QoS Control and Performance Improvement Methods for Optical Burst Switching Networks

Ping DU

DOCTOR OF PHILOSOPHY

Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)

2007 (School Year)

March 2007

QoS Control and Performance Improvement Methods for Optical Burst Switching Networks

Abstract

Optical Burst Switching (OBS) has been proposed as a promising switching technology for the next generation of optical transport network. In this dissertation, we address the issue how to provide QoS control and improve performance in OBS networks.

In order to provide proportional differentiated services without or with absolute constraints in bufferless OBS networks, a Dynamic Wavelength Selection (DWS) scheme is introduced to assign more and longer periods of wavelengths to higher priority classes dynamically and efficiently due to wavelength-sharing. The integration of DWS scheme and another proposed Delayed Burst Assignment (DBA) scheme improves the burst loss performance by giving the burst head packet (BHP) two opportunities of scheduling its data burst (DB).

Furthermore, we propose two novel burst assembly algorithms with traffic shaping functions to reduce the variance of assembled traffic and improve the burst loss performance, which are named advanced timer-based and sliding window-based assembly algorithms.

Finally, an edge buffering based OBS layer fast error recovery scheme is proposed to improve the TCP performance over OBS networks, in which lost bursts are retransmitted at edge nodes without being reassembled.

In general, all the proposed schemes solve many of the fundamental issues faced by optical burst switching networks, thereby making OBS more practical and efficient in the near future.

Acknowledgements

First, I would like to express my deepest appreciation to my research super-advisor, Assoc. Prof. Shunji Abe, for his patient, friendly, and unfailing support over the past three years. Without his invaluable directions, enlightenment, advices and constant supports, this work could never have been completed. Also I would like to thank my sub-advisors, Prof. Shigeki Yamada, Dr. Toru Hasegawa of KDDI R&D Labs, Assoc. Prof. Yusheng Ji, and other dissertation committee members, Prof. Shigeo Urushidani, Prof. Tomohiro Yoneda, Assoc. Prof. Hideki Tode of OSAKA University, for their guidance and their valuable comments.

Furthermore, I would like to extend my highly appreciation to the staffs at NII, especially Ms. Nahoko Iwanaga, Ms. Akiko Uchida, Ms. Yasuko Umebayashi, Ms. Nakata Tami, Ms. Miyuki Kobayashi, and Ms. Sayo Ebihara, for their valuable and fruitful supports during my staying at NII.

I would like to express my utmost appreciation to my parents and my friends for their love and constant support which inspire me to continue my studies. This Ph.D program has been financially supported by NTT DATA Corporation and NII. I would like to thank all people in these organizations that support me to finish my Ph.D program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations and Objectives

Nowadays, the scale of the Internet enlarges rapidly, and the demand for network bandwidth has been increasing remarkably. Dramatically increased amount of the World Wide Web users have brought more information servers online. Furthermore, the types of network services have been largely increased, and the proportion of multimedia technologies integrated by video and audio becomes larger. All these reasons have made the Internet traffic increase rapidly, and the demand for network bandwidth become more urgent than ever.

With the explosive growth of the Internet and the rapid evolution of Dense Wavelength Division Multiplexing (DWDM) technique, optical fiber seems to be the perfect carrier for future high-speed networks. In a DWDM system, each fiber carries multiple communication channels, with each channel operating on a different wavelength [1]. Such an optical transmission system has a potential capacity to provide over 50Tbps bandwidth on a single fiber.

Current networks typically consist of four layers: IP layer for carrying applications and services, ATM (asynchronous transfer mode) layer for traffic engineering, SONET/SDH layer for transport, and DWDM for capacity. When the data stream arrives at a switching point, the optical signal of data is converted into electronic form, and the processing and forwarding are done in the electronic domain. This is known as an Optical-Electronic (O/E) conversion. When the electronic signal of

data is passed to the output port, it is again converted and modulated onto the fiber as an optical signal (Electronic-Optical). Such a switching point is said to perform O/E/O conversion.

In such a network, all communications are limited by the electronic processing capabilities of the system. Although hardware-based high-speed electronic IP routers with capacity up to a few hundred gigabits per second are available now, there is still a serious mismatch between the transmission capacity of WDM fibers and the switching capacity of electronic IP routers.

With IP traffic as the dominant traffic in the networks, the traditional layered network architecture is no longer adapted to the evolution of the Internet. In the multi-layered architecture, each layer may limit the scalability of the entire network, as well as adding the cost of the entire network. As the capabilities of both routers and OXCs (optical cross-connects) grow rapidly, the high data rates of optical transport suggest bypassing the SONET/SDH and ATM layers and moving their necessary functions to other layers. This results in a simpler, more cost-efficient network that can transport very large volumes of traffic. IP over WDM is considered as a promising solution for the next generation network since it has no intermediate layer so that it can void the functionality redundancy of the ATM and SONET/SDH layers.

There are sill some difficulties in realizing all optical networks, such as the optical RAM is ongoing research now, and some technologies and standards have to be designed. So the processing of IP packets in the optical domain is still not practical yet, and the optical router control system is implemented electronically. Nowadays, we are mostly studying the semi-transparent optical transport networks. In optical transport networks, the control messages are processed electronically, and the data are propagated in the high-speed transparent data channels. To realize an IP-over-DWDM architecture, several approaches, such as Wavelength Routing (WR) [2, 3, 74], Optical Packet Switching (OPS) [9,12,14,17] and Optical Burst Switching (OBS) [21, 24, 25], have been proposed. Of all these approaches, optical burst switching (OBS) can achieve a good balance between the coarse-gained wavelength routing and fine-gained optical packet switching, thereby combining others' benefits while

avoiding their shortcomings.

1.2 Contributions of This Dissertation

In this dissertation, we analyze several critical issues affecting optical burst switching networks, such as quality-of-service, burst assembly and TCP over OBS. Our main contributions of this dissertation are as follows:

We address the issue how to provide proportional differentiated services in OBS networks [28, 31, 34]. Firstly, a Dynamic Wavelength Selection (DWS) scheme is introduced to provide proportional differentiated services in bufferless OBS networks by assigning more and longer periods of wavelengths to higher priority classes dynamically and efficiently due to wavelength-sharing. Then, a Delayed Burst Assignment (DBA) scheme is introduced, by which bursts of the high priority class are given a higher probability for reserving bandwidth by scheduling the bursts of the low priority class with a delay to provide QoS in OBS networks. The integration of these two schemes not only provides proportional differentiated services, but also improves the burst loss performance by giving the burst head packet (BHP) two opportunities of scheduling its data burst (DB). We also extend our proposed schemes to provide proportional differentiated services with absolute constraints [33].

Next we analyze the assembled traffic of the Science Information Network (SINET) by using the Fractional Brownian Motion (FBM) model. The analytical results show that the general timer-based and threshold-based assembly schemes could not avoid increasing the burstiness, which will deteriorate the network performance. Thereby we propose two novel burst assembly algorithms with traffic shaping functions to reduce the variance of assembled traffic, which are named advanced timer-based assembly algorithm and sliding window-based assembly algorithm. The simulation results show that both the advanced timer-based and the sliding window-based assembly algorithms have better burst loss performance than the timer-based and the threshold-based assembly algorithms. The simulation results also indicate that the advanced timer-based assembly algorithm has better performance in terms of edge buffering delay than the sliding window-based assembly algorithm does [32, 36, 37].

We also develop a precise analytical model to study the performance of TCP traffic in OBS networks. An edge buffering based OBS layer error recovery scheme is proposed for OBS networks, in which lost bursts are retransmitted at edge nodes. The analytical and simulation results show that our proposed scheme can lighten burst loss ratio for TCP flows of the OBS network [30, 35].

In general, all the proposed schemes solve many of the fundamental issues faced by optical burst switching networks, thereby making OBS more practical and efficient in the near future.

1.3 Organization of This Dissertation

This Ph.D dissertation thus investigates issues for QoS control and performance improvement methods in optical burst switching networks. This document is structured as follows:

Chapter 2 provides an introduction to optical burst switching, with comparisons to wavelength routing and optical packet switching. We also introduce the architecture of OBS networks and burst assembly at the edge nodes, signaling schemes for reserving resources in OBS works.

Chapter 3 addresses the issue how to provide proportional differentiated services in OBS networks. We introduce a Dynamic Wavelength Selection (DWS) scheme and an integration of DWS and DBA (Delayed Burst Assignment) scheme to provide proportional differentiated services in bufferless OBS networks.

Chapter 4 extends the proposed schemes in Chapter 3 to provide proportional differentiated services with absolute constraints.

In Chapter 5, based on the analysis of the assembled traffic characteristics, we present and evaluate two novel burst assembly algorithms, named advanced timer-based and sliding window-based, with traffic smoothing functions to reduce the burstiness of assembled traffic and improve the burst loss performance.

In Chapter 6, we develop an analytical model to evaluate the TCP performance of the OBS network and introduce an OBS layer acknowledgment mechanism and a loss detection and error recovery mechanism.

Chapter 7 concludes the dissertation and identifies the future works.

Chapter 2

Optical Burst Switching

This chapter provides an introduction to optical burst switching, in comparisons with wavelength routing and optical packet switching. We also introduce the architecture of OBS networks and burst assembly at the edge nodes, signaling schemes and contention resolutions for reserving resources in an OBS work.

2.1 Optical Switching Technologies

For the next generation optical transport networks, there are three promising optical switching technologies: wavelength-routing (circuit-switching), optical packet switching, optical burst switching. Circuit and packet switching have been used for many years for voice and data communications. Optical burst switching was introduced only recently and haven't been as well studied as wavelength routing and optical packet switching. In the following, we will introduce optical burst switching in comparisons with wavelength routing and optical packet switching.

2.1.1 Wavelength Routing

Wavelength routing is similar to traditional circuit switching, in which a lightpath [90] is established between the source and the destination before data transmission. During transmission, there is no need for the intermediate nodes to perform complex processing of packet header and buffering of the payload.

Birman et. al [6] points out that wavelength routing networks are not equivalent to electronic circuit switching networks because circuits in a circuit switching network are equivalent whereas lightpaths are not because that a lightpath should take the same wavelength along a path.

When wavelength routing lighpath connections are static, they may not be able to accommodate the highly variable and bursty nature of Internet traffic in an efficient manner. If traffic is varying dynamically, sending this traffic over static lightpaths would result in an inefficient utilization of wavelength. In addition, given a limited number of wavelengths, only a limited number of lightpaths can be established at the same time. On the other hand, if we attempt to set up lightpaths in a very dynamic manner, the network state information will be constantly changing, making it difficult to maintain current network state information. How to assign wavelengths to a lightpath along its route is called Routing and Wavelength Assignment (RWA) [7, 8]. The objective for RWA is to set up lighpaths and assign wavelengths in a manner which minimizes the amount of the blocking of connections or which maximizes the number of connections that are established in the network.

One of the main features of wavelength routing is its two-way reservation process in set-up of lighpaths, where a source sends a request for setting up a lighpath and then receives an acknowledgement back from the corresponding destination.

A lighpath is set up by reserving a fixed period on a wavelength along a path from the source to its corresponding destination. The offset time T, between a set-up request and data transmission, is at least as long as $2P + \triangle$, where P is the one-way propagation delay and $\triangle$ is the total processing delay along the path. The set-up time of a lightpath based on a two-way reservation, which may be tens of milliseconds in a nation-wide backbone network, will be too long for a burst only containing a few number of packets.

2.1.2 Optical Packet Switching

Optical packet switching is suitable for supporting bursty traffic since it allows statistical sharing of the wavelengths among packets belonging to different source

and destination pairs [18–20]. As optical switching technology is developed, we may eventually see the emergence of photonic packet switching networks in which packets are switched and routed independently through the network entirely in the optical domain without being converted back to electronic form at each node. Since all-optical header processing will not be economically viable in the near future due to the immaturity of high-speed optical logic, the optical packet switching approach requires each header to go through O/E conversion for processing and E/O conversion for transmission.

When an optical packet arrives, the optical core node first departs the header from the payload and converts it into electronic domain and processes it electronically [16]. The switching fabric is reconfigured based on the information contained within the header of the packet. Since it takes some time for the header to be processed and for the switch to be reconfigured, the payload should be delayed at the input port. At the output port, the header is converted into optical signal and assembled with the payload into to a new optical packet.

There are many challenges in realizing optical packet switching [10, 11]. One is that there is no optical equivalence of the random access memory (RAM). So an optical packet can only be delayed for a limited amount of time via the use of fiber delay lines (FDLs) before the completion of the header processing. The length of each optical packet, in terms of the product of its transmission time and the speed of light, cannot exceed that of the available FDL in which the optical packet to be "stored".

Another major challenge is the stringent requirement for synchronization [13], both between packets arriving at different input ports of an optical switching fabric, and between packets' header and payload.

2.1.3 Optical Burst Switching

Optical Burst Switching (OBS) is proposed to achieve a good balance between the coarse-gained wavelength routing and fine-gained optical packet switching. It is based on a one-way reservation protocol in which a data burst (a number of packets) follows a corresponding control packet without waiting for an acknowledgement.

Table 2.1: Comparison between three optical switching technologies

Optical Switching Technology	Bandwidth Utilization	Setup Latency	Optical Buffer	Proc/Sync. Overhead
Wavelength Routing	low	high	not required	very low
Optical Packet Switching	very high	very low	required	high
Optical Burst Switching	high	low	not required	low

By reserving resources (wavelengths and FDLs) only for a specified period of time rather than reserving resources for an indefinite period of time, the resources can be allocated in a more efficient manner and a higher degree of statistical multiplexing can be achieved. Thus, optical burst switching is able to overcome some of the limitations of static wavelenth allocation incurred by wavelength routing. Furthermore, since data is transmitted in large bursts, optical burst switching reduces the technological requirement of fast optical switches which is necessary for optical packet switching.

Table 2.1 compares these three optical switching technologies. From this table, we can clearly observe that optical burst switching has the advantages of both wavelength routing and optical packet switching, while potentially avoiding their shortcomings. In the next section, we will introduce the architecture and technologies of OBS in detail.

There are some variations of these three optical switching technologies, such as WaveBand Switching (WBS) [22] and Time Division Switching (TDS) [23]. WBS is a variation of Wavelength Routing, in which multiple wavelengths are grouped into a waveband to simplify the structure of OCX by reducing the required input and output ports which are the main cost of switching fabric. In TDS, each wavelength is divided by time slots and each slot is considered as a switching granularity.

2.2 Architecture and Technologies of OBS

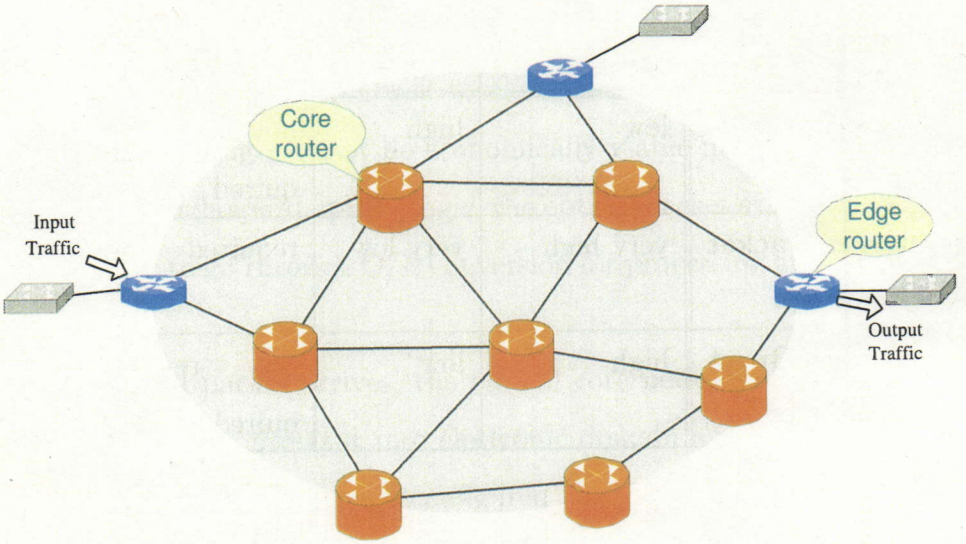


Figure 2.1: OBS network model

In an OBS network (shown in Fig.2.1), the edge routers and core routers connect with each other with WDM links. The edge nodes are responsible for assembling packets into bursts and de-assembling bursts into packets. The core nodes are responsible for routing and scheduling based on the burst header packets.

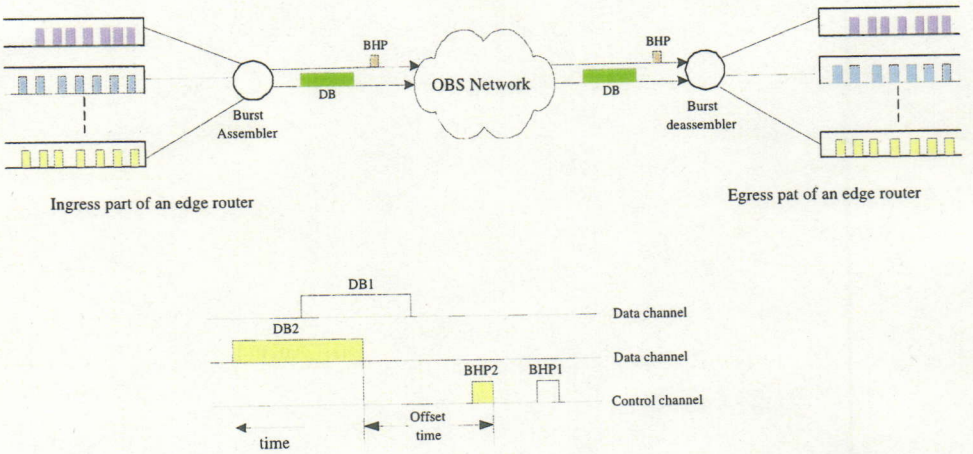


Figure 2.2: Architecture of edge router

An architecture of edge router is shown in Fig.2.2. To eliminate O/E/O (Optical to Electronic to Optical) conversions and electronic processing loads, which are the

bottlenecks of this system, the ingress part of an edge node assembles multiple IP packets with the same egress address into a switching granularity called a burst. A burst consists of a Burst Header Packet (BHP) and a Data Burst (DB). The BHP is delivered on a control channel; its corresponding DB is delivered on a data channel without waiting for a confirmation of a successful reservation. A channel may consist of one wavelength or a portion of a wavelength, in case of time-division or code-division multiplexing [25]. In this dissertation, we assume the words of channel and wavelength have the same meaning.

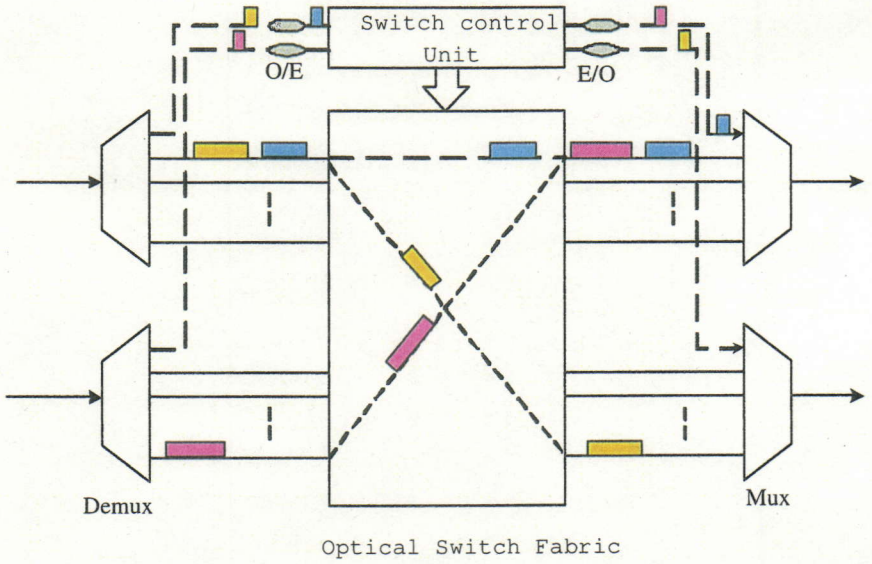


Figure 2.3: Architecture of core router

When a BHP arrives at a core node (shown in Fig.2.3), the core node converts it into an electronic signal, performs routing and configures the optical switching fabric according to the information carried by the BHP. The DB remains in the optical domain without O/E/O conversion when it cuts through the core node.

2.2.1 Burst Assembly Schemes

Burst assembly at the edge router is an important issue for OBS networks. Basically, there are two assembly schemes [25]: *threshold-based* and *timer-based*.

In a timer-based scheme, a timer is started at the initialization of assembly. A burst containing all the packets in the buffer is generated when the timer exceeds

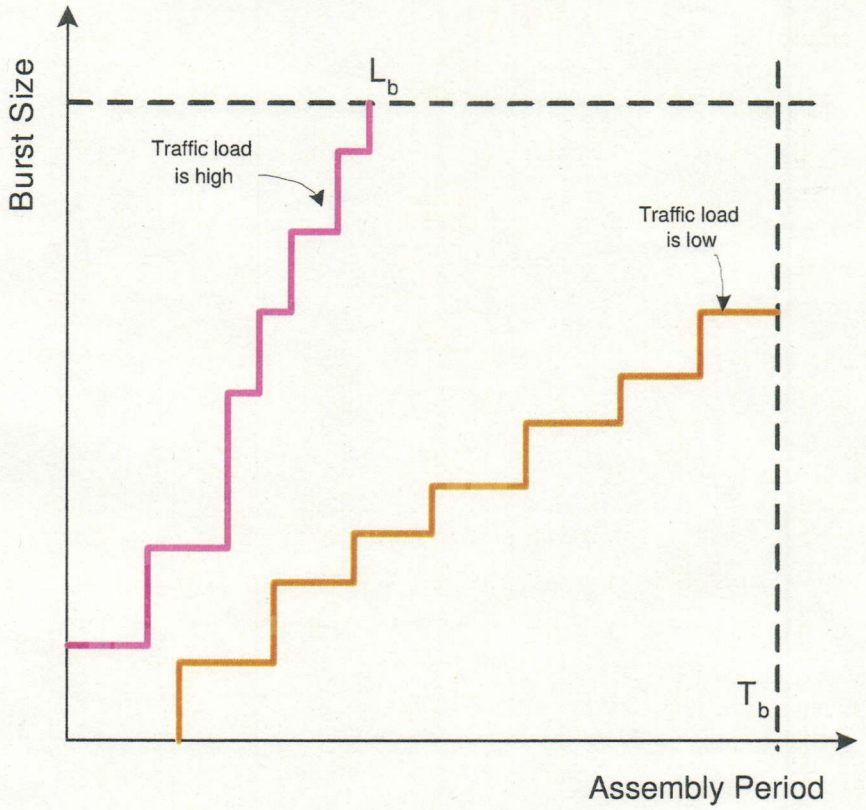


Figure 2.4: Effect of load on timer-based and threshold-based assembly schemes

the burst assembly period T_b. A large time-out value T_b results in a large packet buffering delay at the edge node. On the other hand, a too small T_b results in too many small bursts and a high electronic processing load.

In a threshold-based scheme, a burst is created and sent into the OBS network when the total size of the packets in the queue reaches a threshold value L_b. The shortcoming of the threshold-based scheme is that it does not provide any guarantee on the assembly delay that packets will experience.

The choice of burst assembly algorithms depends on the type of traffic being transmitted. Timer-based algorithms are suitable for time-constrained traffic such as real-time applications because the upper bound of the burst assembly delay is limited. For a time-insensitive application such as file transmission, to reduce the overhead of control packets and increase OBS transmission efficiency, a threshold-based scheme may be more appropriate.

How to choose the appropriate time-out or threshold value for creating a burst

is still an open issue. A smaller assembly granularity leads to a higher number of bursts and a higher number of contentions, but the average number of packets lost per contention is less. Also, it will increase the number of control packets. If the reconfiguration period of optical switching fabric is non-negligible, a smaller assembly granularity will lead to lower network utilization because each switched burst needs a reconfiguration period. On the other hand, a higher assembly granularity will lead to a higher burst assembly delay and the average number of packets lost per contention is larger. There is a tradeoff between the number of contentions and the average number of packets lost per contention. The selection of optimal assembly granularity is strongly correlated to the type of input packet traffic. Cao et al. [82] proposed an Adaptive-Assembly-Period assembly algorithm to adapt assembly periods to match with the TCP congestion control mechanisms and yield a good performance in terms of good-put and data loss ratio. Li et al. [83] pointed out that the worst-case performance of any burst assembly algorithm is primarily determined by the maximum to minimum burst length ratio, followed by the range of offset time.

In the hybrid assembly scheme of timer-based and threshold-based, a burst can be sent out when either the burst length exceeds the desirable threshold or the timer expires. Suppose each edge router has G queues to sort arriving packets. Let the timer of queue $Q[i]$ be denoted by $T[i]$ and the length of $Q[i]$ be denoted by $L[i]$. The hybrid scheme is thus implemented using the following algorithm.

Begin

1. When a packet with length of b arrives to $Q[i]$

If ($Q[i]$ is empty)

Start timer $T[i]$, $L[i] = b$

Else

Push packet into $Q[i]$, $L[i] = L[i] + b$

If ($L[i] > L_b$)

Generate a burst with length $L[i]$ and send it into the OBS network,

$L[i] = 0$, stop timer $T[i]$

End if

End if

2. When $T[i] = T_b$

Generate a burst with length $L[i]$ and send it into the OBS network,

$L[i] = 0$, stop timer $T[i]$

End

After a burst is generated, the burst is buffered in the queue for an offset time before being transmitted to give its BHP enough time to reserve network resource along its route. During this offset time, packets belongs to that queue may continue to arrive. Because the control packet that contains the burst length information has already been sent out, these arriving packets could not be included into the buffered burst. Besides dropping these extra packets, one way to deal with these packets is to perform burst length prediction [84]. Let the control packet carries burst length included a predicted part about the additional packets arrives during the offset time. When the predicted extra burst length is larger than the real extra burst length, part of reserved wavelength will be wasted. Otherwise, a few extra packets are dropped.

Another way is to apply two alternate buffers instead of one buffer for each queue $Q[i]$. When a burst is generated at one buffer, the future arriving packets will be stored at another buffer until the next assembly condition is met.

2.2.2 Wavelength-reservation Schemes

In OBS networks, a control packet is sent ahead to reserve wavelength along its route for its data burst. Widjaja et al. [27] proposed two admission-control protocols for ATM networks. One is called Tell-And-Wait, in which when the source has a burst to transfer, it first tries to reserve the bandwidth along the virtual circuit by sending a short 'request' message. If the requested bandwidth is granted at all the intermediate nodes along its route, an acknowledgement (ACK) from the destination will return to the source after a round-trip delay; otherwise, a negative acknowledgement (NACK) will return to the source. The source transmits its burst on receiving an ACK.

Another is called Tell-And-Go, in which the source transmits a burst immediately as soon as it receives the message from the higher layer. A copy of the message is kept at the source until the source learns that the burst has arrived at the destination successfully. The receiver sends an ACK back to the corresponding source when receives the message successfully. Otherwise, the receiver sends back a NACK so that the source can transmit the same burst again at sometime later.

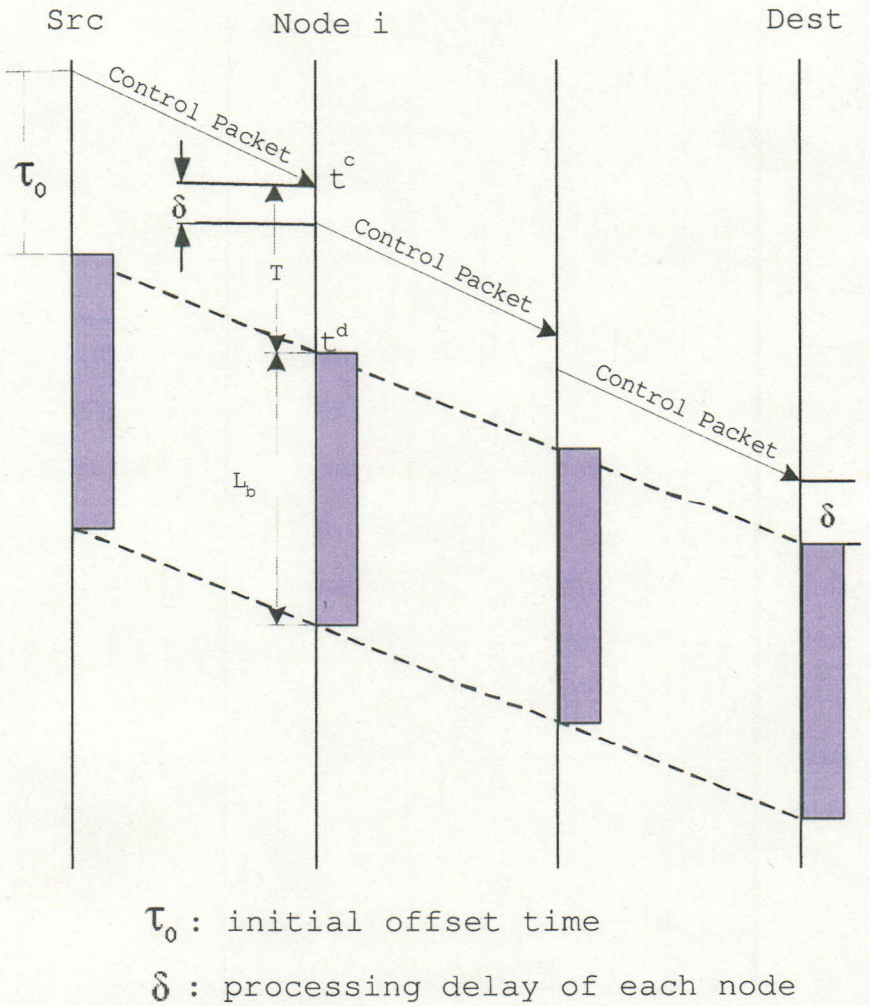


Figure 2.5: Just-Enough-Time (JET) signaling technique

In OBS, a signaling scheme is also required for reserving resources and configuring optical switches for burst transmission. Several wavelength-reservation schemes for OBS, such as Tell-And-Go (TAG) [47], Just-In-Time (JIT) [49] and Just-Enough-Time (JET) [46] have been proposed. In a TAG scheme, a source node sends out a

control packet to inform a burst's arrival. The source node then immediately sends out a data burst. In order to allow time for the processing of the control message and the configuring of the switch at each node, the burst may need to be buffered at each node. In a JIT scheme, the data burst is buffered at the edge node where electronic memory is cheap and abundant, rather than at the intermediate switching nodes where optical delay lines are expensive and limited. A source node sends a SETUP message to reserve wavelength before data transmission and a RELEASE message to release wavelength after data transmission. An intermediate switching node will attempt to reserve wavelength immediately when receiving a SETUP message and release wavelength on receiving a RELEASE message. However, it results in a bad wavelength utilization due to that the bandwidth holding time is bigger than the burst transmission time.

Figure 2.5 illustrates the JET signaling technique. As shown, the ingress node waits for a long offset time before it starts to transmit the data burst. The initial offset time is set to be larger than the total processing time of the BHP along its path. It is calculates as $\tau_o = h \times \delta$, where h is the number of hops between the source and the destination, δ is the per-hop burst header processing time. If at any intermediate node, the reservation is unsuccessful, the burst will be dropped. Comparing with JIT, the BHP in JET contains the information of the offset time and the burst length. So the wavelength at each intermediate node will be reserved at the start of the data burst and will be released at the end of the data burst.

2.2.3 Burst Scheduling Algorithms

When a BHP arrives at a core node, the core node converts it into electronic signal and obtains the burst arrival time and duration of its corresponding data burst from the BHP. A data channel scheduling algorithm is invoked to assign a data channel on the outgoing link to the data burst. The optical switching fabric is reconfigured based on the scheduling results. We assume there are full-wavelength conversions.

To find a suitable wavelength among the candidate wavelengths for an arriving burst, several data channel scheduling algorithms, such as First Fit Unscheduled Channel (FFUC) [25], Latest Available Unused Channel (LAUC) [25], Latest Avail-

able Unused Channel with Void Filling (LAUC-VF) [25], have been proposed.

A. FFUC

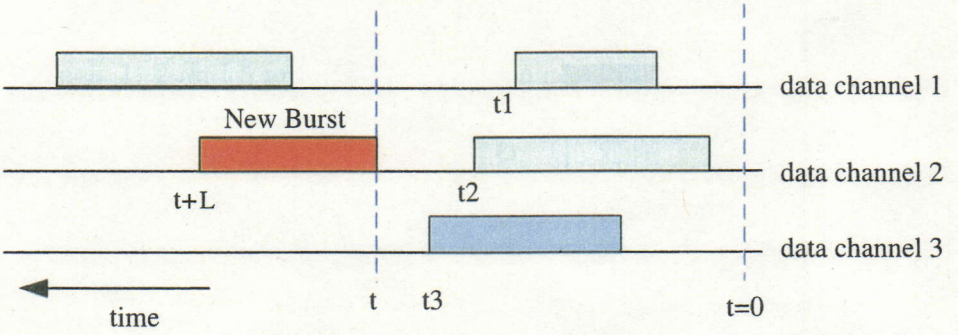


Figure 2.6: Illustration of FFUC data scheduling algorithm

The FFUC (also called Horizon) algorithm keeps the unscheduled time for each data channel. When a BHP arrives, the FFUC algorithm searches all data channels in a fixed order and assigns the burst to the first channel that is available at the arrival time of the data burst. An example of FFUC is shown in Fig.2.6, in which data channel 2 is selected for the new burst because data channel 2 is the first channel that has unscheduled time at the arrival time t of the new burst when we search as the increasing order of channels.

For a link with k channels, the best implementation of the Horizon scheduling algorithm takes $O(logk)$ time to schedule a burst. Accordingly, the Horizon algorithm is relatively simple and has a reasonable good performance in terms of its execution time. However, the Horizon scheduling algorithm results in a low wavelength utilization and a high loss rate. This is due to the fact that the Horizon algorithm does not consider about the generated void intervals between bursts on each data channel.

B. LAUC

Comparing with FFUC, the LAUC increases wavelength utilization by minimizing voids created between bursts by selecting the latest available unscheduled data channel for each arriving data burst. For example, in Fig.2.7, data channel 2 and

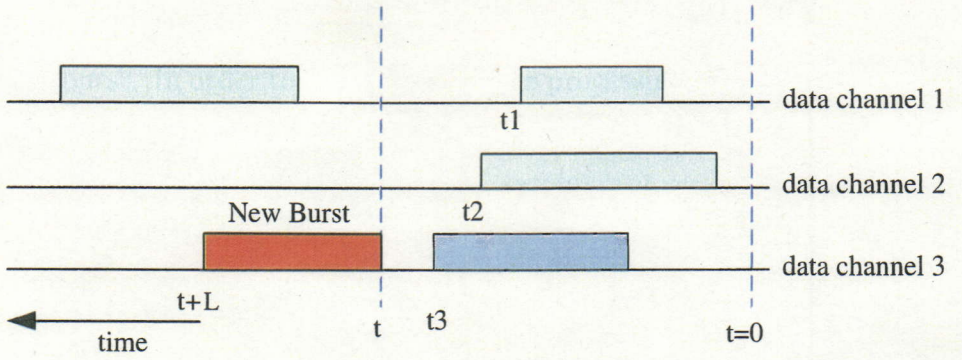


Figure 2.7: Illustration of LAUC data scheduling algorithm

data channel 3 are unscheduled at the arrival time t of the new burst. Data channel 3 will be selected for the new burst because the generated void $(t - t_3)$ on data channel 3 will be smaller than the void $(t - t_2)$ that would have been created if channel 2 was selected. Therefore, LAUC yields better burst loss performance than FFUC.

C. LAUC-VF

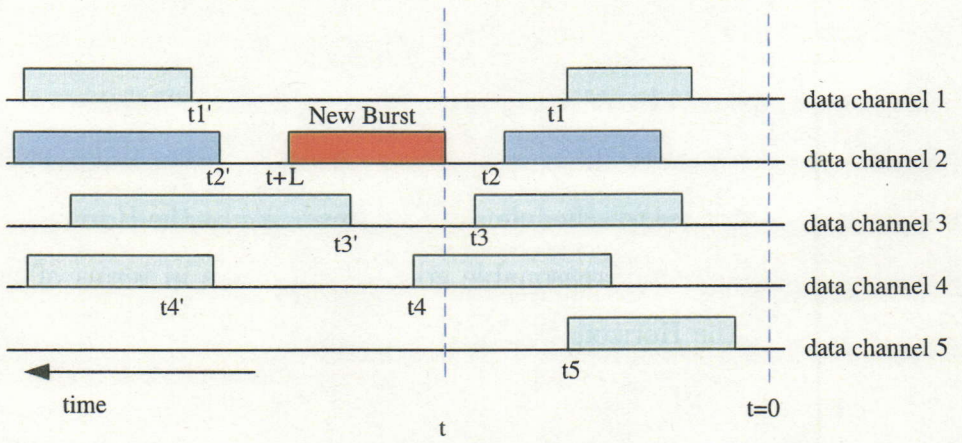


Figure 2.8: Illustration of LAUC-VF data scheduling algorithm

In both FFUC and LAUC scheduling algorithms, the voids between two data burst assignments on one data channel, which are also unused channel capacity, have not been utilized. LAUC-VF (Latest Available Unused Channel with Void Filling) is similar to LAUC except that in which such voids could be filled by new arriving

bursts. In LAUC-VF, the start time and end time of voids for each data channel are maintained. An example of LAUC-VF algorithm is illustrated in Fig.2.8. Given the arrival time t of a new data burst with duration L to the optical switch, the scheduler first finds the outgoing data channels that are available for the time period $(t, t+L)$. Data channels 1, 2 and 5 are available for the coming data burst. Data channel 2 is chosen to carry the new data burst because the void that will be produced between the burst and coming data burst is the minimum value.

LAUC-VF yields a better wavelength utilization and a lower loss rate than the LAUC algorithm. However, even the best known implementation of LAUC-VF has a much longer execution time than the Horizon scheduling algorithm, especially when the number of voids is significantly larger than the number of data channels (which in general is the case).

Several variants of the LAUC-VF algorithm such as Minimum Void [62], Min-SV (Starting Void) [85], Min-EV (Ending Void) [85] algorithms were proposed recently. Minimum Void scheduling algorithm selects the data channel in which a void newly being generated after the burst transmission becomes minimum. Min-SV algorithm can schedule a burst successfully in $O(logm)$ time, where m is the total number of void intervals, as long as there is a suitable void interval. It achieves a loss rate which is at least as low as LAUC-VF, but can run much faster. In fact, its speed can be almost the same as FFUC algorithm. Min-EV tries to minimize the new generated void between the end of new reservation and an existing reservation.

2.2.4 Contention Resolutions

Since data bursts are injected into the OBS backbone by edge routers without waiting for a confirmation of a successful reservation, a burst may contend for the wavelength with other bursts from different burst sources at intermediate nodes, when they want to occupy the same wavelength on the same output port simultaneously. Intermediate OBS nodes are required to resolve possible contention among bursts. Besides dropping contenting bursts directly, contention can be resolved in the following ways:

- `FDL buffering`: a contending burst is delayed for a fixed period by passing

through a FDL [71, 73].

- `Wavelength conversion`: a contending burst is sent on another wavelength through wavelength conversion [46, 77].
- `Deflection routing`: a contending burst is sent to a different output port and then follows an alternative route to the destination [79, 80].
- `Burst segmentation`: a contending burst is broken into segments and the overlapped part of the contending burst is dropped [81].

Table 2.2: Comparison of different contention resolutions

Contention resolution	**Advantages**	**Disadvantages**
FDL buffering	Simple and mature	Increasing end-to-end delay and dimensions of nodes
Wavelength conversion	The most efficient solution	Immature and expensive
Deflection routing	No extra hardware requirement	Out of order arrivals
Burst segmentation	Lower packet loss ratio	Complicated control

Table 2.2 compares these four contention resolutions in detail. Of course, these contention resolutions can be applied jointly.

Chapter 3

Proportional Differentiated Services Models

3.1 Introduction

Realizing some kind of service differentiation is listed as one of the important issues for OBS networks. The differentiated services enable Internet users to provide diverse quality of service (QoS). Higher network usability can be expected as the result of realizing this differentiation. Some methods for realizing differentiated services have been proposed [46, 53, 54]. These methods can be divided broadly into two categories, one is non-proportional method and another is proportional method. In the non-proportional differentiated services model using an extra-offset-time-based [46] QoS scheme, different offset times are assigned to different priority classes without any buffer in the WDM layer. By providing a larger offset time, a higher priority burst is more likely to have wavelength reserved for it because of its early reservation. However, the difference of the burst loss ratios of each class is unstable because it depends on the traffic load. Although we can change the extra offset times' difference to modify the difference of burst loss ratios, a quantitative solution cannot be found in the approach. We call the proportional differentiated model in an OBS network if the burst loss ratio of one service class is proportional to those of other classes regardless of the traffic load. Hence, in the proportional differentiated model, the burst loss ratio of one class is "predictable" if we know that of

another class and is also "controllable" because the network provider can adjust the class differentiation parameters to adjust the burst loss ratios of each class [89]. It can be expected that introducing a proportional differentiated model into an OBS network would be favorable to both network operators and users. Yang et al. [53] introduced an intentional dropping scheme to maintain the proportion of the loss ratios for each class based on a set of predefined parameters. At core nodes, arriving bursts of the lower priority class are dropped when the burst loss ratio of the higher priority class is too high for the proportional differentiated model even when there are idle wavelengths to assign for the lower priority bursts. When the lower priority bursts are dropped, the arrival time of a coming higher priority burst is unknown. This points to a shortcoming of the intentional dropping scheme: the wavelengths "saved" by the dropping of the lower priority bursts will be wasted if no burst of the higher priority class arrives during these "saved" periods. This results in bad wavelength utilization and high blocking probability.

Some other schemes adopt burst-segmentation [55] or wavelength-preemption [54] techniques to achieve QoS support. In [91–93], Dr. Jumpot proposed a bandwidth allocation with wavelength preemption (BA-WP) scheme to provide proportional differentiated service. In the BA-WP scheme, when there are no wavelength available for a higher priority burst, the higher priority burst can preempt the wavelength that has assigned to a lower priority burst and generate a control packet at the core node to inform the up-stream and down-stream nodes this change of wavelength reservation.

In OBS, a BHP is processed and sent to the next hop without waiting for the arrival of its DB. After that, if its DB is truncated or dropped, the BHP is unaware of these changes and cannot update its carried information (e.g., burst length). Any attempt to preempt the reserved resources of lower priority classes by higher priority classes is therefore awkward and inefficient. A solution for preemption and segmentation is to generate and send a trailer control packet to cancel resources reserved by the preempted bursts or the dropped parts of the bursts at core nodes. This results in a complex and difficult implementation, although it does improve wavelength utilization.

In this chapter, we focus on the issue of providing proportional differentiated services with good burst loss performance and simple implementation. First, we introduce a scheme called *Dynamic Wavelength Selection* (DWS) to provide proportional differentiated services in bufferless OBS networks by dynamically assigning more and longer periods of wavelengths to higher priority classes . Compared with general wavelength-continuity-based scheme [59, 60], the DWS scheme can not only adjust the wavelength numbers dynamically when the traffic load changes, but can also utilize the wavelengths more efficiently because the wavelengths are shared among different classes. We also propose another differentiation scheme, named *Delayed Burst Assignment* (DBA) scheme, in which bursts of the lower priority class are processed after a delay to ensure that bursts of the higher priority class have a higher probability of wavelength reservation. In DBA, BHPs are buffered electrically so that fiber delay lines are unnecessary at core nodes.

In the combination of these two schemes, a BHP of a lower priority class is buffered at the core router when it cannot find an available wavelength. It has an opportunity to reschedule its burst to the wavelengths that have been assigned to higher priority classes but have not yet been reserved . The integrated scheme not only provides proportional differentiated services but also achieves lower average dropping probability without any preemption or segmentation mechanisms. Compared with existing approaches, the combined scheme does not need to generate any special packets and needs to maintain only a few parameters at core nodes.

The remainder of this chapter is organized as follows. Section 3.2 and Section 3.3 introduce our proposed DWS and DBA schemes. Section 3.4 discusses the integrated scheme of DWS and DBA. Section 3.5 studies the performance of our proposed schemes.

3.2 Dynamic Wavelength Selection Scheme

Regarding the QoS frameworks suggested by IETF, two broad approaches have been proposed [57]: Integrated Services (IntServ) and Differentiated Services (DiffServ). IntServ achieves QoS guarantees through end-to-end resource reservation for packet

flows at all intermediate nodes. The resource reservation cannot be achieved unless each node is able to participate. The DiffServ model is more scalable than the IntServ model in which the core routers differentiate packets on a class-by-class basis.

To realize DiffServ, a proportional differentiation model seems to be the most promising, owing to its merits. Basically, in a proportional differentiated model, the proportion of burst loss probability of one class to that of another class is maintained at a predefined ratio.

Supposing an OBS network offers N classes of service, we let $\overline{Pb_i}(t, t+\tau)$ be the average burst loss ratio measured in the time interval $(t, t+\tau)$ for class i, where $\tau > 0$ is the monitoring timescale. The calculation accuracy of the average burst loss ratio depends on the monitoring time scale. The average burst loss ratio in a large monitoring time scale cannot response the real-time burst loss ratio when the traffic is bursty [53]. On the other hand, a too small monitoring time scale detracts from the accuracy because only a small number of bursts can be observed in the time scale. Therefore, we assume for now that the monitoring time scale is long enough. We will study the impact of monitoring timescale on the calculation accuracy of burst loss ratio in detail in Section 3.5. The proportional differentiation model requirement is defined in [56] as:

$$\frac{\overline{Pb_i}(t, t+\tau)}{\overline{Pb_j}(t, t+\tau)} = \frac{s_i}{s_j} \quad \forall\, i, j \in \{1 \cdots N\} \tag{3.1}$$

where $s_i (1 \leq i \leq N)$ is the burst loss ratio differentiation parameter of class i as set by the provider. Without loss of generality, we suppose class i has priority over class j when $i < j$.

Proportional differentiation models have two types: single-hop or end-to-end [57]. The end-to-end type needs to maintain the status of each flow at every core node. Here, we study only the single-hop type at each core node.

We assume that there are k data wavelengths for each output link in an OBS node and that there are full wavelength conversions. To find a suitable wavelength among the candidate wavelengths of an arriving burst, several data channel scheduling algorithms, such as First Fit (FF) [25], First Fit with Void Filling (FF-VF) [25],

Latest Available Unused Channel with Void Filling (LAUC-VF) [25] and Minimum Void [62], have been proposed. In this chapter, we adopt LAUC-VF as our algorithm at core nodes due to its high efficiency and simple implementation. In a classless OBS network, there is no constraint on the wavelength searching range.

We introduce the Dynamic Wavelength Selection (DWS) scheme to assign different constraints for different class services in the wavelength searching range. Our scheme assigns a candidate wavelength set for each class. For an incoming burst, the core node can schedule it to be sent only on one of the candidate wavelengths of its class.

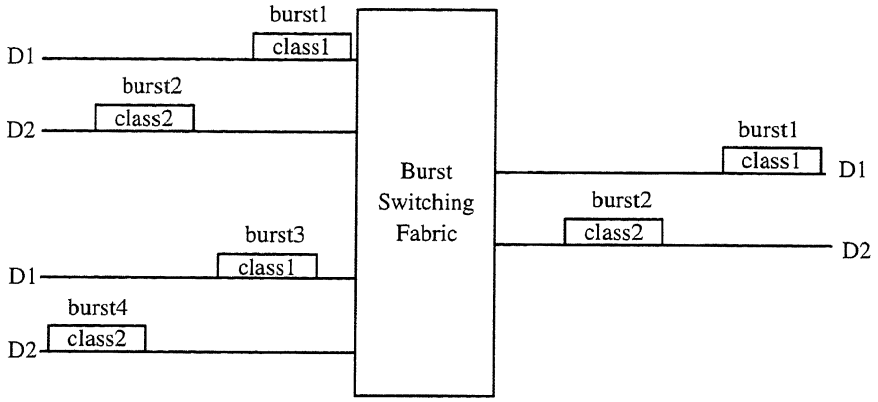
A number of wavelength-continuity-based schemes have already been proposed. In [48], each node assigns wavelengths to arriving bursts based on the wavelength priority information calculated from the wavelength utilization history. This results in a complex implementation because the utilization information of each wavelength needs to be collected in the whole network. In [60], the wavelengths assigned to each class are fixed. Because they could not vary with the traffic load, the system could not provide proportional differentiated services. In [59], each wavelength is occupied exclusively by one class. This results in waste of wavelengths and a high burst loss ratio.

To avoid the shortcomings of the above schemes, we analyze wavelength-continuity-based schemes and make several propositions by giving details about each of them on the list, which are important for our schemes but are disregarded by existing works in the above references. We suppose that $C_i(t)$ is the candidate wavelength set of class i at time t. To avoid the shortcoming of scheme in [59], we first propose:

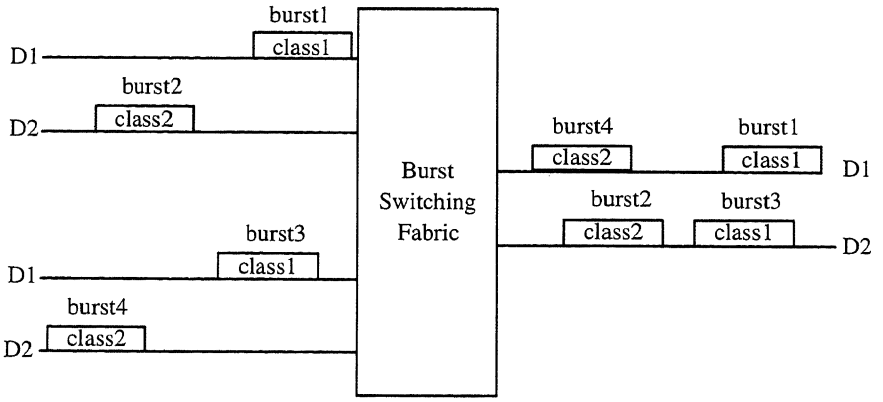
Proposition 1: *To utilize the wavelengths efficiently, the candidate wavelength set of each class $C_i(t)$ should be as large as possible.*

The void (idle period) between two scheduled data bursts on one wavelength is unused wavelength capacity. If a wavelength is occupied exclusively by one class, these voids will be wasted despite bursts of other classes arriving during these periods. To improve wavelength utilization and minimize the dropping probability, multiple classes should be able to share each wavelength.

As an example, suppose a burst switching fabric offers two wavelengths and two classes of services. Fig.3.1(a) illustrates the scenario when we assign wavelength $D1$ to $class1$ and $D2$ to $class2$, the $burst3$ and $burst4$ are dropped due to the wavelengths are occupied by early arriving $burst1$ and $burst2$. Comparing this with Fig.3.1(b), in which we assign $\{D1, D2\}$ to both $class1$ and $class2$, the droppings are avoided and the wavelength utilization is improved.



(a) C1={D1}; C2={D2}



(b) C1={D1,D2}; C2={D1,D2}

Figure 3.1: Example of Proposition 1

Proposition 2: *If $C_j(t) \subset C_i(t)$, class i has a higher priority over class j, for all $i, j = 1 \cdots N$.*

This is because that one class with a larger number of candidate wavelengths has a higher priority, because it has more and longer periods of wavelengths in the

output link for assigning arriving bursts. Different candidate wavelength sets can thus be exploited to provide different priority classes of service and different burst loss probabilities. According to Proposition 1, to maximize wavelength utilization, the highest class is assigned with the whole wavelength set and while lower classes are assigned with different subsets.

To avoid the shortcoming of the scheme of [60] and provide proportional differentiated services, the wavelengths assigned to each class should be varied with the traffic load. The following item is proposed.

Proposition 3: *When $C_j(t) \subset C_i(t)$, if the candidate wavelengths of class j increase, the burst loss ratio of class j is decreased, and the burst loss ratio of class i is increased.*

This item shows that there is no isolation between classes in our DWS model. When the number of candidate wavelengths of a class increases, the burst loss probability of that class will decrease because it has more wavelengths to assign arriving bursts. At the same time, the burst loss probabilities of other classes increase because some periods of their candidate wavelengths are occupied by that class. Thus when we adjust the candidate wavelength number of a class (except the highest class), the proportion of the burst loss ratio of that class to the burst loss ratio of the highest class will change.

Based on the above analyses, we next describe the scheme for proportional service differentiation. To simplify the scheme, we use M burst arrivals to represent the monitoring timescale τ. A FIFO is used to record the status of the recent M bursts. The depth of the FIFO is M bits and its width is $2N$(N is the number of classes) bits. The data structure of the FIFO is organized as in Fig.3.2. For entry j, if the $(N+i)$th bit equals '1', the jth arriving burst belongs to class i. When the ith bit equals '1', it means that the jth arriving burst belongs to class i and is dropped.

For class i, let $recv_i$ be the number of arriving bursts, $drop_i$ be the number of dropped bursts, Pb_i be the burst loss ratio, and a_i be the number of candidate wavelengths. Let k be total number of wavelengths for each output link to carry data burst and B_{EN}^l be the value of the lth bit of entry EN. The scheme is thus realized as the following algorithm.

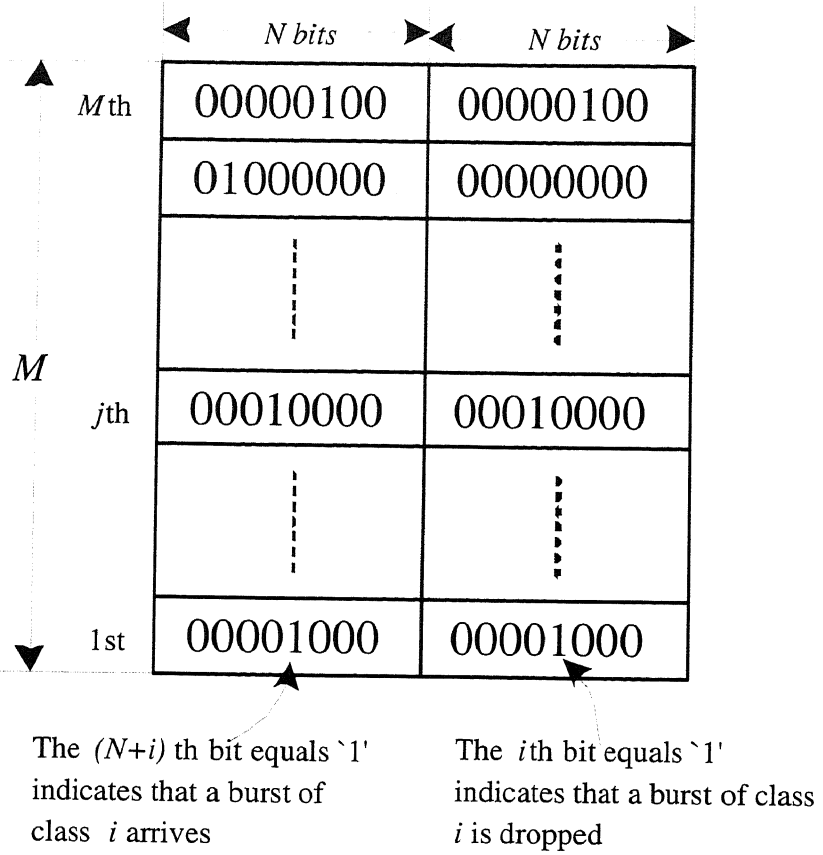


Figure 3.2: Data structure of status of bursts

Begin $a_i = k$, for $i = 1 \cdots N$

Step 1. When a burst of class i arrives

Use LAUC_VF algorithm to schedule burst,

If (schedule succeeds)

Forward it to next hop,

$recv_i++$,

Create a new entry *NEW_EN* with

the $(N+i)$th bit set 1

Else

Discard burst,

$drop_i++$, $recv_i++$

Create a new entry *NEW_EN* with both

the ith and the $(N+i)$th bit set 1

End if

Step 2. Push the entry *NEW_EN* into the FIFO
and pop the oldest entry *OLD_EN*

For i=1 to N

Begin

$recv_i = recv_i - B^{N+i}_{OLD\_EN}$,

$drop_i = drop_i - B^{i}_{OLD\_EN}$,

$Pb_i = drop_i / recv_i$,

If $Pb_i/Pb_1 > s_i/s_1$

if $(a_i < k)$ $a_i = a_i + 1$

Else if $Pb_i/Pb_1 < s_i/s_1$

if $(a_i > 0)$ $a_i = a_i - 1$

End if

End

End

In the above DWS scheme, the status of only the recent M bursts is recorded, so the oldest entry is deleted when a new burst arrives. Also, the variables such as $drop_i$ and $recv_i$ which record the information of the oldest entry are updated. When Pb_i is too high (or low), a_i will be increased (or decreased) to decrease (or increase) Pb_i and maintain the ratio of Pb_i to Pb_1. If there is no candidate wavelength for the arriving burst ($a_i = 0$), the core node will drop the burst immediately.

One advantage of this scheme is that there is no basic requirement on the number of wavelengths. It can work even when there is only one wavelength. Figure 3.3 shows an example of how one wavelength can support two classes. Suppose class 1 has a priority, so it can use this wavelength on all timescales, and class 2 uses only part of this wavelength. For simplicity, we divided the timescale into blocks. In each blocks such as $[0, T_1]$, a sub-block $[0, T_2]$ is shared by class 1 and class 2 and another sub-block $[T_2, T_1]$ is monopolized by class 1. Therefore, by changing the ratio of T_2 to T_1, we can achieve proportional differentiated services.

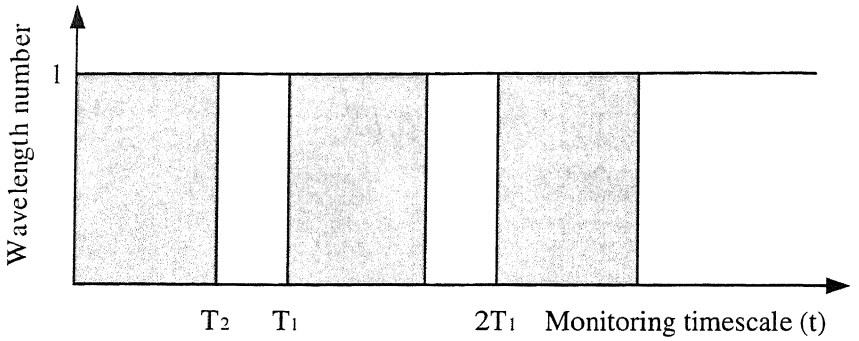


Figure 3.3: Example of one wavelength supporting two classes

3.3 Delayed Burst Assignment Scheme

Delayed Burst Assignment (DBA) is a service differentiation technique in which bursts of lower priority are processed after a delay to guarantee that bursts of higher priority are more likely to have wavelengths reserved for them. A service differentiation technique called the Generalized LAUC-VF (G-LAUC-VF) algorithm was proposed in [61]. It provides differentiated services by maintaining a BHP queue for each class and ensures that BHPs of higher priority are processed before BHPs of lower priority. We define BHP queueing delay as the period a BHP spent waiting in the BHP queue. With this algorithm, the BHP queueing delay delay for the lower priority class is uncontrollable when the higher priority traffic is heavy, and the scheme might deteriorate into a classless one when the traffic load is low (no BHP in queue when the inter-arrival time of BHPs is much longer than the processing time of a BHP). Our DBA scheme is simpler than the G-LAUC-VF algorithm, and it avoids that technique's shortcomings. Moreover, in the DBA scheme, no optical buffer but electronic buffer is needed.

DBA divides BHPs into two types: type **I** and type **II**. Type **I** has priority over type **II**. The bursts of both types have the same offset time. The scheme works as follows.

1. When a BHP of type **I** arrives, it is processed normally and is sent to the next node. When a BHP of type **II** arrives, it is queued in a BHP queue for a waiting time period T_{wait}. Because BHPs are processed electronically, we

can delay them in the electronic components such as Random Access Memory (RAM). During this waiting period, the BHPs of type **I** are processed and wavelengths are reserved by them, resulting in a reducing burst loss ratio for type **I**.

2. When T_{wait} has passed, the BHPs of type **II** are processed and the wavelengths that have not been reserved by type **I** bursts are reserved.

T_{wait} should be included in the extra offset time. If it is not included, the BHP's residual offset time will be less than the processing time of the BHP for its remaining route after it has been buffered at intermediate nodes. In this case, the corresponding DB will be dropped by the core node because the BHP could not be processed before the DB's arrival. When T_{wait} is close to zero, the whole system will deteriorate into a classless one because there is not enough time for processing type **I** BHPs. A large T_{wait}, on the other hand, will cause a large extra offset time and a large end-to-end delay.

3.4 Integrated DWS and DBA Scheme

In the DWS scheme, lower priority traffic cannot use the wavelengths assigned to the higher priority traffic even when the wavelengths are idle. This limitation wastes wavelengths. As for the problem of the DBA scheme, it is difficult to find a suitable value for "T_{wait}" by which it can provide proportional differentiated services.

To solve these problems, we integrated the DWS scheme with the DBA scheme. As described in Section 3.3, we divide bursts into two types. BHPs of type **I** bursts are processed as they arrive and BHPs of type **II** bursts are processed after they have been queued for T_{wait}. In the integrated scheme, a "type **I**" burst refers to each incoming burst, which is scheduled in accordance with the DWS scheme. A "type **I**" burst can be scheduled only to be sent on a wavelength within the candidate wavelength set of its class. If it can be scheduled, its BHP is processed normally and sent to the next hop. If it can not be scheduled when its BHP arrives and if its residual offset time is shorter than a threshold (the minimum time for the remaining route), it is dropped immediately. If its residual time is longer than the threshold,

it is regarded as a "type **II**" burst, whose BHP is to be queued for T_{wait} before it is sent on a wavelength within its rescheduling wavelength set.

In addition to the candidate wavelength set $C_i(t)$ for class i, we also assign a rescheduling candidate wavelength set $C_i^r(t)$. We assign the rescheduling wavelength set with the whole wavelength set when the candidate wavelength is not empty. The rescheduling wavelength set is dynamically adjusted using the flowchart in Fig.3.4. Note that a_i^r denotes the wavelength number of $C_i^r(t)$.

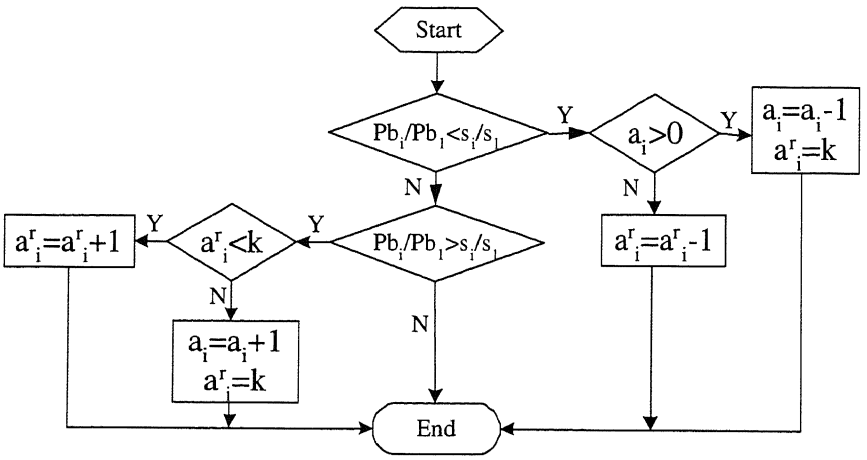


Figure 3.4: Flowchart for assigning rescheduling wavelength set

Bursts will be dropped if no suitable wavelength in $C_i^r(t)$ can be found. Figure 3.5 summarizes the operations of the integrated scheme.

As we assume there is no optical buffer in core nodes, *end-to-end delay = end-to-end propagation delay + end-to-end BHP queueing delay.* Here, the end-to-end BHP queueing delay is defined as the sum of the BHP queueing delays at core nodes during transmission, which should be included in the extra offset time. Let N_{max} be the maximum hop number and T_{offset}^{extra} be the maximum required extra offset time in the OBS network. $T_{offset}^{max} = T_{wait} \times N_{max}$ (i.e., the worst case whereby lower priority BHPs buffered for T_{wait} at each core node).

T_{wait} is a parameter that affects the scheduling performance. When T_{wait} is close to zero, the system will deteriorate into a DWS system. A large T_{wait}, on the other hand, will cause a large extra offset time and a large end-to-end delay. There is a tradeoff between burst loss performance and end-to-end delay. We will study the

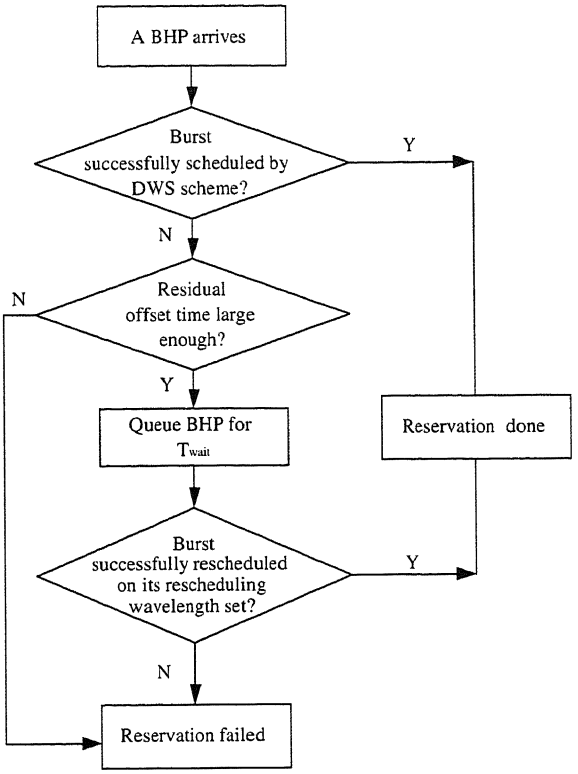


Figure 3.5: Simplified flowchart of integrated scheme

impact of T_{wait} in detail in Section 3.5 by simulations.

To prevent the end-to-end delay from becoming too large, we introduce a new parameter $N_r(N_r \leq N_{max})$, the maximum number of times for a BHP to be queued during transmission. When the number of times a BHP has been queued reaches N_r, the burst will be dropped immediately if it can not find a suitable wavelength in its candidate wavelength set. The required extra offset time is thus limited to: $T_{offset}^{max} = T_{wait} \times N_r$.

3.5 Numerical Results and Discussion

We use OPNET as a simulation tool to study the performance of our schemes and compare them with existing dropping schemes. To check their efficiency, especially when the offset times are varied during transmission, we simulate a multiple hop network with a ring topology as shown in Fig.3.6. The shortest-path-first routing method is used to establish a route between each pair of edge nodes $Ei(i$=1 to 17),

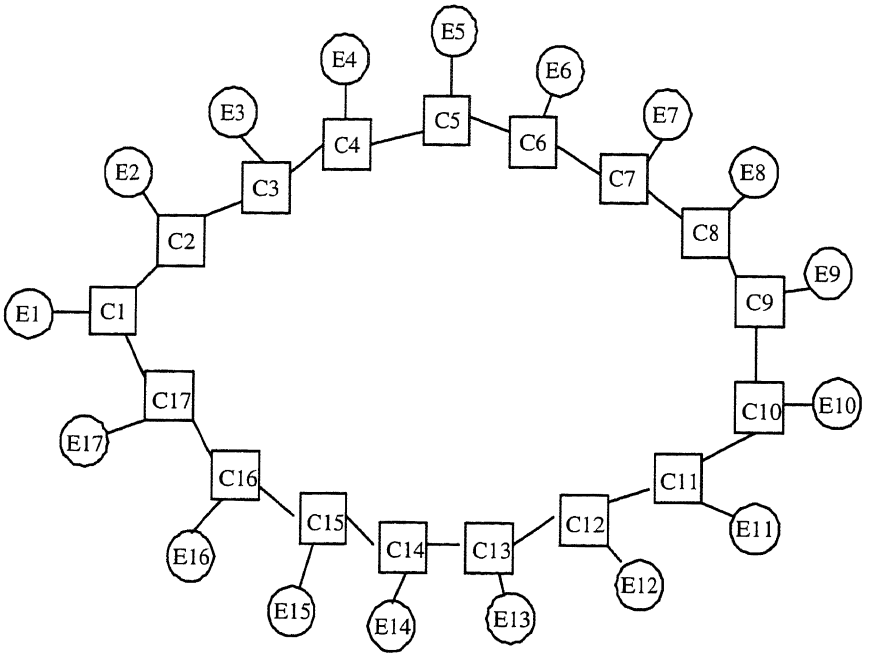


Figure 3.6: Simulation network model for performance evaluations

and the maximum hop distance is 10. Bursts are generated at each edge node Ei. We assume that the burst inter-arrival time follows an exponential distribution and the burst size follows a normal distribution. Note that these assumptions are the same as the ones in [39,40,46]. The average burst size is 50 Kbytes. All bursts are assumed to have the same initial offset time (default value is 5ms if not specified), which is small enough even for real-time applications. For a core node $Ci(i$=1 to 17), we assume that each output link consists of 16 wavelengths with a transmission rate of 1 Gbps per wavelength. Although there is no optical buffer in the simulations, there are full wavelength conversions at the OBS nodes. The basic processing time for BHP at each core node is set to be 0.1 ms. To investigate the service differentiation, we consider four classes, a load distribution of $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4$, and proportional parameters of s_1=1.0, s_2=2.0, s_3=4.0, and s_4=8.0.

To study the calculation accuracy of the burst loss ratio, we do 10 simulations for the DWS scheme (The results for the integrated scheme not shown here are similar) at different monitoring time scales respectively. For each monitoring time scale, we let Max_{pb} be the maximum value of average burst loss ratios, Min_{pb} be the minimum value of average burst loss ratios and, Avg_{pb} be the sample value. Figure

3.7 shows the impact of monitoring time scale on the calculation errors of burst loss ratios for different traffic loads. Here, *Calculation error*=$(Max_{pb} - Min_{pb})/Avg_{pb}$. The figure shows that the calculation error decreases as the monitoring time scale increases. When the monitoring time scale is larger than 6×10^6 burst arrivals, the calculation error is very low (under 0.01) even for a low traffic load (0.3). In the remainder of this chapter, the monitoring time scale is set to 6×10^6 burst arrivals. If we denote traffic load as ρ, the chosen monitoring time scale is about $150/\rho$ seconds. ($\tau = 6 \times 10^6 \times$ *average burst length/(bandwidth of each wavelength* $\times$*wavelength number* $\times\rho$)=$150/\rho$ *seconds.*)

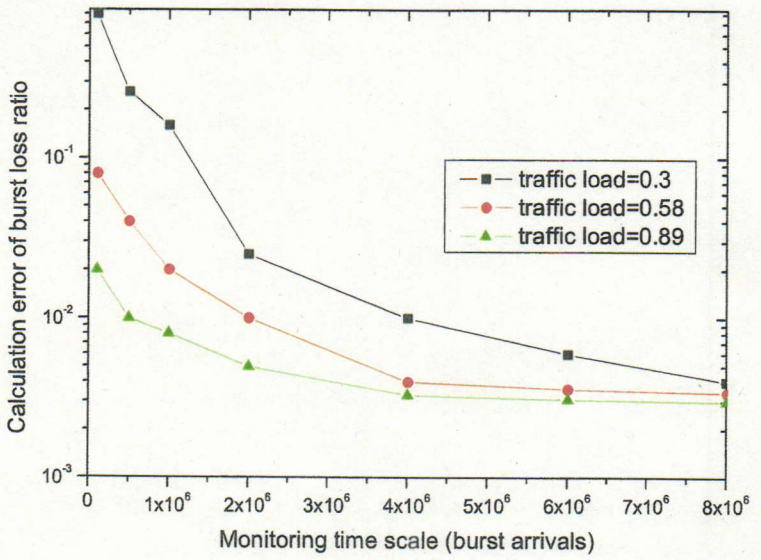


Figure 3.7: Calculation error of burst loss ratio versus monitoring time scale

Table 3.1 shows the proportions of simulated burst loss ratios for the DWS scheme. The results show that the proportions (Pb_i/Pb_1, i=2, 3, 4) of the burst loss ratios of classes 1 to 4 are close to the ratios (s_i/s_1) of predefined parameters and are independent of the traffic load (ρ). Hence, the DWS scheme is proved to achieve proportional differentiated services.

Figure 3.8 compares the burst loss ratio of an intentional dropping scheme (plotted as Drop in the figure) with that of the DWS scheme. The burst loss ratios

Table 3.1: Proportions of simulated burst loss ratios for DWS scheme (s_1:s_2:s_3:s_4=1:2:4:8)

ρ	Pb_2/Pb_1	Pb_3/Pb_1	Pb_4/Pb_1
0.30	2.02	4.07	7.95
0.40	1.97	3.94	7.94
0.50	1.98	3.95	7.95
0.58	1.98	3.96	7.99
0.68	1.98	3.98	7.96
0.76	1.98	3.92	7.99
0.83	2.00	4.00	8.00
0.89	2.02	4.01	8.01
0.93	2.00	4.00	7.96
0.97	1.99	4.00	7.95

increase when the traffic load increases. Intentional dropping and DWS each provide proportional service differentiation for multi-class traffic. However, the DWS scheme has better burst loss performance. For example, when the traffic load changes from 0.3 to 0.8, the burst loss ratio of DWS is only about 50% of that of intentional dropping.

Regarding the performance of the integrated scheme, Figs.3.9 and 3.10 show the impact of N_r and T_{wait} on the average burst loss ratios for a low traffic load (0.3) and for a high traffic load (0.89), respectively. In the figures, the curve "aL" denotes the burst loss ratio when T_{wait} is set to a times the average burst length (L). The initial offset time is set to 30 ms. Note that $N_r = 0$ corresponds to the DWS scheme. These results indicate that as N_r and T_{wait} increase, the burst loss ratio decreases. However, when T_{wait} exceeds the average burst length duration and N_r exceeds 4, the burst loss ratios decrease slowly and eventually become almost the same; the results for other traffic loads, not shown here, are similar. Thus, in the following simulations, we set 0.4 ms (*average burst length duration=average burst length/bandwidth=50 Kbytes/1 Gbps=0.4 ms*) and 4 as the default values of T_{wait}

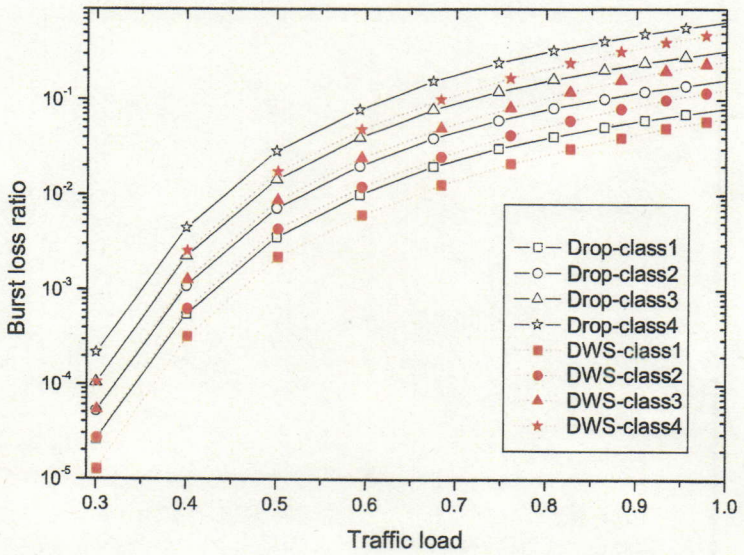


Figure 3.8: Burst loss ratio for Drop and DWS schemes

and N_r.

The results in Table 3.2 show that the proportions of different classes are close to the ratios of predefined parameters and are independent of the traffic load for the integrated scheme. Therefore, the integrated scheme achieves proportional differentiated services for multi-class traffic. Figure 3.11 compares the burst loss performances of the DWS and the integrated schemes. The results show that the integrated scheme has better burst loss performance than does the DWS scheme because that it does not waste wavelengths. For example, the burst loss ratio of the integrated scheme is about 50% that of the DWS scheme when the traffic load changes from 0.3 to 0.6. This is because the BHP of a lower priority class in the integrated scheme will be buffered at the core node when it cannot find an available wavelength and has an opportunity to reschedule its burst to the wavelengths that have been assigned to higher priority classes but have not yet been reserved.

Figure 3.12 shows the end-to-end BHP queueing delay normalized to the average burst length duration for each class during transmission. We can see that the lower priority bursts have larger queueing delay than do the higher priority bursts. For

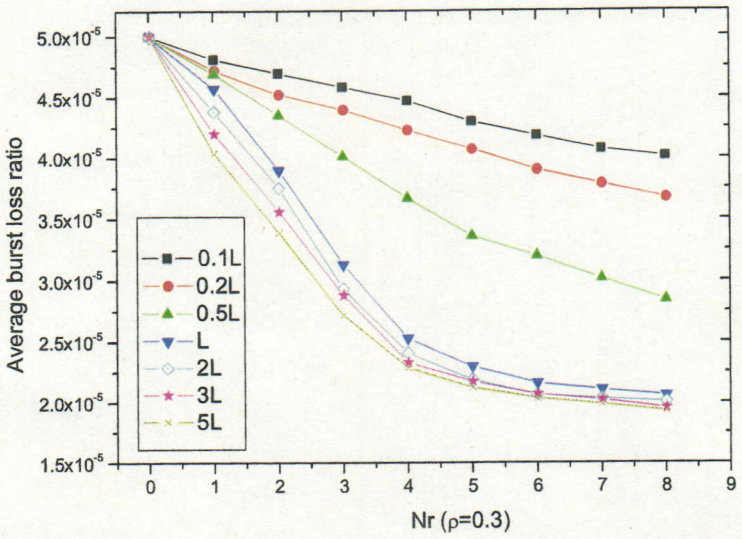


Figure 3.9: Average burst loss ratios versus N_r for integrated scheme with traffic load =0.3

example, the queueing delay is several microseconds for class 2, tens of microseconds for class 3, and hundreds of microseconds for class 4. The simulation results show that although the integrated scheme improves the burst loss performance at the expense of increasing extra offset time, the increase of end-to-end delay is very small (at most hundreds of microseconds) and would be negligible for real applications.

Figures 3.13 and 3.14 illustrate the average burst loss ratio and normalized throughput at each link. *Normalized throughput = throughput/(link capacity)*. Because each burst does not have the same size, the burst loss ratio differs from the bit loss ratio. Thus, the normalized throughput can be used to evaluate the network performance from another angle than burst loss ratio. The integrated scheme has the highest normalized throughput and the lowest average burst loss ratio, whereas the intentional dropping scheme has the lowest normalized throughput and highest loss ratio at all load levels.

Figure 3.15 compares the average end-to-end burst loss ratios. The integrated scheme has not only the best single-hop performance and but also the best end-to-end burst loss performance.

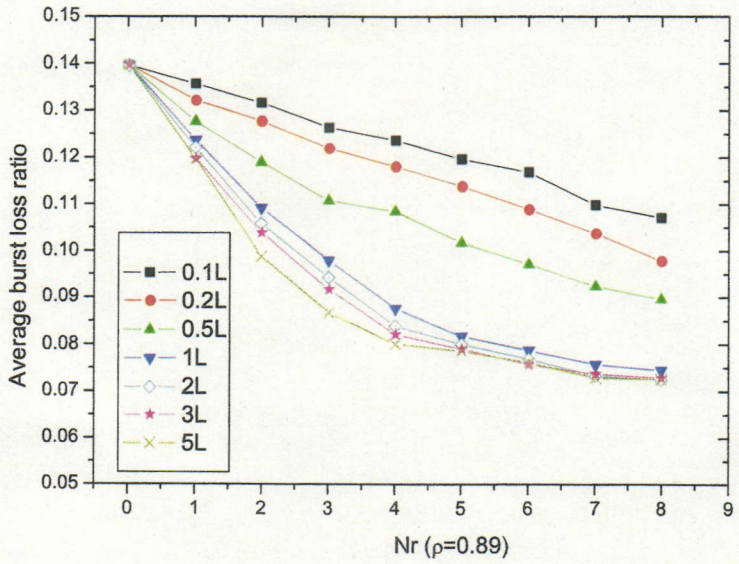


Figure 3.10: Average burst loss ratios versus N_r for integrated scheme with traffic load=0.89

Figure 3.16 shows how the per-hop control affects the end-to-end burst loss ratios of each class for the integrated scheme. The simulation results show that although the integrated scheme cannot guarantee end-to-end proportional differentiated services, it can guarantee that a higher priority class receives better services than does a lower priority class. How to provide end-to-end proportional differentiated services will be the focus of our future study.

3.6 Summary

In this chapter, we described how to provide proportional differentiated services in an OBS network. First, we introduced a priority scheme, called dynamic wavelength selection (DWS), which can efficiently support proportional differentiated services in bufferless OBS networks. We compared the performances of this scheme and the intentional dropping scheme in a series of simulations. The simulation results showed that the DWS scheme has better performance in terms of burst loss ratio

Table 3.2: Proportions of simulated burst loss ratios for integrated scheme (s_1:s_2:s_3:s_4=1:2:4:8)

ρ	Pb_2/Pb_1	Pb_3/Pb_1	Pb_4/Pb_1
0.30	2.03	3.99	7.99
0.40	1.99	4.00	7.98
0.50	2.02	4.02	8.03
0.58	2.02	4.02	8.00
0.68	1.99	3.98	7.96
0.76	2.00	3.99	7.97
0.83	1.99	3.99	7.97
0.89	2.00	4.02	8.01
0.93	1.98	3.99	7.96
0.97	1.99	4.00	8.03

and throughput.

We also proposed a delayed burst assignment scheme and integrated it with the DWS scheme to support proportional differentiated services in OBS networks. The integrated scheme proved to have the best performance. Through simulation, we also found that when the BHP waiting time period (T_{wait}) and the maximum number of times for a BHP to be queued (N_r) exceed some values, the performance is improved smoothly. These results could prevent the end-to-end delay from becoming too large.

As shown in the Appendix, unlike the existing priority schemes such as BA-WP scheme, our integrated scheme does not need any complex burst segmentation or wavelength preemption support, so it is especially suitable for OBS networks because of its simple implementation. Moreover, it provides controllable and predictable proportional differentiated services for each class. Our future research will deal with how to provide end-to-end service differentiation.

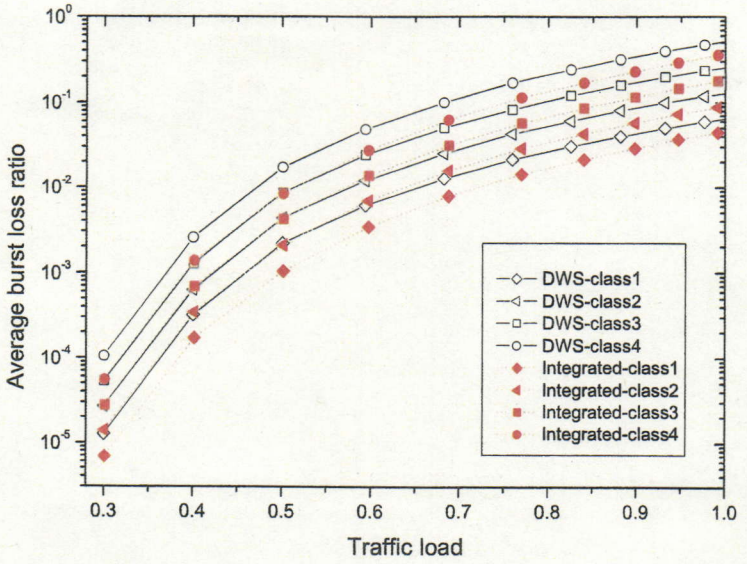


Figure 3.11: Burst loss ratios for DWS and integrated schemes

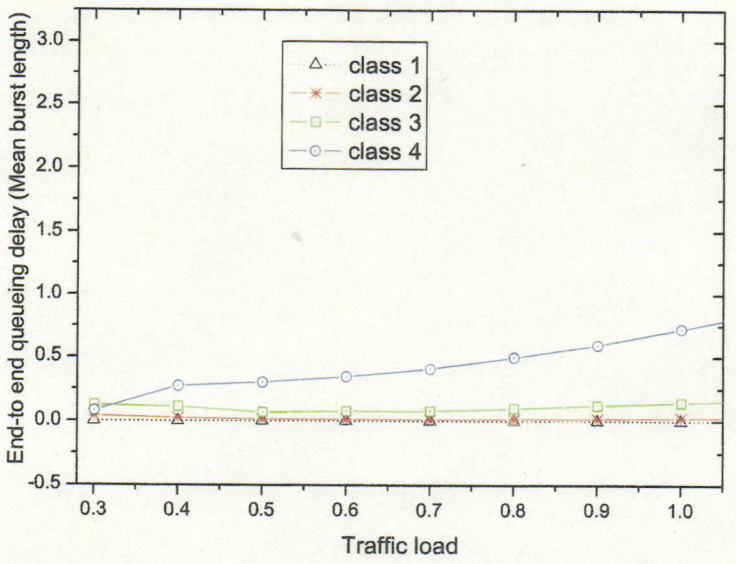


Figure 3.12: End-to-end BHP queueing delay for integrated scheme

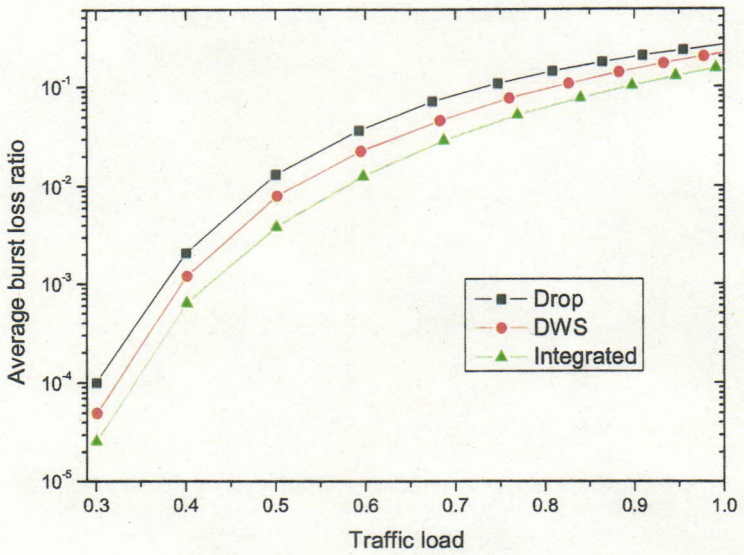


Figure 3.13: Average burst loss ratios for DWS, Drop and integrated schemes

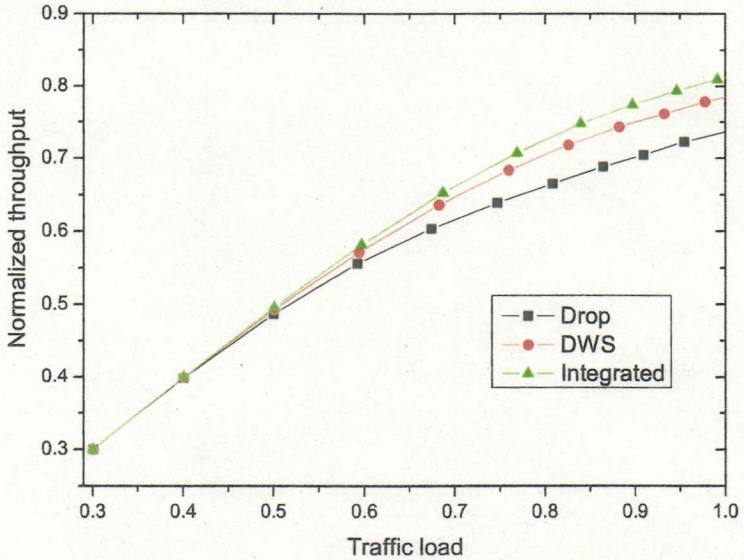


Figure 3.14: Normalized throughput for DWS, Drop and integrated schemes

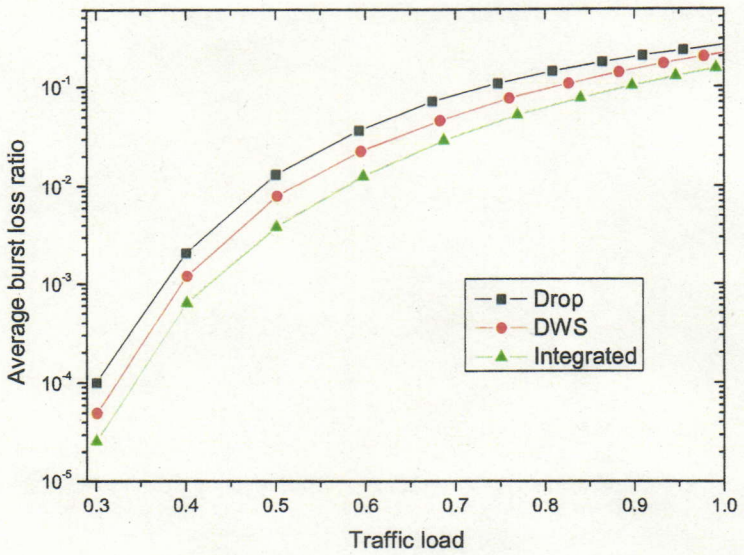


Figure 3.13: Average burst loss ratios for DWS, Drop and integrated schemes

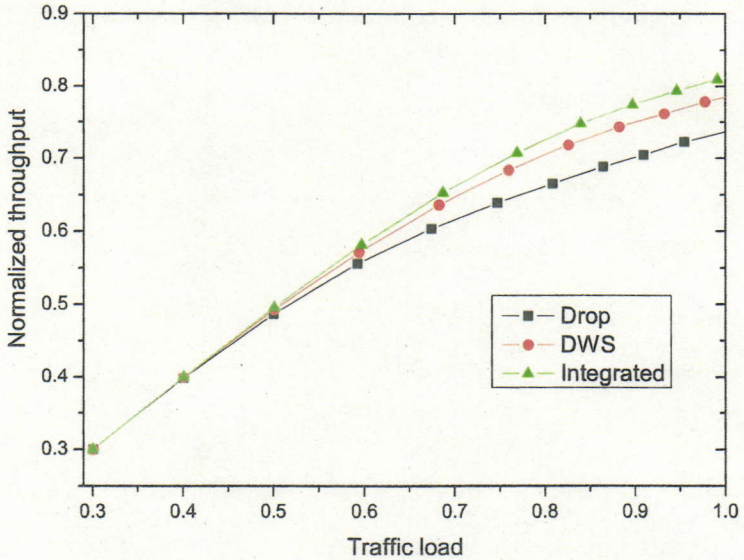


Figure 3.14: Normalized throughput for DWS, Drop and integrated schemes

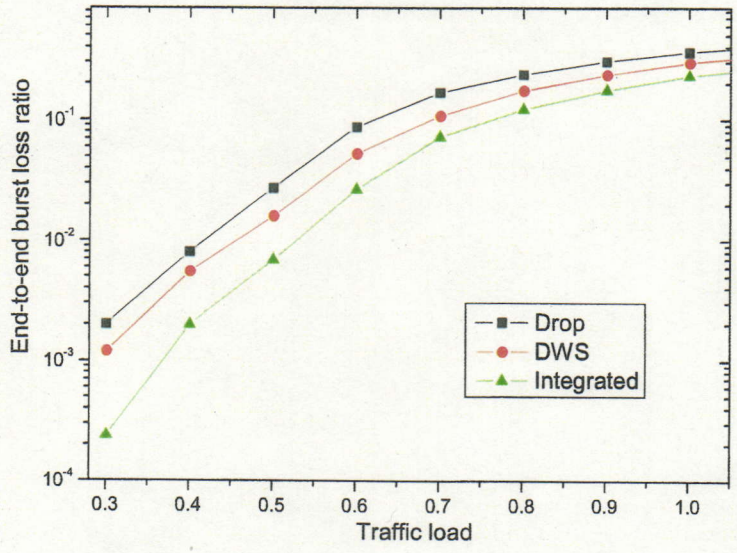


Figure 3.15: End-to-end burst loss ratios for DWS, Drop and integrated schemes

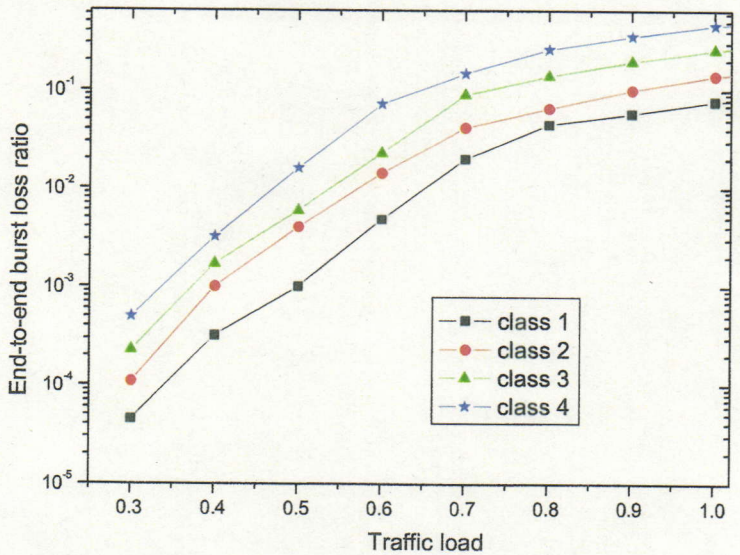


Figure 3.16: End-to-end burst loss ratios for integrated scheme

Appendix: Comparison of integrated scheme with BA-WP scheme

In this appendix, we compared burst loss performance between the BA-WP and the integrated schemes. For simulation, we assume the bursts are generated at the edge nodes following an exponential distribution for burst inter-arrival time and Normal distribution for burst size, the average burst size is 50K bytes. There is no optical buffer but full wavelength conversions at the core node. Each output link of the core nodes consists of 16 wavelengths with a transmission rate of 1Gbps per wavelength. We consider three classes of traffic, the traffic ratios are assumed to be 20%, 20%, and 60% for classes 1, 2 and 3 respectively. The proportional parameters are set as follows: s_1=1.0, s_2=2.0, s_3=4.0.

A. Simulations with star topology

The first simulations are conducted on a star topology as shown in Fig.3.17. Bursts are generated in each edge node and are destined to all other edge nodes.

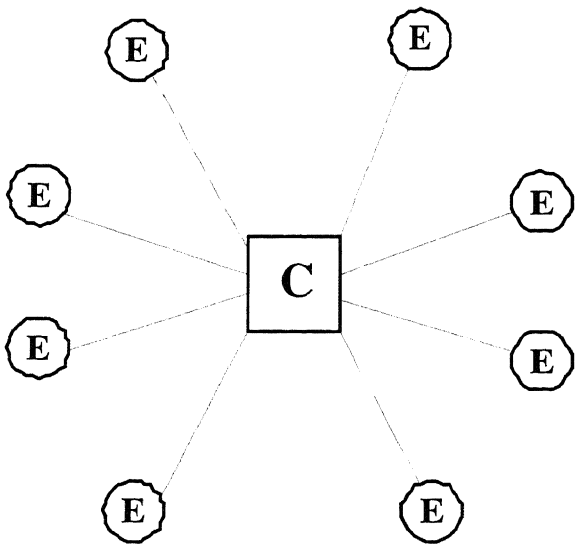


Figure 3.17: Star topology of eight edge nodes and one single core node

It is shown in Fig.3.18 that the integrated scheme has a similar burst loss performance compared to BA-WP scheme. This means that our proposed integrated

scheme can achieve the similar burst loss performance under star topology network without generating control packets at intermediate nodes.

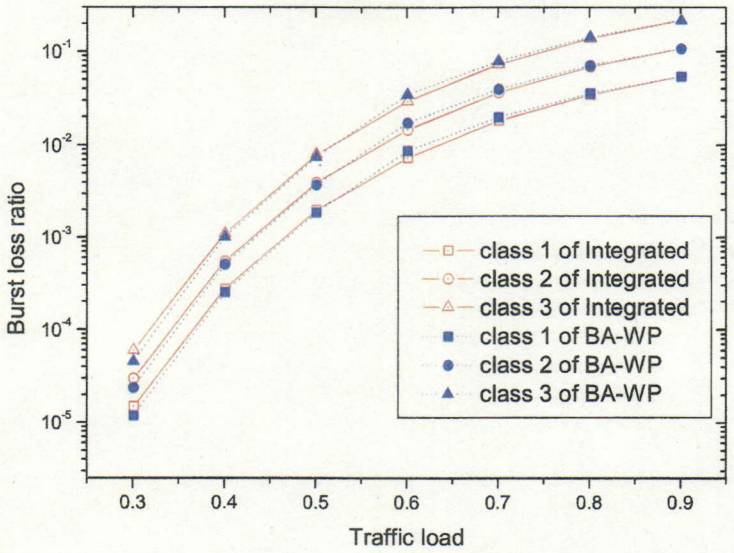


Figure 3.18: Burst loss ratios for BA-WP and integrated schemes

B. Simulation with binary-star topology

We also do simulations with a binary-star topology as shown in Fig.3.19. Bursts are generated in each edge node and are destined to all other edge nodes.

As shown in Fig.3.20, the integrated scheme has a little better burst loss performance than BA-WP scheme does. This is because in BA-WP scheme, the BHP of preempted burst has been already sent to the next core node before preemption happens. The preempted burst will contend the wavelengths with other normal bursts at the next hop node and may succeed in reservation before the cancellation control packet arrives. Although the cancellation control packet can release the wavelength reservation of the preempted burst, some normal bursts have already been dropped before the releasing of wavelengths. This will result in wavelength waste because preempted bursts occupy the wavelengths that should be assigned to normal bursts.

Based on the above analyses, we can draw the conclusion that our proposed integrated scheme has two advantages compared to BA-WP scheme:

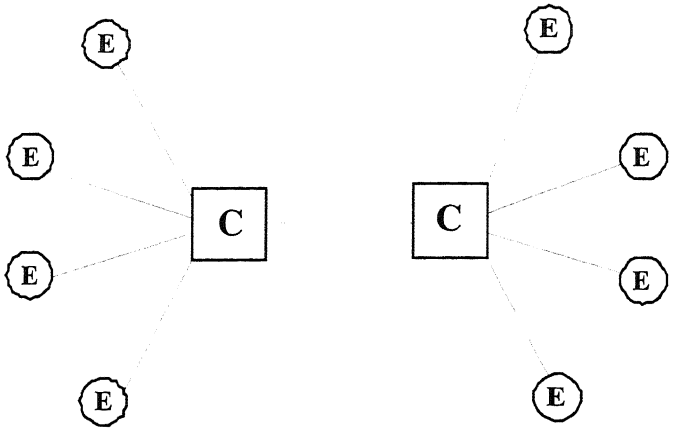


Figure 3.19: Binary-star topology of eight edge nodes and two core nodes

1. The integrated scheme does not need to generate any special packets and needs to maintain only a few parameters at core nodes. So it is especially suitable for OBS networks because of its simple implementation.

2. The performance of integrated scheme will not deteriorate at a multi-hop network while BA-WP scheme does.

The integrated scheme also has two shortcomings comparing with BA-WP scheme:

1. It needs more initial offset time and will results in a longer end-to-end delay.

2. It needs electronic buffering at core routers.

We can deduce that our proposed integrated scheme has better performance than BA-WP scheme in a multi-hop network such as Fig.3.6 because its performance does not deteriorate at a multi-hop network like BA-WP scheme does.

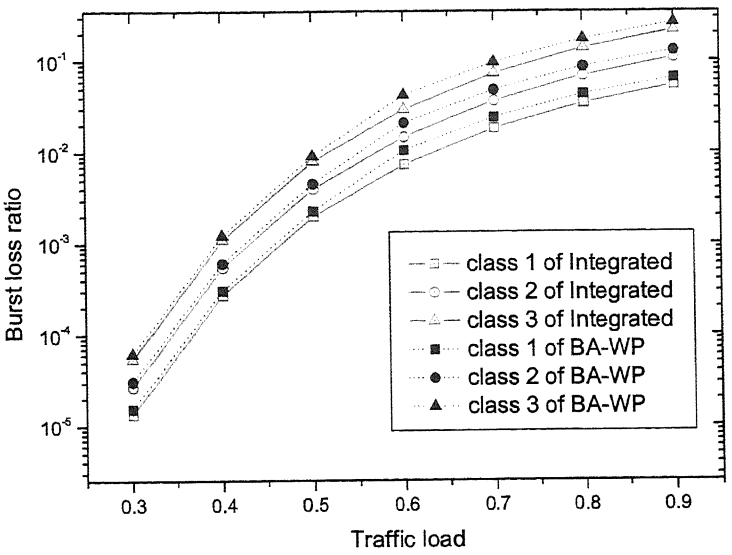


Figure 3.20: Burst loss ratios for BA-WP and integrated schemes

Chapter 4

Proportional Differentiated Services with Absolute Constraints

4.1 Introduction

In Chapter 3, we have introduced several schemes for *relative QoS* model, in which the QoS of one class is defined relatively in comparison to other classes. For example, a higher priority burst is guaranteed to experience lower loss probability than a lower priority burst. However, no upper bound on the loss probability is guaranteed for the higher priority burst. There are many types of traffic that require strict QoS guarantees. For example, a data transfer operation cannot bear packet loss ratio exceeding a certain threshold.

The *absolute QoS* model provides a worst-case QoS guarantee to applications. This kind of hard guarantee is essential to support applications with delay and loss ratio constraints, such as multimedia and mission-critical applications.

Q.zhang et al. [68] proposed an early dropping scheme to drop lower priority bursts to assure higher priority bursts have more probability in reserving wavelength to meet the absolute constraint of higher priority class. However, the absolute QoS model is a crucial QoS model, there is no differentiation among the classes when the traffic load is low. J.Phuritatkul [91] proposed an Adaptive Wavelength Preemption

(AWP) to guarantee absolute QoS. In this approach, higher priority bursts can preempt lower priority bursts adaptively in a probabilistic manner to meet absolute constraints.

Although it has been accepted that proportional differentiated services with absolute constraints is important [69,70], there is no scheme in the literature to provide proportional differentiated services with absolute constraints in OBS.

In this chapter, we focus on the issue of providing proportional differentiated services with absolute constraints. For a system supports both absolute constraints and proportional constraints, we define absolute QoS constraints have higher priority over proportional QoS constraints. When there are conflicts between constraints, the constraints with lower priorities will be relaxed [69, 70].

4.2 Supporting Joint QoS with DWS Scheme

In this section, we extend the DWS scheme introduced in Chapter 3 to support proportional differentiated services with absolute constraints. Besides the parameters introduced in Chapter 3, we add a new parameter Pb_i^{max}, the maximum burst loss ratio at each node for class i. The scheme is thus realized as the following algorithm:

Begin $a_i = k$, for $i = 1 \cdots N$

Step 1. When a burst of class i arrives

Use LAUC_VF algorithm to schedule burst,

If (schedule is success)

Forward it to next hop,

$recv_i++$,

Create a new entry *NEW_EN* with

the $(N+i)$th bit set 1

Else

Discard burst,

$drop_i++$, $recv_i++$,

Create a new entry *NEW_EN* with

the ith and the $(N+i)$th bit set 1

End if

Step 2. Push the entry *NEW_EN* into the FIFO
and pop the oldest entry *OLD_EN*

For i=1 to N

Begin

$recv_i = recv_i - B^{N+i}_{OLD\_EN}$,

$drop_i = drop_i - B^{i}_{OLD\_EN}$,

$Pb_i = drop_i / recv_i$,

If $Pb_{i-1} > Pb^{max}_{i-1}$

$if(a_i > 0)\ a_i = a_i - 1$

Else if $Pb_i/Pb_1 < s_i/s_1$

$if(a_i > 0)\ a_i = a_i - 1$

Else if $Pb_i/Pb_1 > s_i/s_1$

$if(a_i < k)\ a_i = a_i + 1$

End if

End

End

4.3 Supporting Joint QoS with Integrated Scheme

In DWS scheme, we assign some wavelengths only to the higher priority class which could not be used by bursts of the lower priority class. If there is no higher priority burst arriving, these wavelengths will be wasted.

In this section, we extend the integrated scheme of DWS and DBA (refer to Chapter 3) to support joint QoS. As shown in Fig.4.1, the core node first use DWS scheme to process BHP and assign wavelength for its burst. If the arriving BHP could not find an available wavelength for its burst, the BHP will be buffered for T_{wait}. So it has an opportunity to reschedule its burst to the wavelengths in its rescheduling wavelength set in which the wavelengths have been assigned to the higher priority class but haven't been reserved yet. The rescheduling wavelength set is dynamically adjusted by the flowchat in Fig.4.2. In which the absolute constraints

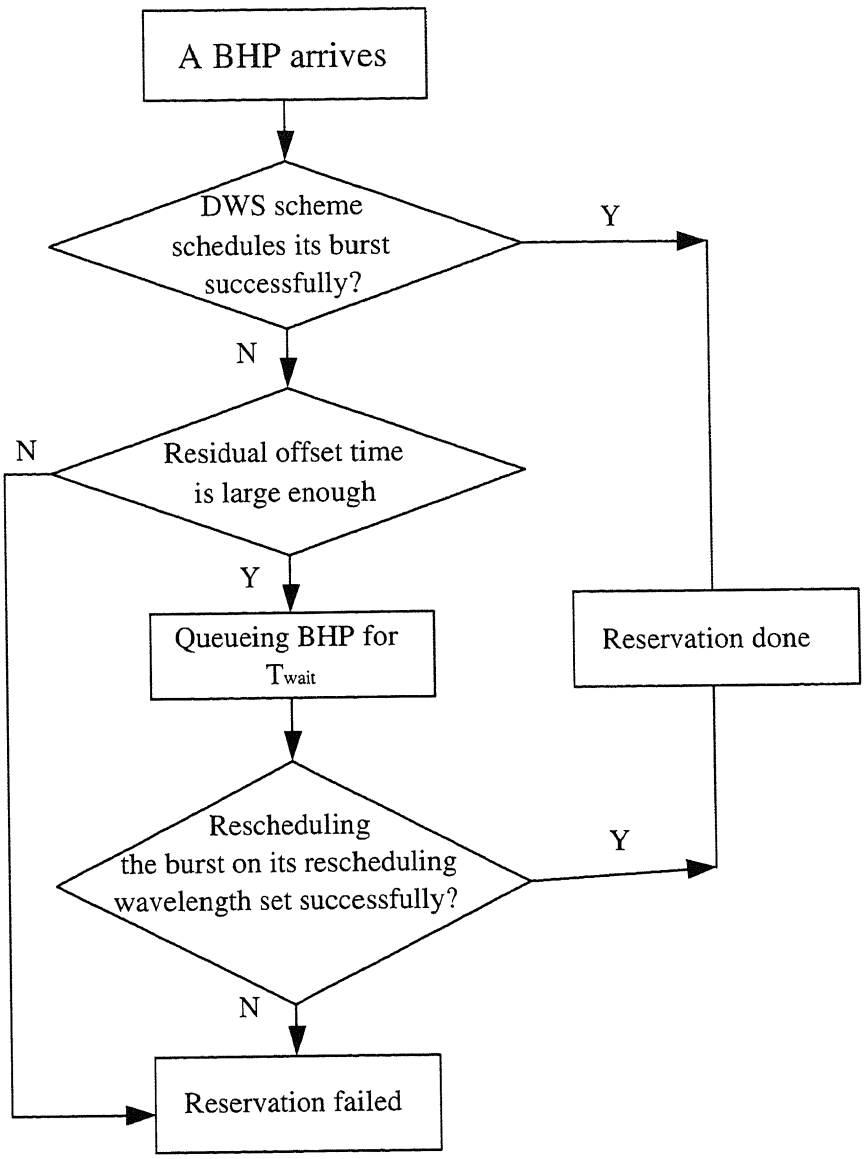


Figure 4.1: Simplified flowchart of integrated scheme

has priority over proportional constraints.

To prevent the end-to-end delay from being too large caused by the BHP buffering at immediate nodes, we also introduce a new parameter $N_r (N_r \leq N_{max})$, the maximum buffering times for each BHP during transmission. When the total buffering times of a BHP reaches N_r, the burst will be dropped when it can not find a suitable wavelength in its candidate wavelength set for the rest of its journey. The required extra offset time is thus limited to: $T_{offset}^{max} = T_{wait} \times N_r$. The value of T_{wait} and N_r are set to the same values as in Chapter 3 in this chapter.

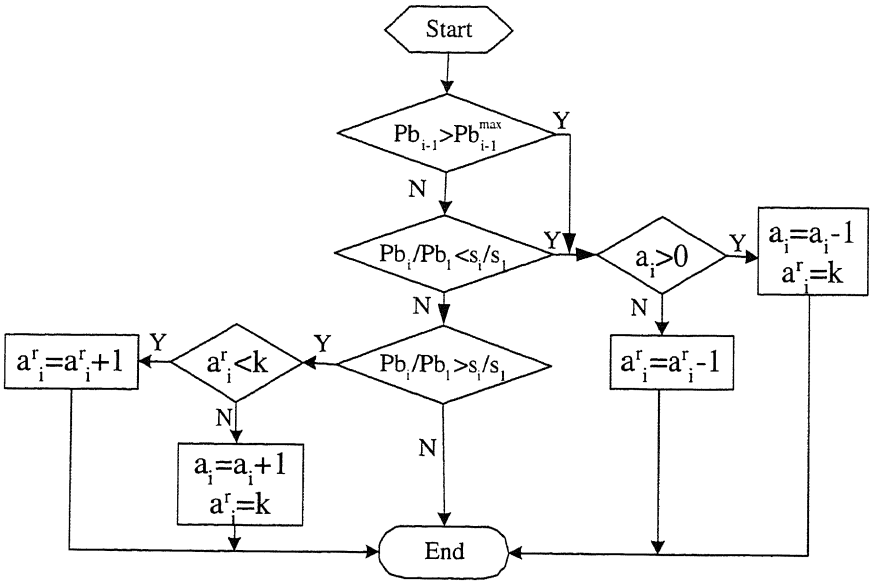


Figure 4.2: Flowchart for rescheduling wavelength set

4.4 Numerical Results and Discussion

In our simulations, we use the same ring topology (shown in Fig.4.3) and environment as in Chapter 3. We consider three classes of traffic, class 1 is the highest one, followed by class 2, then class 3. The traffic ratios are assumed to be 10%, 10%, and 80% for classes 1, 2 and 3 respectively. For proportional constraints, we set the proportional factors $s_1 = 1.0, s_2 = 2.0, s_3 = 4.0$. The absolute loss guarantee for class 1 is $P_1^{MAX} = 0.002$, for class 2 is $P_2^{MAX} = 0.008$. We set T_{wait}=0.4 ms (mean of burst transmission time) and $N_r = 4$ in this chapter.

Figure 4.4 plots the burst loss ratios of class 1 and class 2 for our proposed DWS and integrated schemes. We can see that the burst loss ratios are lower than the absolute constraints we predefined for both schemes. Moreover, the integrated scheme has a better burst loss performance than the DWS scheme does at low traffic loads.

Table 4.1 shows the proportions of simulated burst loss ratios for the DWS and the integrated schemes. The results show that the proportions (Pb_i/Pb_1, i=2,3) of the burst loss ratios of classes 1 to 3 are close to the ratios (s_i/s_1) of predefined parameters and are independent of the traffic load (ρ). Hence, both the DWS and the integrated schemes are proved to achieve proportional differentiated services at

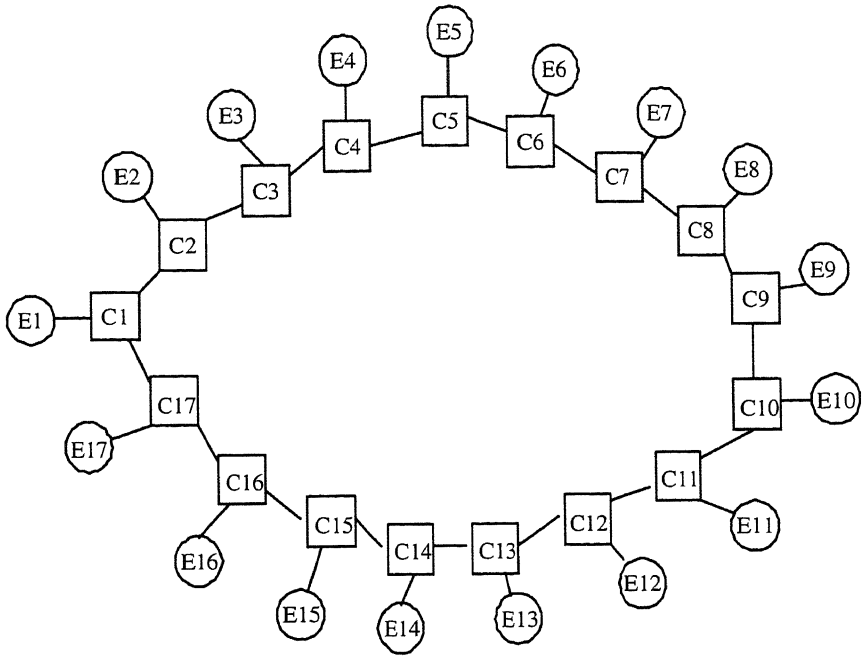


Figure 4.3: Simulation network topology for performance evaluations

low traffic loads.

Figure 4.5 compares the burst loss ratio of intentional dropping scheme (plotted as Drop in the figure) with those of the DWS scheme and the integrated scheme. We can see that the integrated scheme provides joint service differentiation for multi-class traffic and has best burst loss performance than the other two schemes because that it does not waste wavelength. For example, the burst loss ratio of the integrated scheme is about 55% that of the DWS scheme and is 30% that of the Drop scheme when the traffic load is 0.3. This is because the BHP in the integrated scheme will be buffered at the core node when it could not find an available wavelength and has an opportunity to reschedule its burst to the wavelengths which have been assigned to the higher priority class but haven't been reserved yet.

Figure 4.6 illustrates the average burst loss ratio of these approaches. Among the QoS schemes, it can be seen that the integrated scheme has the least average burst loss ratio at all load levels. For intentional dropping scheme, although it has lower average burst loss ratio than DWS scheme at high traffic loads, it could not support absolute guarantees.

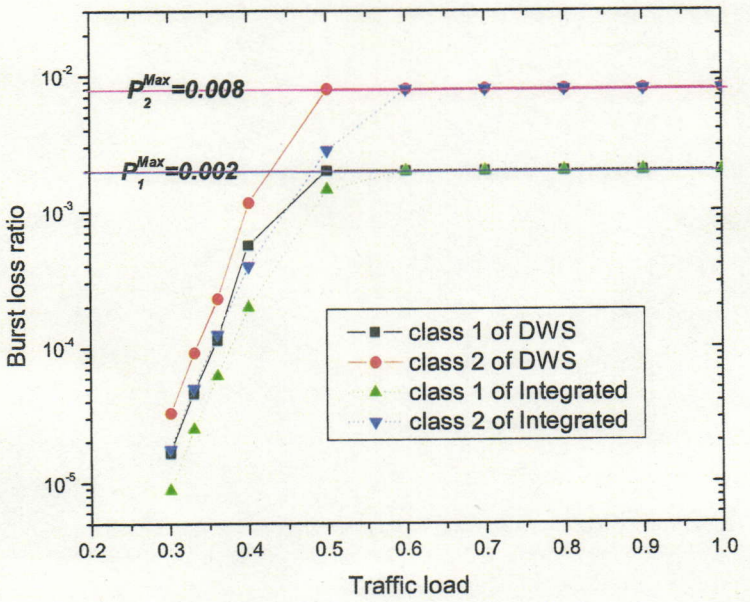


Figure 4.4: Absolute constraints for DWS and integrated schemes

Table 4.1: Proportions of simulated burst loss ratios for DWS and integrated schemes (s_1:s_2:s_3=1:2:4)

	DWS		Integrated	
ρ	Pb_2/Pb_1	Pb_3/Pb_1	Pb_2/Pb_1	Pb_3/Pb_1
0.30	2.00	4.00	1.99	4.00
0.33	2.00	4.01	2.02	3.99
0.36	2.01	3.94	1.99	3.99
0.40	2.06	4.06	2.00	4.00
0.50			1.96	3.89

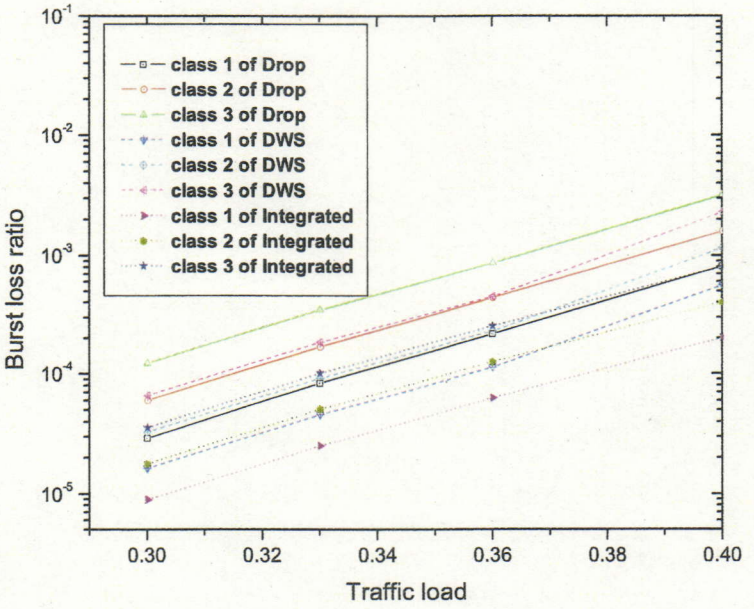


Figure 4.5: Burst loss ratios for different schemes at low traffic loads (0.3 to 0.4)

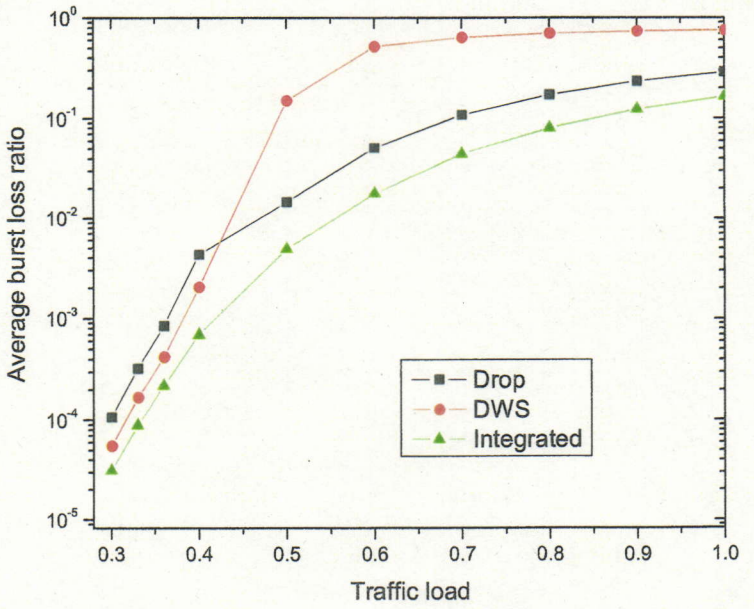


Figure 4.6: Average burst loss ratios for Drop, DWS and integrated schemes

4.5 Summary

In this chapter, we described how to provide proportional differentiated services with absolute constraints in an OBS network. Analysis and simulation results show that the extension of DWS and the integrated scheme of Chapter 3 could not only provide proportional differentiated services at low traffic loads but also absolute differentiated services at high traffic loads..

Chapter 5

Traffic-smoothing Burst Assembly Methods

5.1 Introduction

A challenging issue in OBS is how to assemble IP packets into bursts at ingress nodes. Basically, there are two assembly schemes [25]: *threshold-based* and *timer-based*. In a threshold-based scheme, a burst is created and sent into the optical network when the total size of the packets in the queue reaches a threshold value L_b. In a timer-based scheme, a timer is started at the initialization of the assembly. A burst containing all the packets in the buffer is generated when the timer reaches the burst assembly period T_b. For a *hybrid* algorithm, a burst will be generated when either the timer exceeds T_b or the burst size reaches L_b.

As we assume there is no buffer in OBS networks, a burst loss event will occur if multiple bursts from different input ports are destined for the same output port at the same time. The burst arrival process is determined by the traffic characteristics such as the burst inter-arrival time and the burst length distributions, which are dependent on the burst assembly strategy. For the timer-based assembly algorithm, the burst size is unlimited when too many packets arrive suddenly. A larger burst is more likely to be blocked or to block other bursts during transmission. For the threshold-based and hybrid assembly algorithms, a large number of bursts will be generated and injected into the network in a short period when many packets arrive

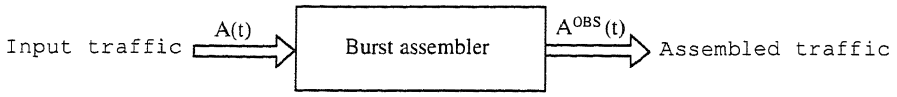


Figure 5.1: Traffic model for burst assembler

suddenly. All of these existing burst assembly algorithms have poor burst loss performances, even at low traffic loads, because the real traffic is bursty [43, 44].

To solve these problems, we will first show how the real traffic characteristics are changed after assembly. Compared with existing analytical methods, we not only analyze the assembled traffic through a theoretical model, but also validate our method by applying it to real traffic. Based on the results, we will present and evaluate two novel burst assembly algorithms with traffic smoothing functions. First, we introduce an assembly scheme, called *sliding window-based*, to smooth the traffic by transmitting bursts at an average rate in a small timescale. We will also propose another traffic-smooth scheme, named *advanced timer-based*, in which the traffic is smoothed by restricting the burst length to a threshold. Compared with existing algorithms, our schemes can improve network performance in terms of the burst loss ratio.

The remainder of this chapter is organized as follows. In Section 5.2, we analyze and simulate the real traffic. In Section 5.3, the proposed algorithms are described. Section 5.4 discusses the findings of our simulations of the proposed schemes, demonstrating their performance.

5.2 Analysis of Assembled Traffic

Recently, the network traffic characteristics have attracted considerable research attention [43, 44, 87], because the traffic properties greatly influence the network's performance. We show a traffic model for a burst assembler in Fig.5.1. Let $\{A(t), t > 0\}$ be the cumulative amount of the input traffic in bits of the burst assembly function arriving during time interval (0, t], and $\{A^{OBS}(t), t > 0\}$ be the cumulative amount of output traffic of the burst assembly function during time interval (0, t], that is, assembled traffic.

We define $X_n(n \in N)$ as a sampled process of input traffic with the unit time of τ. Therefore, $X_n = \{A(n\tau) - A((n-1)\tau), n \in N\}$. We assume X_n to be a wide-sense stationary discrete stochastic process, with constant mean $\mu = E[X_n]$. Let $S_{n,m}$ be the sum of m consecutive numbers from X_n; then

$$S_{n,m} \equiv \sum_{j=(n-1)m+1}^{nm} X_j. \tag{5.1}$$

We define $X_n^{(m)}$ as $X_n^{(m)} \equiv \frac{S_{n,m}}{m}$ and use the variance-time function

$$Var[X_n^{(m)}] = E[(X_n^{(m)} - \mu)^2] \tag{5.2}$$

to represent the variance of input traffic. Here, $E[X_n^{(m)}] = E[X_n] = \mu$.

Similarly, we also define $Y_n(n \in N)$ as a sampled process of assembled traffic with the unit time interval of τ and assume it to be a wide-sense stationary discrete stochastic process. Here, $Y_n = \{A^{OBS}(n\tau) - A^{OBS}((n-1)\tau), n \in N\}$. Then, the variance of assembled traffic can be measured by the variance-time function

$$Var[Y_n^{(m)}] = E[(Y_n^{(m)} - \mu^{OBS})^2], \tag{5.3}$$

where $Y_n^{(m)}$ is the mean value of m consecutive numbers from Y_n, and μ^{OBS} is the mean of $Y_n^{(m)}$ and can be expressed by $\mu^{OBS} = E[Y_n^{(m)}] = E[Y_n]$.

W.E.Leland et al. [43] showed that Internet traffic has fractal characteristics and can be described by a self-similar stochastic process. Self-similarity is a characteristic of traffic in long timescales. The variance of the traffic decreases more slowly, the higher self-similarity of the traffic. The degree of self-similarity can be described by the Hurst parameter H. A larger H indicates a higher self-similarity. In this chapter, we focus on the discussion of "burstiness", which is defined as the variance of traffic in bitrate in small timescales.

Although the characteristics of assembled traffic have already been widely studied [39, 40, 86, 88], there is no agreement on how the burst assembly affects the traffic. In [86], the input traffic was generated according to a Pareto distribution; the researchers simulated the assembled traffic and drew the conclusion that the variance of traffic was increased after assembly and the self-similarity of the traffic was reduced after assembly. However, in [39, 40], the authors assumed the input

traffic following a Poisson distribution and demonstrated that the variance of the assembled traffic decreased after the assembly.

In this chapter, we will study the assembled traffic through different methods. To make our results more reliable, we will first analyze the assembled traffic through a theoretical model, and then validate our method by simulating real traffic.

We assume that the edge routers are loaded with real traffic from the Japan Internet Exchange (JPIX) to SINET [42]. As has already been studied in [41], Fractional Brownian Motion (FBM) [45] was proved to be a good method for analyzing the SINET traffic. The FBM model is defined by

$$A(t) = \lambda t + \sqrt{\lambda a} Z_H(t); \qquad t \in R, \tag{5.4}$$

where λ is the arrival rate for the packets, a is the variance of the coefficient for the arrival packet, and $Z_H(t)$ is the normalized FBM with zero mean and variance of $Var[Z_H(t)] = |t|^{2H}$, where H is the Hurst parameter and satisfies $H \in [0.5, 1)$.

Since Eq.(5.1) can be expressed by $S_{n,m} = A(m\tau)$, the variance-time function in Eq.(5.2) becomes

$$Var[X_n^{(m)}] = Var[\frac{A(m\tau)}{m}] = \lambda a \tau^{2H} m^{-2(1-H)}. \tag{5.5}$$

Figure 5.2 shows the variance-time curve for the real SINET traffic. The variance-time curve is calculated using the variance function provided by the traffic analyzer tools [41] with a sampling time τ set to $10^{-4}s$ (In the following of this chapter, we set $10^{-4}s$ as the default value for the sampling time) for data extracted over 1.2h (14:33-15:45, March 1, 2004).

As shown in Fig.5.2, for large timescales ($m > 200$), the variance-time curve is approximated using the FBM model for the parameter set λ=564.63 Mb/s, $a = 2.61 \times 10^6$, and $H = 0.874$. In small timescales ($m < 200$ and m is large enough to assume the input traffic as a Gaussian process [41] in these timescales), the traffic can be approximated by an FBM model with the parameter set $\lambda = 564.63 Mb/s, a = 8.95 \times 10^5$, and $H = 0.741$.

Next we will study the traffic characteristics of the assembled traffic. R. Morris et al. [52] showed that the traffic from individual sources was not significantly

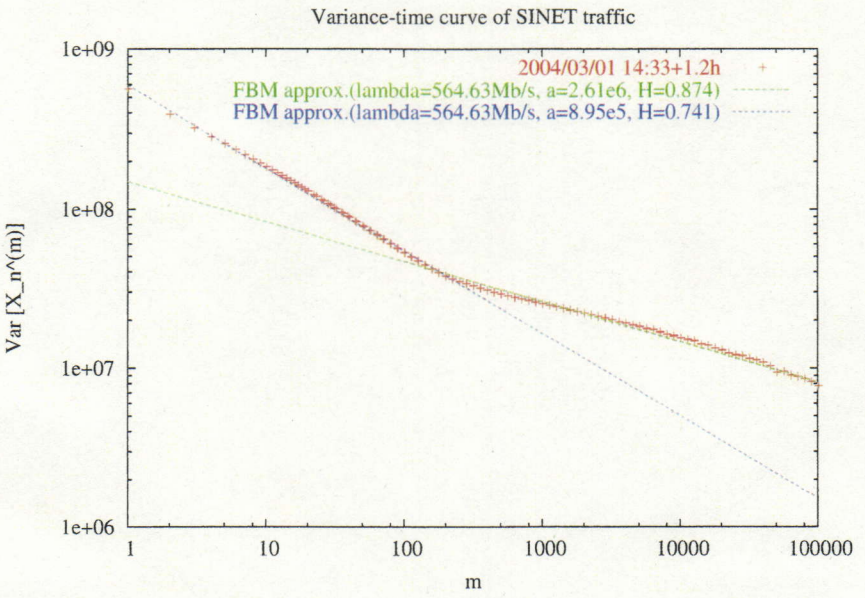


Figure 5.2: Variance-time curve for SINET traffic

correlated. Since the variance of the aggregate of uncorrelated traffic will equal the sum of the individual source's variance, here we only analyze the variance of the assembled traffic of one burst source.

For simplification, we will only analyze the timer-based burst assembly scheme. The other assembly schemes are studied by simulation. With a timer-based burst assembly algorithm, all packets in the assembly buffer will be sent out as a burst when the timer reaches the burst assembly period T_b. Figure 5.3 shows the structure of the timer-based burst assembly mechanism. After a burst is generated, the burst is buffered at the edge node for an offset time before being transmitted to give its BHP enough time to reserve wavelengths along its route. During this offset time period, packets belonging to that queue will continue to arrive. Because the BHP that contains the burst length information has already been sent out, these arriving packets could not be included in the generated burst. Besides dropping these extra packets directly, another way to deal with these packets is to apply two alternate assembly buffers instead of one for each queue. As shown in Fig.5.3, the switch $SPDT_{in}$, which is responsible for selecting the assembly buffer for the arriving IP packets, is switched every T_b for a timer-based burst assembly scheme. When a burst is generated at one buffer, the future arriving packets will be stored

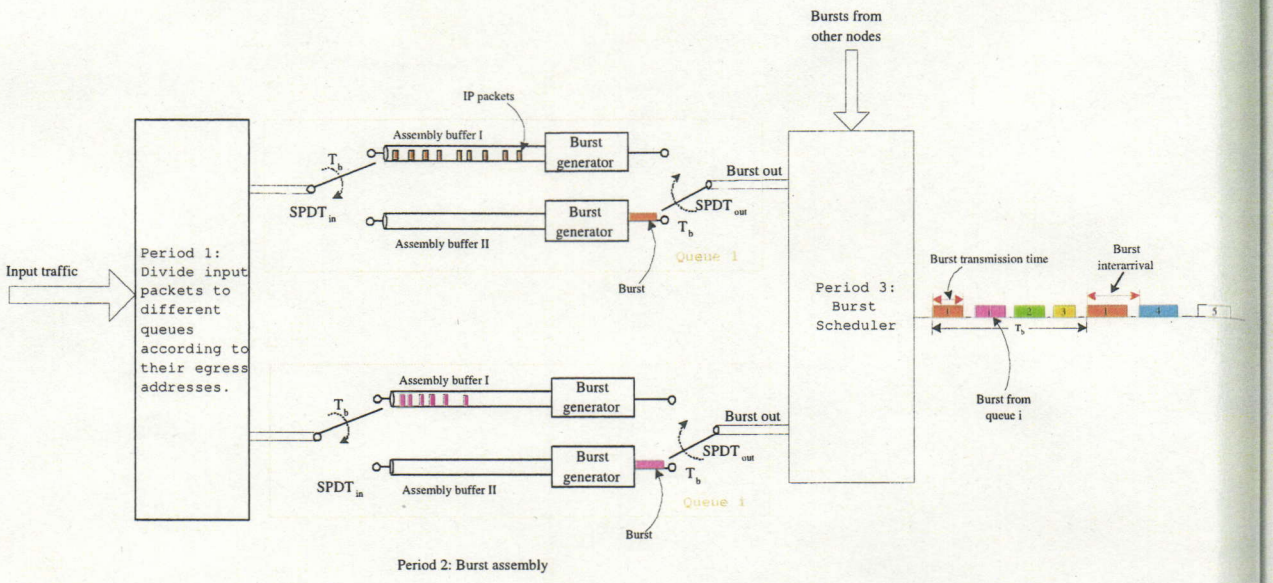


Figure 5.3: Structure of timer-based burst assembly scheme

at another buffer until the next assembly cycle. The switch $SPDT_{out}$ responsible for selecting the buffer for transmitting bursts is also switched every T_b.

Figure 5.4 shows the arrival and departure processes of the assembly queue for the assembled traffic. We assume that the inter-arrival time of the bursts from the same burst source is fixed as T_b. Accordingly, there will be no packets left in the assembly buffer at the time $kT_b (k \in N)$. Therefore, $A^{OBS}(kT_b) = A(kT_b)$. We denote $Q(t)$ as the number of bits that are buffered at the edge router. Then, $A^{OBS}(t) = A(t) - Q(t)$.

For the timer-based assembly algorithm, the $Q(t)$ bits are at most the packets that arrive during $[0, s]$, where $s \in [0, T_b]$. For simplification, we assume s is uniformly distributed in $[0, T_b]$, and $Q(s)$ is a Gaussian process with mean λs and variance $\lambda a s^{2H}$ in not so small timescales [41]. We denote $\overline{E_Q}$ as the mean value of $Q(t)$ observed at any sample point. Then,

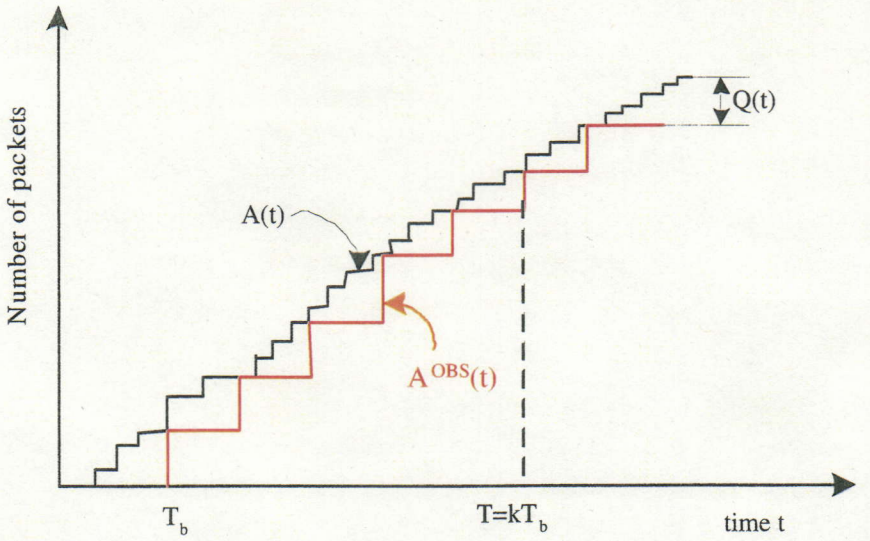


Figure 5.4: Arrival and departure processes of assembly queue

$$\begin{aligned}\overline{E_Q} &= \lim_{T\to\infty}\frac{1}{T}\int_0^T Q(t)dt \\ &= \lim_{k\to\infty}\frac{1}{kT_b}\sum_{i=0}^{k}\int_{iT_b}^{(i+1)T_b} Q(t)dt \\ &= \lim_{k\to\infty}\frac{1}{kT_b}\int_0^{T_b}\sum_{i=0}^{k} Q(s+iT_b)ds. \end{aligned} \tag{5.6}$$

Obviously, $Q(s+iT_b)$ is also a Gaussian process with mean λs and variance $\lambda a s^{2H}$. The series of samples $Q(s+iT_b)$ (for all $i \in N$) construct a stochastic process. The ensemble average for this stochastic process at time s is denoted as $E[Q(s)]$,

$$E[Q(s)] = \lim_{k\to\infty}\frac{1}{k}\sum_{i=0}^{k} Q(s+iT_b) = \lambda s. \tag{5.7}$$

By substituting Eq.(5.7) into Eq.(5.6), we can get,

$$\begin{aligned}\overline{E_Q} &= \lim_{k\to\infty}\frac{1}{kT_b}\int_0^{T_b} kE[Q(s)]ds \\ &= \frac{1}{T_b}\int_0^{T_b}\lambda s ds = \frac{\lambda T_b}{2}. \end{aligned} \tag{5.8}$$

Similarly,

$$\begin{aligned} E[Q^2(s)] &= \lim_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k} Q^2(s + iT_b) \\ &= \lambda^2 s^2 + \lambda a s^{2H}. \end{aligned} \tag{5.9}$$

Let V_Q be the variance of the number of bits for $Q(t)$ observed at any sample point. Then,

$$\begin{aligned} V_Q &= \lim_{T\to\infty} \frac{1}{T} \int_0^T (Q(t) - \overline{E_Q})^2 dt \\ &= \lim_{k\to\infty} \frac{1}{kT_b} \int_0^{T_b} \sum_{i=0}^{k} (Q(t + iT_b) - \overline{E_Q})^2 ds \\ &= \lim_{k\to\infty} \frac{1}{kT_b} \int_0^{T_b} (kE[Q^2(s)] + k\overline{E_Q}^2 - 2k\overline{E_Q}E[Q(s)]) ds \\ &= \frac{\lambda^2 T_b^2}{12} + \frac{\lambda a T_b^{2H}}{2H+1}. \end{aligned} \tag{5.10}$$

Then, the difference between the original traffic and the assembled traffic can be denoted by

$$\begin{aligned} \triangle V(t) &= Var[A^{OBS}(t)] - Var[A(t)] \\ &= Var[A(t) - Q(t)] - Var[A(t)] \\ &= |2cov[A(t), Q(t)] + Var[Q(t)]| \\ &\approx 2Var[A(t)]^{1/2} V_Q^{1/2} + V_Q. \end{aligned} \tag{5.11}$$

To describe the variance of assembled traffic $A^{OBS}(t)$, the variance-time function defined in Eq.(5.3) becomes

$$\begin{aligned} Var[Y_n^{(m)}] &= Var[\frac{A^{OBS}(m\tau)}{m}] \\ &= Var[\frac{A(m\tau)}{m}] + \frac{1}{m^2} \triangle V(m\tau). \end{aligned} \tag{5.12}$$

From Eq.(5.12), we can see there is an increase in the variance in short timescales. This indicates that the timer-based assembly algorithm will increase the variance of traffic. As the timescale increases, the difference between the original and assembled traffic becomes negligible because the traffic does not change significantly for large timescales.

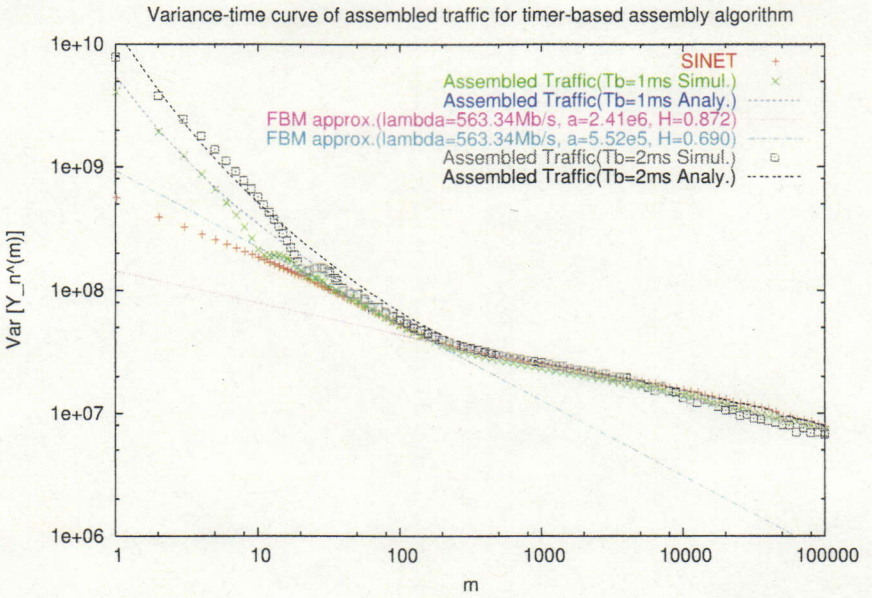


Figure 5.5: Variance-time curve for assembled traffic (timer-based)

Next, we validate our analysis results by simulation. Figure 5.5 shows the variance-time curve for the assembled SINET traffic (same as in Fig.5.2) for the timer-based assembly algorithm with the assembled periods set at 1ms and 2ms. First, we note that the simulation results match the analytical results very well and the variance is increased in short timescales under the timer-based assembly algorithm. And if the variance is approximated by FBM in the area of small timescales ($m < 200$), the Hurst parameter of FBM is smaller than that of the original. A larger assembly period results in a larger variance and a smaller Hurst parameter. The assembled traffic could also be approximated by the FBM model. For example, when the assembly period set at 1ms, the variance-curve of the traffic at large timescales can be approximated using the FBM curve by $\lambda = 563.34$ Mb/s, $a = 2.41 \times 10^6$, and $H = 0.872$, which are almost the same as those of the SINET traffic in Fig.5.2. For small timescales ($10 \leq m < 200$), the approximation curve is calculated using the parameter set λ=564.63 Mb/s, $a = 5.52 \times 10^5$, and $H = 0.690$.

Comparing the simulation results in Figs.5.2 and 5.5, we see that the Hurst parameter of the assembled traffic is smaller than that of the input traffic within the lag range [10, 200), which indicates the Hurst parameter with a range of 1ms to 20 ms is reduced by the burst assembly. However, for timescales bigger than 20

ms, the Hurst parameter and variance remain the same. When the timescales are smaller than the assembly period ($m < 10$), the FBM traffic model is not suitable for real traffic, because the process in this period could not be approximated by a Gaussian process [41].

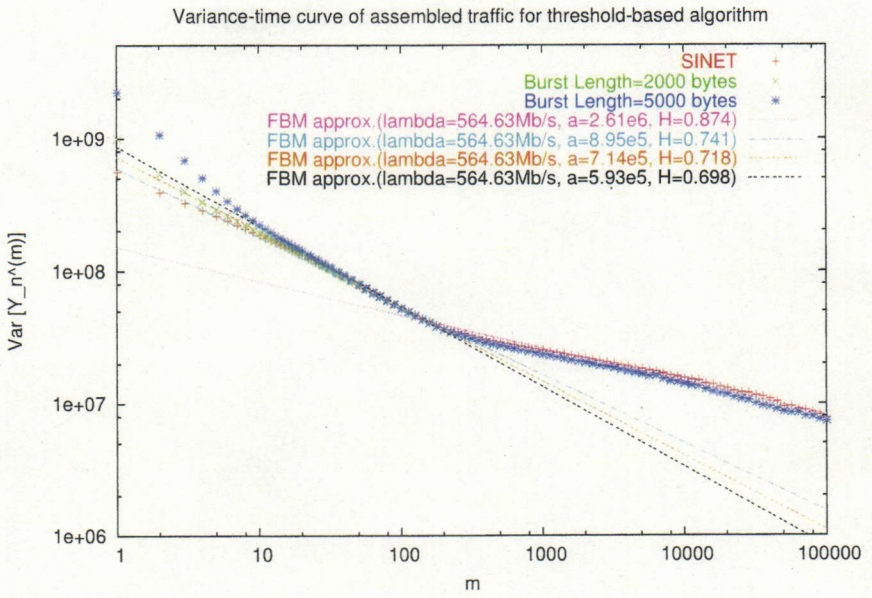


Figure 5.6: Variance-time curve for assembled traffic (threshold-based)

Figure 5.6 shows the variance-time curves of assembled traffic under different assembly thresholds for the threshold-based assembly algorithm. It can be seen that the assembled traffic under threshold-based assembly algorithm can be approximated by the FBM model with the parameter sets shown in the figure. Similar to the results found in Fig.5.5, when we set a larger threshold for the burst assembly length, a larger variance will occur in small timescales. If the variance is approximated by the FBM, the Hurst parameter of the FBM is a smaller value for a larger threshold in small timescales. Seen from Fig.5.6, the Hurst parameter of the FBM equals 0.698 for assembly length of 5000 bytes and equals 0.718 for assembly length of 2000 bytes in small timescales. Additionally, the variance curve of the input traffic and the output traffic overlap at timescales larger than 20 ms, which leads to the same Hurst parameter value ($H = 0.874$) and indicates unchanged self-similarity.

For a hybrid assembly algorithm, the simulation results (not shown here) are similar. This is because the hybrid assembly algorithm is similar to the timer-based

assembly algorithm at low traffic loads and similar to the threshold-based assembly algorithm at high traffic loads.

5.3 Operation of Proposed Burst Assembly Algorithms

The simulation and analytical results in Section 5.2 show that both the timer-based and threshold-based assembly algorithms could not reduce the variance of the real traffic, but do increase it in small timescales. In this chapter, the word burstiness is defined as the variance of the bitrate in small timescales. A larger burstiness indicates that the traffic is burstier and more likely to exceed the capacity of the OBS network and it results in burst loss events. So, a larger burstiness implies a higher burst loss ratio in bufferless OBS networks.

One way to reduce the burst loss ratio is to control the burst sources at the edge nodes and thereby inject the bursts more smoothly into the network. In this section, we will propose two novel burst assembly algorithms with traffic smoothing functions, to reduce the burstiness of the assembled traffic.

5.3.1 Sliding Window-based Burst Assembly Algorithm

We first propose a scheme, called *sliding window-based*, to reduce the burstiness of traffic by smoothing the size of the bursts. In this scheme, we store a small amount of traffic at the edge router and the output traffic is determined by the amount of stored traffic. Let T_{edge}^{max} be the preset maximum tolerant value for the edge buffering delay. Here, the edge buffering delay refers to the period a packet spent waiting at the edge node. As shown in Fig.5.7, let $B(t) = A(t-T_{edge}^{max})$ represent the cumulative traffic delayed for T_{edge}^{max}. The curve of $A^{OBS}(t)$ should lie between $B(t)$ and $A(t)$ because of the bounds on the assembly delay. At the initial phase of assembly, the arriving packets will be stored at the edge node until the timer exceeds T_{edge}^{max}. Then, the edge router sends traffic out at an average rate in the timescale $[0,T_{edge}^{max}]$. At time t, the rate of the output traffic is set to the average rate in $[t, t+T_{edge}^{max}]$. When

the bursts are sent out periodically, the burst size is set to the average number of packets arriving during assembly period T_b.

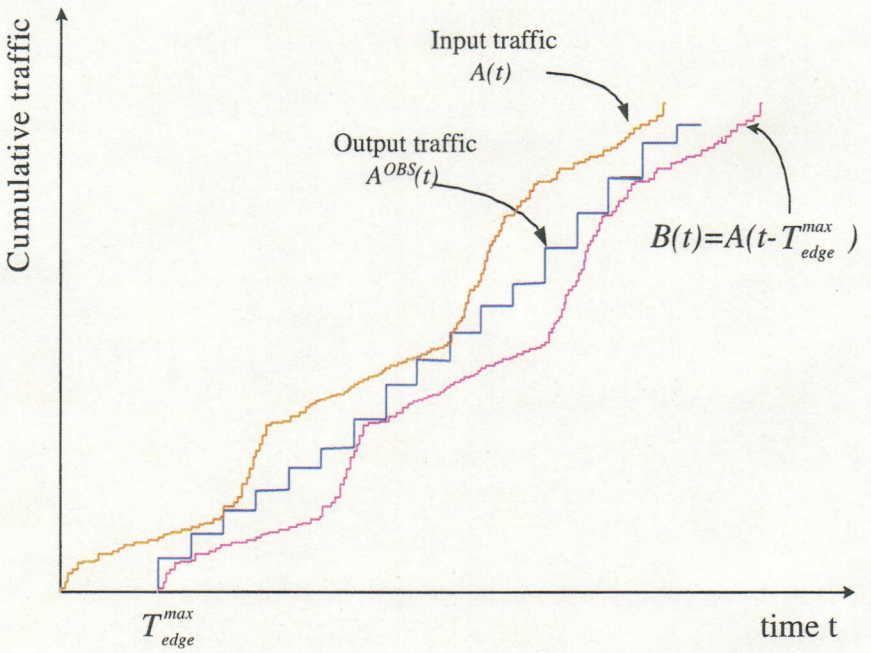


Figure 5.7: Delay constraints for burst assembly

Suppose each edge router has G queues to sort the arriving packets. Let the timer of queue $Q[i]$ be denoted by $T[i]$ and the length of $Q[i]$ be denoted by $L[i]$. The threshold for generating a burst is $L_{th}[i]$. When the value of the queue length $L[i]$ is smaller than $L_{th}[i]$, all packets in $Q[i]$ will be assembled into a burst. Otherwise, a burst is generated with the length of $L_{th}[i]$ and the other packets are left in $Q[i]$. The scheme is thus implemented using the following algorithm.

Begin

1. When a packet with a length of b arrives at $Q[i]$

If ($Q[i]$ is empty)

Start timer $T[i]$, $L[i] = b$

Else

Push packet into $Q[i]$,

$L[i] = L[i] + b$

End if

2. When $T[i] = T_{edge}^{max}$

If $(L[i] > L_{th}[i])$

$L_b = L_{th}[i]$, $L[i] = L[i] - L_{th}[i]$,

Set timer $T[i] = T_{edge}^{max} - T_b$

Else

$L_b = L[i]$, $L[i] = 0$,

$T[i]$=0, stop timer $T[i]$

End if

Generate a burst with length L_b and send it into the OBS network

End

In the above description, the threshold $L_{th}[i]$ is calculated by the following equation, in which D is defined as T_{edge}^{max}/T_b and L_{min} is defined as the preset minimum threshold for the burst size.

$$L_{th}[i] = \begin{cases} L[i]/D & (L[i] \geq L_{min} \times D) \\ L_{min} & (L[i] < L_{min} \times D) \end{cases} \tag{5.13}$$

According to the first line of Eq.(5.13), we will assemble $1/D$ parts of the packets in $Q[i]$ to a burst each time the length of the assembly queue is greater than $L_{min} \times D$. When the traffic load suddenly increases (or decreases), the increase (or decrease) of the burst size only equals $1/D$ of the one under a timer-based assembly algorithm. This makes the burst size smoother and therefore, the traffic smoother. To avoid too many small bursts, we set L_{min} as the minimum threshold for the burst size, as shown in the second line of Eq.(5.13). Whenever a queue is empty, the packets would experience an initial delay T_{edge}^{max}.

5.3.2 Advanced Timer-based Burst Assembly Algorithm

Besides the sliding window, another way to reduce the burstiness is a peak rate restriction. Conceptually, the number of bursts simultaneously arriving at an input port is most likely to reach a maximum value when the traffic is at a peak rate. Reducing the number of overlapping bursts on a link is for each ingress node to restrict the assembled traffic to a specified rate.

In a timer-based assembly scheme, because the bursts are generated periodically, the traffic rate can be restricted by restricting the burst length to a threshold. Based on this idea, we propose a scheme, called *advanced timer-based*, to reduce the burstiness of traffic.

The scheme is thus implemented using the following algorithm, in which the relative parameters are defined to be the same as in the sliding window-based assembly algorithm.

Begin

1. When a packet with a length of b arrives at $Q[i]$

If ($Q[i]$ is empty)

Start timer $T[i]$, $L[i] = b$

Else

Push packet into $Q[i]$,

$L[i] = L[i] + b$

End if

2. When $T[i] = T_b$

If ($L[i] > L_{th}[i]$)

$L_b = L_{th}[i]$, $L[i] = L[i] - L_{th}[i]$,

$T[i]$=0, restart timer $T[i]$

Else

$L_b = L[i]$, $L[i] = 0$,

$T[i]$=0, stop timer $T[i]$

End if

Generate a burst with length L_b and send it into the OBS network

End

In the above description of the advanced timer-based assembly algorithm, the threshold $L_{th}[i]$ is calculated by the following equation:

$$L_{th}[i] = \begin{cases} (1+\alpha) \times \overline{L_M[i]} & (\overline{L_M[i]} \geq L[i]/\beta) \\ (1+\alpha) \times L[i]/\beta & (\overline{L_M[i]} < L[i]/\beta). \end{cases} \tag{5.14}$$

where $\overline{L_M[i]}$, the mean length of the recent M bursts of $Q[i]$, is used to predict the

average size of the arriving bursts, α is the parameter of the redundancy degree for the prediction of the arriving burst size, and β is defined as: $\beta = T_{edge}^{max}/T_b$.

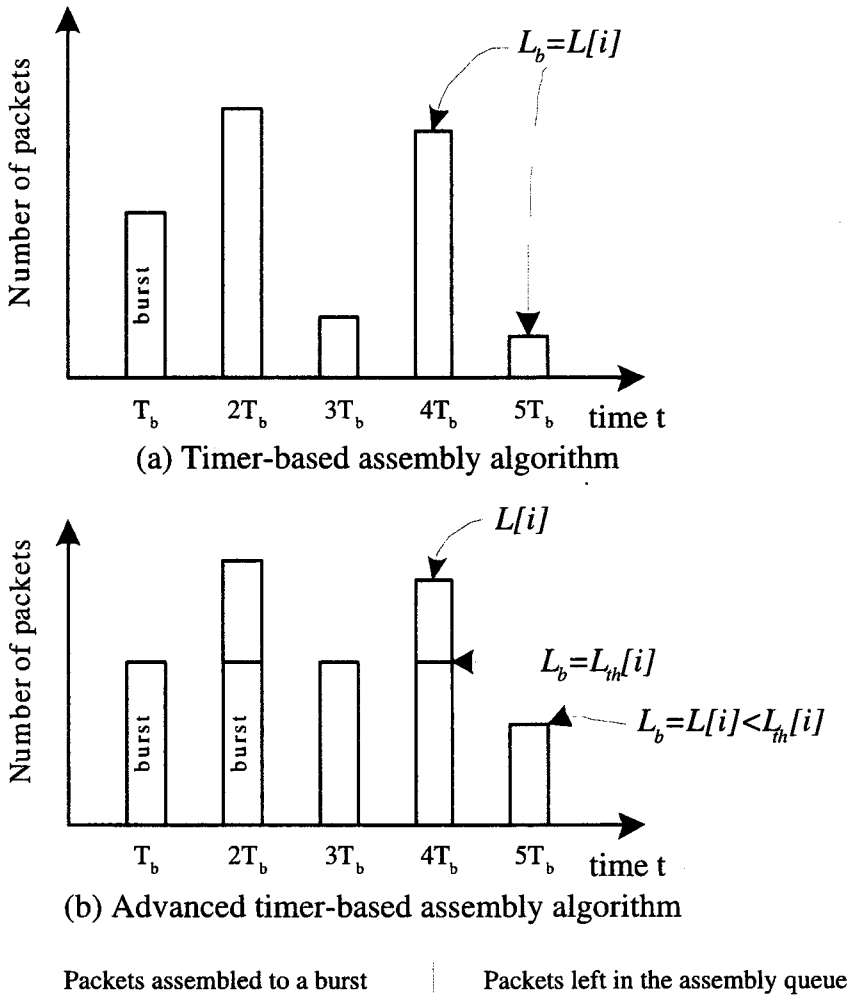


Figure 5.8: Comparison of timer-based and advanced timer-based algorithms

Figure 5.8 shows a comparison of the timer-based and advanced timer-based assembly algorithms. For the timer-based assembly algorithm, all packets in the assembly buffer will be multiplexed to a burst every assembly period. This makes for various burst sizes because the number of packets that arrive in each assembly period varies. On the other hand, the burst size in the advanced timer-based assembly algorithm is restricted by the threshold $L_{th}[i]$. After an $L_{th}[i]$ length of packets are assembled into a burst, the other packets will be left in the assembly queue for a future assembly process. So, the advanced timer-based scheme can avoid a sudden increase in the burst size and makes the burst sent out more smooth than

the timer-based assembly algorithm does.

Take into consideration the choice of $L_{th}[i]$ to take advantage of the effects of the peak rate restriction. It is clear that the restriction of on the peak rate should be bigger than the average rate and the threshold $L_{th}[i]$ must exceed the average burst length $\overline{L_M[i]}$, otherwise the traffic would be blocked at the edge nodes. As $L_{th}[i]$ is close to $\overline{L_M[i]}$ ($\alpha \rightarrow 0$), the transmission is almost the same as in a CBR transmission. However, this will result in enormous backlogs at the edge routers. The choice of α is a tradeoff between the effects of the peak rate restriction and the edge buffering delay. This will be discussed in detail in Section 5.4. When too many packets suddenly arrive, if we still assemble packets into bursts using a small threshold $L_{th}[i]$, the packets will suffer a large edge buffering delay. By setting $L_{th}[i]$ according to the second line of Eq.(5.14), the system could quickly increase the assembly threshold and keep the maximum edge buffering delay under T_{edge}^{max}.

5.4 Numerical Results and Discussion

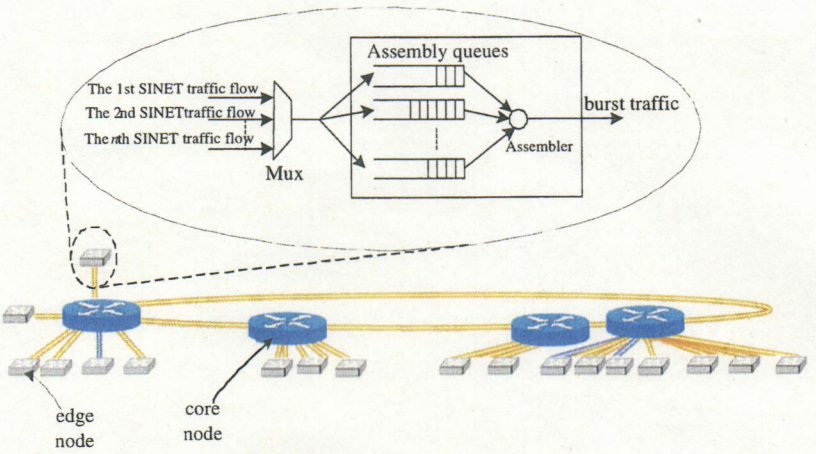


Figure 5.9: Simulation topology of SINET network

To evaluate the performance of the sliding window-based and advanced timer-based assembly algorithms, we use the SINET network topology [50] (Fig.5.9) as our simulation model. In our simulation topology, we assume both the link connecting two core nodes and the link connecting a core node and an edge node are bidirectional

links, in which each direction consists of 16 wavelengths with a transmission rate of 1 Gbps per wavelength. Although there is no optical buffer in the simulations, there are full wavelength conversions among the data channels at the OBS nodes. In addition, we assume that there is a route path for every edge node pair which is computed by a shortest path algorithm. For an output link, the traffic load is defined as the ratio of the average arriving bitrate of traffic to the capacity of the output link. Here, the capacity of one output link equals 16 Gbps.

For simplicity, we only study the burst loss events on the links connecting two core nodes. So, in the following simulations, the traffic load refers to the average traffic load on the links connecting two core nodes. As shown in Fig.5.9, the multiplexing of real SINET traffic is used as the input traffic for edge routers to generate bursts. We adjust the number n of SINET traffic flows of each edge node to adjust the input traffic volume and make the traffic evenly distributed among the links connecting two core nodes so that we can apply different traffic loads to the SINET network. Each SINET traffic source flow was obtained during 14:33-15:45, March 1, 2004. The average traffic bitrate of 564.63 Mb/s and other characteristics are shown in Fig.5.2. The basic processing time for a BHP at each core node is chosen to be 0.1 ms. The offset time is uniformly distributed in [0.5ms, 1ms], which not only includes the basic part for the electronic processing of the control message at the intermediate nodes, but also includes the extra offset time necessary to avoid a systematic synchronization between different burst sources [51]. We set T_b=1ms as the default value for the burst assembly period, and T_{edge}^{max}=5ms as the maximum tolerant burst edge buffering delay. For the threshold-based and hybrid algorithm of the simulations, the threshold of the burst size is set to have the same mean burst size as the sliding window-based assembly algorithm has.

Regarding the performance of the advanced timer-based assembly algorithm, we set $M = 50$. Figure 5.10 shows the impact of parameter α on the burst loss ratios for different traffic loads under the SINET topology in Fig.5.9. It shows that the burst loss ratio decreases as α decreases. On the other hand, the average edge buffering delay increases as α decreases, as shown in Fig.5.11. However, the edge buffering delays are mostly less than T_{edge}^{max} (5ms), even at small α. How to choose

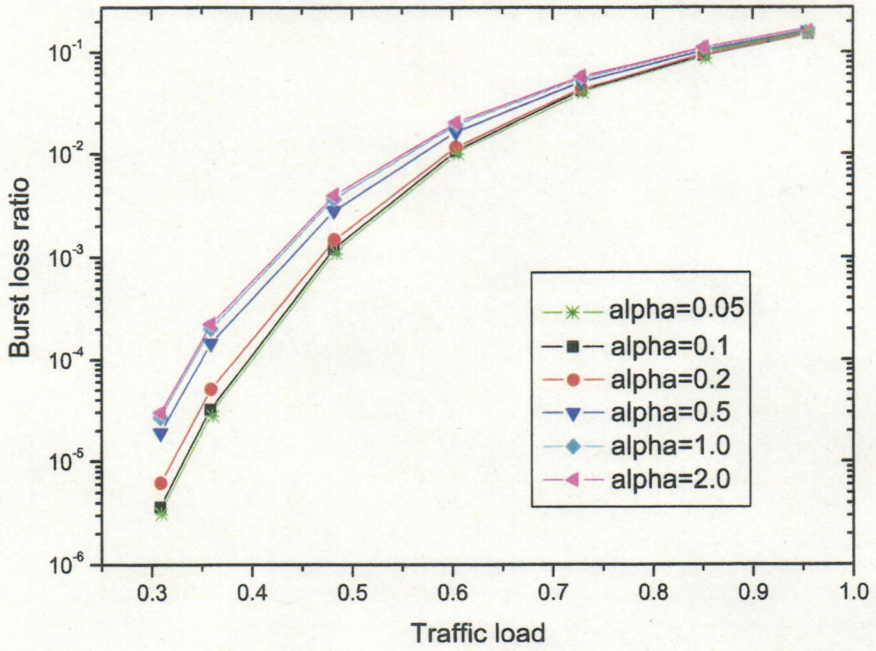


Figure 5.10: Impact of α on burst loss ratio for advanced timer-based algorithm

α is a tradeoff between the burst loss ratio and the average edge buffering delay. A suitable α should be able to achieve a low burst loss ratio while not increasing the edge buffering delay too much. From Figs.5.10 and 5.11, we can see that when α is smaller than 0.1, the burst loss ratio decreases slowly and is almost the same as the one for $\alpha = 0.1$ while the average edge buffering delay obviously increases. So in the following simulations, we set 0.1 as the default value of α. As shown in the Appendix (Figs.5.21 and 5.22), $\alpha = 0.1$ also achieves a good balance for the simulations with the SINET traffic data during another time period.

Figure 5.12 compares the burst size distribution for the timer-based assembly algorithm with the sliding window-based and advanced timer-based assembly algorithms. Here we set L_{min}=4 Kbytes for the sliding window-based assembly algorithm. As has been assumed, the mean values of the burst size for these three algorithms are the same. However, the curves of the sliding window-based and advanced timer-based algorithms decay more quickly than the curve for the timer-based assembly algorithm does. This indicates that the burst size variances of our proposed algorithms are smaller than that of the timer-based assembly algorithm and most bursts under our proposed algorithms have a similar size, thereby making

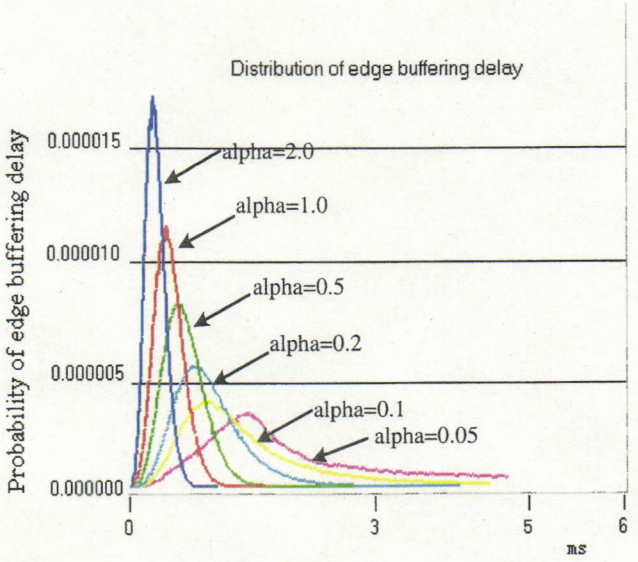


Figure 5.11: Impact of α on edge buffering delay for advanced timer-based algorithm (traffic load=0.5)

the traffic smooth.

We next use the variance of bitrate to describe the traffic characteristics. Figure 5.13 shows the variances of assembled traffic under different assembly algorithms. The variances in small timescales for the sliding window-based and advanced timer-based assembly algorithms are clearly smaller than those of the timer-based, threshold-based, and hybrid algorithms. It also shows that all the assembled traffic flows under different assembly algorithms can be approximated by FBM models with the parameter sets shown in Fig. 5.13.

Figure 5.14 compares the burst loss ratios under different assembly algorithms. The simulation results show that the sliding window-based and advanced timer-based assembly algorithms have better burst loss performances than the timer-based, threshold-based, and hybrid assembly algorithms, due to their traffic smoothing effect. For example, the burst loss ratios of our proposed schemes are about 10 times lower than those for the other schemes when the traffic load is 0.3.

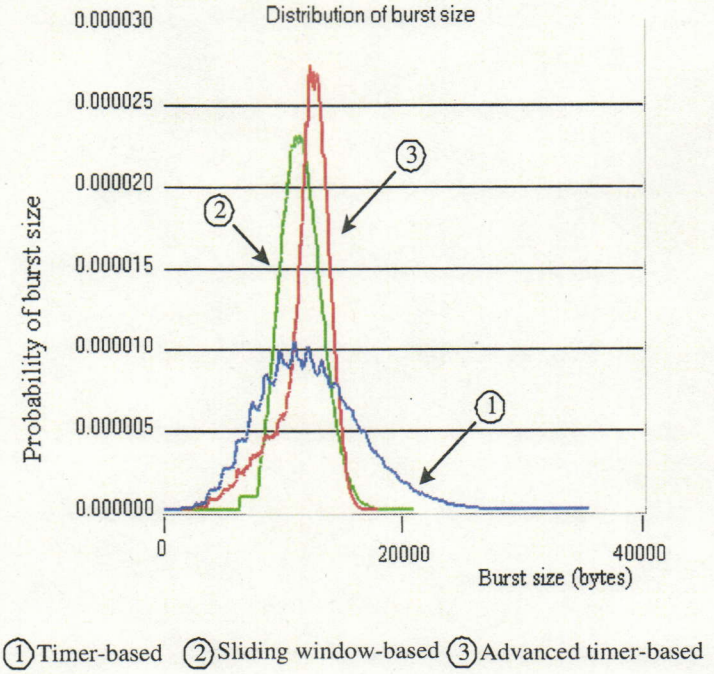


Figure 5.12: Burst size distribution for different assembly algorithms (traffic load=0.5)

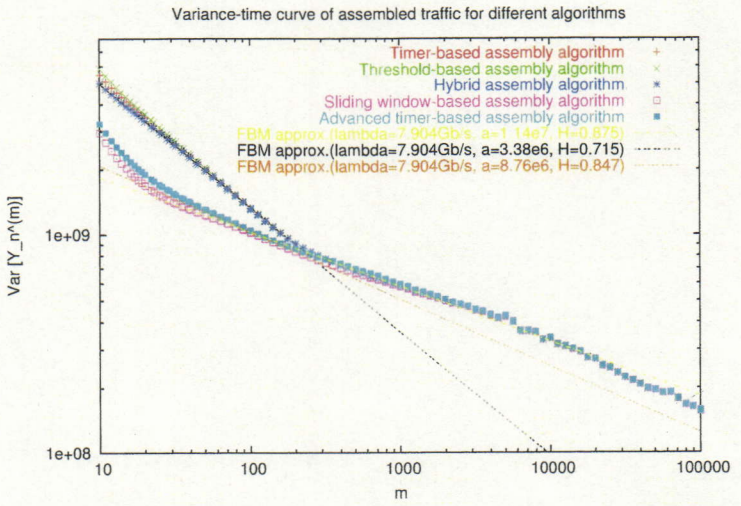


Figure 5.13: Variance-time curve for different assembly algorithms (traffic load=0.5)

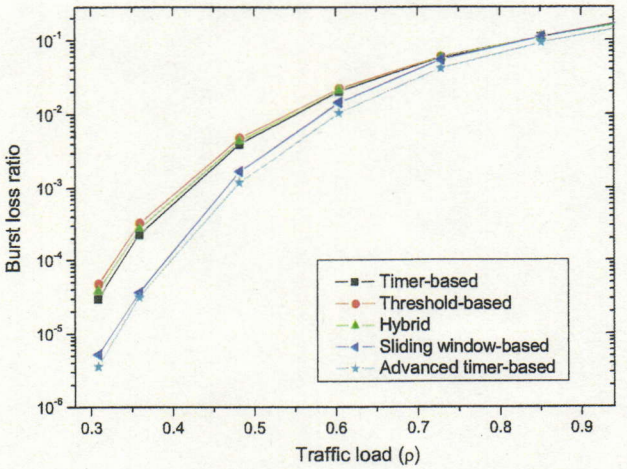
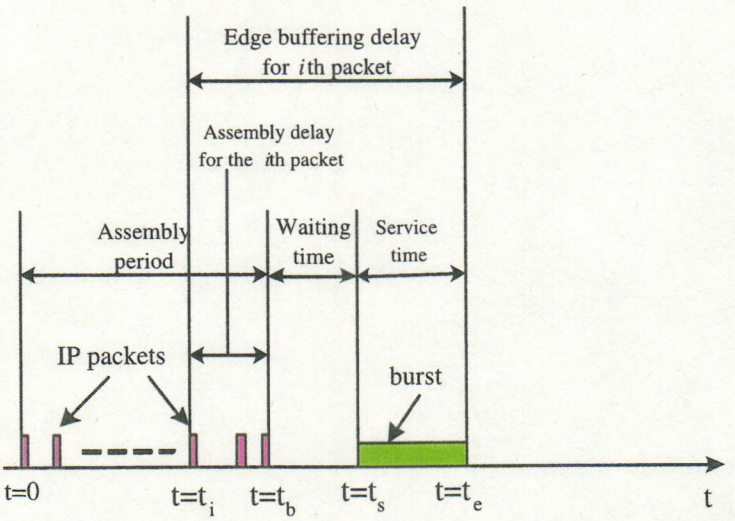


Figure 5.14: Burst loss ratios for different assembly algorithms



t=0 : The start time the assembly period
$t=t_i$: The arrival of the ith packet.
$t=t_b$: Assembly the packets into a burst
$t=t_s$: Start of transmission of the burst
$t=t_e$: End of transmission of the burst

Figure 5.15: Time-based burst assembly process of packets at edge node

Figure 5.15 shows the edge buffering delay that the ith packet experiences at the edge node ($[t_i, t_e]$), which consists of three parts. The first period is the assembly delay ($[t_i, t_b]$), which refers to the period spent from its arrival to its being assembled into a burst. Assume t=0 is the start time of the assembly period, $t = t_i$ is the arrival time of the ith packet. So the assembly delay of the ith packet can be calculated by $t_b - t_i$.

After assembly, the burst (not IP packets) will be buffered at the edge node waiting for being transmitted. We call this period as waiting time and calculate it by $t_s - t_b$. Because the offered traffic at the edge node is very small even the traffic load at core node is close to one so that the waiting time of burst after assembly are negligible.

The third part of edge buffering delay is transmission period (service time)($[t_s, t_e]$). According to Fig.5.12, the burst has an average length of 10000 bytes. So the average service time of a burst can be calculate by *average burst length duration=average burst length/bandwidth=10000 Kbytes/1 Gbps=0.08 ms.* So the average transmission time of most bursts are also negligible comparing to the assembly delay.

Of course, when many packets arrive in the assembly period, the burst length is larger than 10000 bytes. However there is a probability for this event. And here I would like to evaluate this probability by a simple model.

According to Fig.5.12, the burst length has a Gaussian-similar distribution. We first use Normal Quantile-Quantile (Q-Q) Plot [94] to graphically compares the distribution of burst length samples to the Normal distribution. When the theoretical Gaussian distribution matches the distribution of measured burst length samples, the quantile-quantile plot will give a straight line. The Q-Q plots results in Fig.5.16 show that curve of burst length samples fits theoretical Gaussian distribution very well.

We then use One-sample Kolmogorov-Smirnov test (KS-test) [96, 97] to determine whether the distribution of burst length samples and Gaussian distribution (with the same mean and variance) differ significantly. The test result D= 0.0639, which is the maximum vertical deviation between the cumulative fraction plot of the distribution of burst length sample and that of Gaussian distribution, indicates

that there is very little discrepancy between the burst size distribution and Gaussian distribution. We also do another further test, called Shapiro-Wilk normality test [95, 97], to test the null hypothesis whether the burst length samples comes from a Normal distribution. The result W=0.999 shows that the test result does not reject the null hypothesis, where the statistic W is calculated as follows

$$W = \frac{(\sum_{i=1}^{n} a_i L_b(i))^2}{\sum_{i=1}^{n}(L_b(i) - \overline{L_b})^2}, \tag{5.15}$$

in which the $L_b(i)$ are the ordered sample values ($L_b(1)$ is the smallest) and the a_i are constants generated from the means, variances and covariances of the order statistics of a sample of size n(n=5000) from a Normal distribution. All these tests show that there is very little discrepancy between the burst size distribution and Gaussian distribution.

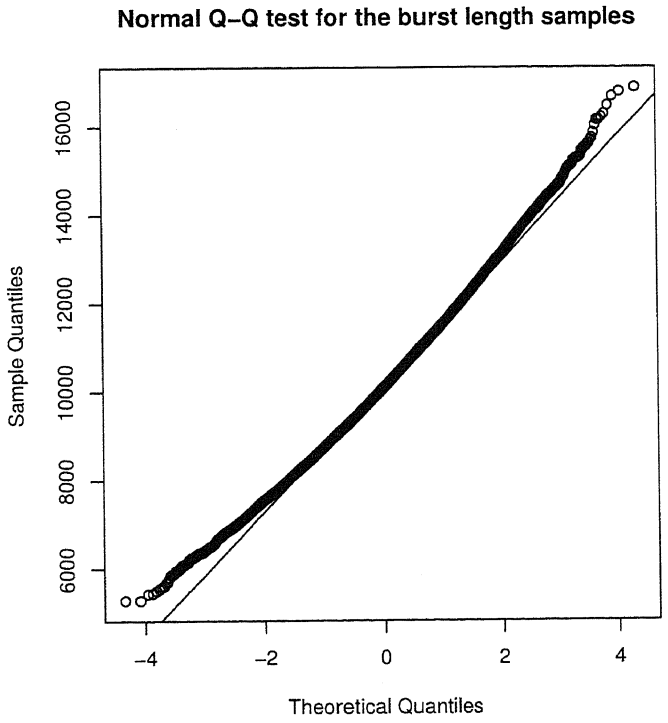


Figure 5.16: Q-Q normal test for observed burst lengths for sliding-window based algorithm

Then we applied Gaussian distribution to fit the curve of burst length distribution in Fig.5.17. The results show that it fits well for sliding-window based assembly algorithm. According to my analyzed parameters($\overline{L_b}$ is average and σ^2 is variance for the

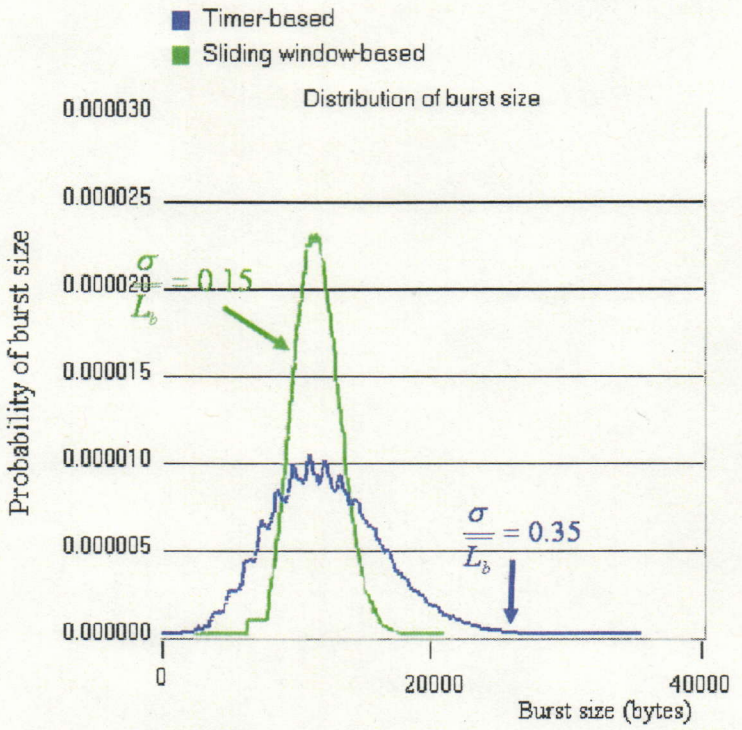


Figure 5.17: Fitting observed burst lengths with Gaussian distribution

burst size), we get P(BL >20000bytes)=$1.62e^{-11}$ and P(BL >30000bytes)=$1.1e^{-40}$, where BL is the burst length. Even when the burst length reaches 30000 bytes at a probability of $1.1e^{-40}$, the service time is only 0.24 ms, which can be omitted comparing with 5ms.

For the sliding-window based assembly algorithm, t_b is predefined to 5ms in this chapter, so the edge buffering delay $(t_b - t_i)$ are mostly limited to 5ms. According to the sliding-window based algorithm, each time we only assemble 1/5 parts of packets in the queue of the edge node into a burst. So the edge buffering delays are mostly distributed from 4ms to 5ms.

We have also compared the edge buffering delays of our proposed sliding window-based and advanced timer-based assembly algorithms with those of the threshold-based and hybrid assembly algorithms (Fig.5.19). Simulation results show that hybrid assembly algorithm has the best performance in terms of burst edge buffering delay due to that its maximum edge buffering delay is limited by the burst assembly period (T_b). Simulation results show that the probabilities over T_{edge}^{max} (5ms) of the

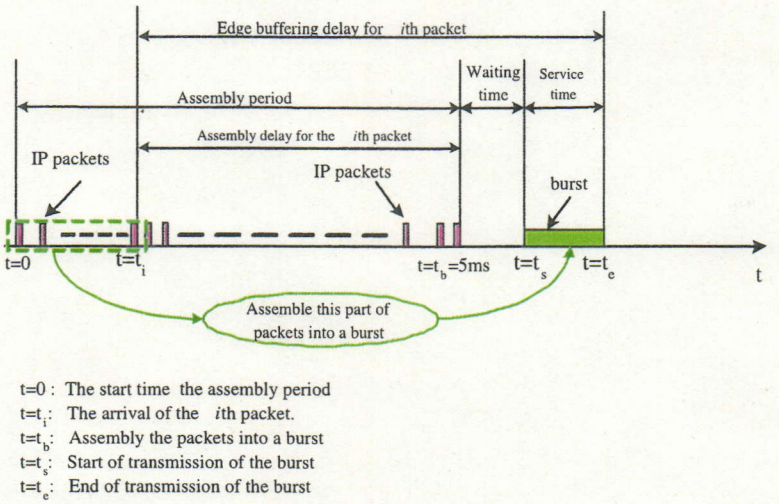


Figure 5.18: Burst assembly process for sliding-window based algorithm

edge buffering delay are very small while the edge buffering delay of the threshold-based assembly algorithm is out of control. It can also be seen that the average edge buffering delay of the advanced timer-based assembly algorithm is smaller than that of the sliding window-based assembly algorithm due to the initial delay suffered by the sliding window-based assembly algorithm.

As we have mentioned in the first paragraph of Section 5.4, the threshold of the burst size of threshold-based and hybrid algorithms is set to have the same mean burst size as those of sliding window-based and timer-based assembly algorithms. For each edge node, we assume that the electronic buffering is large enough and will not overflow; the departure time of assembled burst is negligible compared to assembly period. For an ingress edge node, let the arriving rate of packets be λ_i and the mean burst size of sliding-window algorithm be Lb_i. It takes an average period T_i for the edge router to gather packets of Lb_i size, where $T_i = Lb_i/\lambda_i$. When the arriving rate of packets is increased to λ_j, and the mean burst size of timer-based algorithm is increased to Lb_j. Obviously we can get $\frac{Lb_i}{Lb_j} = \frac{\lambda_i}{\lambda_j}$ for the same assembly period T_b. It takes an average period T_j for the edge router to gather packets of Lb_i size, where $T_j = Lb_j/\lambda_j = T_i$. So the edge buffering delay is approximately independent of traffic load.

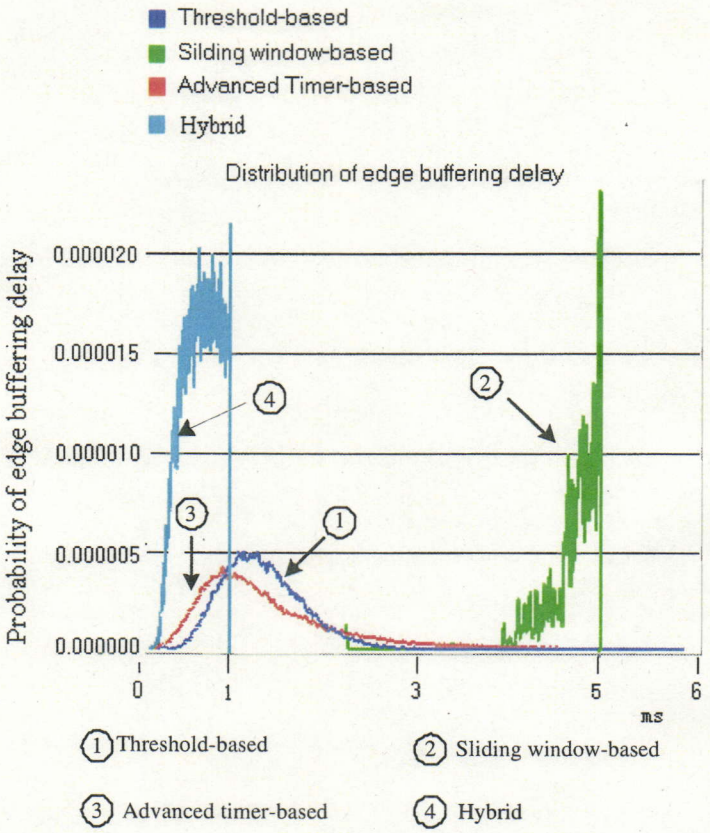


Figure 5.19: Edge buffering delay for different assembly algorithms (traffic load=0.5)

5.5 Summary

In this chapter, we used a Fractional Brownian motion (FBM) model to analyze the effect of OBS assembly mechanisms on the burstiness of SINET traffic. Our simulation and analytical results showed that both the timer-based and threshold-based assembly algorithms could not reduce the burstiness of real traffic. Therefore, we proposed two novel burst assembly algorithms with traffic smoothing functions: the sliding window-based assembly algorithm and the advanced timer-based assembly algorithm. Simulation results showed that the advanced timer-based assembly was best, because it was not only able to reduce the burst loss ratio, but could also control the edge buffering delay. The sliding window-based algorithm was also able to reduce the burst loss ratio, especially under low traffic loads at the expense of only a small initial delay.

Appendix: Some results based on SINET data during another time period

In this appendix, we add some results based on the SINET data that were obtained during 14:50-15:50, February 27, 2004 as supplements for our proposed schemes.

Figure 5.20 shows that the SINET traffic during this period can also be approximated by FBM models with different parameter sets. The characteristics of each traffic flow are shown in this figure.

Figures 5.21 and 5.22 show that when α is smaller than 0.1, the burst loss ratio decreases slowly and is almost the same as the one for $\alpha = 0.1$ while the average edge buffering delay obviously increases. So 0.1 is also a preferred value for α for the new simulations.

Figures 5.23 and 5.24 indicate that both the burst size variances and the burstiness of our algorithms are smaller than those of the timer-based, threshold-based, and hybrid assembly algorithms and the assembled traffic can also be approximated by FBM with the parameter sets shown in Fig.5.24.

The results in Figs.5.25 and 5.26 show that the sliding window-based and advanced timer-based assembly algorithms not only bound most edge buffering delays to the preset T_{edge}^{max} (5ms) but also have better burst loss performance than the timer-based, threshold-based, and hybrid assembly algorithms.

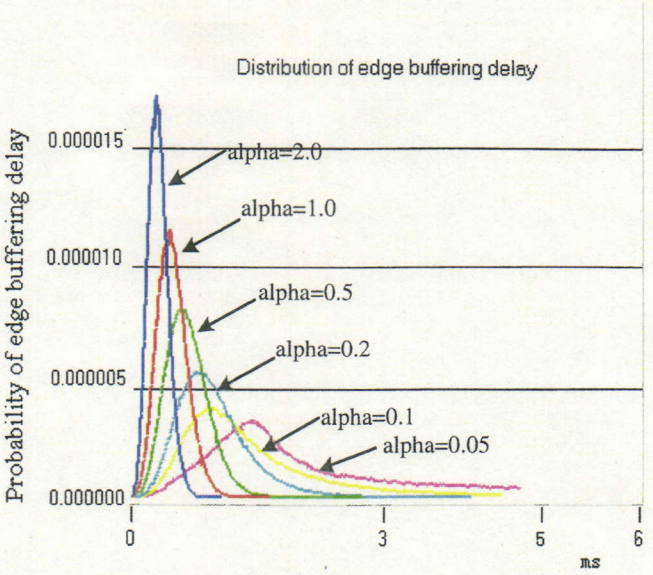


Figure 5.22: Impact of α on edge buffering delay for advanced timer-based algorithm (traffic load=0.5)

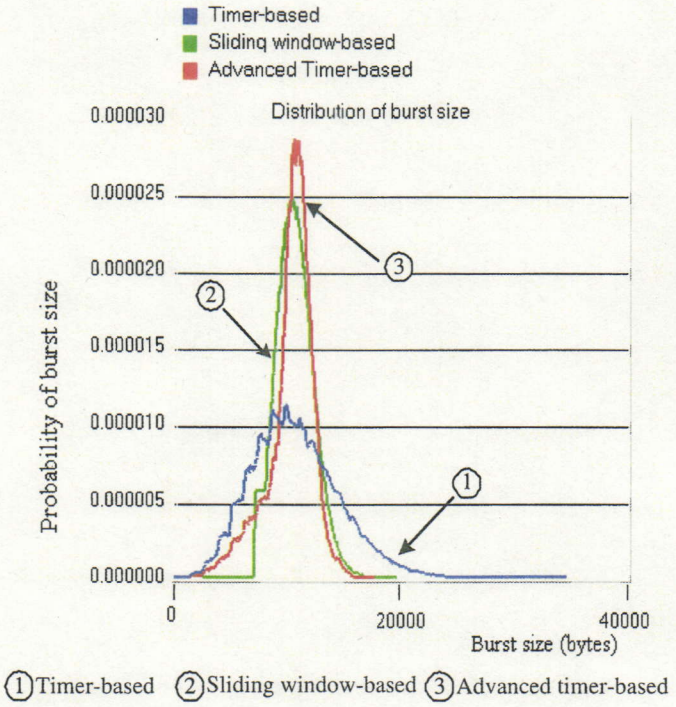


Figure 5.23: Burst size distribution for different assembly algorithms (traffic load=0.5)

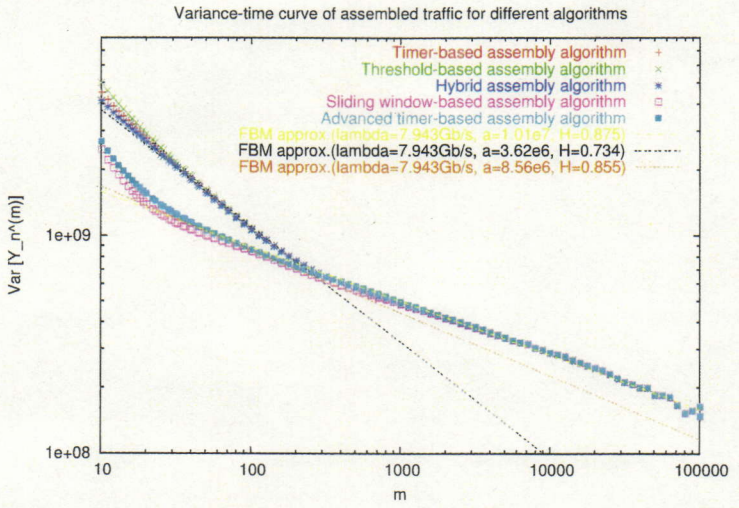


Figure 5.24: Variance-time curve for different assembly algorithms (traffic load=0.5)

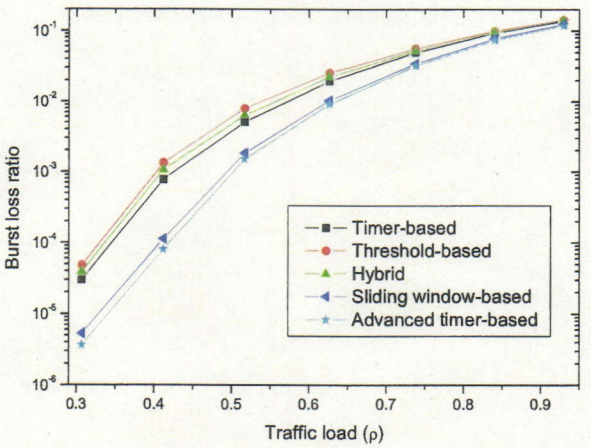


Figure 5.25: Burst loss ratios for different assembly algorithms

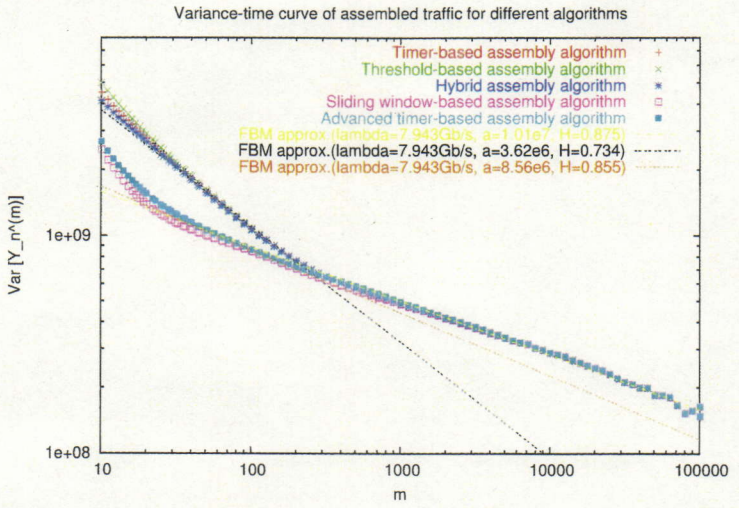


Figure 5.24: Variance-time curve for different assembly algorithms (traffic load=0.5)

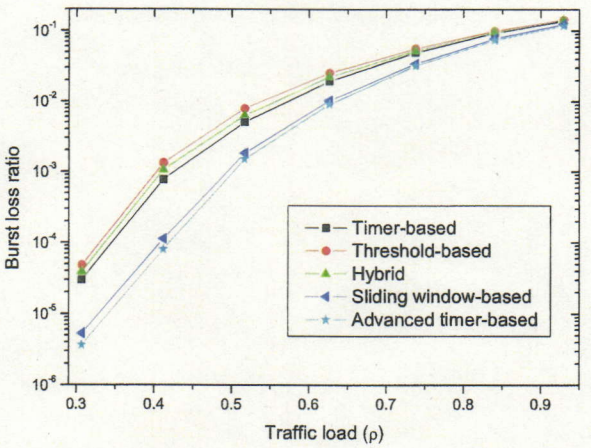


Figure 5.25: Burst loss ratios for different assembly algorithms

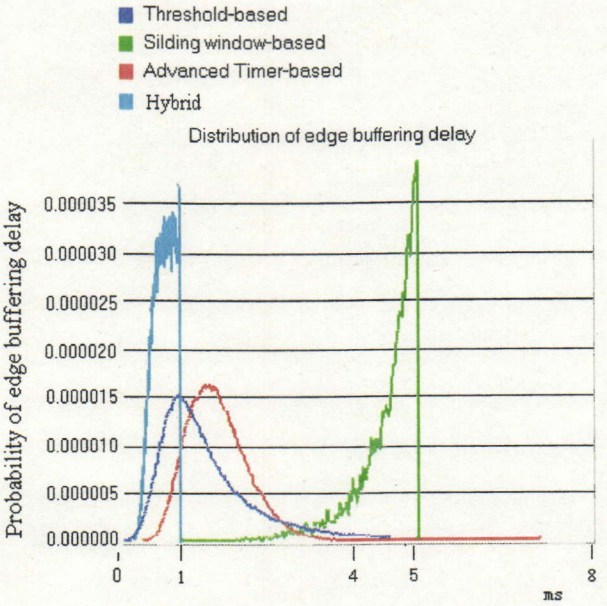


Figure 5.26: Edge buffering delay for different assembly algorithms (traffic load=0.5)

Chapter 6

TCP over OBS

6.1 Introduction

The Transmission Control Protocol (TCP) is a connection-oriented transport protocol and TCP traffic accounts for approximately 90% of total Internet traffic and foreseeingly will be the dominant traffic in future optical networks. We believe that it is very important to study the TCP performance on OBS networks. Comparing with traditional packet switching networks, the OBS layer has two effects on the functionality of transport layer.

- `Effect of burst dropping`: The probability of burst dropping depends on the traffic load of network. In a traditional electronic network, packet loss is largely due to the overflow of buffers. So the packet loss events are correlated especially during a round trip time. In an OBS network, a burst loss event is caused when several bursts are destined to the same output port simultaneously. So burst loss events are not so correlated. When a burst is dropped, many packets included in this burst will be dropped. This may result in a network wide dropping in throughput.

- `Effect of burst assembly delay`: Another factor affecting performance is the delay that a packet experiences during the assembly process before its associated burst is transmitted.

In [67], the OBS researchers proposed a single-link model to analyze the OBS performance. However, it could not evaluate the performance of an OBS network with multiple TCP connections.

In this chapter, we will develop a precise analytical model to evaluate the TCP performance of OBS network. The remainder of the chapter is organized as follows. Section 6.2 introduces TCP Reno mechanism. Section 6.3 introduces our analytical model. In Section 6.4, we introduce a burst acknowledgment mechanism; then, a loss detection and error recovery mechanism is proposed. Section 6.5 simulates the performance of our proposed mechanisms.

6.2 Review of TCP Reno

There are several versions of TCP running on different hosts (operation systems). In this chapter, we will only discuss TCP "Reno" because it is the most popular implementations in Internet and there is no significant difference between different TCP versions in terms of performance.

TCP-Reno is a window-based flow control protocol [65]. The TCP transmitter uses receiver window (RCVWND) to advertise the receiver window and the TCP sender adjusts congestion window (CWND) by perceiving the network congestion. The value of TCP sender window equals the minimum value of RCVWND and CWND. While there is no packet lost, the TCP transmitter is in congestion avoidance phase and the value of TCP sender window is increased by one during every b RTTs until it reaches RCVWND. Many TCP receiver implementations send one cumulative acknowledgment packet for two consecutive packets received, so b is typically 2 [64]. In this chapter, we will also set b to 2.

When there are packets lost, TCP Reno has two schemes for detecting the loss of data, which are triple-duplicate ACKs (shown in Fig.6.1) and timeouts (shown in Fig.6.2). When the TCP transmitter receives more than three duplicate ACKs, it indicates there is a light network congestion and the fast retransmission and fast recovery algorithms will be invoked to recover the losses without waiting for retransmission time-out. The Reno TCP transmitter sets the slow start threshold

to half of the current CWND and retransmits the missing segment. CWND is then increased by one segment on reception of each ACK. When the retransmission time-out occurs before the TCP transmitter receives three duplicate ACKs, it indicates that there is a heavy congestion in the network. The TCP source will respond by retransmitting the lost segment and entering a slow start phase. During slow start phase, CWND is increased by 1 MSS (Maximum Segment Size) for every acknowledgment packet. Once CWND exceeds slow start threshold, the TCP source enters into a congestion avoidance phase.

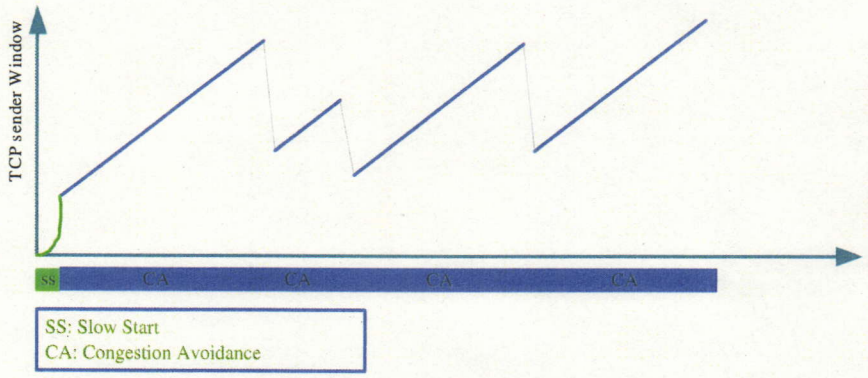


Figure 6.1: Burst loss indication by receiving triple-duplicate acknowledgments

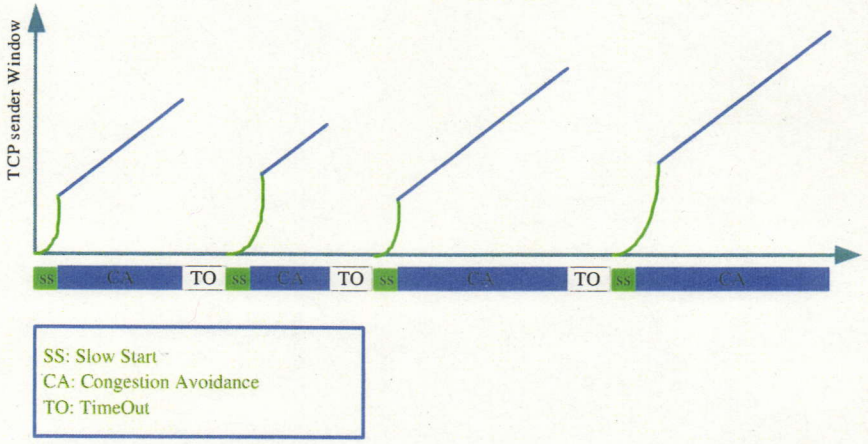


Figure 6.2: Burst loss indication when time-out occurs

6.3 TCP Performance Evaluation Model for OBS Networks

In this section, we develop an analytical model for evaluating the TCP performance of OBS network that exclusively affected by the burst loss events.

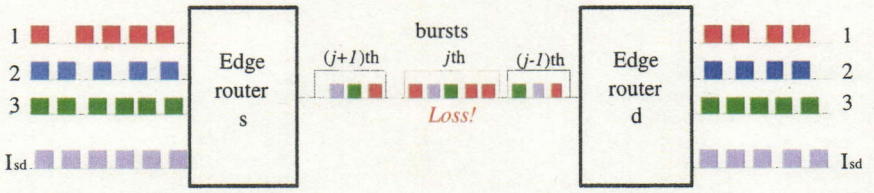


Figure 6.3: Burst assembly model

For one pair of edge router s and d (Fig.6.3), we assume I_{sd} as the TCP connection number between s and d, M denotes the average assembly granularity . We also suppose the average TCP sender window is W, and burst blocking probability is pb_{sd}. If there is one burst (the jth burst in Fig.6.3, we denote it by $burst_j$) dropped in the OBS network, many TCP connections will be caused to loss packets at the same time. Suppose there are k_i IP packets of the ith ($1 \leq i \leq I_{sd}$) TCP connection in this dropped burst. We may get

$$k_i = \min(\frac{Band_i \times T_b}{\overline{L_{IP}}}, W). \tag{6.1}$$

Where $Band_i$ is the access bandwidth of the ith TCP connection. T_b is the assembly period, $\overline{L_{IP}}$ is the average length of IP packets.

We define γ as the *correlation factor* of the ith TCP connection, and expressed by $\gamma = E[k_i]/W$. Suppose each TCP flow is independent and of identical distribution. So in $burst_j$, the IP packets belong to $M/(W\gamma)$ different TCP connections. Then, we get IP packets loss probability of the ith TCP connection caused by this burst loss event,

$$p_{IP} = \begin{cases} pb_{sd} \times [1 - \frac{\binom{I_{sd}-1}{M/(W\gamma)}}{\binom{I_{sd}}{M/(W\gamma)}}] & (M \leq W(I_{sd}-1)) \\ pb_{sd} & (M > W(I_{sd}-1)). \end{cases} \tag{6.2}$$

The TCP sender of the ith TCP connection can recover this error by fast recovery algorithm (when receiving triple-duplicate acknowledgment packets) or slow start algorithm (when time-out occurs). Generally, fast retransmission scheme can eliminate about half the coarse-grain timeouts.

J.padhye et. al [66] has proposed a model to evaluate the TCP performance. In this chapter, I would like to give a simpler one. Firstly, we assume the loss indications caused by burst loss events are exclusively "triple-duplicate acknowledgment packets". TDP (Triple-duplicate Period) is used to denote the period between two loss events.

According to TCP protocol, TCP sender window will increase when it receives acknowledgment packets until a loss event happens or the value of window reaches W_{max} (the maximum window can be used by TCP sender). If a loss event happens before the TCP sender window reaches W_{max}, there is no window limitation for TCP sender window during this TDP. The TDP can be described as Fig.6.4(a) and be simplified to Fig.6.4(b).

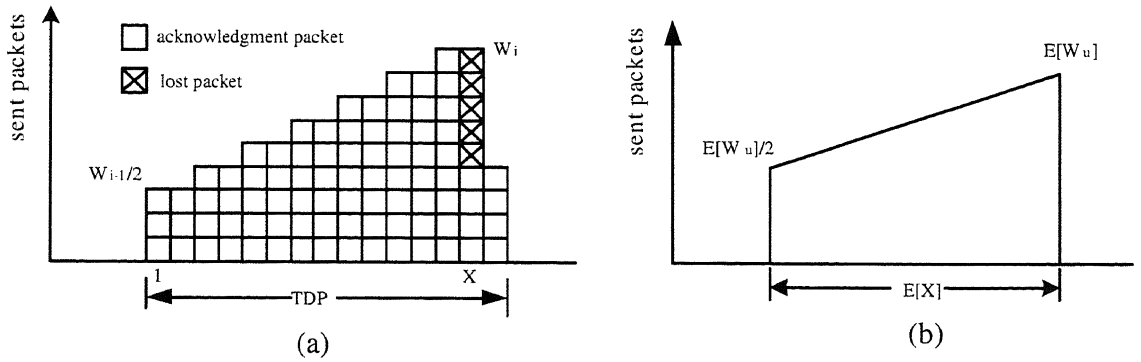


Figure 6.4: Triple-duplicate period without window limitation

In Fig.6.4(b), $E[X]$ is the average number of RTT(Round Trip Time) in one TDP, $E[W_u]$ is the maximum value of TCP sender window. We use $E[Y]$ to denote the average number of IP packets sent in one TDP. We have

$$E[Y] = \frac{E[X]}{2} \times (E[W_u] + \frac{E[W_u]}{2}) = \frac{1}{p_{IP}}. \tag{6.3}$$

It can be observed that $E[W_u]$=$E[X]$, then we get

$$E[X] = \sqrt{\frac{4}{3 \times p_{IP}}}. \tag{6.4}$$

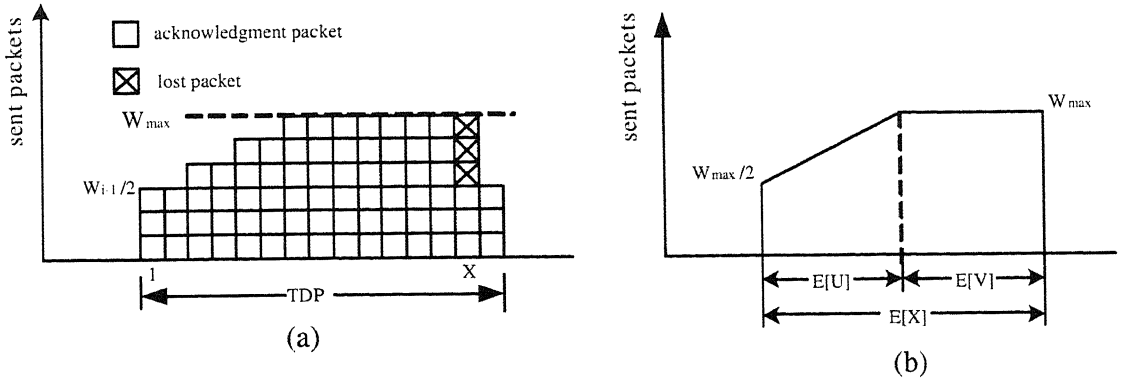


Figure 6.5: Triple-duplicate period with window limitation

According to Eq.(6.4), if $p_{IP} < \frac{4}{3 \times W_{max}^2}$, the sender window will reach W_{max} during this TDP. Then TDP can be described as Fig.6.5(a) and be simplified to Fig.6.5(b).

In Fig.6.5(b), $E[U]$=W_{max} and $E[V]$=$E[X]$–$E[U]$. Then the average number of IP packets sent in this period can be expressed by

$$E[Y]=\frac{E[U]}{2}(W_{max}+\frac{W_{max}}{2})+W_{max}E[V]=\frac{1}{p_{IP}}. \tag{6.5}$$

By substituting $E[U]$ and $E[V]$ into Eq.(6.4), we get

$$E[X] = \frac{\frac{1}{p_{IP}} + \frac{1}{4} \times W_{max}^2}{W_{max}}. \tag{6.6}$$

We utilize B_i to denote the throughput of ith TCP connection. B_i can be expressed by

$$B_i = \frac{E[Y] \times \overline{L_{IP}}}{E[X] \times RTT}. \tag{6.7}$$

Where $\overline{L_{IP}}$ is the average length of IP packets, in which RTT is the round trip time and is defined as $RTT = RTT_0 + 2T_b + 2offset\ time$. Where RTT_0 is the normal round trip time without assembling.

In conclusion, the throughput of the ith TCP flow is expressed by

$$B_i = \begin{cases} \frac{\overline{L_{IP}}}{RTT \times p_{IP} \times \sqrt{\frac{4}{3 \times p_{IP}}}} & (p_{IP} > \frac{4}{3W_{max}^2}) \\ \frac{W_{max} \times \overline{L_{IP}}}{RTT(1+\frac{1}{4} \times W_{max}^2 \times p_{IP})} & (otherwise). \end{cases} \tag{6.8}$$

Now we will discuss the probability that IP loss indication is time-out. Assume the IP packets of the ith TCP connection in one RTT distribute in N bursts. Let k_i

be the number of IP packets of the ith TCP connection in $burst_j$ (shown in Fig.6.3), a_i be the number of IP packets of the ith TCP connection in the earlier $(j-1)$ bursts and b_i be the number of IP packets of the ith TCP connection in the later $(N-j)$ bursts.

According to TCP protocol, if $burst_j$ is dropped in the OBS network, the ith TCP receiver will send back a_i+b_i acknowledgment packets with the same sequence number. We utilize Q to denote the probability that loss indication is time-out, and Q is expressed by

$$Q=\begin{cases} 0 & (W>M+2) \\ E[p(a_i+b_i=2/k_i\neq 0)]=\frac{q^M}{1-q^M} & (W=M+2) \\ \sum_{l=1}^{2}E[p(a_i+b_i=l/k_i\neq 0)]=\frac{q^M+M(1-q)q^M}{1-q^M} & (W=M+1) \\ \sum_{l=0}^{2}E[p(a_i+b_i=l/k_i\neq 0)]=\frac{\sum_{j=W-2}^{W}\binom{M}{j}q^j(1-q)^{M-j}}{\sum_{j=1}^{W}\binom{M}{j}q^j(1-q)^{M-j}} & (W\leq M), \end{cases} \tag{6.9}$$

where W is the size of TCP sender window, q is the probability that a burst belongs to the ith TCP flow. They can be calculated by $W=a_i+k_i+b_i$ and $q=1/I_{sd}$.

Observe that when $M < W \times I_{sd}$, Q is very small and approximately equals zero. For example, $Q=4.0\times 10^{-95}$ when I=10, W=100, and M=100; $Q=3.4\times 10^{-124}$ when I=20, W=100, and M=100. So we have the conclusion that time-out indication is negligible in the case that IP packets loss event is caused by a burst loss event.

Throughput of each TCP connection is decided by Eq.(6.8) and the total throughput of the burst link between s and d can be described as $B_{sd}=\sum_{i=1}^{I_{sd}}B_i$.

The common burst assembly is a combination of timer-based and threshold-based approaches [25]. The assembly period can be expressed by

$$T_b=\begin{cases} T_{max} & (B_{sd}\leq L_{max}\times T_{max}) \\ \frac{B_{sd}}{L_{max}} & (otherwise), \end{cases} \tag{6.10}$$

where T_{max} is the maximum assembly period for creating a burst, L_{max} is the maximum burst length.

The average sender window in Eq.(6.1) can be expressed by $W=E[Y]/E[X]$, where $E[X]$ is given in Eq.(6.4) and Eq.(6.6) and $E[Y]=1/p_{IP}$.

From the discussion above, we may have a function of k_i, p_{IP}, B_{sd}, T_b and W, which is expressed by

$$F(k_i, p_{IP}, B_{sd}, T_b, W) = 0. \tag{6.11}$$

The analytical solution of the nonlinear Eq.(6.11) can be got through successive substitution by using Newton-Raphson method.

6.4 OBS Layer Error Recovery Scheme

6.4.1 Statement of Problems

TCP is intended to provide a reliable transport layer over an unreliable network layer. TCP includes mechanisms for acknowledging received data and resending lost data. It also provides a flow control mechanism that reduces the sending rate if congestion is detected in the network. However, there are some limitations of TCP recovery mechanism in OBS networks. Firstly, due to that one burst contains multiple packets belong to different TCP links, a burst dropping event may cause many TCP connections to loss packets and start recovery mechanism at the same time. This may result in Global Synchronization. Secondly, TCP recovery mechanism is slow (shown in Fig.6.6) and inefficient because it needs to wait for timeout and reassemble lost packets into bursts. In this section, we introduce a novel burst layer fast restoration scheme which is based on a burst acknowledgment mechanism.

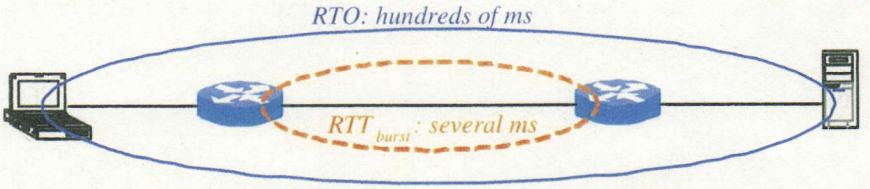


Figure 6.6: Illustration of limitations of TCP over OBS

General OBS network are connectionless networks. An ingress router will transmit bursts into an OBS network as it receives IP packets. An egress router doesn't send correct receipt of transmitted bursts back to the ingress router. There is no

loss detection or error recovery mechanism on the burst layer in OBS networks. In the following, we will first introduce a burst acknowledgment scheme.

6.4.2 Burst Acknowledgment Scheme

Generally, a basic BHP cannot support any acknowledgment function. Therefore, we present our Burst Head Packet (BHP) format first. An example of the BHP format is shown in Fig.6.7. Where ***Dest_Edge*** denotes the id of the egress edge

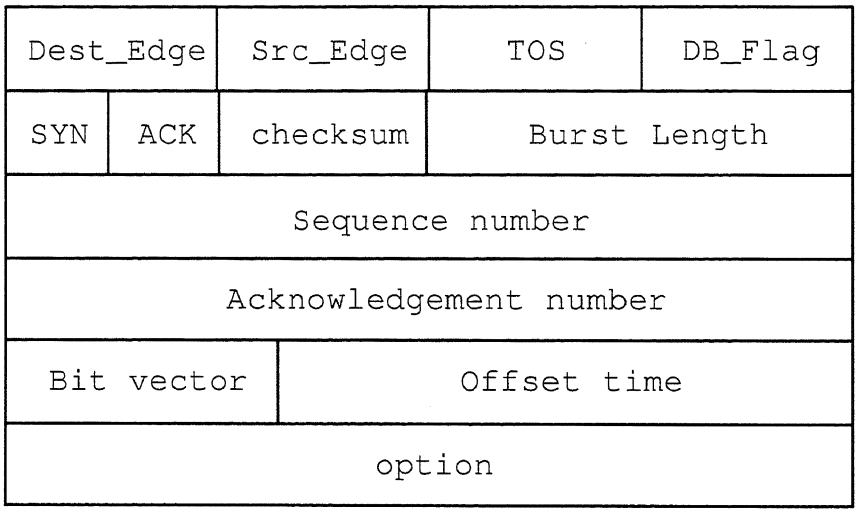


Figure 6.7: Example of BHP format for burst acknowledgment mechanism

router, ***Src_Edge*** denotes the id of the ingress edge router. ***TOS*** is burst type field; ***DB_flag*** is the id of the channel used by corresponding data burst. ***Burst Length*** denotes the length of payload. ***Offset time*** is the gap between the BHP and corresponding DB.

The fields we introduced above are the same as those of general OBS networks. In addition to these fields, we also append several new fields which are necessary for our scheme. ***SYN*** is sequence number field significant; ***ACK*** is Acknowledgment number field significant. ***Sequence number*** field will be filled with a monotonically increasing number when ACK field is set. ***Acknowledgment number*** field will be filled with the received Sequence number when ACK field is set. An n-bit ***Bit vector*** indicates the arrival status of the earlier n bursts. The burst acknowledgment scheme works as follows:

1. Because TCP and UDP traffic have different QoS attributes, TCP packets and UDP packets are assembled to different bursts. For each TCP burst, the ingress node sets SYN bit of the BHP and fills the Sequence number with a monotonically increasing number.

2. When an egress node receives a marked burst successfully, it creates an acknowledgment burst. An acknowledgment burst only contains a BHP with no payload and will not occupy any data channels when it is transmitted in the OBS network. The ACK bit is set and Acknowledgment number field is filled with a suitable value. Bit vector is composed of n bits. If the ith bit is set 1, it indicates that the earlier $(n-i+1)$th burst has been received successfully. Of course, the Src_Edge and Dest_Edge field should also be filled with suitable values. ACK and SYN can't be set at the same time.

3. When an ingress edge node receives an acknowledgment burst, it can perform loss detection function based on the information carried by the acknowledgment burst.

6.4.3 Edge Buffering Based OBS Layer Retransmission Scheme

In this subsection, we will present an OBS layer error recovery scheme by use of edge electronic buffering. An OBS network needs to connect with other types of networks. These client networks are often electrical. As we have introduced before, there is no optical RAM, we can only use expensive Fiber Delay Lines (FDLs) to achieve limited buffering. However, we can also use inexpensive and abundant electronic buffering in the interfaces between OBS and client networks. The OBS layer error recovery scheme works as follows:

1. For each marked burst, we store a copy of this burst in the electronic buffer of the ingress node.

2. When the ingress node receives an acknowledgment burst, it checks the value of the Bit Vector. If there are some bits of the vector equal zero, it means the corresponding bursts have been dropped during the transmission period. The

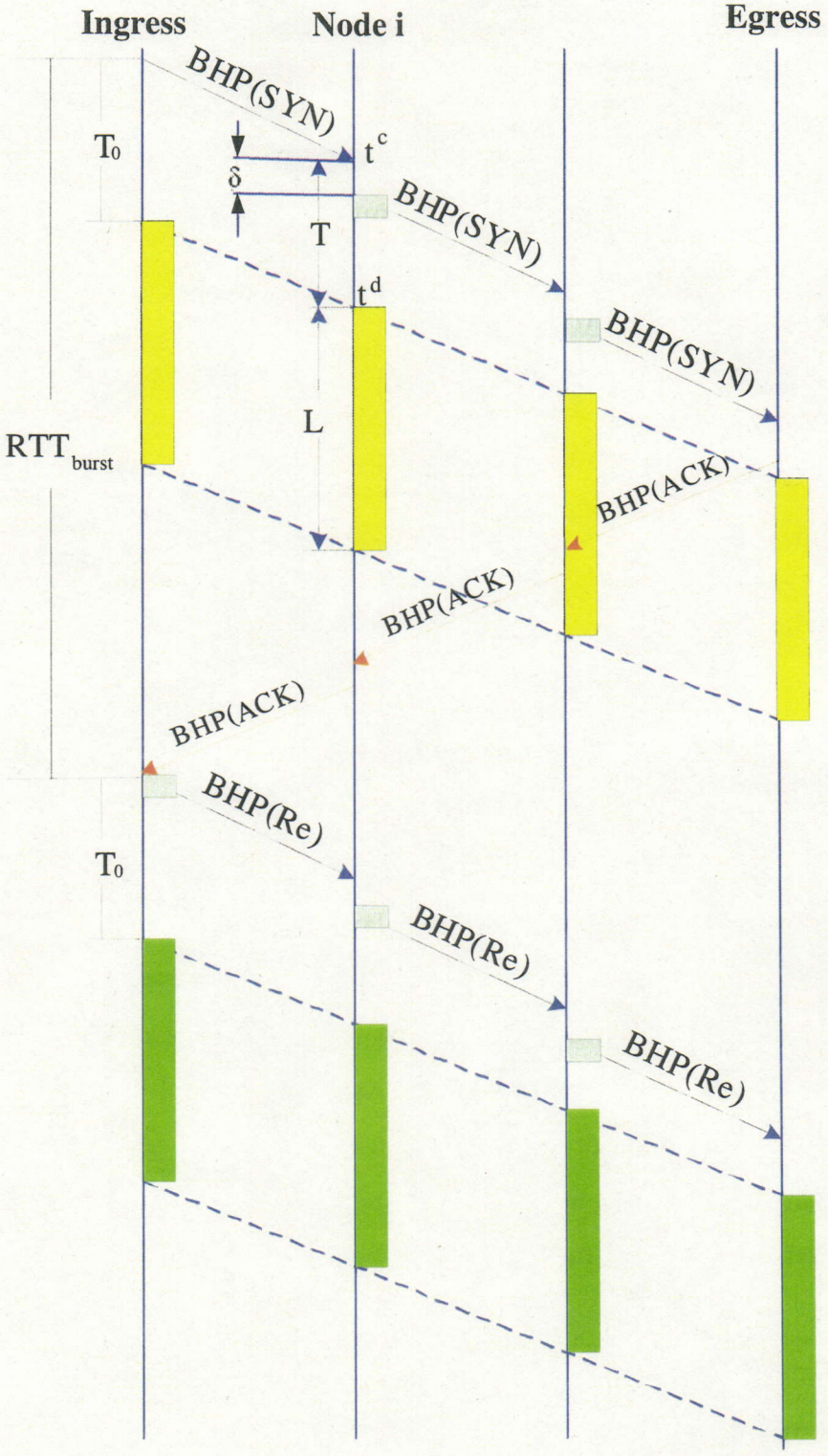


Figure 6.8: Fast restoration mechanism in OBS

ingress node will retransmit these bursts. To avoid the burst being hold for a too long period and the TCP packets being disordered too much, the buffer which is occupied by these n bursts will be released whether corresponding bursts could been transmitted successfully. If the retransmitted bursts are also lost, the error will be recovered by the TCP layer.

Next we will discuss the electronic memory that is needed by our proposed scheme. There are two signaling schemes in OBS networks, Tell-And-Go (TAG) [47, 49] and Just-Enough-Time (JET) [63]. In this chapter, we adopt JET as our signaling mechanism. In JET, a large offset time is used to compensate the time for processing BHP and configuring optical switching fabrics before its DB arrives at the immediate nodes. Thus, no fiber delay lines are necessary at the intermediate nodes. The transmission delay from an ingress node to an egress node is mainly propagation delay.

$$propagation\ delay = distance/light\ speed. \tag{6.12}$$

We describe the period which is between the ingress router sends a burst and receives an acknowledgment burst as RTT_{burst}, and get

$$RTT_{burst} = 2 \times propagation\ delay + T_{extra}. \tag{6.13}$$

Where T_{extra} is the processing delay for the egress node to create an acknowledgment burst. In this chapter, we suppose it is very small and negligible when comparing to propagation delay. So, if there is no burst loss event, the ingress node only needs to hold the bursts for RTT_{burst}. When some bursts lost, the edge node needs to hold the burst for a longer period and the buffer will be released by a later burst. If the burst loss event is detected by the later kth $(1 \leq k \leq n)$ burst, the burst assembly period is T_b, then the lost burst holding period equals $RTT_{burst} + k \times T_b$.

Suppose the burst lost events are independent and the burst loss ratio is pb, the switching capacity of the edge node is C, then the maximum electronic memory which is required for buffering bursts can be expressed by

$$Memory = C \times (RTT_{burst} + pb \times (RTT_{burst} + k \times T_b)). \tag{6.14}$$

For example, if the distance between an ingress node and an egress node is 300 km, the propagation delay approximately equals 1ms. Assume the switching capacity of the edge node is 10 Gbits/s, the burst loss probability is 0.001, assembly delay is 1ms, $k = 10$. It only needs approximately 20.2 Mbits electronic memory for buffering bursts.

In many TCP implementations, the value of Retransmission Timeout (RTO) (typically, $RTO = 0.5sec$) is much larger than burst retransmission delay, so our electrical buffering mechanism can recover burst loss error and avoid reassembling burst before TCP time-out occurs. When more than n sequential bursts loss, our proposed mechanism cannot detect the loss event. However, the probability is very small, and equals pb^n. It is negligible unless when there are some mechanical errors and pb is very large.

So in an OBS network that adopts our OBS layer error recovery scheme, a burst will be lost only when its retransmitted burst is also lost. Assume the probability is pb', $pb' = pb \times pb$. And IP packets loss probability in Eq.(6.2) can be replaced by

$$p_{IP} = \begin{cases} pb_{sd} \times pb_{sd} \times [1 - \frac{\binom{I_{sd}-1}{M/(W\gamma)}}{\binom{I_{sd}}{M/(W\gamma)}}] & (M \leq W(I_{sd}-1)) \\ pb_{sd} \times pb_{sd} & (M > W(I_{sd}-1)), \end{cases} \tag{6.15}$$

where pb_{sd}, M, I_{sd}, W are defined as before.

6.5 Numerical Results

In this section, we study the performance of our proposed TCP models and the OBS layer error recovery mechanism presented in the previous sections by simulation. The simulated scenario is shown in Fig.6.9. We use ftp traffic as input to this network. Assume $W_{max} = 100$, TCP connection number $I_{sd} = 100$, access bandwidth of each TCP flow is 100 Mbits/s, the maximum assembly length L_{max}=150 Kbytes, the maximum assembly period T_{max}=1ms, RTT=10 ms, RTT_{burst}=2ms.

The results in Fig.6.10 show the performance of the conventional OBS network under TCP recovery scheme and an OBS network under our OBS-layer retransmission mechanism. First, we observe a good agreement between the analytical and the simulation results. We also observe that as the burst loss ratio increases, the

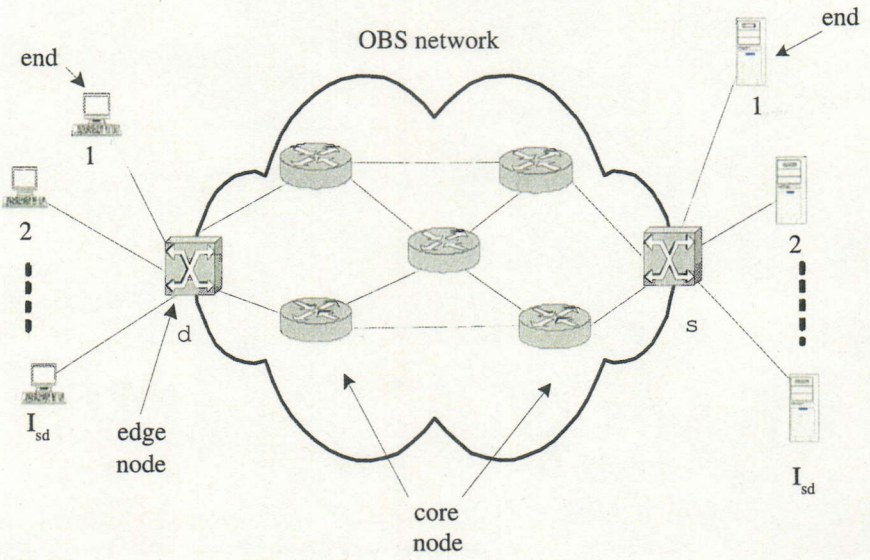


Figure 6.9: Simulation network topology

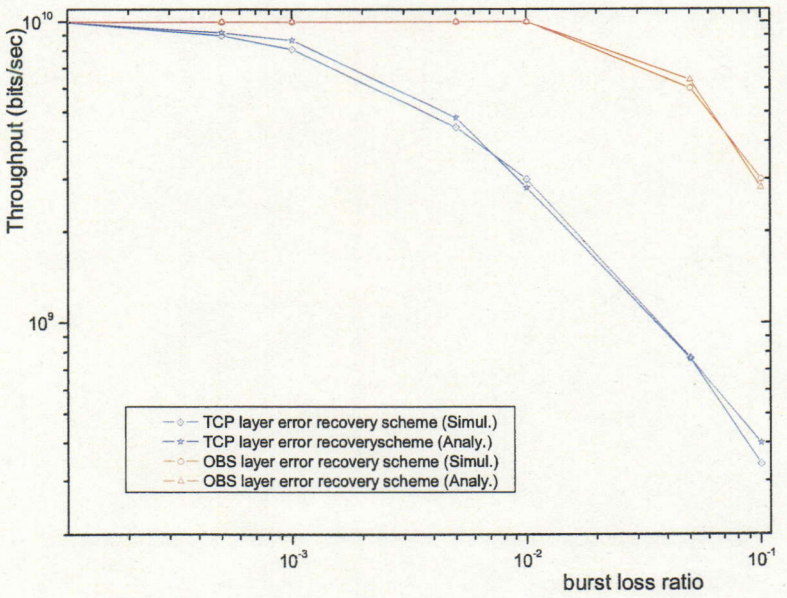


Figure 6.10: Throughput vs burst loss ratio

throughput between the edge node s and d decreases. At a low burst loss ratio, such as 0.005, the burst loss errors are almost recovered by our OBS layer retransmission recovery mechanism. At a high burst loss ratio, such as 0.05, the throughput of the conventional OBS network and the throughput of the OBS network under OBS-layer retransmission mechanism are 7.6×10^8 bits/s and 6.4×10^9 bits/s respectively. So although a network with an OBS layer retransmission recovery mechanism couldn't recover all errors, it has better performance than a network under TCP error recovery mechanism.

According to Eq.(6.14), in the OBS network under OBS-layer retransmission error recovery mechanism, the occupied electronic buffer also varies with the burst loss ratio. At low burst loss ratio, the memory is mainly occupied by the normal bursts, and will increase with the increase of burst loss ratio slightly. However, at high burst loss ratios (greater than 0.01), when the burst loss ratio increases, the occupied electronic buffer will decrease due to the decrease of the throughput.

6.6 Summary

In this chapter, an analytical model, allowing evaluating the TCP performance of an OBS network, has been presented. The results of this analytical model fit accurately with simulation results. The developed model is very useful for designing OBS networks or evaluating the performance of OBS networks.

We have also proposed an OBS layer loss detection and error recovery mechanism by use of electrical buffering for OBS networks. The analytical and simulation results show that our proposed scheme can improve the TCP performance of the OBS network.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

In this dissertation, we have studied QoS control and performance improvement methods in OBS networks. The terminology of our study is presented first.

We described how to provide proportional differentiated services in an OBS network. First, we introduced a priority scheme, called dynamic wavelength selection (DWS), which can efficiently support proportional differentiated services in bufferless OBS networks. We compared the performances of this scheme and the intentional dropping scheme in a series of simulations. The simulation results showed that the DWS scheme has better performance in terms of burst loss ratio and wavelength utilization.

We also proposed a delayed burst assignment scheme and integrated it with the DWS scheme to support proportional differentiated services in OBS networks. It was shown that the integrated scheme has the best performance. Through simulation, we also found that when the BHP waiting time period (T_{wait}) and the maximum buffering times (N_r) exceed some values, the performance is improved smoothly. These results could prevent the end-to-end delay from becoming too large.

We applied a Fractional Brownian motion (FBM) model to analyze the effect of OBS assembly mechanisms on the burstiness of SINET traffic. Our simulation and analytical results showed that both the timer-based and the threshold-based assembly algorithms could not reduce the bustiness of real traffic in short timescales.

We therefore proposed two novel burst assembly algorithms with traffic smoothing functions: the advanced timer-based assembly algorithm and the sliding window-based algorithm. The simulation results have shown that the advanced timer-based assembly is best, because it is able to not only reduce the burst loss ratio but also control the edge buffering delay. The sliding window-based algorithm was also able to reduce the burst loss ratio, especially at low traffic loads at the expense of a small initial delay.

An analytical model, allowing evaluating the TCP performance of an OBS network, has been presented. The results of this analytical model fit accurately with simulation results. The developed model is very useful for designing OBS networks or evaluating the performance of OBS networks.

We have also proposed a loss detection and error recovery mechanism by use of electrical edge buffering for OBS networks. The analytical and simulation results show that our mechanism can improve the TCP performance of the OBS network.

7.2 Future Works

The following are some of the possible areas of future work based on individual chapters in the dissertation.

In Chapter 3, we have achieved per-hop proportional differentiation in terms of burst loss ratio. However, it is not straightforward to achieve end-to-end proportional differentiated burst loss ratio via that single-hop approach. The mismatch is illustrated by the following example, used in [57,58]. In a two-hop network, suppose the loss ratio of class i is $s_i \times l_a$ at node a and $s_i \times l_b$ at node b, respectively; then the end-to-end loss ratio relations is:

$$\begin{aligned} \frac{l_i}{l_j} &= \frac{s_i \times l_a + (1 - s_i \times l_a) \times s_i \times l_b}{s_j \times l_a + (1 - s_j \times l_a) \times s_j \times l_b} \\ &= \frac{s_i \times (l_a + l_b) - l_a \times l_b \times s_i^2}{s_j \times (l_a + l_b) - l_a \times l_b \times s_j^2} \neq \frac{s_i}{s_j}. \end{aligned} \tag{7.1}$$

From Eq.(7.1), the end-to-end loss ratios will deviate from the predefined factors on individual nodes.

One possible solution proposed in [58] is that the dropping decision at each intermediated node should calculate loss ratios per-flow instead of packet arrival at this node. Our future research will deal with how to provide end-to-end service differentiation with DWS and DBA schemes.

In Chapter 6, we proposed a novel burst layer fast restoration scheme to improve TCP performance. It can also be used for UDP flows in future work. As an unreliable transport layer protocol, UDP has neither congestion control nor error recovery mechanism. Unlike TCP flows, UDP senders don't require the correct receipt of all transmitted packets. However, as the Internet applications become diverse, reliability requirement also becomes varied. Many applications needn't retransmit all lost packets, but if we can transmit some special packets, the performance will be improved distinctly.

Bibliography

[1] S.Chatterjee and S.Pawlowski, "All-optical networks," Communications of the ACM, vol.42, no.6, pp.74–83, June 1999.

[2] D.Banerjee and B.Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," IEEE Journal on Selected Areas in Communications, vol.14, pp.903–908, June 1996.

[3] H.Zang, J.P.Jue and B.Mukherjee, "A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks," SPIE Optical Networks Magazine, vol.1, no.1, January 2000.

[4] I.Chlamtac, A.Farago and T.Zhang, "Lightpath(wavelength) routing in large WDM networks," IEEE Journal on Selected Areas in Communications, vol.14, no.5, pp.909–913, June 1996.

[5] I.Chlamtac, A.Ganz and G.Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's," IEEE Transactions on Communications, vol.40, pp.1171–1182, July 1992.

[6] A.Birman, "Computing approximate blocking probabilities for a class of all-optical networks," IEEE Journal on Selected Areas in Communications, vol.14, pp.852–857, June 1996.

[7] E.Karasan and E.Ayanoglu, "Performance of WDM transport networks," IEEE Journal on Selected Areas in Communications, vol.16, pp.1081–1096, September 1998.

[8] S.Xu, L.Li, and S.Wang, "Dynamic routing and assignment of wavelength algorithms in multifiber wavelength division multiplexing networks," IEEE Journal on Selected Areas in Communications, vol.18, pp.2130–2137, October 2000.

[9] B.Li, A.Ganz and C.M.Krishna, "An in-band signaling protocol for optical packet switching networks," IEEE Journal on Selected Areas in Communications, vol.18, no.10, pp.1876–1884, October 2000.

[10] D.Blumenthal, P.Prucnal and J.Sauer, "Photonic packet switches - architectures and experimental implementations," in Proceedings of the IEEE, vol.82, pp.1650–1667, November 1994.

[11] C.Qiao and M.Yoo, "Choices, Features and Issues in Optical Burst Switching(OBS)," Optcal Networking Magazine, vol.1, no.2, pp.36–44, April 2000.

[12] S.Yao, S.J.B.Yoo, B.Mukherjee and S.Dixit, "All-optical packet switching for metropolitan area networks: Opportunities and challenges," IEEE Communication Magazine, vol.39, pp.142–148, March 2001.

[13] M.C.Cardakli and A.E.Willner, "Optical packet and bit synchronization of a switching node using fbg optical correlator," in Proceedings, Optical Fiber Communication Conference and Exhibit(OFC), March 2001.

[14] F.Callegati, M.Casoni, C.Raffaelli and B.Bostica,"Packet optical networks for high-speed TCP-IP backbones," IEEE Communications Magazine, vol.37, no.1, pp.124–129, January 1999.

[15] T.S.El-Bawab and J.D.Shin, "Optical Packet Switching in Core Networks: Between Vision and Reality," IEEE Communications Magazine, vol.40, no.9, pp.60–65, September 2002.

[16] C.Guillemot and M.Renaud et al., "Transparent Optical Packet Switching: The European ACTS KEOPS Project Approach," IEEE/OSA Journal of Lightwave Technology, vol.16, no.12, pp.2117–2134, December 1998.

[17] M.J.OMahony, D.Simeonidou, D.Hunter and A.Tzanakaki, "The application of optical packet switching in future communication networks," IEEE Communications Magazine, vol.39, pp.128–135, March 2001.

[18] D.K.Hunter and I.Andronovic, "Approaches to optical Internet packet switching," IEEE Communications Magazine, vol.38, pp.116–122, September 2000.

[19] F.Callegati, A.C.Cankaya, Y.Xiong and M.Vandenhoute, "Design issues of optical IP routers for Internet backbone applications," IEEE Communications Magazine, vol.37, pp.124–128, December 1999.

[20] A.Jourdan, D.Chiaroni, E.Dotaro, G.J.Eilenberger, F.Masetti and M.Renaud, "The perspective of optical packet switching in IP dominant backbone and metropolitan networks," IEEE Communications Magazine, vol.39, pp.136–141, March 2001.

[21] S.Amstutz, "Burst switching - an introduction," IEEE Communications Magazine, vol.21, pp.36–42, November 1983.

[22] X.Cao, Anand.V, Y.Xiong and C.Qiao, "A study of waveband switching with multilayer multigranular optical cross-connects," IEEE Journal on Selected Areas in Communications, vol.21, no.7, pp.1081–1095, September 2003.

[23] T.S.P.Yum, F.Tong and K.T.Tan, " An architecture for IP over WDM using time-division switching," IEEE/OSA Journal of Lightwave Technology, vol.19, no.5, pp.589–595, May 2001.

[24] C.Qiao and M.Yoo, "Optical Burst Switching (OBS)-A new paradigm for an optical Internet," Journal of High Speed Networks, vol. 8, no.1, pp.69–84, January 1999.

[25] Y.J.Xiong, M.Vandenhoute and H.C.Cankaya, "Control architecture in optical burst-switched WDM networks," IEEE Journal on Selected Areas in Communications, vol.18, no.10, pp.1838–1851, October 2000.

[26] M.Yoo and C.Qiao, "A novel switching paradigm for buffer-less WDM networks," IEEE Communications Magazine, 1999.

[27] I.Widjaja, "Performance analysis of burst admission-control protocols," IEE proceeding of Communications, vol.142, pp.7–14, February 1995.

[28] P.Du and S.Abe, "Dynamic Wavelength Selection and Delayed Burst Assignment Schemes for Optical Burst Switching Networks," Submitted to IEICE Transactions on Communications.

[29] P.Du and S.Abe, "Traffic Analysis and Traffic-smoothing Burst Assembly Methods for the Optical Burst Switching Network," Accepted by IEICE Transactions on Communications.

[30] P.Du and S.Abe, "TCP Performance Analysis of Optical Burst Switching Networks with a Burst Acknowledgment Mechanism," APCC2004, vol.2, pp.621–625.

[31] P.Du and S.Abe, "A Proportional Differentiated Services Scheme for Optical Burst Switching Networks," IASTED International Conference, OCSN2005, pp.126–131.

[32] P.Du and S.Abe, "Sliding window-based Burst Assembly Method in Optical Burst Switching Networks," ICON2006, pp.355–360.

[33] P.Du and S.ABE, "Burst Assembly Method with Traffic Shaping for the Optical Burst Switching Network," In Proceedings, IEEE GLOBECOM 2006.

[34] P.Du and S.Abe, "DDCS-based Proportional Differentiated Services for OBS Networks," Technical Report of IEICE, OCS2005–18.

[35] P.Du and S.Abe, "An Edge Buffering Based Fast Restoration Scheme for Optical Burst Switching Networks," Society Conference of IEICE, SE29–30, 2005.

[36] P.Du and S.Abe, "Traffic Characteristics of Assembled Burst Traffic for Optical Burst Switching Networks," Technical Report of IEICE, NS2005–10.

[37] P.Du and S.Abe, "An Advanced Timer-based Burst Assembly Algorithm with Traffic Shaping in Optical Burst Switching Networks," General Conference of IEICE, BS-8-1, 2006.

[38] P.Du and S.Abe, "A Rescheduling Scheme for Providing Joint QoS in Optical Burst Switching Networks," Society Conference of IEICE, SE20-14, 2006.

[39] M.Izal and J.Aracil, "On the influence of self-similarity in optical burst switching traffic," in Proceedings, IEEE GlOBECOM 2002.

[40] X.Yu, Y.Chen and C.Qiao,"A Study of traffic statistics of assembled burst traffic in optical burst switched networks," in Proceedings, SPIE Optical Networking and Communication Conference (OptiComm) 2002, pp. 149–159, 2002.

[41] S.Abe, T.Hasegawa and S.Asano,"Traffic analysis and network bandwidth provisioning toools for academic information networks," Progress of Informatics, No.1, pp.83–91, March 2005.

[42] http://www.sinet.ad.jp

[43] W.E.Leland, M.S.Taqqu, W.Willger and D.V.Wilson,"On the self-similar nature of Ethernet traffic (extended version)," IEEE/ACM Transactions on Networking, vol.2, pp.1–15, February 1994.

[44] V.Paxson ans S.Floyd, "Wide-area traffic: the failure of Poisson modeling," IEEE/ACM Transactions on Networking, vol.3, pp.226–244, June 1995.

[45] I.Norros,"On the use of fractional brownian motion in the theory of connectionless network," IEEE Journal on Selected Areas in Communications, vol.13, no.6, pp.953–962, 1995.

[46] M.Yoo, C.Qiao and S.Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," IEEE Journal on Selected Areas in Communications, no.10, pp.2062–2071, October 2000.

[47] A.Detti and M.Listanti, "Application of Tell & Go and Tell & Wait Reservation Strategies in an Optical Burst Switching Network: a performance Comparison," Proceedings of IEEE Int'l Conf.on Telecommunication (ICT), vol.2, pp.540–548, June 2000.

[48] X.Wang, H.Morikawa and T.Aoyama, "Priority-based wavelength assignment algorithm for burst switched WDM optical networks," IEICE Transaction on Communications, vol.E86-B, no.5, pp.1508–1514, 2003.

[49] J.Y.Wei, J.L.Pastor, R.S.Ramanurthy and Y.Tsai, "Just-in-time signaling for WDM optical burst switching networks," IEEE Journal of Lightwave Technology, vol.18, pp.2019–2037, December 2000.

[50] http://www.sinet.ad.jp/sinet/sinet_kaisen_chizu_1.htm

[51] H.Chaskar, S.Verma and R.Ravikanth, "A framework to support IP over WDM using optical burst switching," In Proceedings of Optical Networks Workshop, 2000.

[52] R.Morris and D.Lin, "Variance of aggregated web traffic," in Proceedings, IEEE INFOCOM 2000.

[53] Y.Chen, M.Hamdi and Danny H.Tsang, "Proportional QoS over OBS networks," in Proceedings, IEEE GLOBECOM 2001.

[54] H.C.Cankaya, S.Charcranoon and T.S.El-Bawab, "A preemptive scheduling technique for OBS networks with service differentiation," in Proceedings, IEEE GLOBECOM 2003, no.1, pp.2704–2708, December 2003.

[55] V.M.Vokkarane and J.P.Jue, "Prioritized burst segmentation and composite burst-assembly techniques for QoS support in optical burst-switched networks," IEEE Journal on Selected Areas in Communications, no.7, pp.1198–1209, September 2003.

[56] C.Dovrolis and P.Ramanathan, "A case for relative differentiated services and the proportional differentiation model," IEEE Network Magazine, no.5, pp.26–34, October 1999.

[57] Y.Chen, M.Hamdi, D.H.K.Tsang and C.Qiao, "Proportional differentiation-A scalable QoS approach," IEEE Communications Magazine, vol.41, pp.52–58, June 2003.

[58] A.Kumar, J.Kaur and H.M.Vin, "End-to-End Proportional Loss Differentiation," Techical report TR-01-33, Univ.of TX Austin, September 2001.

[59] I.Ogushi, S.Arakawa, M.Murata and K.Kitayama, "Parallel Reservation Protocols for Achieving Fairness in Optical Burst Switching," in Proceedings of IEEE Workshop on High Performance Switching and Routing, pp.213–217, 2001.

[60] B.Zhou, M.Bassiouni and G.Li, "Improving fairness in optical-burst-switching networks," Journal of Optical Networks, vol.3, no.4, pp.214–228, March 2004.

[61] A.Kaheel, T.Khattab, A.Mohamed and H.Alnuweiri, "Quality-of-Service Mechanisms in IP-over-WDM Networks," IEEE IEEE Communications Magazine, vol.40, no.12, December 2002.

[62] M.Iizuka, M.Sakuta, Y.Nishino and I.Sasase, "A Scheduling Algorithm Minimizing Voids Generated by Arriving Bursts in Optical Burst Switched WDM Network," in Proceedings, IEEE GLOBECOM 2002.

[63] M.Yoo and C.Qiao, "Just-enough-time (JET): a high speed protocol for bursty traffic in optical network," IEEE/LEOS Conference on Technologies for a Global Information Infrastructure, pp.26–27, August 1997.

[64] W.Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, January 1997.

[65] W.Stevens, "TCP/IP Illustrated, Volume 1: The protocols," Addison-Wesley Professional Computing Series.

[66] J.Padhye and V.Firoiu, "Modeling TCP Reno performance: a simple model and its empirical validation," IEEE/ACM Transaction on Networking, vol.8, no.2, pp.133–145, April 2000.

[67] A.Detti and M.Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks," in Proceedings, IEEE INFOCOM 2002.

[68] Q.Zhang, V.M.Vokkarane, B.Chen and J.P.Jue,"Early drop and wavelength grouping schemes for providing absolute QoS differentiation in optical burst-switched networks," in Proceedings, GLOBECOM 2003.

[69] C.Dovrolis and P.Ramanathan,"Proportional differentiated services: Delay differentiation and packet scheduling," IEEE/ACM Transaction on Networking, vol.10, pp.12–26, February 2002.

[70] Y.Chen, M.Hamdi, D.H.K.Tsang and C.Qiao, "Proportional QoS provision: a uniform and practical solution," in Proceedings, IEEE ICC 2002, vol.4, pp.2363–2367, May 2002.

[71] D.K.Hunter, M.C.Chia and I.Andonovic, "Buffering in Optical Packet Switches," IEEE/OSA Journal of Lightwave Technology, vol.16, no.12, pp.2081–2094, December 1998.

[72] T.Sakamoto, "Variable Optical Delay Circuit Using Wavelength Converts," IEEE Electronics Letters, vol.37, pp.454–455, 2001.

[73] D.K.Hunter, W.D.Cornwell, T.H.Gilfedder, A.Franzen and I.Andonovic, "SLOB: A switch with large optical buffers for packet switching," IEEE/OSA Journal of Lightwave Technology, vol.16, no.10, pp.1725–1736, October 1998.

[74] I.Chlamtac, A.Fumagalli and L.G.Kazovsky et al., "CORD: Contention resolution by delay lines," IEEE Journal on Selected Areas in Communications, vol.14, no.5, pp.1014–1029, June 1996.

[75] Z.Haas, "The Staggering Switch: An electronically controlled optical packet switch," IEEE/OSA Journal of Lightwave Technology, vol.11, no.5/6, pp.925–936, May 1993.

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

[77] B.Ramamurthy and B.Mukherjee, "Wavelength conversion in WDM networking," IEEE Journal on Selected Areas in Communications, vol.16, no.7, pp.1061–1073, September 1998.

[78] A.Acampora and I.Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," IEEE Transactions On Communications, vol.40, no.6, pp.1082–1090, June 1992.

[79] F.Forghieri, A.Bononi and P.Prucnal, "Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks," IEEE Transactions On Communications, vol.43, no.1, pp.88–98, January 1995.

[80] A.Bononi, G.A.Castanon and O.K.Tonguz, "Analysis of hot-potato optical networks with wavelength conversion," IEEE/OSA Journal of Lightwave Technology, vol.17, no.4, pp.525–534, April 1999.

[81] V.Vokkarane, J.P.Jue and S.Sitaraman, "Burst segmentation: an approach for reducing packet loss in optical burst switched networks," in Proceedings, IEEE ICC 2002, vol.5, pp.2673–2677, 2002.

[82] X.Cao, J.Li, Y.Chen and C.Qiao, "TCP/IP packets assembly over optical burst switching network," in Proceedings, IEEE GLOBECOM 2002.

[83] J.Li, C.Qiao, J.Xu and D.Xu, "Maximizing throughput for optical burst switching networks," in Proceedings, INFOCOM 2004.

[84] Y.Chen, C.Qiao and X.Yu, "Optical Burst Switching (OBS): A New Area in Optical Networking Research," IEEE Network Magazine. vol.18, pp.16–23, May 2004.

[85] J.Xu, C.Qiao, J.Li and G.Xu, "Efficient channel scheduling algorithms in optical burst switched networks," in Proceedings, IEEE Infocom 2003.

[86] A.Ge, F.Callegati and L.Tamil, " On optical burst switching and Self-similar traffic," IEEE Communications Letters, March 2000.

[87] L.Tancevski, A.Ge, G.Castanon and L.Tamil, "A new scheduling algorithm for asynchronous, variable length IP traffic incorporating void filling," In Proceedings, OFC 1999.

[88] G.Hu, K.Dolzer and C.Gauger, "Does burst assembly really reduce the self-similarity," In Proceedings, OFC 2003.

[89] C.Dovrolis and P.Ramanathan, "Dynamic Class Selection: From Relative Differentiation to Absolute QoS," In Proceedings, IEEE Int'l Conf. Network Protocols, November 2001.

[90] I.Chlamtac, A.Gam and G.Karmi, "Lightpath Communications: An Approach to High Bandwidth Optical WANs," IEEE Transactions on Communications, vol.40, no.7, pp.1171–1182, 1992.

[91] J.Phuritatkul, "Quality of Service Provisioning in WDM Optical Burst Switching Networks," Ph.D dissertation, September 2005.

[92] J.Phuritatkul, Y.Ji and Y.Zhang, "Blocking Probability of a Preemption-based Bandwidth-Allocation Scheme for Service Differentiation in OBS Networks," IEEE/OSA Journal of Lightwave Technology, vol.24, no.8, pp.2986–2993, 2006.

[93] J.Phuritatkul and Y.Ji, "Resource Allocation Algorithms for Controllable Service Differentiation in Optical Burst Switching Networks," IEICE Transactions on Communications, vol.E88-B, no.4, pp.1424–1431, 2005.

[94] JAIN.R, "The Art of Computer Systems Performance Analysis," JohnWiley & Sons, 1991.

[95] S.Shapiro and B.Wilk, "An Analysis of Variance Test for Normality (Complete Samples)," Biometrika, vol.52, No. 3/4, pp.591–611, 1965.

[96] Chakravarti, Laha and Roy, "Handbook of Methods of Applied Statistics," Volume I, John Wiley and Sons, pp.392–394, 1967.

[97] N.Venables and M.Smith, "An Introduction to R," http://cran.r-project.org/doc/ manuals/R-intro.pdf.

List of Publications

Transactions and Journals

1. Ping DU and Shunji ABE, "Traffic Analysis and Traffic-smoothing Burst Assembly Methods for the Optical Burst Switching Network," Accepted by IEICE Transactions on Communications.

2. Ping DU and Shunji ABE, "Dynamic Wavelength Selection and Delayed Burst Assignment Schemes for Optical Burst Switching Networks," Submitted to IEICE Transactions on Communications.

International Conference Proceedings

1. Ping DU and Shunji ABE, "Burst Assembly Method with Traffic Shaping for the Optical Burst Switching Network," In proceeding of Globecom2006.

2. Ping DU and Shunji ABE, "Sliding window-based Burst Assembly Method in Optical Burst Switching Networks," ICON2006, pp.355–360.

3. Ping DU and Shunji ABE, "A Proportional Differentiated Services Scheme for Optical Burst Switching Networks," OCSN2005, pp.126–131.

4. Ping DU and Shunji ABE, "TCP Performance Analysis of Optical Burst Switching Networks with a Burst Acknowledgment Mechanism," APCC2004, vol.2 pp.621–625.

Technical Reports and Posters

1. Ping DU and Shunji ABE, "A Rescheduling Scheme for Providing Joint QoS in Optical Burst Switching Networks," Society Conference of IEICE, SE20-14, 2006.

2. Ping DU and Shunji ABE, "An Advanced Timer-based Burst Assembly Algorithm with Traffic Shaping in Optical Burst Switching Networks," General

Conference of IEICE, BS-8-1, 2006

3. Ping DU and Shunji ABE, "Traffic Characteristics of Assembled Burst Traffic for Optical Burst Switching Networks," Technical Report of IEICE, NS2005-10.

4. Ping DU and Shunji ABE, "An Edge Buffering Based Fast Restoration Scheme for Optical Burst Switching Networks," Society Conference of IEICE, SE29-30, 2005.

5. Ping DU and Shunji ABE, "DDCS-based Proportional Differentiated Services for OBS Networks," Technical Report of IEICE, OCS2005-18.

6. Ping DU and Shunji ABE, "Burst Multiplexing Methods with Traffic Smoothing for the Optical Burst Switching Network," NII Open House, June, 2006.

7. Ping DU and Shunji ABE, "Burst Assembly Algorithm Using Burst-length Prediction for the Optical Burst Switching Network," NII Open House, June, 2005.

8. Ping DU and Shunji ABE, "A Selective Burst Acknowledgment Mechanism for Optical Switching Network," NII Open House, May, 2004.