}$=0.95, the system will never switch from DTDM to CTDM. However, this system must switch from DTDM to CTDM in order to obtain a better performance. On the other hand, choosing a smaller value for $C_{U,T}$, may avoid using the benefits of DTDM. We suggest to use $0.8 \leq C_{U,T} \leq 0.9$.

Chapter Nine: Conclusions

In this dissertation, we have developed a framework to manage the bandwidth in a QoS-capable slotted all-optical packet-switched network where the core switches do not use any optical buffer for contention resolution. A new ingress switch architecture and control structure have been designed. Packet-scheduling algorithm is used in each ingress switch to provide differentiation between traffic streams from different source routers. In addition, a new technique is formulated to access to the optical network where retransmission of dropped slots in OPS network is supported.

We have designed a new and more efficient packet scheduler (called the OCGRR) to provide packet differentiation among DiffServ classes in our ingress switch. In order to have a fair bandwidth allocation to streams within a class, the streams are isolated and saved in dedicated buffers. The major achievement of OCGRR is to reduce the inter-transmission time from the same-stream in order to achieve a lower jitter and startup latency.

Since cost may be a big design consideration, we have investigated contention avoidance schemes based on both hardware and software. In the inexpensive software-based schemes, the control part of a edge/core switch has the main role to reduce loss in the network. In the hardware-based schemes, the additional number of fibers and wavelength channels can provide less traffic collision. Base on these approaches, we have combined their merits to provide a cost-effective design. Of particular interest is the PR (Prioritized Retransmission) technique as an inexpensive contention resolution scheme which can limit the number of retransmissions for a slot. We have analyzed its performance, and demonstrated that it can provide a better TCP throughput, especially for the long-hop connections.

We have developed and formulated an even slot distribution method for slotted optical networks, and used it to implement our DTDM algorithm. DTDM can provide a better delay differentiation, throughput, and distribution than the conventional CTT scheme. Finally, we have compared our DTDM plus PR algorithms with a reservation-

based scheme in the AAPN network. We then designed our ITDM scheme that integrates these two protocols in order to provide a better performance for traffic classes.

9.1 Future Work

The following are some interesting aspects for further research:

1) Study end-to-end QoS in a DiffServ domain by using OCGRR at the core routers. If QoS for a class in a given network path deteriorates during congestion, the schedulers in core routers along the path may cooperate with each other for improvement by adjusting the scheduling parameters C_2 to C_ψ or by requesting arrival rates to be adjusted by the edge routers. As another research area, one can study OCGRR in OBS edge switches or EPON edge switches.
2) Study and implement a suitable scheduling algorithm in a core switch to manage the optical buffers in such a way that both contention resolution and CST implementation are achieved.
3) Investigate other criteria for the PR technique including distance, importance of traffic carried in a slot, etc. In addition, one can combine PR with QoS handling schemes proposed for OPS networks. Finally, one can study the performance of TCP using PR in an OBS network.
4) Study more accurate bandwidth-provisioning schemes to achieve a better performance for DTDM, especially in reducing queuing delay.

References

[AAPN06] Home page of AAPN, http://www.aapn.mcgill.ca/eng/index.html, accessed in 2006.

[AfCo96] Y.Afek, M.Cohen, E.Hallman, and Y.Mansour, "Dynamic bandwidth allocation policies," *Proc. IEEE Infocom'96*, San Francisco, CA, Mar.1996, pp.880–87.

[AgLe03] F.Agharebparast and V.C.M.Leung, "A new traffic rate estimation and monitoring algorithm for the QoS-enabled Internet," *Proc. IEEE Globecom 2003*, San Francisco, USA, Dec.2003.

[AnOw93] T.Anderson, S.Owicki, J.Saxe, and C.Thacker, "High speed switch scheduling for local area networks", *ACM Transactions on Computer Systems*, vol.11, no.4, Nov.1993, pp.319-352.

[BaDe04] E.Baert, C.Develder, D.Colle, F.D.Turck, M.Pickavet, and P.Demeester, "Routing strategies to minimize packet loss in an optical packet switched network with recirculating FDL buffers," *Photonic Network Comm.*, vol.7, no.3, 2004, pp.279-300.

[BaKl02] P.Barford, J.Kline, D.Plonka, and A.Ron, "A signal analysis of network traffic anomalies," *Proc. ACM Internet Measurement Workshop*, Marseilles, France, Nov.2002.

[BeFi03] M.Beshai, J.Fitchett, and A.Graves, "Structures of high-capacity reliable telecommunication networks," *Proc. DRCN 2003*, Banff, Canada, Oct.2003.

[BeGa87] D.Bertsekas and R.Gallager, *Data Networks*, Prentice Hall, 1987.

[Bert99] D.P.Bertsekas, *Nonlinear Programming*, 2nd Edition, Athena Scientific, 1999.

[BeTs01] B.Bensaou, D.H.K.Tsang, and K.T.Chan, "Credit-based fair queueing (CBFQ): a simple service-scheduilng algorithm for packet-switched networks," *IEEE/ACM Trans. on Networking*, vol.9, no.5, Oct.2001, pp.591-604.

[BeZh96] J.C.R.Bennett and H.Zhang, "WFQ: worst-case fair weighted fair queueing" *Proc. IEEE Infocom*, pp.120-128, San Francisco, CA, March 1996.

[BiGi04] A.Bianco, P.Giaccone, E.Leonardi, and F.Neri, "A framework for differential frame-based matching algorithms in input-queued switches," *Proc. IEEE Infocom 2004*, Hong Kong, 2004, pp.1147-1157.

[BiGu03] A.Bianco, M.Guido and E.Leonardi, "Incremental scheduling algorithms for WDM/TDM networks with arbitrary tuning latencies", *IEEE Journal on Selected Areas in Communications*, vol.51, no.3, March 2003, pp. 464-475.

[BjNo03] S.Bjornstad, M.Nord, et al., "Optical burst and packet switching: node and network design, contention resolution and quality of service", *Proc. IEEE International Conference on Telecommunications (ConTEL2003),* Zagreb, Croatia, Jun.2003.

[BjØv05] S.Bjørnstad and H.Øverby, "Quality of service differentiation in optical packet/burst switching: A performance and reliability perspective," *Proc.*

International Conf. on Transparent Optical Networks, Barcelona, Spain, Jul.2005.

[BlBl98] S.Blake, D.Black, M.Carlson, E.Davies, Z.Wang, and W.Weiss, "An architecture for differentiated services", *RFC 2475*, Dec.1998.

[BlLe02] F.J.Blouin, A.W.Lee, A.J.M.Lee, and M.Beshai, "Comparison of two optical-core networks," *Journal of Optical Networking*, vol.1, no.1, Jan.2002, pp.56-65.

[BrCh05] E.V.Breusegem, J.Cheyns, D.D.Winter, D.Colle, M.Pickavet, P.Demeester, and J.Moreau, "A broad view on overspill routing in optical networks: a real synthesis of packet and circuit switching?," *Elsevier Optical Switching and Networking*, vol.1, no.1, 2004, pp.51-64.

[BrCl94] R.Braden, D.Clark, and S.Shenker, "Integrated Services in the Internet architecture: an overview," *Internet informational RFC 1633,* Jun.1994.

[BrCl02] N.Brownlee and K.C.Claffy, "Understanding Internet traffic streams: dragonflies and tortoises," *IEEE Comm. Magazine*, vol.40, no.10, Oct.2002, pp.110 –117.

[CAID06] Cooperative Association of Internet Data Analysis (CAIDA), "Packet size reports," http://www.caida.org/analysis/AIX/plen_hist, accessed in March 2006.

[CaCe03] F.Callegati, W.Cerroni, C.Raffaelli, and P.Zaffoni, "DWDM for QoS management in optical packet switches," *Lecture Notes in Computer Science*, vol.2601, 2003, pp.447-459.

[CaCe04] F.Callegati, W.Cerroni, G.Corazza, C.Develder, and M.Pickavet, "Scheduling algorithms for a slotted packet switch with either fixed or variable length packets," *Photonic Network Communications*, vol.8, no.2, Sep.2004, pp.163-176.

[CaCo02] F.Callegati, G.Corazza, and C.Raffaelli, "Exploitation of DWDM for optical packet switching with quality of service guarantees", *IEEE Journal on Selected Areas in Communications*, vol.20, no.1, Jan.2002, pp.190-201.

[CaLu04] M.Casoni, E.Luppi, and M.L.Merani, "Impact of assembly algorithms on end-to-end performance in optical burst switched networks with different QoS classes", *Proc. Workshop on Optical Burst Switching*, San Josè, CA, USA, Oct.2004.

[CaPa03] D.Careglio, J.S.Pareta, and S.Spadaro, "Optical slot size dimensioning in IP/MPLS over OPS networks", *Proc. IEEE International Conference on Telecommunications (ConTEL 2003),* Zagreb, Croatia, Jun.2003.

[CaRa03] D.Careglio, A.Rafel, J.Solé-Pareta, S.Spadaro, A.M.Hill, and G.Junyent, "Quality of service strategy in an optical packet network with a multi-class frame-based scheduling", *Proc. IEEE International Workshop on High Performance Switching and Routing (HPSR 2003),* Torino, Italy, June 2003.

[CaSo05] D.Careglio, J.Solé-Pareta, and S.Spadaro, "Novel contention resolution technique for QoS support in connection-oriented optical packet switching", *Proc. IEEE ICC 2005*, Seoul, Korea, May 2005.

[ChCh99a] F.S.Choa and H.J.Chao, "All-optical packet routing-architecture and implementation," *Photonic Network Commun.*, vol.1, no.4, 1999, pp.303-311.

[ChCh99b] C.-S.Chang, W.-J.Chen, and H.Y.Huang, "On service guarantees for input

buffered crossbar switches: a capacity decomposition approach by Birkhoff and von Neumann," *Proc. IEEE/IFIP Seventh International Workshop on Quality of Service (IWQoS'99)*, London, UK, Jun.1999.

[ChCh00] C.S.Chang, W.J.Chen and H.Y.Huang, "Birkhoff-von Neumann input buffered crossbar switches," *Proc. IEEE Infocom'00,* vol.3, March 2000, pp.1624-1633.

[ChCh01] C.S.Chang, W.J.Chen, and H.Y.Huang, "Birkhoff-von Neumann input-buffered crossbar switches for guaranteed-rate services," *IEEE Transactions on Communications*, vol.49, 2001, pp.1145-1147.

[ChCo01] T.Chich, J.Cohen, and P.Fraingniaud, "Unslotted deflection routing: a practical approach and efficient protocol for multi-hop optical networks", *IEEE/ACM Transactions on Networking*", vol.9, no.1, Feb.2001, pp.47-59.

[ChEl97] I.Chlamtac, V.Elek, A.Fumagalli, and C.Szab'o, "Scalable WDM network architecture based on photonic slot routing and switched delay lines", *Proc. IEEE Infocom '97*, Kobe, Japan, Apr.1997.

[ChEl06] S.Charcranoon, T.S.El-Bawab, J.D.Shin, and H.C.Cankaya, "Group scheduling for multi-service Optical Burst Switching (OBS) networks," *Photonic Network Communications*, vol.11, no.1, Jan.2006, pp.99-110.

[ChFu94] I.Chlamtac, and A.Fumagalli, "QUADRO-Star: a high performance optical WDM star network", *IEEE Trans. on Comm.*, vol.42, no.8, Aug.1994, pp.2582-2590.

[ChFu96] I.Chlamtac, A.Fumagalli, et.al., "CORD: contention resolution by delay lines," *IEEE Journal on Selected Areas Communication*, vol.14, Jun.1996, pp.1014-1029.

[ChFu99] I.Chlamtac, A.Fumagalli, and G.Wedzinga, "Slot routing as a solution for optically transparent scalable WDM wide area networks", *Photonic Network Communications,* vol.1, no.9, 1999, pp.9-21.

[Chia04] D.Chiaroni, "Optical packet switching networks," *Proc. SPIE Network Architectures, Management, and Applications II*, vol.5626, Beijing, China, Nov.2004.

[ChLe04] C.S.Chang, D.S.Lee, and Y.J.Shih, "Mailbox switch: a scalable two-stage switch architecture for conflict resolution of ordered packets," *Proc. IEEE Infocom 2004*, Hong-Kong, Mar.2004.

[ChVl06] K.Christodoulopoulos, K.Vlachos, E.Varvarigos, L.Stampoulidis, and E.Kehayas, "Efficient burst reservation protocol: a hybrid signaling protocol for efficient burst-level reservations and quality-of-service differentiation in optical burst switching networks," *Journal of Optical Networking,* vol.5, no.3, Mar.2006, pp.147-158.

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

[DaHa98] S.L.Danielsen, P.B.Hansen, and K.E.Stubkjaer "Wavelength conversion in optical packet switching," *Journal of Lightwave Technology*, vol.16, no.12, Dec.1998, pp.2095-2108.

[Dasi00] L.A.DaSilva, "Pricing for QoS-enabled networks: a survey," *IEEE Communications Surveys & Tutorials*, 2nd Quarter, 2000, pp.2-8.

[Deve02] C.Develder, "Node Architectures for Optical Packet and Burst Switching", *Proc. COIN-PS 2002*, Cheju Island, Korea, Jul.2002.

[DhTa01] M.Dhodhi, S.Tariq and K.Saleh, "Bottlenecks in next generation DWDM-based optical networks," *Computer Comm. Journal,* vol.24, no.17, 2001, pp.1726-1733.

[DiCh99] J.Diao and P.L.Chu, "Analysis of partially shared buffering for WDM optical packet switching," *Journal of Lightwave Technology*, vol.17, no.12, Dec.1999, pp.2461-2469.

[DiDe03] L.Dittmann, C.Develder, D.Chiaroni et al., "The European IST project DAVID: a viable approach toward optical packet switching," *IEEE Journal on Selected Areas in Communications*, vol.21, no.7, Sept.2003, pp.1026–1040.

[Dolz02] K.Dolzer, "Assured Horizon - a new combined framework for burst assembly and reservation in optical burst switched networks", *Proc. European Conference on Networks and Optical Communications*, Darmstadt, Germany, June 2002.

[DoSt02] C.Dovrolis, D.Stiliadis, and P.Ramanathan, "Proportional differentiated services: delay differentiation and packet scheduling," *IEEE/ACM Trans. on Networking*, vol.10, no.1, 2002, pp.12-26.

[DuPu06] Y.Du, T.Pu, H.Zhang, and Y.Guo, "Adaptive load balancing routing algorithm for optical burst-switching networks," *Proc. IEEE/OSA OFC 2006*, Anaheim, CA, Mar.2006.

[DwSm02] A.Dwivedi, and D.F.Smith, "Strategies for optimizing optical networks," *Proc. SPIE Optical Transmission Systems and Equipment for WDM Networking*, vol.4872, Boston, USA, July 2002.

[ElCh03] H.Elbiaze, T.Chahed, T.Atmaca, and G.Hébuterne, "Shaping self-similar traffic at access of optical network," *Performance Evaluation*, vol.53, no.3-4, 2003, pp.187-208.

[ElMe03] M.Elhaddad, R.Melhem, T.Znati, and D.Basak, "Traffic shaping and scheduling for OBS-based IP/WDM backbones," *Proc. SPIE Optical Networking and Communication Conference 2003*, Dallas, TX, vol.5285, Oct.2003, pp.336-345.

[ElSh02] T.S.El-Bawab and J.Shin, "Optical packet switching in core networks: between vision and reality," *IEEE Comm. Magazine*, vol. 40, no. 9, Sep.2002, pp.60-65.

[ErLi03a] V.Eramo, M.Listanti, and M.Tarola, "Advantages of input wavelength conversion in optical packet switches," *Proc.IEEE Globecom2003*, San Francisco, CA, Dec.2003.

[ErLi03b] V.Eramo, M.Listanti, and P.Pacifici "A comparison study on the number of wavelength converters needed in synchronous and asynchronous all-optical switching architectures," *J. of Lightwave Tech.*, vol.21, no.2, Feb.2003, pp.340-355.

[ErLi05] V.Eramo, M.Listanti, and A.Valletta, "Advantages of hybrid input/output wavelength conversion in optical packet switches," *Photonic Network Communications*, vol.10, no.2, Sep.2005, pp.233-252.

[EURE05] EURESCOM Organization, P700-series, www.eurescom.de/~pub-

deliverables/P700-Series/P709/D3/Vol7/p709d3vol7.pdf, accessed in Nov.2005.

[FaAd05] C.Fan, S.Adams, and M.Reisslein, "The FT–FR AWG network: a practical single-hop metro WDM network for efficient uni- and multicasting," *IEEE Journal of Lightwave Technology,* vol.23, no.3, Mar.2005, pp.937-954.

[FaHe04] J.Fang, W.He, and A.K.Somani, "Optimal Light Trail Design in WDM Optical Networks," *Proc.IEEE ICC2004*, Paris, Jun.2004. pp.1699-1703.

[FaMa04] C.Fan, M.Maier, and M.Reisslein, "The AWG||PSC network: a performance enhanced single-hop WDM network with heterogeneous protection," *IEEE/OSA Journal of Lightwave Technology*, vol.22, no.5, May 2004, pp.1242-1262.

[FaVo03] F.Farahmand, V.M.Vokkarane, and J.P.Jue, "Practical priority contention resolution for slotted optical burst switching Networks," *Proc. First International Workshop on Optical Burst Switching (WOBS 2003)*, Dallas, TX, Oct.2003.

[FaZh04] F.Farahmand, Q.Zhang, and J.P.Jue, "A feedback-based contention avoidance mechanism for optical burst switching networks," *Proc. 3rd International Workshop on Optical Burst Switching*, San Jose, CA, October 2004.

[FlPa01] S.Floyd and V.Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Transactions on Networking*, vol.9, no.4, Aug.2001, pp.392-403.

[GaCh02] B.Ganguly, and V.Chan, "A scheduled approach to optical flow switching in the ONRAMP optical access network testbed," *Proc. IEEE/OSA OFC 2002*, Anaheim, California, March 2002.

[Gaug03] C.M.Gauger, "Trends in optical burst switching," *Proc. SPIE ITCOM 2003*, Orlando, USA, Sep.2003.

[Gaug04] C.M.Gauger, "Optimized combination of converter pools and FDL buffers for contention resolution in optical burst switching," *Photonic Network Communications*, vol.8.2, Sep.2004, pp.139-148.

[GeRa03] O.Gerstel and H.Raza, "Merits of low-density WDM line systems for long-haul networks," *IEEE/OSA Journal of Lightwave Technology*, vol.21, no.11, Nov.2003, pp.2470–2475.

[Gole94] S.J.Golestani, "A self-clocked fair queueing scheme for broadband applications," *Proc. IEEE Infocom* 1994, Toronto, June 1994, pp.636-646.

[GoMa06] J.V.Gontan, P.P.Marino, J.V.Alonso, and J.G.Haro, "A feasibility study of GMPLS extensions for synchronous slotted Optical Packet Switching networks," *WGN5: Workshop in G/MPLS networks*, Girona, Spain, Mar.2006.

[Goya97] P.Goyal, H.M.Vin and H.Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switched networks," *IEEE Trans. on Networking*, vol.5, no.5, Oct.1997, pp.690-704.

[GrGo01] P.Gravey, S.Gosselin, et al., "Multiservice optical network: main concepts and first achievements of the ROM program," *Journal of Lightwave Technology*, vol.19, no.1, Jan.2001, pp.23-31.

[Guo01] C.Guo, "SRR: an O(1) time-complexity packet Scheduler for flows in multiservice packet networks," *IEEE/ACM Trans. on Networking*, vol.12, no.6, Dec.2004, pp.1144-1155.

[GuZh05] A.Gupta, L.Zhang, and S.Kalyanaram "Spot pricing framework for loss guaranteed Internet service contracts," *Proc. IEEE Consumer Comm. and Networking Conf. (CCNC)*, Les Vagas, Nevada, Jan 2005, pp.553-555.

[HaFa05] A.Habib, S.Fahmy, and B.Bhargava, "Monitoring and controlling QoS network domains," *ACM/Wiley International Journal of Network Management*, vol.15(1), Jan.2005, pp.11-29.

[Hage98] T.Hagerup, "Sorting and searching on the word RAM," *Proc. 15th Symposium on Theoretical Aspects of Computer Science (STACS 1998), Lecture Notes in Computer Science*, vol.1373, pp.366–398.

[HeBa99] J.Heinanen, F.Baker, W.Weiss, and J.Wroclawski, "Assured forwarding PHB group", *RFC 2597*, June 1999.

[HeEl04] S.Herbst, J.Elbers, C.Fuerst, H.Griesser, H.Wernz, and C.Glingener, "Benefits and challenges in transparent optical networks," *Proc. SPIE Optical Transmission Systems and Equipment for WDM Networking III,* vol.5596, Philadelphia, Pennsylvania, USA, Oct.2004.

[HsLi02] C.F.Hsu, T.L.Liu, and N.F.Huang, "Performance analysis of deflection routing in optical burst-switched networks", *Proc. IEEE Infocom 2002*, New York, June 2002.

[HuAn00] D.K.Hunter and I.Andonovic: "Approaches to optical Internet packet switching," *IEEE Communication Magazine*, vol. 38, no. 9, Sept.2000, pp.116-122.

[HuCh98] D.K.Hunter, Meow C.Chia, and Ivan Andonovic, "Buffering in optical packet switches" *Journal of Lightwave Tech.*, vol.16, no.12, Dec.1998, pp.2081-2094.

[HuDo03] G.Hu, K.Dolzer, and CMGauger: "Does burst assembly really reduce the self-similarity," *Proc. IEEE/OSA OFC 2003,* Atlanta, Mar.2003.

[HuKa03] A.Huang, O.Kabranov and D.Makrakis, "Performance analysis of a novel optical network architecture - PetaWeb," *Proc. IEEE International Conf. on Comm. Technology (ICCT)*, Beijing, China, Apr.2003.

[HuKo06] G.Hu and M.Köhn, "Evaluation of packet delay in OBS edge nodes," *Proc. on Transparent Optical Networking (ICTON)*, Nottingham, UK, Jun.2006.

[HuLi00] N.F.Huang, G.H.Liaw, and C.P.Wang, "A novel all optical transport network with time-shared wavelength channels," *IEEE Journal on Selected Areas in Communications*, vol.18, no.10, Oct.2000, pp.1863–1875.

[HuVo05] X.Huang, V.M.Vokkarane, and J.P.Jue, "Burst Cloning: a proactive scheme to reduce data loss in optical burst-switched networks," *Proc. IEEE ICC 2005,* Seoul, South Korea, May 2005.

[IRPM06] Intel 80386 Reference Programmer's Manual, 80386 Instruction Set, http://pdos.csail.mit.edu/6.828/2005/readings/i386/c17.htm, accessed in Jun.2006.

[ItTa02] Y.Ito, S.Tasaka and Y.Ishibashi, "Variably weighted round robin queueing for core IP routers," *Proc. IPCCC 2002*, Phoenix, Arizona, USA, Apr.2002, pp.159-166.

[JaNi99] V.Jacobson, K.Nichols, and K.Poduri, "An Expedited Forwarding PHB", *RFC 2598*, June 1999.

[JeAy96] G.Jeong and E.Aynoglu, "Comparison of wavelength-interchanging and

wavelength-selective cross-connects in multiwavelength all-optical networks," *Proc. IEEE Infocom 1996*, San Francisco, CA, Mar.1996, pp.156–163.

[JiAr03] Liang Ji, T.N.Arvanitis, and S.I.Woolley, "Fair weighted round robin scheduling scheme for diffserv network," *IEE Electronic Letters*, vol.39, no.3, Feb.2003, pp.333- 335.

[JiYa06] M.Jin and O.W.W.Yang, "A TDM solution for all-photonic overlaid-star networks," *Proc. Conference on Information Sciences and Systems (CISS 2006),* Princeton University, Princeton, NJ, Mar.2006.

[KaGu98] S.Kamat, R.Guerin, V.Peris, and R.Rajan, "Scalable QoS provision through buffer management," *Proc. ACM SIGCOMM'98*, Vancouver, Canada, Oct.1998.

[KaKh02] A.Kaheel, T.Khattab, A.Mohamed, and H.Alnuweiri, "Quality-of-Service mechanisms in IP-over WDM networks," *IEEE Communications Magazine*, vol.40, no.12, Dec.2002, pp.38-43.

[KaLa00] K.Kar, T.V.Lakshman, D.Stiliadis, and L.Tassiulas, "Reduced complexity input buffered switches," *Proc. Hot Interconnects VIII*, Palo Alto, Aug.2000.

[KaPa02] S.Kanhere, A.Parekh, and H.Sethu, "Fair and efficient packet scheduling using elastic round robin," *IEEE Trans. on Parallel and Distributed Systems,* vol.13, no.3, Mar.2002, pp.324-336.

[Kart03] S.V.Kartalopoulos, *DWDM: Networks, Devices, and Technology*, Wiley-IEEE Press, 2003.

[KaSe01] S.S.Kanhere and H.Sethu, "Fair, efficient and low-latency packet scheduling using nested deficit round robin," *Proc. IEEE Workshop on High Performance Switching and Routing*, May 2001.

[KaSi91] M.Katevenis, S.Sidiropoulos, and C.Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," *IEEE Journal on Selected Areas in Communications*, vol.9, no.8, Oct.1991, pp.1265–1279.

[KeKo05] I.Keslassy, M.Kodialam, T.V.Lakshma, and D.Stiliadis, "On guaranteed smooth scheduling for input-queued switches," *IEEE/ACM Trans. on Networking,* vol.13, no.6, Dec.2005, pp.1364–1375.

[KhMo02] T.Khattab, A.Mohamed, A.Kaheel, and H.Alnuweiri, "Optical packet switching with packet aggregation", *Proc. IEEE SoftCom 2002*, Venice, Italy, Oct.2002.

[KiHs06] J.Kim, Y.L.Hsueh, L.G.Kazovsky, R.Rabbat, T.Hamada, and C.F.Su, "Spatial reuse on the optical burst transport network," *Proc. IEEE/OSA Optical Fiber Communication (OFC),* Anaheim, CA, Mar.2006.

[KiMu05] S.Kim, B.Mukherjee, and M.Kang, "Integrated congestion-control mechanism in optical burst switching networks," *Proc. IEEE GLOBECOM 2005,* St. Louis, MO, USA, Nov.2005.

[Kotu04] I.Kotuliak, "Optical networks access node performance," *Proc. 46th International Symposium Electronics in Marine (ELMAR-2004)*, Zadar, Croatia, Jun.2004.

[KrPe02] G.Kramer and G.Pesavento, "Ethernet Passive Optical Network (EPON): building a next-generation optical access network", *IEEE Comm. Magazine*, vol.40, no.2, Feb.2002, pp.66-73.

[KwLe98] T.Kwon, S.Lee, and J.Rho, "Scheduling algorithm for real-time burst traffic using dynamic weighted round robin," *Proc. IEEE International Symposium on Circuits and Systems*, Monterey, USA, Jun 1998.

[LaEd03] C.Ladas, R.M.Edwards, M.Mahdavi, and G.A.Manson, "TCP retransmission prioritization for rapid recovery in slow and lossy networks," *Proc. European Personal Mobile Comm. Conf.*, Glasgow, UK, 2003.

[LeLi95] K.-C.Lee, and V.O.K.Li, "Optimization of a WDM optical packet switch with wavelength converters," *Proc. IEEE Infocom'95*, Boston, pp.423-430, Jun.1995.

[LeMi04] L. Lenzini, E. Mingozzi, and G. Stea, "Tradeoffs between low complexity, low latency and fairness with Deficit Round Robin schedulers", *IEEE/ACM Transactions on Networking,* vol.12, no.4, Aug.2004, pp.681-693.

[LeSr03] S.Lee, K.Sriram, H.Kim, and J.Song, "Contention-based limited deflection routing in OBS networks," *Proc. IEEE Globecom 2003*, San Francisco, CA, USA, Dec.2003, pp.2633-2637.

[LiAn01] D.Liu, N.Ansari, and E.hou, "A novel fairness criterion for input queued switches," *Proc. IEEE MILCOM 2001*, McLean, VA, Oct.2001.

[LiAn04] J.Liu and N. Ansari, "The impact of the burst assembly interval on the OBS ingress traffic characteristics and system performance," *Proc. IEEE International Conference on Communications (ICC 2004)*, Paris, France, Jun.2004.

[LiFa03] W.Liu, Y.Fang, and Y.Kwon, "Performance enhancement in multi-hop wireless ad hoc networks," *Proc. IEEE VTC Fall 03*, Orlando, FL, Oct.2003.

[LiMo05] Y.Liu, G.Mohan, and K.C.Chua, "A dynamic bandwidth reservation scheme for a collision-free time-slotted OBS network," *Proc. WOBS 2005,* Boston, Oct. 2005.

[LiVi05] X.Liu, A.Vinokurov, and L.G.Mason, "Performance comparison of OTDM and OBS scheduling for Agile All-Photonic Network", *Proc. IFIP 2005 Conference on Metropolitan Area Networks*, Viet Nam, Apr.2005.

[LiXi04] Y.Li, G.Xiao, and H.Ghafouri-Shiraz, " On the benefits of multifiber optical packet switch," *Microwave and Optical Technology Letter*, vol.43, no.5, Dec.2004, pp.376-378.

[LiYe06] J.Li and K.L.Yeung, "Burst coning with load balancing," *Proc. IEEE/OSA OFC 2006*, Anaheim, CA, Mar.2006.

[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*, San Francisco, CA, USA, Dec.2003.

[LuHu05a] H.Luo, G.Hu, and L.Li, "Contention resolution in optical burst switching networks: a unified study of wavelength conversion and optical buffer schemes," *Proc. SPIE Optical Transmission, Switching, and Subsystems II,* vol.5625, Beijing, Nov.2004.

[LuHu05b] Z.Lu, D.K.Hunter, and I.D.Henning, "Contention resolution scheme for slotted optical packet switched networks", *Proc. 9th conference on Optical Network Design and Modelling (ONDM)*, Milan, Italy, Feb.2005.

[LuHu05c] Z.Lu, D.K.Hunter, and I.Henning, "Edge traffic smoothing in optical packet switched networks", *Proc. London Communications Symposium (LCS 2005)*,

London, UK, Sep.2005.

[LuHu06] Z.Lu, D.K.Hunter, and I.D.Henning, "Congestion control scheme in optical packet switched networks," *Proc. IEEE/OSA OFC 2006*, Anaheim, CA, Mar.2006.

[LuZh04] J.Luo, Z.Zhang, S.Qiu, and J.Wang, "ROBS: a novel architecture of reliable optical burst switching with congestion control," *Proc. SPIE APOC 2004*, vol.5626, Beijing, China, Nov.2004.

[MaBi00] M.A.Marsan, A.Bianco, E.Leonardi, F.Neri, and A.Nucci, "Simple on-line scheduling algorithms for all-optical broadcast-and-select networks," *IEEE European Trans. Telecommunications*, vol.11, Jan.2000, pp.109–116.

[MaBo04a] A.Maach, G.Bochmann, and H.T.Mouftah, "Congestion control and contention elimination in optical burst switching," *Telecommunication Systems*, vol.27, no.2-4, Oct.2004, pp.115-131.

[MaBo04b] A.Maach, G.Bochmann, and H.Mouftah, "Contention avoidance in optical burst switching", *Proc. 3rd International Conference on Networking, ICN'04*, Guadeloupe, French Caribbean, Mar.2004.

[MaKu03] X.Ma and G.S.Kuo, "Optical switching technology comparison: optical MEMS vs. other technologies," *IEEE Communications Magazine*, vol.41, no.11, Nov.2003, pp.S16-S23.

[MaMo01] J.Mao, W.M.Moh, and B.Wei, "PQWRR scheduling algorithm in supporting of DiffServ", *Proc. ICC 2001*, Helsinki, Finland, June 2001, pp.679-684.

[MaRe02] M.Maier, M.Reisslein, and A.Wolisz, "A hybrid MAC protocol for a metro WDM network using multiple free spectral ranges of an arrayed-waveguide grating", *Computer Networks,* vol.41, no.4, Mar.2003, pp.407-433.

[MaSh00] M. MacGregor and W.Shi, "Deficits for bursty latency-critical flows: DRR++," *Proc. IEEE ICON 2000*, Singapore, Sept.2000, pp.287-293.

[MaVi05] L.Mason, A.Vinokurov, N.Zhao, and D.Plant, "Topological design and dimensioning of Agile All Photonic Networks," *Computer Networks (Elsevier),* Special issue on Optical Networking, vol.50, no.2, Feb.2006, pp.268-287.

[Mcke99] N.McKeown, "The iSLIP scheduling algorithm for input-queue switches," *IEEE Trans. Networking*, vol.7, no.2, Apr.1999, pp.188–201.

[Medh02] D.Medhi, "QoS routing computation with path caching: a framework and network performance," *IEEE Comm. Magazine*, vol.40, no.12, Dec.2002, pp.106-113.

[MiIn04] H.Miyoshi, T.Inoue, and K.Yamashita, "D-CRED: a dynamic bandwidth allocation scheme in gigabit-ethernet passive optical networks," *Proc. ISSLS2004, Edinburgh*, UK, March 2004.

[MMIS06] Motorolla MC680x0 Instruction Set, Bit Field Find First One instruction, http://oldwww.nvg.ntnu.no/amiga/MC680x0_Sections/bfffo.HTML, accessed in 2006.

[Modi99] E.Modiano, "Random algorithms for scheduling multicast traffic in WDM broadcast-and-select networks", *IEEE/ACM Transactions on Networking*, vol.7, no.3, Jun.1999, pp.425-434.

[MoNa00] E.Modiano and A.Narula-Tam, "Mechanisms for providing optical bypass in WDM-based networks," *SPIE Optical Networks Magazine*, no.1, Jan.2000,

pp.9-16.

[MoPa03] J.H.Moon, J.P.Park, and M.S.Lee, "Hybrid bandwidth allocation algorithm to support multiple services in Ethernet PON," *Proc. ICACT 2003*, Phoenix Park, Korea, Jan.2003.

[NoBj03] M.Nord, S.Bjørnstad, and C.M.Gauger, "OPS or OBS in the core network?," *Proc. IFIP Conference on Optical Network Design and Modeling, Budapest, Feb.*2003.

[Odly03] A.M.Odlyzko, "Internet traffic growth: sources and implications", *Proc. SPIE Optical Transmission Systems and Equipment for WDM Networking II,* vol.5247, Orlando, FL, Sep.2003, pp.1-15.

[OmSi01] M.O'Mahony, D.Simeonidou, D.Hunter, and A.Tzanakaki, "The application of optical packet switching in future communications networks", *IEEE Communications Magazine*, vol.39, no.3, Mar.2001, pp.128-135.

[OPNE06] Home page of http://www.opnet.com/products/modeler/home.html; accessed in 2006.

[PaFl95] V.Paxson and S.Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Trans. on Networking,* vol.3, no.3, Jun.1995, pp.226 – 244.

[PaOb02] G.I.Papadimitriou, M.S.Obaidat, and A.S.Pomportsis, "Advances in optical networking," *International Journal of Communication Systems*, vol.15, no.2-3, 2002, pp.101-113.

[PaPa04] C.Papazoglou, G.Papadimitriou, and A.Pomportsis, "Design alternatives for optical-packet-interconnection network architectures," *OSA Journal of Optical Networking,* vol.3, no.11, Nov.2004, pp.810-825.

[PaSt82] C.H.Papadimitriou and K.Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, PH. 1982.

[PaYo02] D.H.Park and M.Yoo, "A comparative study on the burst assemble schemes for optical burst switched networks," *Proc. SPIE Optical Networking II,* vol.4910, 2002.

[PeSc96] S.Pejhan, M.Schwartz, and D.Anastassiou, "Error control using retransmission schemes in multicast transport protocols for real-time media," *IEEE/ACM Trans. on Networking,* vol.4, no.3, Jun.1996, pp.413-427.

[Pisi03] D.Pisinger, "Dynamic programming on the word RAM," *Algorithmica*, vol.35(2), 2003, pp.128-145.

[Pota02] M.J.Potasek, "All-optical switching for high bandwidth optical networks", *Optical Networks Magazine*, vol.3, no.6, 2002, pp.30-43.

[PuPe05] V.S.Puttasubbappa and H.G.Perros, "Quality of service in an optical burst switching ring," *Photonic Network Communications*, vol.9, no.3, May 2005, pp.357-371.

[RaTu03] J.Ramamirtham and J.Turner, "Time sliced optical burst switching," *Proc. IEEE Infocom 2003,* San Francisco, Mar.2003, pp 2030–2038.

[RaTu04] J.Ramamirtham and J.Turner "Design of time sliced optical burst routers," *Proc. IEEE/OSA Optical Fiber Communication (OFC)*, Los Angeles, CA, Feb.2004.

[RaYa05a] A.G.P.Rahbar and O.Yang, "An integrated TDM architecture for AAPN networks," *Proc. SPIE Photonics North*, vol.5970, Toronto, Canada, Sep.2005.

[RaZa03] C.Raffaelli, and P.Zaffoni, "Packet assembly at optical packet network access and its effects on TCP performance," *Proc. IEEE High Performance Switching and Routing (HPSR.)*, Torino, Italy, June 2003.

[RaZa04] C.Raffaelli and P.Zaffoni, "Effects of slotted optical packet assembly on end-to-end performance", *Lecture Notes in Computer Science*, Springer, vol.3079, Jun.2004.

[RaZa06] C.Raffaelli, P.Zaffoni, "TCP Performance in Optical Packet-Switched Networks," *Photonic Network Communications*, vol.11, no.3, May 2006, pp.243–252.

[ReRe01] R.Rejaie and A.R.Reibman, "Design issues for layered quality-adaptive Internet video playback", *Proc. Tyrrhenian International Workshop on Digital Communications, IWDC 2001*, Sep.2001, Taormina, Italy.

[SaBr00] K.Samaras, D.C.O'Brien, and D.J.Edwards, "Analytical calculation of throughput of ALOHA based protocols in optical wireless data networks," *IEE Proc. part J-Optoelectronics*, vol.147, 2000, pp.322-328.

[SaMu98] D.Saha, S.Mukherjee, and S.K.Tripathi, "Carry-over round robin: a simple cell scheduling mechanism for ATM networks," *IEEE/ACM Transactions on Networking*, vol.6, no.6, 1998, pp.779-796.

[SaPa06] P.Sarigiannidis, G.Papadimitriou, and A.Pomportsis, "CS-POSA: a high performance scheduling algorithm for WDM star networks," *Photonic Network Communications*, vol.11, no.2, Mar.2006, pp. 211-227.

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

[ShSu98] H.Shimonishi and H.Suzuki, "Performance analysis of Weighted Round Robin cell scheduling and its improvement in ATM networks," *IEICE Transactions on Communications,* vol.E81-B, no.5, May 1998, pp.910-918.

[ShVa96] M.Shreedhar and G.Varghese, "Efficient fair queuing using Deficit Round Robin", *IEEE/ACM Trans. On Networking*, vol.4, no.3, June 1996, pp.375-385.

[ShYa04] S.Shah-Heydari and O.W.Yang, "Traffic protection in agile all-photonic networks," *Proc. SPIE, Network Architectures, Management, and Applications II (APOC04),* vol.5626, Beijing, China, Nov.2004.

[SiBa03] P.Siripongwutikorn, S.Banerjee, and D.Tipper, "A survey of adaptive bandwidth control algorithms," *IEEE Comm. Surveys*, vol.5, no.1, 2003, pp.2-14.

[SiMo04] V.Sivaraman, D.Moreland, and D.Ostry, "Ingress traffic conditioning in slotted optical packet switched networks", *Proc. Australian Telecommunication Networks and Applications Conference (ATNAC 2004),* Sydney, Dec.2004.

[SiMo05] V.Sivaraman, D.Moreland, and D.Ostry, "A novel delay-bounded traffic conditioner for optical edge switches", *Proc. IEEE Workshop on High Performance Switching and Routing (HPSR)*, Hong Kong, May 2005.

[SoMi04] A.K.Somani, M.Mina, and L.Li, "On trading wavelengths with fibers: a cost performance based study", *IEEE/ACM Transactions on Networking*, vol.12, no.5, Oct.2004, pp.944-951.

[SrSo02] M.Sridharan and A.K.Somani, "Design for upgradability in mesh-restorable optical networks," *Optical Networks Magazine*, vol.3, no.3, May/June 2002, pp.77–87.

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

[Stev94] W.R.Stevens, *TCP/IP Illustrated Volume I: The Protocols*, Addison-Wesley 1994.

[Stev01] W.R.Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," *RFC 2001, IETF*, Jan. 1997.

[StMa02] A.Striegel and G.Manimaran, "Packet scheduling with delay and loss differentiation," *Computer Communications*, vol.25(1), Jan.2002, pp.21-31.

[SuLa99] B.Suter, T.V.Lakshman, D.Stiliadis, and A.K.Choudhury, "Buffer management schemes for supporting TCP in gigabit routers with per-flow queueing," *IEEE Journal on Selected Areas in Comm.*, vol. 17, no.6, June 1999, pp.1159-1169.

[TaLu04] W.Tan, Y.Luo, S.Wang, D.Xu, Y.Pan, and L.Li, "A new scheduling algorithm to provide proportional QoS in optical burst switching networks," *Proc. SPIE Optical Transmission, Switching, and Subsystems II,* vol.5625, Beijing, China, Nov.2004.

[Tane02] A.S.Tanenbaum, *Computer Networks*, Prentice Hall, 4th edition, 2002.

[TaYe00] L.Tancevski, S.Yegnanarayanan, G.A.Castañon, L.Tamil, F.Masetti, and T.McDermont, "Optical routing of asynchronous, variable length packets", *IEEE Journal of Select Areas on Comm.*, vol.18, Oct.2000, pp.2084-2093.

[ThMi97] K.Thompson, G.J.Miller, and R.Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network*, vol.11, no.6, 1997, pp.10-23.

[ThVo03] G.P.V.Thodime, V.M.Vokkarane, and J.P.Jue, "Dynamic congestion-based load balanced routing in optical burst-switched networks," *Proc. Globecom 2003*, San Francisco, California, Dec.2003.

[TsLi01] S.Tsao and Y.Lin, "Preorder deficit round robin; a new scheduling algorithm for packet switched networks," *Computer Networks*, vol.35, no.2-3, Feb.2001, pp.287-305.

[TuZh99] R.S.Tucker and W.D.Zhong, "Photonic packet switching: An overview," *IEICE Trans. Electronics*, vol.E82-C, Feb.1999, pp.202–212.

[UnØv04] A.Undheim, H.Øverby, and N.Stol, "*Absolute QoS in synchronous optical packet switched networks," Proc. Norsk Informatikk Konferanse*, Stavanger, Norway, Nov.2004.

[VeCh00] S.Verma, H.Chaskar, and R.Ravikanth, "Optical burst switching: a viable solution for terabit IP backbone," *IEEE Network*, vol.14, no.6, Nov./Dec.2000, pp.48-53.

[VoJu03] V.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 Comm.*, vol.21, no.7, Sep.2003, pp.1198-1209.

[VoHa02] V.M.Vokkarane, K.Haridoss, and J.P.Jue, "Threshold based burst assembly policies for QOS support in optical burst switched networks," *Proc. SPIE OptiComm* 2002, Boston, vol.4874, July 2002, pp.125-136.

[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, Taipei, Taiwan, Nov.2002.

[Wagn06] Prof. Kelvin Wagner home page, Low-loss Acousto-optic Photonic Switch, http://drip.colorado.edu/~kelvin/ao_switch.html, accessed in Jun.2006.

[WaLi94] Y.T.Wang, T.P.Lin, and K.C.Gan, "An improved scheduling algorithm for weighted round-robin cell multiplexing in an ATM switch," *Proc. IEEE SUPERCOMM/ICC '94*, New Orleans, LA, USA, May 1994.

[WaMa03] L.Wang, M.Ma, and M.Hamdi, "Efficient protocols for multimedia streams on WDMA networks", *IEEE/OSA Journal of Lightwave Technology*, vol.21, no.10, Oct.2003, pp.2123–2144.

[WaMo00] X.Wang, H.Morikawa, and T.Aoyama, "Deflection routing protocol for burst switching WDM mesh networks", *Proc. SPIE/IEEE Terabit Optical Networking: Architecture, Control, and Management Issues*, Boston, USA, Nov.2000.

[WaMo02] X.Wang, H.Morikawa, and T.Aoyama, "On wavelength assignment and deflection routing for contention resolution in burst switched photonic networks," *Proc. Optoelectronics and Communications Conference,* Yokohama, Japan, Jul.2002.

[WaSh01] H.Wang, C.Shen, and K.Shin, "Adaptive-weighted packet scheduling for premium service", *Proc. IEEE ICC 2001*, Helsinki, Finland, June 2001.

[WeSi02] B.Wen and K.M.Sivalingam, "Routing, wavelength and time-slot assignment in time division multiplexed wavelength-routed optical WDM networks," *Proc. IEEE Infocom,* New York, NY, USA, June 2002.

[WiSa04] I.Widjaja and I.Saniee," Simplified layering and flexible bandwidth with TWIN", *Proc. SIGCOMM04 Workshop*, Portland, USA, Aug 2004.

[WoHa03] W.K.Wong, Z.Haiying, and V.C.M.Leung, "Soft QoS provisioning using the token bank fair queuing scheduling algorithm," *IEEE Wireless Communications,* vol.10, no.3, Jun 2003, pp.8-16.

[WoTa04] W.K.Wong, H.Tang, and V.C.M.Leung, "Token bank fair queuing: a new scheduling algorithm for wireless multimedia services", *Int'l Journal of Communication Systems*, vol.17, Aug.2004, pp.591-614.

[XiNi99] X.Xiao and L.M.Ni, "Internet QoS: a big picture", *IEEE Network,* vol.13, no.2, 1999, pp.8-18.

[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, Oct.2000, pp.1838-1851.

[XuPa03] F.Xue, Z.Pan, Y.Bansal, and J.Cao, "End-to-end contention resolution schemes for an optical packet switching network with enhanced edge routers," *J. of Lightwave Tech.,* vol.21, no.11, Nov.2003, pp.2595-2604.

[XuPa04] F.Xue, Z.Pan, H.Yang, J.Yang, J.Cao, K.Okamoto, S.Kamei, V.Akella, and S.J.B.Yoo, "Design and experimental demonstration of a variable length optical packet routing system with unified contention resolution," *IEEE/OSA Journal of Lightwave Tehcnology*, vol.22, no.11, Nov.2004, pp.2570-2581.

[XuYa02] F.Xue, S.Yao, B.Mukherjee, and S.J.B.Yoo, "The performance improvement in optical packet-switched networks by traffic shaping of self-similar traffic,"

Proc. IEEE/OSA OFC 2002, Anaheim, CA, Mar.2002.

[XuYo02] F.Xue, and S.J.B.Yoo, "Self-similar traffic shaping at the edge router in optical packet-switched networks", *Proc. ICC 2002 - IEEE International Conference on Communications*, vol.25, no.1, Apr.2002, pp.2449-2453.

[YaLi05] J.Yang, J.Li, Q.Zeng, G.Zhu, and T.Ye, "Influence of packet scheduling algorithms on optical packet switch," *Proc. SPIE Optical Transmission, Switching, and Subsystems II*, vol.5625, Beijing, China, Nov.2004.

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

[YaXu02] S.Yao, F.Xue, B.Mukherjee, S.J.B.Yoo, and S.Dixit, "Electrical ingress buffering and traffic aggregation for optical packet switching and their effect on TCP-level performance in optical mesh networks," *IEEE Comm. Magazine,* vol.40, no.9, Sep.2002, pp.66-72.

[YaYj03] L.Yang, Y.Jiang, and S.Jiang, "A probabilistic preemptive scheme for providing service differentiation in OBS networks," *Proc. IEEE Globecom 2003*, San Francisco, USA, Dec.2003.

[YaYo00] S.Yao, S.J.B.Yoo, B.Mukherjee, and S.Dixit, "All-optical packet-switched networks: a study of contention-resolution schemes in an irregualr mesh network with variable-sized packets," *Proc. SPIE* vol.4233, *OPTICOMM 2000*, Plano, TX., Oct.2000.

[YaYo01a] S.Yao, S.Yoo, and B.Mukherjee, "A comparison study between slotted and unslotted all-optical packet-switched network with priority-based routing," *Proc. IEEE OSA Optical Fiber Conference,* 2001.

[YaYo01b] S.Yao, S.J.B.Yoo, and B.Mukherjee, "All-optical packet switching for metropolitan area networks: opportunities and challenges," *IEEE Comm. Magazine*, Mar.2001.

[YaZe05] J.Yang, Q.J.Zeng, J.Li, and G.Zhu, "Performance improvement for optical packet switch with shared buffers," *Proc. SPIE Optical Transmission, Switching, and Subsystems II*, vol.5625, Beijing, China, Nov.2004.

[YaZe06] J.Yang, Q.Zeng, T.Ye, and G.Zhu, "A totally shared optical packet switch: dimensioning, control algorithm and performance," *Photonic Network Communications,* vol.11, no.2, Mar.2006, pp.173–185.

[YoQi99] M.Yoo and C.Qiao, "Optical Burst Switching (OBS): a new paradigm for an optical Internet", *J. of High Speed Networks*, vol.8, no.1, 1999, pp.69- 84.

[YoQi00a] M.Yoo, C.Qiao and S.Dixit, "A comparative study of contention resolution policies in optical burst switched WDM networks", *Proc. Conf. on Terabit Optical Networking: Architecture, Control and Management Issues*, Boston, MA, Nov.2000, SPIE vol.4213, pp.124-135.

[YoQi00b] M.Yoo, C.Qiao, and S.Dixit, "QoS performance in IP over WDM networks", *IEEE J. Selected Areas in Communications (JSAC), Special Issue on the Protocols for Next Generation Optical Internet*, vol.18, no.10, Oct.2000, pp.2062-2071.

[YoXu03] S.J.B.Yoo, F.Xue, Y.Bansal, et al., "High-performance optical-label switching packet routers and smart edge routers for the next generation Internet," *IEEE Journal on Selected Areas in Comm.*, vol.21, no.7, Sep.2003, pp.1041-1051.

[YuAn05] L.Yuanqiu and N.Ansari, "Bandwidth allocation multiservice Access on EPONs," *IEEE Comm. Magazine*, vol.43, no.2, Feb.2005, pp.s16-s21.

[YuQi04] X.Yu, C.Qiao, and Y.Liu, "TCP implementation and false time out detection in OBS networks," *Proc. IEEE Infocom,* Hong Kong, Mar.2004.

[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*, vol.1, no.1, Jan.2000, pp.47-60.

[ZhLu03] Z.Zhang, J.Luo, Q.Zeng, and Y.Zhou, " Novel threshold-based burst assembly scheme for QoS support in optical burst switched WDM networks," *Proc. SPIE*, vol.5244, Orlando, FL, USA, Sep.2003.

[ZhMa02] C.Zhang and M.MacGregor, "Scheduling latency-critical traffic: a measurement study of DRR+ and DRR++", *Proc. IEEE High Performance Switching and Routing (HPSR 2002)*, Kobe, Japan, June 2002.

[ZhVo03] Q.Zhang, V.Vokkarane, B.Chen, and J.Jue, "Early drop and wavelength grouping schemes for providing absolute QoS differentiation in optical burst-switched networks," *Proc. IEEE GLOBECOM 2003*, San Francisco, CA, Dec.2003.

[ZhVo05] Q.Zhang, V.M.Vokkarane, Y.Wang, and J.P.Jue, "Evaluation of burst retransmission in optical burst-switched networks," *Proc. IEEE Broadnets 2005,* Boston, USA, Oct.2005.

[ZhYa03] P.Zhou and O.Yang, "How practical is optical packet switching in core networks?" *Proc. IEEE GLOBECOM 2003,* vol.22, no.1, San Francisco, CA, USA, Dec.2003.

[ZhYa99] P.Zhou and O.Yang, "A feasible scheduling algorithm for per-VC queuing ATM switches," *Proc. ICATM99*, Colmar, France, Jun. 1999.

[ZhYo05] L.Zhaobiao, P.Yong, Z.Min, and Y.Peida, "Comparison study of length-threshold, timer-based and hybrid algorithm for optical packet assembly," *Proc. SPIE Optical Transmission, Switching, and Subsystems II*, vol.5626, 2005, pp.428-435.

[Øver03] H.Øverby, "An adaptive service differentiation algorithm for optical packet switched networks," *Proc. International Conference on Transparent Optical Networks (ICTON)*, Warsaw, Poland, Jul.2003.

[Øver04] H.Øverby, "Network layer packet redundancy in optical packet switched networks," *OSA Optics Express*, vol.12, no.20, October 2004, pp.4881-4895.

[Øver05] H.Øverby, "Packet loss rate differentiation in slotted optical packet switched networks," *IEEE Photonics Tech. Letters*, vol.17, no.11, Nov.2005, pp.2469-2471.

[ØvSt04] H.Øverby and N.Stol, "Quality of service in asynchronous buffer-less optical packet switched networks," *Kluwer Telecommunication Systems*, vol.27, no.2-4, Oct.-Dec.2004, pp.151-179.

[ØvSt06] H.Øverby, N.Stol, "QoS differentiation in asynchronous buffer-less optical packet switched networks," *Wireless Networks*, vol.12, no.3, Jun.2006, pp.383-394.

Appendix A: The CTT Scheme in *Arch1*

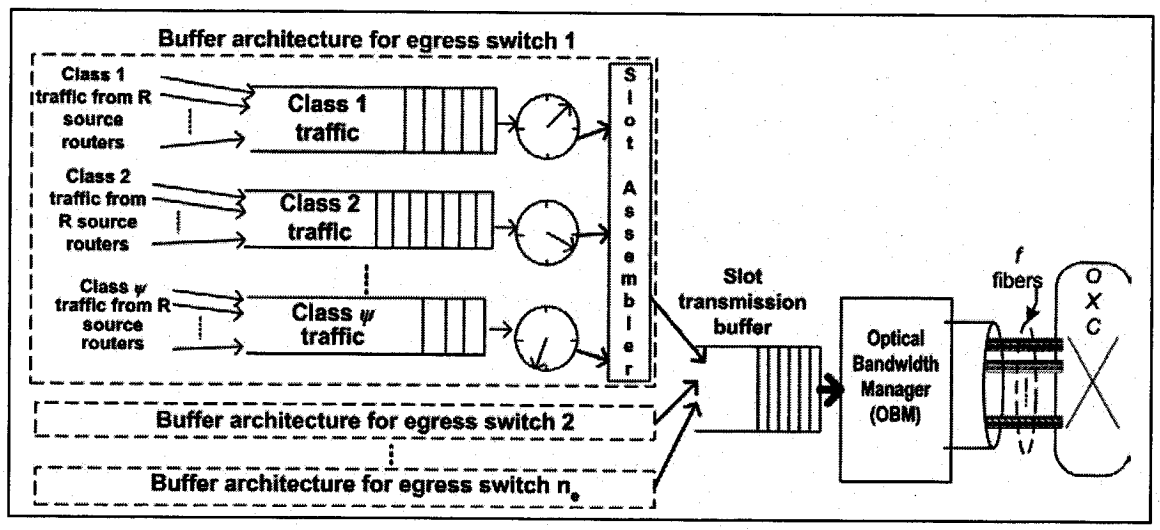


Fig.A.1: Ingress Switch Architectures (*Arch1*)

Fig.A.1 depicts the common CTT scheme, e.g., [RaZa03] in *Arch1* used in our comparison study in Chapter 6. In this architecture, all same-class traffic from different sources going to the same egress switch is saved in one big buffer. In this architecture, ψ buffers are maintained for each egress switch if there are ψ classes in network. Therefore, for n_e egress switches in the network, each ingress switch maintains n_e ψ buffers.

Let the average traffic arrival rate to class i of a given egress switch be λ_i packets/sec. We assign the time-out T_i to the class i. A slot from class i traffic can be generated whenever either the time-out event happens or the available traffic is enough to make a slot. Under the Poisson traffic arrival, the former happens when $N_1=\lambda_i T_i+1$ packets arrive for class i, while the latter happens when there are $N_2=\left\lfloor \frac{B_c S_T}{\ell} \right\rfloor$ packets in class i buffer [KhMo02]. Therefore, the slot generation rate from class i, will be $\lambda_{s,i}=\frac{\lambda_i}{min(N_1,N_2)}$. Since an ingress switch can serve up to fW slots (the slots can be destined to any egress switch) per S_T+S_O units of time, the average slot service rate for each egress switch is given by $\mu_s=\frac{fW}{(S_T+S_O)n_e}$. We also have $\rho=\sum_{i=1}^{\psi}\lambda_{s,i}/\mu_s$ on average for each egress switch.

Appendix B: Proof of Theorems and Lemmas in Chapter 3

Appendix B.1: Proof of Lemma 3.1

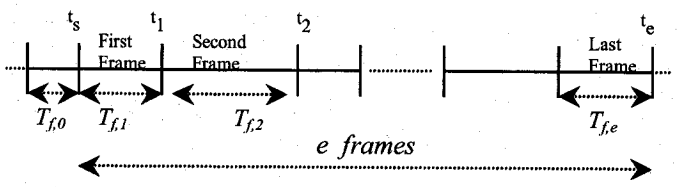


Fig.B.1: The Diagram used in Lemma 3.1.

Proof: Let the available grant and used-grant at time t_s be $G(t_s)$ and $U(t_s)$ respectively, and there are e frames starting from t_s to t_e (Fig.B.1). Also let q_s be the acquired quantum at the first frame

$$q_s = K\lambda(t_s + T_f) - (G(t_s) + U(t_s)) \quad . \tag{B.1}$$

Therefore, the grant and the used-grant are updated to $G'=G(t_s)+q_s$ and $U'=U(t_s)$ respectively. Now let the stream transmit M_1 bits, and based on this transmission, the available grant is reduced to $G=G'-M_1=G(t_s)+q_s-M_1$ and the used-grant is increased to $U=U'+M_1=U(t_s)+M_1$. This grant pair will be used at the beginning of the second frame.

Let the second frame start at time $t_1=t_S+T_{f,1}$ where $T_{f,1}$ is the actual duration of the frame starting at t_s and ending at t_1 that may be different from T_f. At this time, $G(t_1)=G$ and $U(t_1)=U$ denote the available grant and the amount of the used-grant until time t_1 respectively for the stream. By considering Eqs. (3.6) and (B.1), the acquired quantum at the beginning of the second frame, is

$$q_1 = K\lambda(t_1+T_f) - (G(t_1)+U(t_1)) = K\lambda(t_s+T_{f,1}+T_f) - (G(t_s)+q_s - M_1 + U(t_s)+M_1) = K\lambda(t_s+T_{f,1}+T_f)$$
$$-(U(t_s)+G(t_s)+q_s) = K\lambda(t_s+T_{f,1}+T_f) - K\lambda(t_s+T_f) = K\lambda T_{f,1}.$$

In all subsequent frames, the situation is similar to the calculations we made for the second frame so that the gained quantum at the beginning of frame i equals to $q_{i-1} = K\lambda T_{f,i-1}$, where $T_{f,i-1}$ is the actual frame period for frame $i-1$. Hence, the total acquired quantum, Q_T, between $[t_s, t_e]$ equals to $Q_T = q_s + \sum_{i=1}^{e-1} K\lambda T_{f,i} = q_s + K\lambda(t_e - t_s - T_{f,e})$,

where the first term denotes the acquired quantum at time t_s, the summation denotes the acquired quantum up to the beginning of frame e, and $T_{f,e}$ denotes the real period of the last frame before t_e. The quantum acquired at time t_s can be rewritten as $q_s = K\lambda T_{f,0}$, since we derived this equation for any frame, where $T_{f,0}$ is the frame before time t_s. Thus, $Q_T = K\lambda(t_e - t_s) + K\lambda(T_{f,0} - T_{f,e})$. Since $T_{f,0}$ and $T_{f,e}$ are the actual frame periods and $\Delta T_f = T_{f,0} - T_{f,e} \in [-T_{f,max}, +T_{f,max}]$, we have $Q_T \in [K\lambda(t_e - t_s) - K\lambda T_{f,max}, K\lambda(t_e - t_s) + K\lambda T_{f,max}]$.

Appendix B.2: Proof of Lemma 3.2

Proof: In the worst case, let the total transmitted bits in a frame be Γ-1. Since the number of the transmitted bits is still less then Γ, another stream can transmit one packet. In the worst case, this stream will transmit the largest packet. Therefore, $\Gamma_{max} = \Gamma + L_{max} - 1$ and the longest frame period is $T_{f,max} = \frac{\Gamma_{max}}{C} = \frac{\Gamma + L_{max} - 1}{C}$. Using Eq.(3.5), considering $L_{max} = 30\ell$ and assuming $\sum_{j=1}^{\Psi} C_j \le 3$,

$$T_{f,max} \le \frac{3(R+1)\frac{L_{max}}{30} + L_{max}}{C} = \frac{L_{max}}{K}\frac{K}{C}(1 + 0.1(R+1)) \quad . \tag{B.2}$$

Considering Eq.(3.3) and $\sum_{j=1}^{\Psi} C_j \ge 1 + \frac{1}{R}$, we have $\frac{K}{C} = \frac{1}{\sum_{j=1}^{\Psi} C_j \sum_{i=1}^{R} \lambda_{j,i}} \le \frac{1}{\sum_{j=1}^{\Psi} C_i \lambda_{min} R} \le \frac{1}{(R+1)\lambda_{min}}$. Thus, we obtain $T_{f,max} \le \frac{L_{max}}{K\lambda_{min}}(0.1 + \frac{1}{R+1})$.

Appendix B.3: Proof of Lemma 3.3

Lemma 3.3: The minimum grant, G_{min}, at the end of a logical frame is 1-L_{max}.

Proof: When a stream has a positive grant of even one bit, it can transmit a packet even with the maximum length L_{max}. Now the length of the transmitted packet is added to the used-grant and reduced from the available grant. Therefore, the smallest grant becomes 1-L_{max}, which is a negative grant. When the grant is negative, the stream cannot transmit any packet.

Appendix B.4: Proof of Lemma 3.4

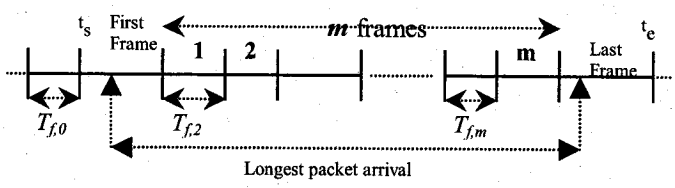


Fig.B.2: Maximum Grant Accumulation

Proof: By considering q_s from the first lemma, the available grant at time t_s becomes $G=(K\lambda t_s - U(t_s))+K\lambda T_f$. To calculate the maximum grant accumulation, we consider the worst case in which the stream has no packet, and a packet with size L_{max} starts arriving at the stream during the first frame showed in Fig.B.2. The longest packet may take m intermediate logical frames to arrive, especially when the frame length is small due to having a few streams. However, for a large number of streams, the frame length may be higher and m=0. Let $T_{f,i}$ denote the actual frame length of the logical frame that starts at time t_{i-1} and finishes at time t_i. Based on the quantum calculation we showed in Lemma 3.1, at the last frame the grant is adjusted to $G=(K\lambda t_s - U(t_s)+K\lambda T_f)+\sum_{i=1}^{m+1} K\lambda T_{f,i}$, where the first term denotes the available grant at t_s and the second term shows acquired quantum during m intermediate frames. Assuming $(K\lambda t_s - U(t_s)+K\lambda T_f)+K\lambda T_{f,1} \le 2K\lambda T_{f,max}$, we have $G \le 2K\lambda T_{f,max} + K\lambda\sum_{i=2}^{m+1} T_{f,i}$. Since the grant received in the intermediate frames is less than L_{max}, we have $G \le 2K\lambda T_{f,max} + K\lambda\frac{L_{max}}{C_{in}}$, where C_{in} is the line speed of the router input port that carries the stream. By assuming C_{in}=C, the maximum accumulated grant, G_{max}, at time t_e becomes $G_{max}=\frac{K}{C}\lambda L_{max} + 2K\lambda T_{f,max}$. For a large number of streams we have $G_{max}=2K\lambda T_{f,max}$ because the frame lengths are longer and there is no intermediate frame *(m=0).*

Appendix B.5: Proof of Theorem 3.1

Theorem 3.1: The OCGRR scheduling algorithm is fair.

Proof: We use the fairness index I_F [Gole94] to obtain the OCGRR fairness between streams i and j:

$$I_F = \left| \frac{W_i(\tau_1,\tau_2)}{R_i} - \frac{W_j(\tau_1,\tau_2)}{R_j} \right| \le K' \qquad \text{(B.3)}$$

where $W_i(\tau_1, \tau_2)$ denotes the bit service received by stream i during period (τ_1, τ_2), R_i is the relative share of the bandwidth allocated to stream i, i.e., $R_i=\lambda_i/\lambda_{min}$ by considering Eq.(3.2), and K' is the bound for the fairness index. A scheduler is perfectly fair if the difference in normalized service offered to any two backlogged streams i and j in the period (τ_1, τ_2) is 0, i.e., $K'=0$, in Eq.(B.3). Let τ_1 and τ_2 denote the beginning of two frames. We assume streams i and j are within the same class and they are always non-empty in (τ_1, τ_2). OCGRR has the worst fairness index when $K \le 1$ where the grant of streams may become negative. Two cases may happen for the streams:

The Best Case: Here, stream i has started the period with the maximum grant of $G_i(\tau_1)=G_{i,max}$ at time τ_1 and has finished the period with the smallest grant, $G_i(\tau_2)=G_{i,min}$, at time τ_2. The stream has gained the quantum of $\lambda_i K(\tau_2 - \tau_1 + \Delta T_f)$ during this period (Lemma 3.1). Since stream i is always non-empty, the total grant can be used. Therefore, the maximum service received by stream i is $W_{i,max}(\tau_1, \tau_2) = \lambda_i K(\tau_2-\tau_1+\Delta T_f)+G_{i,max} - G_{i,min}$.

The Worst Case: In this case, stream j enters the period with the minimum grant of $G_j(\tau_1)=G_{j,min}$ at time τ_1. In the worst case, the stream has a packet with size L_{max} to send at the last frame, but the frame finishes before sending the packet. Moreover, the stream has obtained some quantum of $\lambda_j K(\tau_2 - \tau_1 + \Delta T_f)$ during this period. The minimum service received by stream j is $W_{j,min}(\tau_1, \tau_2) = \lambda_j K(\tau_2 - \tau_1 + \Delta T_f) + G_{j,min} + G_{j,min}$.

By referring to Eq.(B.3), the fairness index I_F becomes

$$I_F \le \left| \frac{W_{i,max}(\tau_1,\tau_2)}{R_i} - \frac{W_{j,min}(\tau_1,\tau_2)}{R_j} \right| \le \left| \frac{K\lambda_i(\tau_2-\tau_1+\Delta T_f)+G_{i,max}-G_{i,min}}{R_i} - \frac{K\lambda_j(\tau_2-\tau_1+\Delta T_f)+2G_{j,min}}{R_j} \right| \le \left| \frac{G_{i,max}-G_{i,min}}{R_i} - \frac{2G_{j,min}}{R_j} \right|.$$

By using Lemmas 3.3 and 3.4, the short-term fairness index during period (τ_1, τ_2) becomes

$$I_{F(OCGRR)} \le 2kT_{f,max} + \frac{K}{C}L_{max} + \frac{L_{max}}{R_i} + 2\frac{L_{max}}{R_j} \le (0.2 + \frac{3}{R+1})L_{max} + \frac{L_{max}}{R_i} + 2\frac{L_{max}}{R_j} \quad .$$

For a large number of streams, term $\frac{K}{C}L_{max}$ is not counted (Lemma 3.4) and the bound becomes smaller. The short-term fairness index of OCGRR is comparable with the

fairness index of DRR, $I_{F(DRR)} \le L_{max} + \frac{L_{max}}{R_i} + \frac{L_{max}}{R_j}$ [ShVa96]. The long-term fairness index is obtained when $\tau_1=0$ and $\tau = \tau_2 \to \infty$. Since the initial grant pairs are $G_{i,max}=0$ and $G_{j,min}=0$ at $\tau_1=0$, thus, the long-term fairness index is $I_{F(\tau\to\infty)} \le \left| \frac{K\lambda_i \tau + L_{max}}{R_i} - \frac{K\lambda_j \tau - L_{max}}{R_j} \right| \le \frac{L_{max}}{R_i} + \frac{L_{max}}{R_j}$. This is a better fairness index bound comparing to DRR.

Appendix B.6: Proof of Theorem 3.2

Proof: We provide the latency bound for a single-class system to compare with DRR without considering the shared buffer. The latency bound is the maximum time it takes for the first packet of a newly backlogged stream (called "tagged" packet) to be served completely [KaPa02]. In OCGRR, the latency bound $\hat{L}_i$ for stream i can be written as $\hat{L}_i = T_G + \sum_{j=1}^{r} P_j / C + P_i / C$, where T_G is the time it takes for the stream to obtain positive grant if its current grant is negative; r is the number of backlogged streams before stream i in the *ActiveList*; P_j is the transmitted packet length in bits from stream j; and the last term denotes the tagged packet transmission time. Note that when $K>1$, the streams have always enough grant and therefore we have $T_G=0$.

To obtain $\hat{L}_i$, we consider the worst case. Let t_0 be the frame beginning time in which the tagged stream, with one packet of size L_{max} and $G=1$, is at the head of the *ActiveList*. After scheduling the last packet at time $t_0 + L_{max}/C$, the tagged stream is removed from the *ActiveList* with $G=1-L_{max}$. At the same time, the tagged packet arrives at a new busy period. However, the transmission of the tagged packet depends on the available grant at the next frame. Since the grant of the tagged stream has already been reduced to $G=1-L_{max}$, the stream must obtain the total grant of at least L_{max} to be able to transmit the tagged packet. This may take several frames.

Let $T_{G,max}$ be the maximum waiting time to obtain a positive grant when $K \le 1$. In each subsequent frame, the tagged stream can acquire some quantum proportional to the previous actual frame length (see Lemma 3.1). Hence, by assuming the longest frame lengths, the gained quantum at each new frame equals to $q = K\lambda T_{f,max}$. Let n frames be

required to obtain the total quantum of L_{max}. To be able to transmit the tagged packet, the total accumulated grant of stream i at the n-th frame must satisfy $(1-L_{max})+nK\lambda T_{f,max} \geq 1$. Hence, we have $n \geq \frac{L_{max}}{K\lambda T_{f,max}} \Rightarrow n_{min} = \left\lceil \frac{L_{max}}{K\lambda T_{f,max}} \right\rceil$. Considering the arrival time of the tagged packet at time $t_0 + \frac{L_{max}}{C}$ and $X \leq \lceil X \rceil \leq X+1$, we have $T_{G,max} = n_{min}T_{f,max} - \frac{L_{max}}{C} = \left\lceil \frac{L_{max}}{K\lambda T_{f,max}} \right\rceil T_{f,max} - \frac{L_{max}}{C} \leq \frac{L_{max}}{K\lambda} + T_{f,max} - \frac{L_{max}}{C}$. Now the stream becomes backlogged and in the worst case, it becomes the last stream in the *ActiveList*. Moreover, all previous streams (at most $r=R-1$ streams) are planning to transmit the packets with size L_{max}. Thus, we have $\hat{L}_i \leq T_{G,max} + (R-1)L_{max}/C + P_i/C$. By considering Eq.(B.2) with R streams and $\sum_{j=1}^{\Psi} C_j = 1$, we have $\hat{L}_{OCGRR} \leq \frac{L_{max}}{C}\left(\sum_{j=1}^{R} \frac{\lambda_j}{\lambda_{min}} + \frac{31R+1}{30}\right)$. To compare with DRR latency bound [KaPa02], we consider the same rate for all streams resulting in $\hat{L}_{OCGRR} \leq \frac{L_{max}}{C}\frac{61R+1}{30}$. Comparing to $\hat{L}_{DRR} \leq 2\frac{L_{max}}{C}R + \frac{L_{max}-R}{C}$, the OCGRR bound is better when $R < \frac{29L_{max}}{L_{max}+30} \approx 29$.

Under a multi-class traffic, by reducing C_2 to C_ψ, the frame length is reduced and K is increased, and the frequency of scheduling from the higher priority traffic will be increased. This leads to a lower latency bound for higher priority classes than lower priority.

Appendix B.7: Proof of Theorem 3.3

Theorem 3.3: The per-packet work complexity[12] of OCGRR is *O(1)*.

Proof: When a packet for stream i in class j arrives at the system, it is inserted into buffer i inside class j buffers with an *O(1)* operation. Then, if the stream is not in *ActiveList j*, but has enough grants, it is appended to *ActiveList j*. The *append* process can be implemented in *O(1)* operation. The packets are also selected with a complexity of *O(1)* inside class j because *ActiveList j* points to the backlogged streams within that class. After finishing each round inside a frame, the new round is started from a location referenced by the header of *ActiveList j*. Similarly, new frames start from the location

referenced in the header of *ActiveList j* or just the last unprocessed stream within class *j*. Therefore, to de-queue the packets, a constant work of single traversing of *ActiveList j* is required. Since the per-packet work complexity for en-queuing and de-queuing the packet is *O(1),* the work per-packet complexity of OCGRR is *O(1).*

[12] The *work* is defined as the maximum of the time complexities to en-queue or de-queue a packet [ShVa96].

Appendix C: The DRR, DRR+, DRR++ and PQWRR Algorithms

We shall briefly outline DRR, DRR+, DRR++ and PQWRR algorithms that are used in comparison with our OCGRR algorithm in Chapter 3.

C.1. The DRR Algorithm [ShVa96]

DRR adds a quantum defined by $Q_i = \lambda_i L_{max} / \lambda_{min}$, where λ_{min} is the smallest AAR among all streams, to the credit of active stream i (a stream with at least one packet) at each round. Then it visits the active streams in a round robin. In each visit, an active stream transmits a number of packets as far as its credit is greater than the packet size of the head of the stream. The remainder from the previous credit is added to the quantum for the next round. When a stream becomes empty, its grant is reset to zero.

C.2. The DRR+ Algorithm [ShVa96]

There are two higher priority (we denote it by EF) and lower priority (we denote it by BE) classes in DRR+. Each EF stream guarantees to send at most one quantum worth traffic at every T seconds. When an EF stream has more traffic than its contract (one quantum) within T, it will be treated as a BE stream. Otherwise, it is treated as an EF stream. The period T is considered to service one quantum from each one of the EF streams as well as one quantum's worth for every other BE stream. When a packet arrives in an empty EF stream, DRR+ adds the stream to its *ActiveList,* but only before BE streams. When a stream becomes empty, its grant is reset to zero.

C.3. The DRR++ Algorithm [MaSh00]

Similar to DRR+, there are two higher priority and lower priority classes in DRR++. In DRR++, the contract for an EF stream is an agreement to send less than one quantum during a scheduling round. Unlike DRR+, if a stream has more traffic than its contract, it is simply remained backlogged and still will be treated as an EF stream. In DRR++, a Priority Transmission Queue (*PTQ*) is used to save the traffic of all higher priority

streams. Moreover, traffic in all BE streams are scheduled using the DRR algorithm in a Transmission Queue (*TQ*). In each transmission process, *PTQ* is first checked from the head of the *PTQ* for any eligible EF packet for transmission. An EF packet is eligible for transmission if its relevant EF stream has sufficient grants. If there is an eligible EF packet, it is first transmitted. Otherwise, one BE packet is served from *TQ*. The same quantum calculation as DRR+ is used for DRR++. Similar to DRR+, when an EF or BE stream becomes empty, the stream's grant is zeroed.

C.4. The PQWRR Algorithm [MaMo01]

PQWRR uses the PQ scheduling for EF traffic and the WRR scheme for AF and BE traffic with class weights based on their contracted bandwidth. In this scheme, EF traffic is prioritized over AF+BE traffic. In WRR, the mean packet length in each class must be considered in calculating the class weight. The class weight is converted to a small integer number, from the method mentioned in [ItTa02], representing the number of packets to be served from the class in each scheduler round. The WRR scheduler visits the class queues in a round robin manner. When visiting class i, $min(W_i, Q_i)$ packets are served from class i, where W_i is the class i associated (integer) weight and Q_i is the queue size of class i.

Appendix D: Proof of Lemmas in Chapter 4

Appendix D.1: Proof of Lemma 4.1

Proof: From Eq.(4.2) we have

$$R_{SL}(n,E)=\sum_{c=2}^{n-E}\frac{(n-E-1)!(n-1)^{n-E}}{n^{n-E-1}}\frac{c-1}{(n-1)^c c!(n-c-E)!}\ , \tag{D.1}$$

and

$$R_{SL}(n,E+1)=\sum_{c=2}^{n-E-1}\frac{(n-E-2)!(n-1)^{n-E-1}}{n^{n-E-2}}\frac{c-1}{(n-1)^c c!(n-c-E-1)!}\ , \tag{D.2}$$

where the term n-E in Eq.(D.2) is zero. The terms inside the both summations for $c\in[2,n-E]$ are compared. One term of Eq.(D.2) is divided over the similar term in Eq.(D.1) inside the summation. After simplification, we obtain

$$k=\frac{n(n-c-E)}{(n-E-1)(n-1)}=\frac{n^2-nE-nc}{n^2-nE+E-2n+1}\ ,\ \forall c\in[2,n-E]\ .$$

By considering $c \geq 2$, we have E-$2n$+1>-nc. Therefore, we obtain k<1. In other words, each term inside the summation in Eq.(D.2) is less than the similar term in Eq.(D.1). Thus, the summation in Eq.(D.2) is less than the summation in Eq.(D.1). By this way, the lemma is proved.

Appendix D.2: Proof of Lemma 4.2

Proof: Eq.(4.4) provides the loss rate for one single channel. Let E_i denote the average number of the empty slots on wavelength i. Then, $(n-E_i)R_{SL}(n,E_i)$ slots are dropped on average. Thus, the average loss rate of the switch, $R_{SL,SW}(n,E)$, is obtained from

$$R_{SL,SW}(n,E)=\frac{\sum_{i=0}^{W-1}(n-E_i)R_{SL}(n,E_i)}{\sum_{i=0}^{W-1}(n-E_i)}\ . \tag{D.3}$$

When the traffic is uniformly distributed on all W wavelengths, we have E_0=E_1=...=E_n=E . Therefore, at the balanced traffic load on the wavelengths we have

$$R_{SL,SW}(n,E)=\frac{\sum_{i=0}^{W-1}R_{SL}(n,E)}{W}=R_{SL}(n,E)\ .$$

Appendix D.3: Proof of Lemma 4.3

Proof: Eq.(D.3) can be expanded as

$$R_{SL,SW}(n,E)=f(E_0)+f(E_1)+\ldots+f(E_{W-1})\ ,$$

$$\text{where}\quad f(E_k)=\frac{(n-E_k)R_{SL}(n,E_k)}{\gamma_k-E_k}\ ,\quad \gamma_k=nW-\sum_{i=0,i\neq k}^{W-1}E_i \tag{D.4}$$

Since all terms in $f(E_k)$ are positive, then $R_{SL,SW}(n,E)$ is minimum when each term in the right hand side of Eq.(D.4) is minimum. To find the minimum of $f(E_k)$, we have

$$\frac{df(E_k)}{dE_k}=0\Rightarrow\frac{(n-\gamma_k)R_{SL}(n,E_k)+(n-E)(\gamma_k-E)\frac{d(R_{SL}(n,E_k))}{dE_k}}{(\gamma_k-E_k)^2}=0\quad,\quad 0\le k\le W-1.$$

Using Eq.(4.4) to calculate $R_{SL}(n,E_k)$ and simplifying, we obtain

$$-\frac{\gamma_k}{n}\left(\frac{n}{n-1}\right)^{n-E}+(\gamma_k-E_k)ln\left(\frac{n}{n-1}\right)+1=0\ .$$

Now after rearranging,

$$\gamma_k-E_k=\frac{\frac{E_k}{n}\left(\frac{n}{n-1}\right)^{n-Ek}-1}{ln\left(\frac{n}{n-1}\right)-\frac{1}{n}\left(\frac{n}{n-1}\right)^{n-Ek}}=u(E_k)\ .$$

The value of γ_k-E_k for all k is constant, i.e., $\gamma_0-E_0=\gamma_1-E_1=\ldots=\gamma_{W-1}-E_{W-1}=\sum_{i=0}^{W-1}(n-E_i)$. This implies that $u(E_0)=u(E_1)=\ldots=u(E_{W-1})$. On the other hand, by evaluating $u(E_k)$ we find that it is a monotonic decreasing function for $E_k\le n-1$. Thus, we conclude that $E_0=E_1\ldots=E_{W-1}$, which shows the balanced load on the wavelengths will lead to the lowest slot loss rate.

Appendix D.4: Proof of Lemma 4.4

Proof: Consider Eq.(4.3) for the case $c>k$ and call it $\Psi'(n,c,E,k)$, where

$\Psi'(n,c,E,k)=\sum_{c=k}^{n-E}\frac{c-1}{(n-1)^c c!(n-c-E)!}$. There is an outer constant coefficient in Eq.(4.2) independent of c and equal for all terms inside the summation in Eq.(4.2). By dividing both sides of $\Psi'(n,c,E,k)$ over the first term inside the summation on the second side, we obtain

$$\frac{\Psi'(n,c,E,k)}{\frac{k-1}{(n-1)^k k!(n-E-k)!}}=1+\Theta(n,c)\ ,\ c>k\ ,$$

where *1* at the right hand side represents the loss rate due to k-slot collisions and $\Theta(n,c)$ represents the loss rate for more than k-slot collisions. Thus

$$\Theta(n,c)=\sum_{c=k+1}^{n}\frac{\frac{c-1}{(n-1)^c c!(n-E-c)!}}{\frac{k-1}{(n-1)^k k!(n-E-k)!}}\ .$$

By considering that $\frac{(n-E-k)!}{(n-E-c)!}=\prod_{i=k}^{c-1}(n-E-i)<(n-1)^{c-k}$, $\Theta(n,c)$ is simplified to

$$\Theta(n,c)=\sum_{c=k+1}^{n}\frac{k!(c-1)\prod_{i=k}^{c-1}(n-E-i)}{(k-1)(n-1)^{c-k}c!}<\sum_{c=k+1}^{n}\frac{k!(c-1)}{(k-1)c!}\ .$$

By further simplifying,

$$\Theta(n,c)<\frac{k!}{k-1}\left(\sum_{c=k}^{n-1}\frac{1}{c!}-\sum_{c=k+1}^{n}\frac{1}{c!}\right)<\frac{k!}{k-1}\left(\frac{1}{k!}-\frac{1}{n!}\right)<\frac{1}{k-1}<1\ .$$

Therefore, the role of the k-slot collision in dropping slots is larger than the role of all c-slot collisions where $c>k$. For example, by considering k=2, the loss rate due to a two-slot collision is larger than the loss rate due to the sum of 3-slot collision, 4-slot collision,..., until n-E slot collision. Thus, if a device is used to resolve two-slot collisions, the slot loss rate will significantly reduce. However, this significance is exponentially reduced when k increases. Therefore, to achieve a very low loss rate, many contention resolution devices will be required.

Two approaches are candidate for implementing a k-slot resolution: wavelength converters and multiple fibers. In the former case, k-*1* full wavelength converters ($k \le W$)

are used at each output port of the core switch. In the latter case, ingress switches are connected to the core switch with *k-1* more fibers.

Consider two architecture scenarios: the first scenario has one fiber and *h* wavelengths inside the fiber, and the second scenario has *h* fibers and each fiber with one wavelength. Note that the latter scenario has two noticeable advantages: a higher network survivability and a less contention rate. Moreover, the cost of the second scenario is less than the architecture that uses the first scenario plus wavelength converters at the core switch to achieve the same loss rate reduction as the first scenario [SoMi04].

Appendix D.5: Proof of Lemma 4.5

Proof: Using Eq.(4.4), the drop reduction ratio can be obtained by the following equation:

$$r = \frac{R_{SL}(n,E)}{R_{SL}(n,0)} = \frac{\frac{\alpha^{n-1}}{n-1} - \frac{\alpha^n - \alpha^E}{n-E}}{\frac{\alpha^{n-1}}{n-1} - \frac{\alpha^n - 1}{n}} = \frac{\frac{\alpha^{n-1}}{n-1} - \frac{\alpha^n - \alpha^E}{n-E}}{\frac{1}{n}} = \frac{n\alpha^E - E\alpha^n}{n-E},$$

where $\alpha = \frac{n}{n-1}$. By replacing $E = np_e$ in the above equation and simplification, r is simplified to

$$r = \frac{(\alpha^n)^{Pe} - p_e\alpha^n}{1-p_e}.$$

Now for a large n and considering that $\lim_{n->\infty} \alpha^n = e$, the maximum drop reduction ratio is given by

$$r_{max} = \frac{e^{Pe} - ep_e}{1-p_e}.$$

Appendix E: Building MUX/DMUX Modules

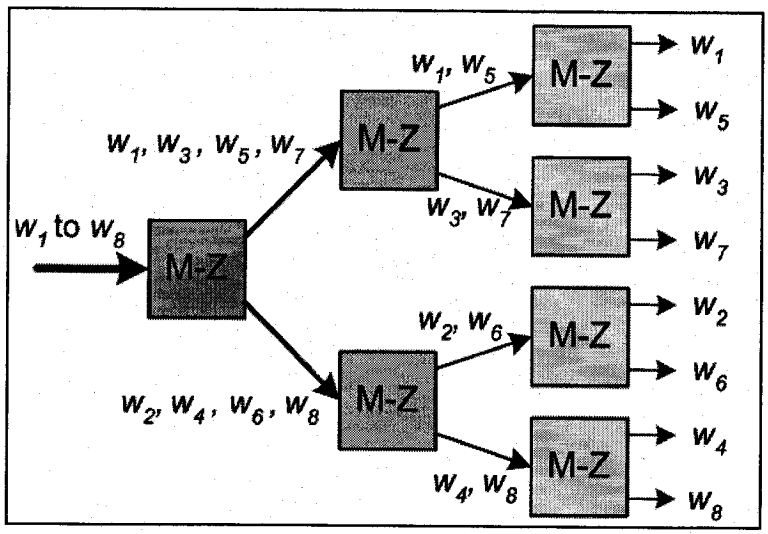


Fig.E.1: Cascaded M-Z Filters to Make 1x8 DMUX (similar to Fig.2.34 [Kart03])

Fig.E.1 shows a 1x8 demultiplexer module by cascading a number of channel splitters (each constructed with an M-Z filter) detailed in [Kart03]. Note that in a reverse case, channel inter-leavers are cascaded to construct a multiplexer module [Kart03, StBa00]. Channel splitter and inter-leaver can also be constructed using Multilayer Interference (MI) filters [StBa00]. Then, an integrated array of MI filters can make a DMUX or MUX [StBa00].

Now suppose M-Z filters are used to construct a demultiplexer module. Let $N_{MZ}(\hat{W})$ be the required number of filters for demultiplexing $\hat{W}$ channels. Clearly, for one channel we do not need any filter. For $\hat{W}>1$ channels (where $\hat{W}$ may be either an even or an odd number), we need one filter to divide $\hat{W}$ channels into two parts. Then, we require $N_{MZ}(\lfloor \hat{W}/2 \rfloor)$ filters for $\lfloor \hat{W}/2 \rfloor$ channels and $N_{MZ}(\hat{W}-\lfloor \hat{W}/2 \rfloor)$ filters for the remaining $\hat{W}-\lfloor \hat{W}/2 \rfloor$ channels. Therefore, one can formulate $N_{MZ}(\hat{W})$ by the following recursive equation:

$$N_{MZ}(\hat{W}) = \begin{cases} 1+N_{MZ}(\lfloor \hat{W}/2 \rfloor)+N_{MZ}(\lceil \hat{W}/2 \rceil) & \text{if } \hat{W}>1 \\ 0 & \text{if } \hat{W}=1 \end{cases}$$

Using mathematical induction, one can easily prove $N_{MZ}(\hat{W})=\hat{W}-1$, and therefore, we have $C_{MD}(\hat{W})=(\hat{W}-1)C_{MZ}$, where C_{MZ} is a cost of one filter.

Appendix F: Reduced Complexity Linear Search Algorithm

The status of slot-sets (in terms of empty or non-empty) within a frame can be represented by an array of bits in which we assign one bit to each slot-set. When this bit is set (reset), the slot-set is empty (non-empty). This type of memory management has been used for different sorting and searching algorithms, e.g., [Hage98, Pisi03].

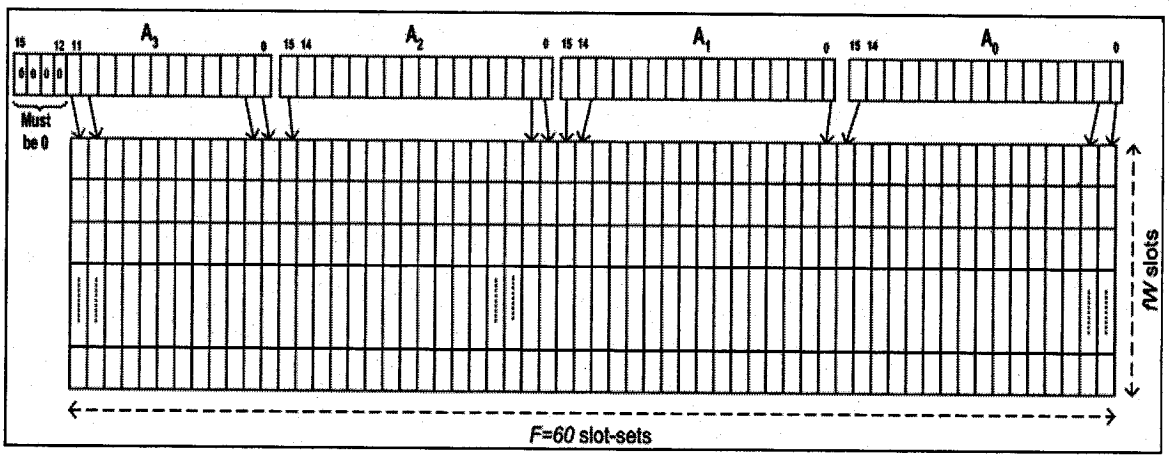


Fig.F.1: An Example to the Reduced Complexity Linear Search Algorithm

To manage the bits, Boolean operators (such as binary *and,* binary *or*, etc.) must be used. However, the Boolean operators can only be applied on a word of size ω bits (depending on the system processor) where ω=16bits, ω=32bits or higher. Therefore, the array of bits must be managed by an array including $N_W = \left\lceil \frac{F}{\omega} \right\rceil$ words. For example as illustrated in Fig.F.1, if the frame includes F=60 slot-sets and ω=16bits, then we require an array of N_W=4 words (named as A_3, A_2, A_1 and A_0) to maintain the status of the slot-sets. All bits in a word are numbered from left to right as ω−1, ω-2,...,1,0 (see Fig.F.1). The words in the array are numbered in a similar way as N_W-1, N_W-2, ..,1,0 (see Fig.F.1). The slot-sets in the frame are also numbered from left to right. Based on this numbering, the status of slot-set s is maintained by the bit $k=s \% \omega$ in word A_m where $m=\left\lfloor \frac{s}{\omega} \right\rfloor$.

Before the slot assignment procedure, all words used in the array must be set to 1, except those bits that are not used. For example, the four unused bits in word A_3 (see Fig.F.1) must be set to 0, which is a requirement in the following code.

```
1.  Function SelectNextEmptySlotSet (S, step)
2.  {
3.    s = (S - step) % F   ///obtain the next desired slot-set, may re-circulate to the frame end
4.    if (SlotSetCapacity[s]!=0) return s;   ///if slot-set s is empty return
5.    k = s % ω
6.    v=m = ⌊ s / ω ⌋
7.    a = A_m and (2^k-1)          ///binary and operation to reset all bits from k to ω-1
8.    while (a=0)                  ///loop until finds a word which indicates an empty slot
9.      {
10.       m = (m-1) % N_W          ///obtain index for the next word, may re-circulate
11.       if ( m = v )
12.          a = A_m and (2^ω -2^(k+1))  ///binary and operation to reset all bits from 0 to k+1
                                         ///in the word that indicates the status of slot-set s
13.       else
14.          a = A_m                 //// an other word that does not indicate the status of slot-set s
15.     }
16.   b = getPosMS_Bit (a)          /// gets the position of the most significant set bit in a
17.   return  m x ω + b
18. }
```

Fig.F.2: Pseudo Code of the Reduced Complexity Linear Search Algorithm

Fig.F.2 shows the pseudo code to implement the function of *SelectNextEmptySlotSet()* used in the DTDM algorithm in Section 6.3.2, which returns the reference of the next empty slot-set within the frame. Line-4 obtains the reference of the next desired slot-set. If this slot-set is empty, it is returned to the DTDM algorithm. Otherwise, Line-5 and Line-6 compute the bit number in the relevant word (k) and the reference of the relevant word (m and v) respectively to maintain the status of slot-set s as discussed. Then, Line-7 resets all bits that are numbered from k to ω-1 to 0 and puts the results in parameter a. When a is not 0, it means that there are some bits equal to 1 in word A_m representing empty slot-set(s). In this case, the loop is skipped. Otherwise, the loop searches for a non-zero word. This loop may be executed N_W times in the worst case. After the loop, Line-16 finds the position of the most significant set bit (i.e. 1) in a and saves it in b. To compute b, one can use the BSR instruction[13] in Intel processors [IRPM06] or BFFFO instruction in Motorola processors [MMIS06]. Finally, Line-17 calculates the slot-set number within the frame using m, ω and b, and then returns it to the

[13] BSR instruction requires two operands. It searches the source operand (the second operand) for the most significant set bit (1 bit). If a most significant 1 bit is found, its bit index is stored in the destination operand (the first operand which is a register). Note that BFFFO is a similir instruction in Motorola processors.

DTDM algorithm.

Note that the complexity of the linear search algorithm using the method presented here can be reduced significantly to $O(N_W)=O(\left\lceil \frac{F}{\omega} \right\rceil)\approx O(1)$, especially when ω is a large number. For example, we have ω =64 in an Intel's 64-bit processor. Now if $F \leq 64$, the complexity is exactly O(1). If $F \leq 128$, the complexity is $O(2) \approx O(1)$. In practice, we should choose $F \leq 128$ in the DTDM protocol in order to provide bandwidth accurately.

Appendix G: The Birkhoff and von Neumann (BvN) Algorithm

Here, we summarize the key operation of the BvN decomposition algorithm [ChCh99b, ChCh00, ChCh01] used in Section 7.2. The idea of this algorithm is to decompose a matrix in a number of permutation matrices. A permutation matrix with entries of just 0 and 1 is a matrix in which the summation of each row or column is always 1.

After collecting traffic demand from all ingress switches at the core switch, the slot assignment process is simultaneously determined for all ingress switches. To perform the scheduling, the traffic demand matrix must first be converted to a doubly stochastic matrix R in which the summation of each row and the summation of each column is equal to 1.0. The lowest complexity for this conversion is $O(n^2)$ where n is the number of ingress switches connected to the core switch. Then by using the BvN decomposition algorithm, the permutations matrices and relative times of the permutations being scheduled (will be denoted by permutation weight) are extracted from the doubly stochastic matrix R in at most $n^2-2n+2 = O(n^2)$ iterations. For each iteration, a bipartite matching (perfect matching) should be computed with the complexity [PaSt82] of $O(n^3)$ via the alternating path or $O(n^{2.5})$ via the maximum flow algorithm. After decomposition, we have $R = \sum_{i=1}^{k} u_i P_i$, where k is the number of permutation matrices, u_i is the permutation weight, and P_i is a permutation matrix. A permutation matrix can be easily mapped to a cross-bar switch. Each permutation should receive service in proportion to its weight.

This algorithm can guarantee the minimum service requirement to any source-destination pair. However, the main issue with the matrix decomposition algorithms is its higher complexity. Based on the above discussion, the complexity of CTDM for one channel is $O(n^2)+ O(n^2)$ x $O(n^{2.5})= O(n^{4.5})$, an exponentially increasing complexity.

Appendix H: OPNET Modeling

The OPNET [OPNE06] Modeler is an environment for modeling and simulation of communication networks, protocols, and applications. OPNET provides three hierarchical levels to model a communication network including network level, node level, and process level. One can present the topology of a communication network in the network level. The network level consists of node and link objects. The node model can describe the architecture of individual network elements by displaying the data flow between functional modules, where a module could be protocol layers, algorithms, and buffers. Each module can generate, send, and receive packets from other modules. Finally, the process level uses a Finite State Machine (FSM) approach to implement the detail of each module by using C/C++ code.

The models presented in Chapter 2, Fig.7.1, and Appendix A have been used in our performance evaluations in this dissertation. In Chapter 3, we simulate the OCGRR packet scheduler in a node level based on the model presented in Fig.2.3. In Chapter 7, we have used a single-star network model as shown in Fig.7.1 for performance evaluation. In other chapters, we provide performance evaluations in network level based on the general model presented in Fig.2.1.