國立交通大學

資訊工程學系

博士論文

全光近屬封包交換 IP-over-WDM 網路之訊務控制技術與效能分析

Traffic Control and Performance Analyses for Optical Coarse Packet-Switched IP-over-WDM Networks

研究生：施汝霖

指導教授：楊啟瑞 博士

中華民國九十五年七月

全光近屬封包交換 IP-over-WDM 網路之訊務控制技術與效能分析

Traffic Control and Performance Analyses for Optical Coarse Packet-Switched IP-over-WDM Networks

研究生：施汝霖　　**Student: Ju-Lin Shih**
指導教授：楊啟瑞　博士　　**Advisor: Dr. Maria C. Yuang**

國立交通大學　資訊學院

資訊工程學系

博士論文

A Thesis
Submitted to Department of Computer Science
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in

Computer Science
July 2006
Hsinchu, Taiwan, R.O.C.

中華民國九十五年七月

全光近屬封包交換 IP-over-WDM 網路之訊務控制技術與效能分析

研究生：施汝霖　　　　　　　指導教授：楊啟瑞　博士

國立交通大學　資訊工程學系

中文摘要

對於IP-over-WDM網路而言，全光近屬封包交換網路(OCPS)技術已設計來克服全光封包交換的限制。藉由使用內頻控制進行叢集交換，同時採用訊務控制強化技術，以提供高頻寬使用率與服務品質保證。在此篇論文裡，先簡單介紹全光近屬封包交換網路技術，接著提出所設計的服務品質強化訊務控制機制，於入口路由器做封包集結動作時，提供延遲等級區分與遺失等級區分技術，以應用於全光近屬封包交換網路上。根據這兩個目的，此機制可稱之為(ψ,τ)封包排程器/流量調節器，其中ψ與τ分別代表最大的叢集大小與最長叢集組合時間。為了提供延遲等級區分，IP封包資料流選定一個延遲相關之權重，(ψ,τ)封包排程器根據這些權重與大小為ψ的虛擬視窗，集結封包為叢集。每個延遲等級之延遲保證上限，可以藉由正式規範的逐步服務曲線來量化。為了提供遺失等級區分，(ψ,τ)流量調節器分配較大的叢集尺寸給較高遺失優先權等級者，以促進訊務調節效果。為了檢查此效果與遺失表現之關係，此論文分析並導出了(ψ,τ)流量調節器的輸出程序，封包輸入流則模組化為具有批次輸入的雙態Markov Modulated Bernoulli Process。分析結果顯示(ψ,τ)流量調節器的叢集輸出間距時間的變化程度的減少，與叢集大小是有關的。最後，此論文做了個模擬實驗，環境設定為24節點的美國ARPANET網路與16節點的4x4-torus網路，並比較全光近屬封包交換技術與全光叢集交換技術之遺失率表現。模擬結果顯示，透過叢集尺寸調整，(ψ,τ)

流量調節器可以有效的區分遺失等級，與使用外頻控制與偏移時間服務品質策略之全光叢集交換技術相比，全光近屬封包交換技術可以呈現優越的封包遺失率予高優先權等級，與較佳的遺失訊務等級區分。

Traffic Control and Performance Analyses for Optical Coarse Packet-Switched IP-over-WDM Networks

Student: Ju-Lin Shih *Advisor*: Dr. Maria C. Yuang

Department of Computer Science
National Chiao Tung University, Taiwan

Abstract

For IP-over-WDM networks, Optical Coarse Packet Switching (OCPS) has been proposed to circumvent optical packet switching limitations by using in-band-controlled per-burst switching and advocating traffic control enforcement to achieve high bandwidth utilization and Quality-of- Service (QoS). In this thesis, we first introduce the OCPS paradigm. Significantly, we present a QoS-enhanced traffic control scheme exerted during packet aggregation at ingress nodes, aiming at providing delay and loss class differentiations for OCPS networks. Serving a dual purpose, the scheme is called (ψ,τ)-Scheduler/Shaper, where ψ and τ are the maximum burst size and burst assembly time, respectively. To provide delay class differentiation, for IP packet flows designated with delay-associated weights, (ψ,τ)-Scheduler performs packet scheduling and assembly into bursts based on their weights and a *virtual window* of size ψ. The guaranteed delay bound for each delay class is quantified via the formal specification of a *stepwise* service curve. To provide loss class differentiation, (ψ,τ)-Shaper facilitates traffic shaping with larger burst sizes assigned to higher loss priority classes. To examine the shaping effect on loss performance, we analytically derive the departure process of (ψ,τ)-Shaper. The aggregate packet arrivals are modeled as a two-state Markov Modulated Bernoulli

Process (MMBP) with batch arrivals. Analytical results delineate that (ψ, τ)-Shaper yields substantial reduction, proportional to the burst size, in the coefficient of variation of the burst inter-departure time. Furthermore, we conduct extensive simulations on a 24-node ARPANET network and a 16-node 4x4-torus network to draw packet loss comparisons between OCPS and Just-Enough-Time (JET)-based OBS. Simulation results demonstrate that, through burst size adjustment, (ψ, τ)-Shaper effectively achieves differentiation of loss classes. Essentially, compared to JET-based OBS using out-of-band control and offset-time-based QoS strategy, OCPS is shown to achieve invariably superior packet loss probability for a high priority class, facilitating better differentiation of loss traffic classes.

Acknowledgements

I am deeply grateful to my dearest advisor, Prof. Maria C. Yuang, for her great guidance and encouragement during the preparation of this thesis. I also like to particularly acknowledge and thank Prof. Jean-Lien Chen, Prof. San-Liang Lee, Prof. Cheng-Shang Chang, Prof. Duan-Shin Lee, Prof. Ying-Dar Lin and Prof. Ming-Feng Chang, for their patient reviewing and giving valuable comments and suggestions. I would like to thank all the members in Broadband and Optical Network Laboratory for their assistance. Finally, I would like to express my deepest gratitude to my families for their love and encourage during the preparation of this thesis.

Table of Contents

List of Figures

Symbols

ψ	Maximun burst size/Window Size
τ	Maximum burst assembly time
M	Number of destination*loss classes
N	Number of delay classes
W	Number of wavelengths
w_i	Normalized weight of delay class i
$\delta(t,\theta)$	Stepwise Function of time t and delay θ
$\theta_{\min}$	The minimum value of θ
T_k^{Π}	Time when actual $k \cdot G$ service amount is received
T_k^{Δ}	Time when at least $k \cdot G$ service amount is received
T_k^{δ}	The kth ascending point in stepwise function
$\Pi(t)$	The amount of service actually received by a class at time t
$\Delta(t)$	Stepwise service curve
S	General server to guarantee the stepwise service curve
R	Optical link rate
$\overline{L}$	Offer load/mean arrival rate
B	Burstiness
α	Probability of changing from state H to L in a slot
β	Probability of changing from state L to H in a slot
λ_H	Probability of having a batch arrival at state H
λ_L	Probability of having a batch arrival at state L
$P_{i,j}$	State change probability from state i to state j
$b_H(m)$	Batch size distribution at state H
$b_L(m)$	Batch size distribution at state L
$\overline{b}_H$	mean batch size of $b_H(m)$
$\overline{b}_L$	mean batch size of $b_L(m)$

$\tilde{t}$ random variable represent the burst inter-departure time

$\tilde{s}$ random variable represent the burst size

$P_{\tilde{t},\tilde{s}}(t,s)$ Joint distribution of $\tilde{t}$ and $\tilde{s}$

$\tilde{q}_{y_k}^{k}$ number of packets left in queue behind by the k^{th} departing burst, say at time slot t_k, under the condition that the arrival process is in state y_k at t_k

$\tilde{u}_{z|y}$ number of packets that arrive during the burst inter-departure interval, under the condition that the arrival process changes from state *y* prior to the beginning of the interval, to state *z* at the end of the interval

$\tilde{v}_{z|y}^{n}$ number of packets that arrive during the transmission time of an *n*-packet burst, namely *n* slots, under the condition that the arrival process changes from state *y* prior to the beginning of the time interval, to state *z* at the end of the interval

$c_{r_t|r_0}^{t}(m)$ probability that *m* packets have arrived in an interval of *t* slots, under the condition the arrival process changes from state r_0 prior to the beginning of the interval, to state r_t at the end of the interval

λ_i Arrival rates of class *i*

μ_i Service rates of class *i*

$\tilde{n}_i$ Total number of class-*i* bursts in the system

LP_i Loss probability for class *i*

$\pi_{n_1,\cdots,n_Y}$ Joint distribution of $\tilde{n}_i$

Acronyms

BAT	Burst Assembly Time
BATr	Burst Assembly Timer
BoB	Begin of Burst
CoV	Coefficient of Variation
EoB	End of Burst
FDL	Fiber Delay Line
FIFO	First In First Out
FCFS	First Come First Serve
HER	Header Eraser
HET	Header Extractor
IP	Internet Protocol
JET	Just Enough Time
JIT	Just In Time
MMBP	Markov Modulated Bernoulli Process
OBS	Optical Burst Switching
OCS	Optical Circuit Switching
OCPS	Optical Coarse Packet Switching
OLSP	Optical Label Switched Path
OPS	Optical Packet Switching
QCP	QoS Control Processor
QoS	Quality of Service
SASK	Superimposed Amplitude Shift Keying
TLS	Tunable Laser Source
WDM	Wavelength Division Multiplexing

Chapter 1. Introduction

The ever-growing demand for Internet bandwidth and recent advances in optical Wavelength Division Multiplexing (WDM) technologies [1] brings about fundamental changes in the design and implementation of the next generation IP-over-WDM networks or optical Internet. Current applications of WDM mostly follow the Optical Circuit Switching (OCS) paradigm by making relatively static utilization of individual WDM channels. Optical Packet Switching (OPS) technologies [2-5], on the other hand, enable fine-grained on-demand channel allocation and have been envisioned as an ultimate solution for data-centric optical Internet. Nevertheless, OPS currently faces some technological limitations, such as the lack of optical signal processing and optical buffer technologies, and large switching overhead. In light of this, while some work [4,6,7] directly confronts the OPS limitations, others attempt to tackle the problem by exploiting different switching paradigms, in which Optical Burst Switching (OBS) [8-18] has received most attention.

OBS [8] was originally designed to efficiently support all-optical bufferless [9,10] networks while circumventing OPS limitations. By adopting per-burst switching, OBS requires IP packets to be first assembled into bursts at ingress nodes. The most common packet assembly schemes are based on timer [18], packet-count threshold [10], and a combination of both [10,13,19]. Essentially, major focuses in OBS have been on one-way out-of-band wavelength allocation (e.g., Just-In-Time (JIT) [11], and Just-Enough-Time (JET) [9,12]), and the support of Quality of Service (QoS) for networks without buffers [9,10] or with limited Fiber-Delay-Line (FDL)-based buffers [14]. Particularly in the JET-based OBS scheme that is considered most effective, a control packet for each burst payload is first transmitted out-of-band, allowing each switch to perform just-in-time configuration before the burst arrives. Accordingly, a wavelength is reserved only for the duration of the burst. Without waiting for a positive acknowledgment from the destination node, the burst payload follows its control packet immediately after a predetermined offset time, which is path (hop-count) dependent and theoretically designated as the sum of intra-nodal processing delays.

In the context of supporting QoS in bufferless OBS networks, the work in [9] employs a prioritized extra offset-time method. In the method, a high loss priority class is given a larger extra offset time, allowing the high priority class to make earlier wavelength reservation than lower priority classes. The method effectively provides different grades of loss performance, but at the expense of a drastic increase in the end-to-end delay particularly for high priority classes. Besides, as discussed in [20], the method undergoes the unfairness and near-far problems. Especially due to the near-far problem, a low priority burst with a longer path to travel may end up with the same or larger offset time than that of a high priority burst, resulting in obstacles to QoS burst truncation [21] in switching nodes. The prioritized burst segmentation approach proposed in [10], different from most approaches, adopts the assembly of different priority packets into a burst in the order of decreasing priorities. Should contention occur in switching nodes, the approach supports burst truncation rendering lower-priority packets toward the tail be dropped or deflected with higher probability. The approach achieves low packet loss probability for high priority classes, with the price of excessive complexity paid during burst scheduling in switching nodes.

OBS gains the benefits of OCS and OPS. However, its offset-time-based design results in three complications. First, the determination of the offset time is a design dilemma. A large offset time incurs excessive packet delay. A small offset time may fail to make wavelength reservation prior to the burst arrival. This fact renders deflection routing (via longer paths) infeasible during contention resolution. Second, to enable efficient reservation of wavelengths, JET-based OBS requires the offset-time and burst length information to be included in the control packet, to provide a switch with the exact time and duration that the burst arrives and lasts, respectively. At each switching node along the path, such information needs to be maintained for future configuration until the burst arrives. Besides, the offset time is required to be decremented at every switching node and the burst length needs to be updated should burst truncation occur. Evidently, such design results in significantly increased complexity [15]. Third, the inclusion of the burst length information in control packets, together with the near-far problem described above, OBS gives rise to a difficulty in supporting QoS burst truncation. For example, consider a case that there is a high priority burst that arrives after a low priority burst and potentially collides with the low priority burst. If the control packet of the low priority burst has already departed, its length can no longer be updated. In this case, the switching node is left no choice but to truncate the high priority rather than the low priority burst. This type of operation is referred to as *restricted* QoS burst truncation.

These three OBS design complications are the primary motivators behind the design of the Optical Coarse Packet Switching (OCPS) paradigm [22-24]. While OBS can be viewed as a more efficient variant of OCS; OCPS can be considered as a less stringent variant of OPS. Similar to OBS, OCPS is aimed at supporting all-optical per-burst switched networks, which are labeled-based [12], QoS-oriented, and either bufferless or with limited FDL-based buffers. Unlike OBS using offset-time-based out-of-band control, OCPS (see Figure 1) adopts in-band control in which the header and payload are together transported via the same wavelength. More specifically, in an OCPS network, IP packets belonging to the same loss class and the same destination are assembled into bursts at ingress routers. A header for a burst payload, which carries forwarding (i.e., label) and QoS (e.g., priority) information, is modulated with the payload based on the newly designed Superimposed Amplitude Shift Keying (SASK) technique [25]. Besides, they are time-aligned during modulation via necessary padding added to the header. They are re-aligned in switching nodes should burst truncation occur. Such design eliminates the payload length information from the header, and thus as will be shown, facilitates restriction-free QoS burst truncation in switching nodes. The entire burst is then forwarded along a pre-established Optical Label Switched Path (OLSP). At each switching node, the header and payload are first SASK-based demodulated [25]. Each burst payload is switched according to the label information in the header. While the header is electronically processed, the burst payload remains transported optically in a fixed-length FDL achieving constant delay and data transparency.

The main focus of my thesis is on QoS-enhanced traffic control exerted during packet burstification at ingress nodes, aiming at providing delay and loss class differentiations for OCPS networks. In this work, optical switches are assumed buffer-less and all wavelengths are shared using wavelength converters [3,26]. Regarding delay performance, due to the absence of buffering delay in core switches, the end-to-end delay performance is solely determined by the burstification delay. Considering the assembly of packets from flows with different delay requirements, the problem becomes the scheduling of these packets during burstification. At first thought, existing scheduling disciplines [27,28,29] are possible candidates. These schemes have placed emphasis on the design of scalable *packet* schedulers achieving fairness and delay guarantees. All packets follow the exact departure order that is computed according to virtual finishing times being associated with packets. Nevertheless, in the case of burstification, considering tens or hundreds of packets in a burst, the exact position of packets within a burst is no longer relevant. Most existing scheduling schemes thus become economically unviable. Regarding loss performance, rather than exploring reactive contention resolution mechanisms [20], in this work we focus on the design of traffic shaping with QoS provisioning.

In this thesis, we present a dual-purpose traffic control scheme, called (ψ,τ)-Scheduler/Shaper. Notice that from the packet burstification perspective, it is simply a timer and threshold combined scheme, where ψ and τ are the maximum burst size (packet count) and maximum burst assembly time, respectively. To provide delay class differentiation, for IP packet flows designated with delay-associated weights, (ψ,τ)-Scheduler performs packet scheduling and assembly into bursts based

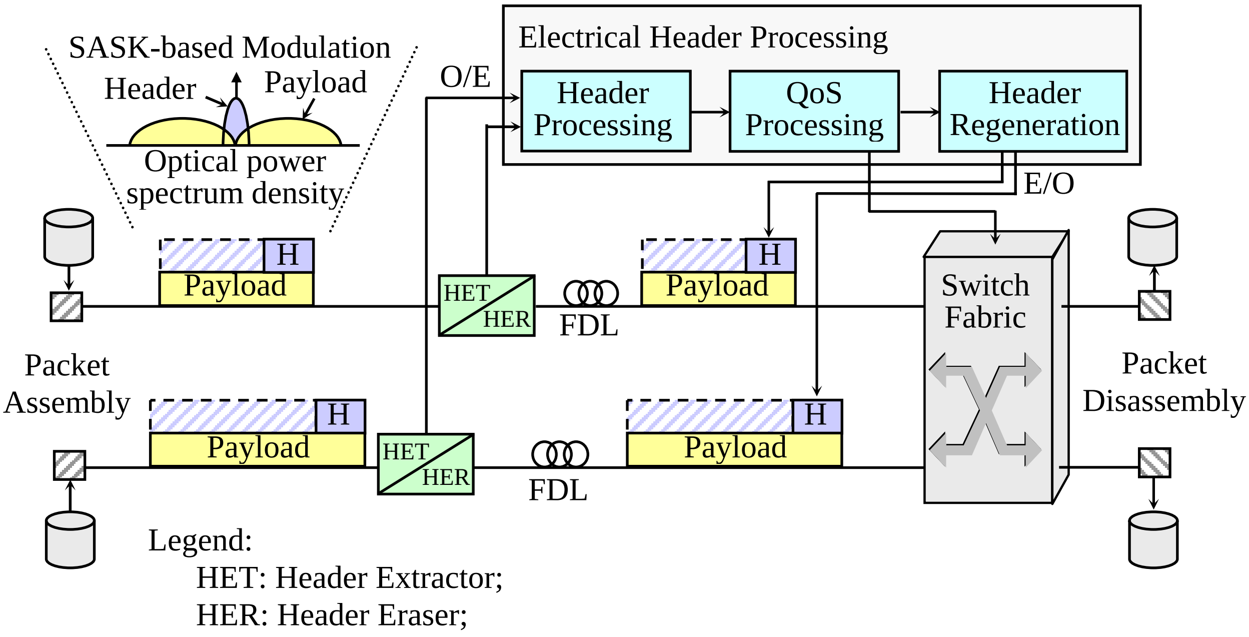


Figure 1. Optical Coarse Packet Switching (OCPS).

on their weights and a *virtual window* of size ψ. The Scheduler exerts simple First In First Out (FIFO) service within the window and assures weight-proportional service at the window boundary. The guaranteed delay bound for each delay class is quantified via the formal specification of a *stepwise* service curve [27]. The mean delay and 99% delay bound for each delay class are also demonstrated via simulation results.

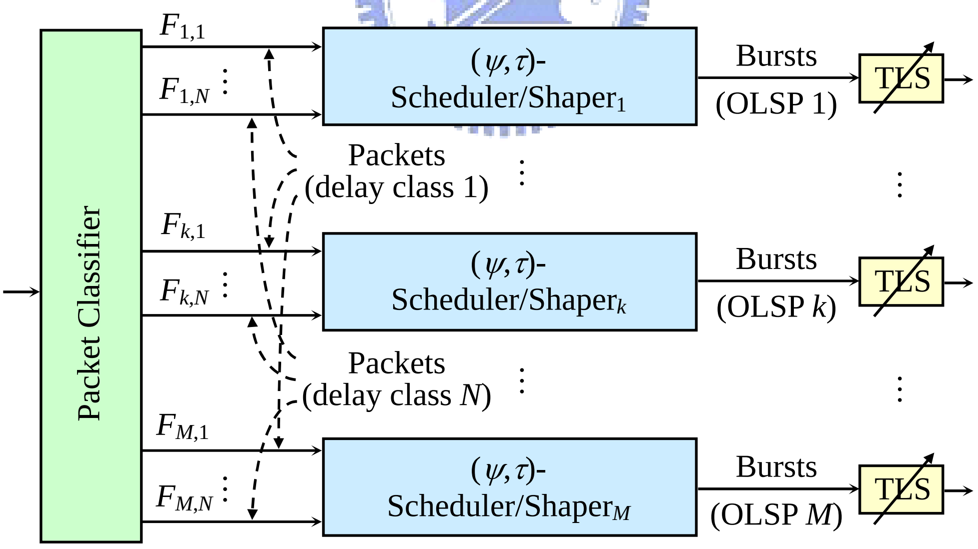
To provide loss class differentiation, (ψ,τ)-Shaper facilitates traffic shaping with a larger burst size (ψ) assigned to a higher priority class. To examine the shaping effect on loss performance, we analytically derive the departure process of (ψ,τ)-Shaper. The aggregate packet arrivals are modeled as a two-state Markov Modulated Bernoulli Process (MMBP) with batch arrivals. Analytical results delineate that (ψ,τ)-Shaper yields substantial reduction in the Coefficient of Variation (CoV) of the burst inter-departure time. The greater the burst size, the more reduction in the CoV. Furthermore, extensive simulations are conduct on a 24-node ARPANET network and a 4x4-torus network to draw loss performance comparisons between OCPS and JET-based OBS. Simulation results demonstrate that, through burst size adjustment, (ψ,τ)-Shaper effectively achieves differentiation of loss classes. Essentially, owing to enabling restriction-free QoS burst truncation in switching nodes, OCPS is shown to achieve superior packet loss probability for a high priority class, and facilitate better differentiation of traffic classes, compared to JET-based OBS.

The remainder of this thesis is organized as follows. In Chapter 2, we introduce the (ψ,τ)-Scheduler/Shaper system architecture. In Chapter 3, we describe the (ψ,τ)-Scheduler design, the stepwise service curve, and show the worst and 99%

delay bounds for each delay class. In Chapter 4, we present a precise departure process analysis for (ψ,τ)-Shaper to analytically delineate the shaping effect on departing traffic characteristics. In Chapter 5, we demonstrate the provision of loss class differentiation, and draw packet loss comparisons between OCPS and JET-based OBS via network-wide simulation results. Finally, concluding remarks are made in Chapter 6.

Chapter 2. (ψ,τ)-Scheduler/Shaper System Architecture and Design Concept

In any ingress node, incoming packets (see Figure 2) are first classified on the basis of their destination, loss, and delay classes. Packets belonging to the same destination and the same loss class are assembled into a burst. Thus, a burst contains packets of various delay classes. In the figure, we assume there are *M* destination*loss classes and *N* delay classes in the system. For any one of *M* destination*loss classes, say class *k*, packets of flows belonging to *N* different delay classes are assembled into bursts through (ψ,τ)-Scheduler/Shaper$_k$ according to their pre-assigned delay-associated weights. Departing bursts from any (ψ,τ)-Scheduler/Shaper are optically transmitted, and forwarded via their corresponding, pre-established OLSP.



Legend:

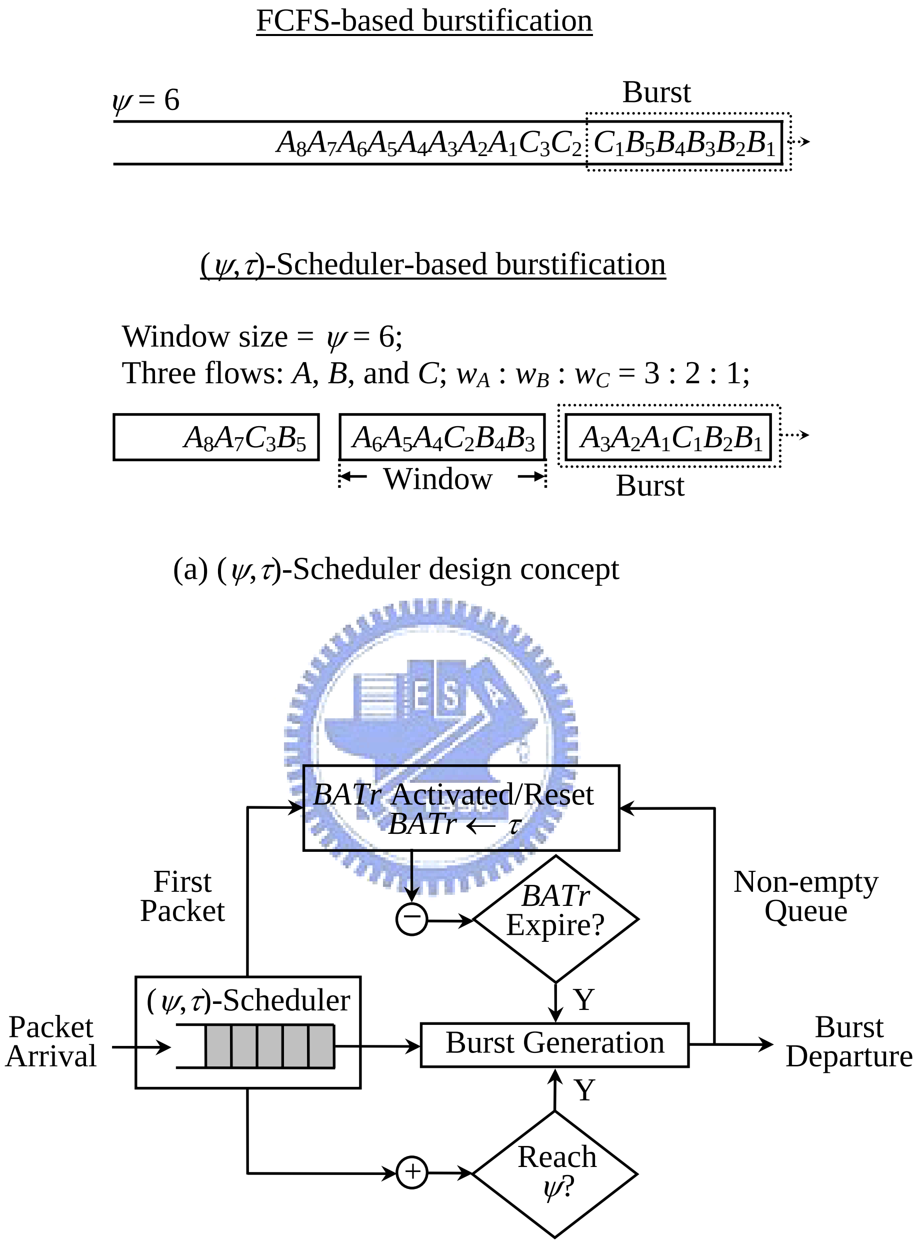
$F_{d,y}$: Packet flow of destination*loss class *d* and of delay class *y*;
OLSP : Optical Label Switched Path;
TLS : Tunable Laser Source;

Figure 2. (ψ,τ)-Scheduler/Shaper system architecture.

Essentially, (ψ,τ)-Scheduler/Shaper is a dual-purpose scheme. It is a scheduler for packets, abbreviated as **(ψ,τ)-Scheduler**, which performs the scheduling of different delay class packets into back-to-back bursts. On the other hand, it is a shaper for bursts, referred to as **(ψ,τ)-Shaper**, which determines the sizes and departure times of bursts.

(ψ,τ)-Scheduler performs packet scheduling on their weights and a *virtual window* of size ψ. The Scheduler exerts simple FIFO service within the window and assures weight-proportional service at the window boundary. The design concept of (ψ,τ)-Scheduler is shown in Figure 3(a). While the First Come First Serve (FCFS)-based burstification simply aggregates the first six arriving packets into a burst, (ψ,τ)-Scheduler assembles the first window into a burst to assure weight-proportional service of different delay class.

The functional design diagram of the (ψ,τ)-Shaper is shown in Figure 3(b). IP packets arrive at the system queue. If the total number of packets reaches ψ before the burst assembly time exceeds τ, a burst of size ψ is generated and transmitted. The burst assembly time is controlled by τ, which is activated at two different instants. Basically, burst assembly timer (BATr) is triggered by the first arriving packet to an empty queue and is set by τ, then BATr starts counting down. The second instant is occurred after finishing transmission of a burst leaving a non-empty queue. BATr is re-activated and set by τ. While the BATr expire and system queue size is less than ψ, a burst consisted by all the packets in the queue is generated and transmitted.



(a) (ψ, τ)-Scheduler design concept

(b) (ψ, τ)-Shaper: functional design

Figure 3. (ψ, τ)-Scheduler/Shaper design concept.

Chapter 3. (ψ,τ)-Scheduler and Delay QoS

In the (ψ,τ)-Scheduler system, each delay class is associated with a pre-determined weight [27]. A higher delay priority class is given a greater weight, which corresponds to a more stringent delay bound requirement. In addition, we assume all packets are of fixed size of one unit. Generally, (ψ,τ)-Scheduler performs scheduling of packets in accordance with their weights and a *virtual window* of size ψ. The weight of a class corresponds to the maximum number of packets of the class that can be accommodated in a window, or burst in this case. Such window-based scheduling allows simple FIFO service within the window and assures weight-proportional service at the window boundary. In the sequel, we present the design and algorithm, followed by the specification of the stepwise service curve from which the guaranteed delay bound can be obtained.

3.1 Scheduling Design and Algorithm

Upon arriving, packets of different classes are sequentially inserted in a sequence of virtual windows. The window size, which is set as the maximum burst size, ψ, together with the weight (w) of a class, determines the maximum number of packets (i.e., quotas) from this class that can be allocated in a window. For a class, if there are sufficient quotas, its new packets are sequentially placed in the current window in a FIFO manner. Otherwise, its packets are placed in an upward window in accordance to the total accumulated quotas. A burst is formed and departs when the burst size reaches ψ or the Burst Assembly Timer (BATr) (set as τ initially) expires. For convenience, class weights are normalized to the window size. Namely, $\sum w_i = \psi$, where w_i is the normalized weight of class i.

The operation of (ψ,τ)-Scheduler can be best explained via a simple example illustrated in Figure 4. For ease of illustration, the normalized weights are set as integers in the example. Initially, five packets from three classes (*X*, *Y*, and *Z*) arrive at time 1, and four of them are placed in the first virtual window except Y_2 due to having only one quota in a window. The BATr is activated and set as BATr = $\tau = 3$. At the end of time 1, a burst of size $\psi = 4$ packets departs. The same operation repeats until the end of time 4. Notice that there are four packets in the system, which are placed in three consecutive virtual windows. A burst is still generated at the end of time 4. This explains why the "virtual" window is named. Finally, at time 8, a burst of size three is generated due to time out of the BATr.

The detailed algorithm of (ψ,τ)-Scheduler is outlined in Figure 5. First, the system performs the Initialization operation whenever the system changes from being idle to busy due to packet arrivals. The quota of each class is initialized as its normalized weight, and the BATr is activated and set to be the value of τ. The algorithm then asynchronously performs two tasks repeatedly: Arrival and Departure. The Arrival task handles the insertion (Enqueue) of newly arriving packets in appropriate virtual windows; whereas the Departure task removes (Dequeue) the generated burst from the queue. If the queue remains non-empty, the BATr is reset to the τ value. It is worth noting that the algorithm works under non-integer normalized weights which are practically the case in real systems.

Assumptions:

$\psi = 4$, $\tau = 3$, $R = 4$;

Three classes: X, Y, Z; $w_X : w_Y : w_Z = 2 : 1 : 1$;

Time	Packet Arrival	Virtual-Window Queue			BATr	Burst Departure
1	$Z_1Y_2Y_1X_2X_1$		Y_2	$Z_1Y_1X_2X_1$	A_a→3	$Z_1Y_1X_2X_1$
2	$Z_2X_4X_3Y_4Y_3$	Y_4	Y_3	$Z_2X_4X_3Y_2$	R_d→3	$Z_2X_4X_3Y_2$
3			Y_4	Y_3	R_d→3	
4	Y_5Z_3	Y_5	Y_4	Z_3Y_3	2	$Y_5Y_4Z_3Y_3$
5	Z_4			Z_4	A_a→3	
6	Z_5		Z_5	Z_4	2	
7			Z_5	Z_4	1	
8	X_5		Z_5	X_5Z_4	0	$Z_5X_5Z_4$

Legend:

x_n: The n_{th} packet of class x ($x = X$, Y, or Z);

A_a: Activated by the first packet arrival;

R_d: Reset by burst departure;

Figure 4. (ψ,τ)-Scheduler: an example.

```
Variable
    $w_i$ : normalized weight of class $i$ ($\Sigma w_i = \psi$);
    $cw$ : index of currently served window;
    $lw_i$ : index of window containing the last class $i$'s packet;
    $q_i$ : net quota for class $i$;
    $P_i$ : newly arriving packet from class $i$;
    $Bu$ : the generated burst;
    BATr: burst assembly timer;

Initialization()    /* idle to busy */
    $cw \leftarrow 1$;
    for (each class $i$) do    $lw_i \leftarrow 1$;    $q_i \leftarrow w_i$;    endfor
    BATr $\leftarrow \tau$;

Arrival($P_i$)    /* a newly arriving packet from class $i$ */
        Determine the window $P_i$ can be placed;
    if ($lw_i < cw$)    $lw_i \leftarrow cw$;    $q_i \leftarrow w_i$;    endif
    while ($q_i < 1$) do    $lw_i \leftarrow lw_i + 1$;    $q_i \leftarrow q_i + w_i$;    endwhile
        Place packet in window $lw_i$ and update quota;
    Enqueue($P_i$, $lw_i$);    $q_i \leftarrow q_i - 1$;

Departure($Bu$)    /* BATr expires or packet count $\geq \psi$ */
        Remove burst $Bu$ from the head of the queue;
    Dequeue($Bu$);
        Update information;
    $cw \leftarrow$ index of the next window with packets;
    if (queue is not empty)    BATr $\leftarrow \tau$;    endif
```

Figure 5. (ψ, τ)-Scheduler: the algorithm.

3.2 Worst Delay Bound Guarantee- Stepwise Service Curve

The service curve specification [27,29] has been widely used as a flexible methodology for resource allocation to satisfy diverse delay and throughput guarantees. Prevailing packet scheduling schemes are mostly work conserving exhibiting continuous-wise service curves. In contrast, the (ψ,τ)-Scheduler is a non-work-conserving server, in which packets do not depart from the system before the burst is generated. My objective is to characterize the stepwise nature of the service curve for the non-work-conserving system, (ψ,τ)-Scheduler.

In the sequel, we first define the stepwise function and introduce the stepwise service curve guaranteed by a general server, *S*. Then we specify the stepwise service curve guaranteed for a delay class by (ψ,τ)-Scheduler in Theorem 1. Finally we provide the worst delay bound in two different forms based on the theorem. Throughout this section we assume that there are *N* delay classes in the system, and the optical link capacity is *R* packets/slot. For ease of description, the normalized weight of any class is assumed greater than or equal to one.

Definition 1: A ***stepwise function*** $\delta(t,\theta)$ of time *t* and delay θ, under jump *G* and incremental interval *I*, is defined as

$$\delta(t,\theta)=\begin{cases} k\cdot G, & T_k^{\delta}\le t<T_{k+1}^{\delta},\ \text{and}\ k\ge 0 \\ 0, & 0\le t<T_0^{\delta} \end{cases}, \tag{1}$$

where T_k^{δ} is the *k*th ascending point, defined as $T_k^{\delta}\equiv\theta+k\cdot I$.

Accordingly, a stepwise function is uniquely determined by three parameters, *G*, *I*, and θ The significance of such stepwise function is that it corresponds to a quasi-constant-bit-rate service, in which a fixed amount (*G*) of service can be offered per every time period (*I*), after a minimum delay of time θ.

As depicted in Figure 6, under a general server, *S*, let $\Pi(t)$ denote the amount of service actually received by a class at time *t*. In addition, denote T_k^{Π} the time instant at which the received service exceeds *k* times of service granularity, *G*. Namely, $T_k^{\Pi} \equiv \min\{t : \Pi(t) \geq k \cdot G\}$, for all $k \geq 0$. For example in Figure 6, a *G* amount of service corresponds to the finishing transmission of four packets. Due to batch service, server *S* actually finishes a two-packet (=0.5*G*) transmission at t_2, and a total of six-packet (=1.5*G*) transmission at t_6. Thus, T_1^{Π} is equal to t_6 which is the

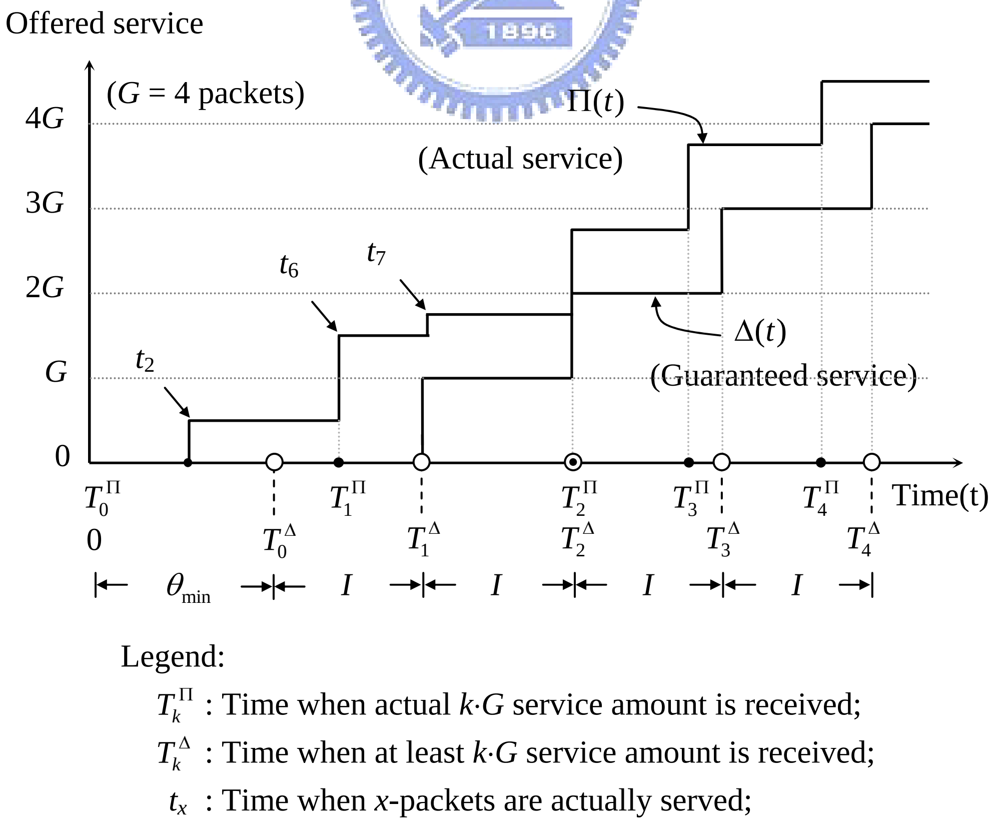


Figure 6. Concept of stepwise service curve.

earliest time upon which 1*G* (four-packet) service has been received.

The problem of seeking guaranteed service becomes the determination of a stepwise function $\Delta(t)$ which is the greatest lower bound of all possible scenarios of $\Pi(t)$ (see Figure 6). $\Delta(t)$ is called the *stepwise service curve*, guaranteed by *S*, defined as follows.

Definition 2: A **stepwise service curve** $\Delta(t)$ under *G* and *I*, guaranteed by general server *S*, is defined as

$$\Delta(t) \equiv \sup_{\theta \in E} \left\{ \delta(t,\theta) \right\}, \ \forall t \geq 0, \tag{2}$$

where $E = \left\{ \theta : \delta(t,\theta) \leq \Pi(t), \ \forall t \geq 0 \right\}$. The supremum of Equation (2) uniquely occurs at the minimum value of θ, denoted as $\theta_{\min}$.

Notice that the above uniqueness and minimum properties of $\theta_{\min}$ rest on the fact that, by fixing θ, function $\delta(t,\theta)$ is monotonically increasing with *t*; and by fixing *t*, the function is monotonically decreasing with θ. My main goal is to determine the stepwise service curve guaranteed by (ψ,τ)-Scheduler for a class, say class *i*. To this end, one way of approaching it is to find the minimum service amount achieved at any given time, i.e., to find *y*-axis service amount for any given *x*-axis time *t*. Another way, which is what we adopt here, is to determine the maximum time required before a given service amount is received, i.e., to find *x*-axis time value for any given *y*-axis service amount. For rigorousness, the above statement is outlined in the next lemma.

Lemma 1: If server *S* guarantees a stepwise service curve $\Delta(t)$ with $\theta_{\min}$ taken by Definition 2. If for all stepwise functions $\delta(t,\theta_i)$, $\forall i \geq 0$, defining $\theta^*_{\min}$ by

$$\theta^*_{\min} \equiv \inf\left\{\theta_i \geq 0 : T_k^{\Pi} \leq T_k^{\delta},\ \forall \delta(t,\theta_i),\ \forall i,k \geq 0\right\}, \tag{3}$$

then $\theta^*_{\min} = \theta_{\min}$.

The proof of Lemma 1 is in Appendix A. To find the stepwise service curve for class *i*, three parameters, *G*, *I*, and $\theta_{\min}$ have to determine first. It is simple to perceive that service granularity *G* for class *i* is equal to the normalized weight, w_i, of the class. Second, the worst time period that w_i amount of service can be at least offered is the maximum burst assembly time, *τ*, plus the burst transmission time, namely ψ / R. Therefore, it arrive at $I = \tau + \psi / R$. The problem left is to find $\theta_{\min}$, which is given in the following theorem, with the proof shown in Appendix B.

Theorem 1: A stepwise service curve guaranteed by (*ψ*,*τ*)-Scheduler for class *i*, is $\Delta_i(t)$ in which $G = w_i$, $I = \tau + \dfrac{\psi}{R}$ and $\theta_{\min} = \left(1 + \left\lceil \dfrac{N}{\psi} \right\rceil\right) \cdot \left(\tau + \dfrac{\psi}{R}\right)$.

Based on Theorem 1, we are now in the position to derive the worst delay bound for different delay classes of traffic. Notice that, the work [27] provided an absolute delay bound, subject to the constraint that arriving packets are leaky-bucket regulated. In this work, due to the lack of traffic regulation, a time-independent delay bound is unachievable. In the end, the worst delay bound for each class is provided in two forms.

In the first form, we present a time-dependent worst delay bound for a packet, given the class of the packet. As shown in Figure 7, we delineate two guaranteed service curves for class 1 with w_1=3 and class 2 with w_2=1, respectively, based on Theorem 1. Suppose the forth packet (P^4) from the beginning of a busy period arrives at t_4. According to the theorem, if the packet is of class 1, the worst delay bound until packet P^4 served is $\theta_{\min}+2\cdot I-t_4$, and if the packet is of class 2, the worst delay bound until packet P^4 served is $\theta_{\min}+4\cdot I-t_4$. Accordingly, for the j^{th} packet P_i^j of class *i* arriving at time t_j from the beginning of a busy period, the worst delay bound is $\theta_{\min}+\left\lceil \frac{j}{w_i} \right\rceil \cdot I-t_j$, where $\theta_{\min}$ and *I* are given in Theorem 1.

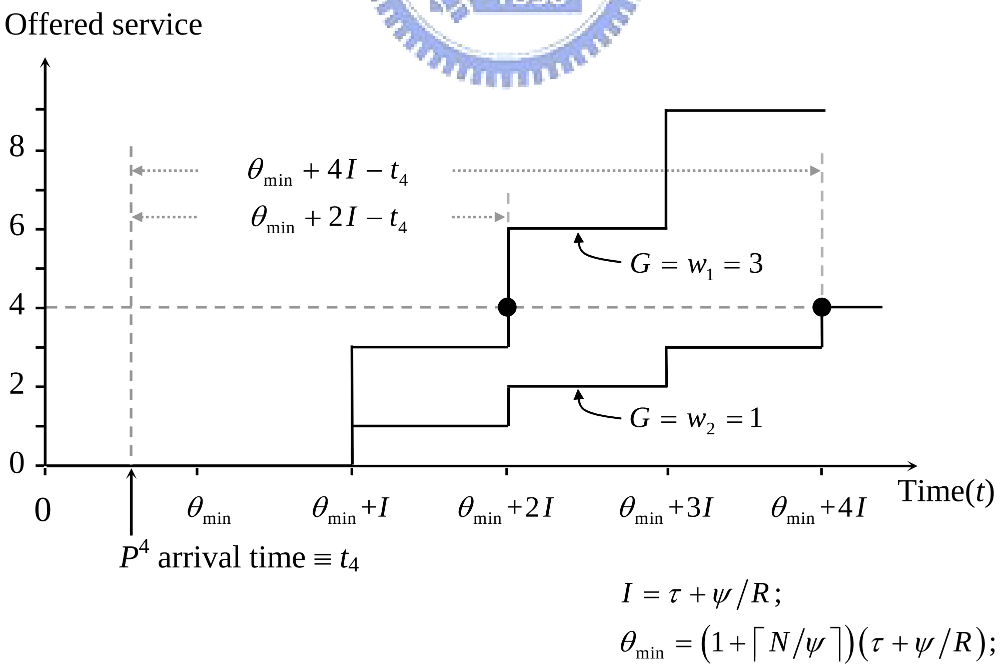
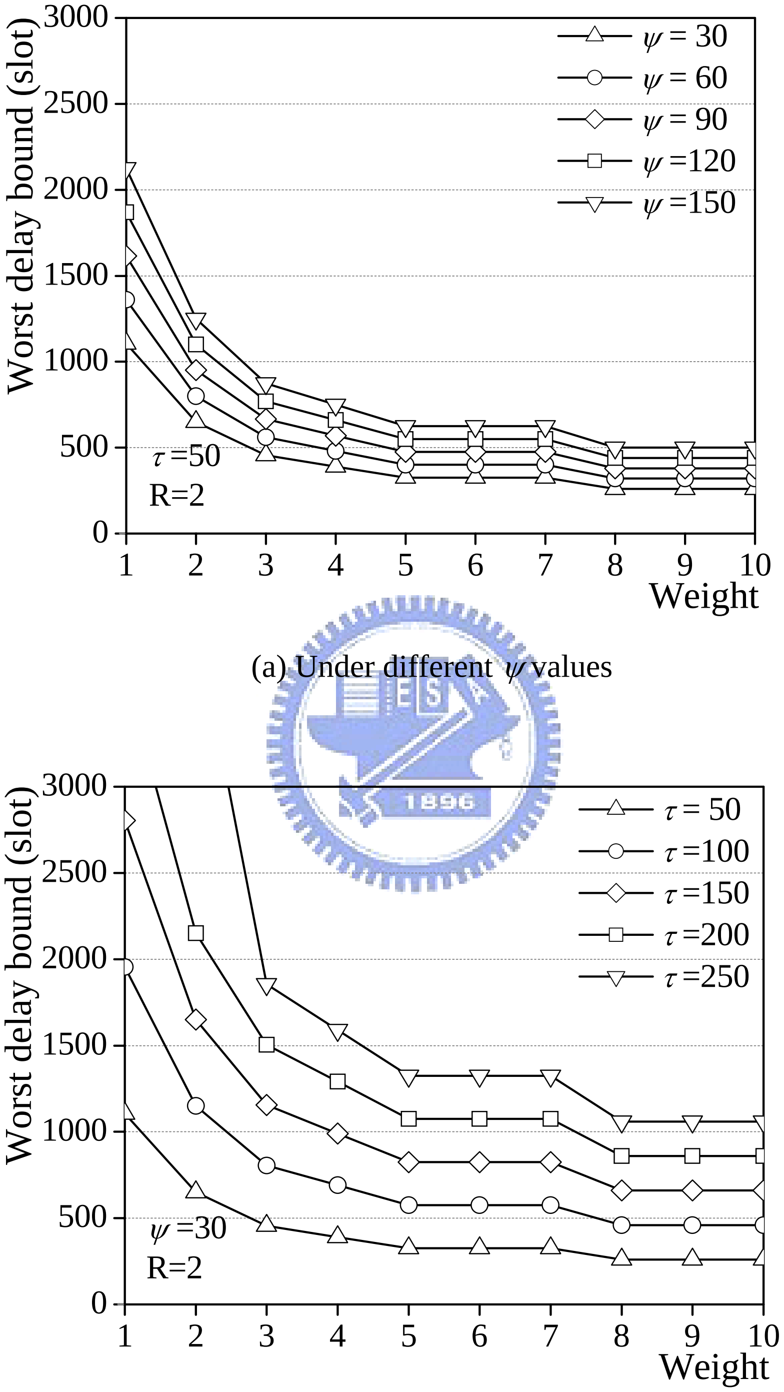


Figure 7. (ψ,τ)-Scheduler's stepwise service curves for two classes.

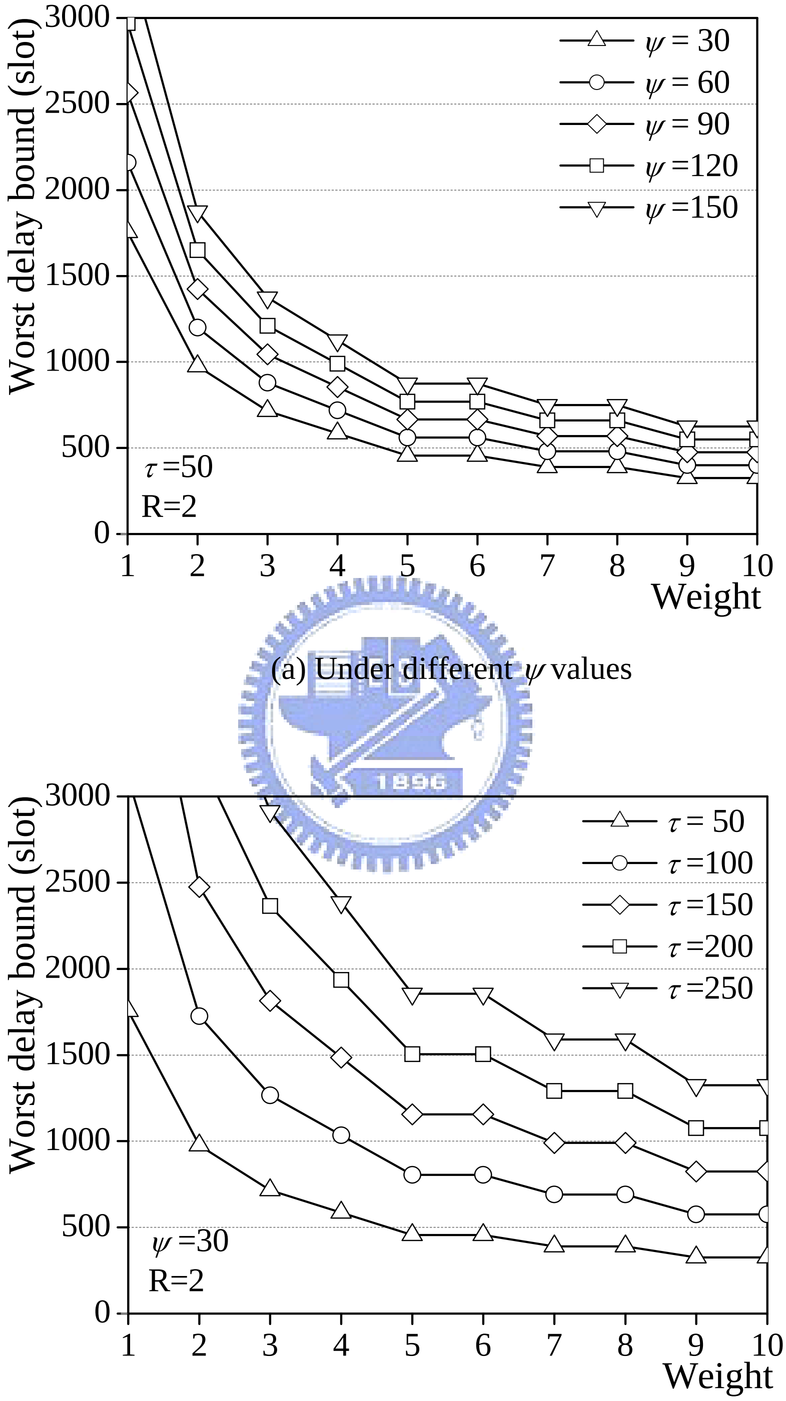
In the second form, we provide the worst delay bound of an observed packet (of one class) that arrives along with a bulk of packet arrivals that belong to any traffic classes. Based on Theorem 1, we plot in Figures 8 and 9 the worst delay bound as a function of the normalized weight for the observed packets (15th packet and 25th packet), under a bulk arrival of 25 packets (including the observed packets). In the setting, we assume that the optical link capacity is 2 packets per slot. we reveal from the figure that the worst delay bound grows while the burst size (ψ from 30 to 150) or the burst assembly time (τ from 50 to 250) increase, and the dramatically declines as the class weight increases under all (ψ, τ) settings. In Figure 8, the worst delays of the15th packet under weights 5, 6, and 7 are the same. It is caused by that the 15th packet is transmitted on the same (3rd) burst. Significantly, such worst delay bound is guaranteed irrelevant to the weight and class distributions of other packets that arrive in the same bulk. This partially illustrates the significance of service curve in providing delay and throughput guarantees.



(a) Under different ψ values

(b) Under different τ values

Figure 8. Worst delay bound of the 15th packet in bulk arrival.



(a) Under different ψ values

(b) Under different τ values

Figure 9. Worst delay bound of the 25th packet in bulk arrival.

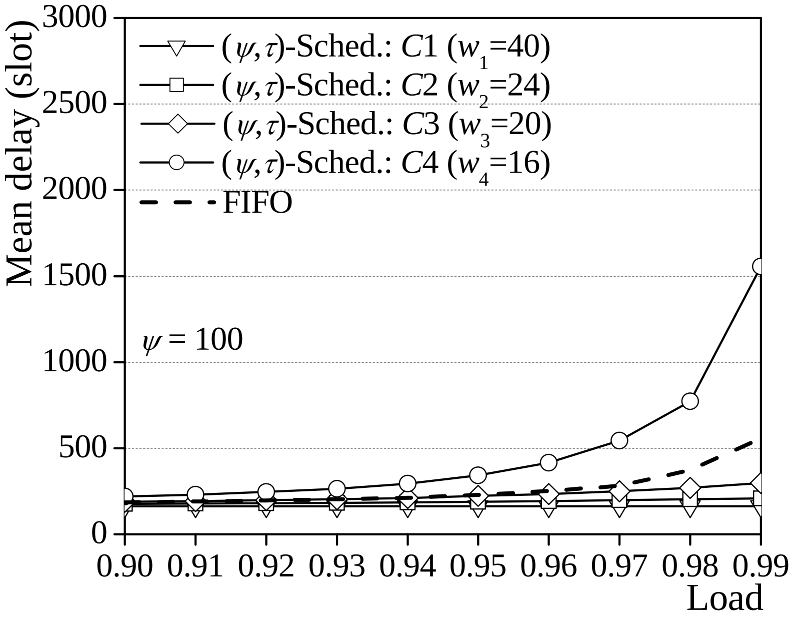
3.3 Delay QoS Provision

In addition to the deterministic worst delay bound, we also seek stochastic delay performance metrics to gain more insights into the effectiveness of the weight-based scheduling on delay QoS provisioning. To this end, we carried out event-based simulations in which the mean packet delay and 99% delay bound (in units of slots) were measured.

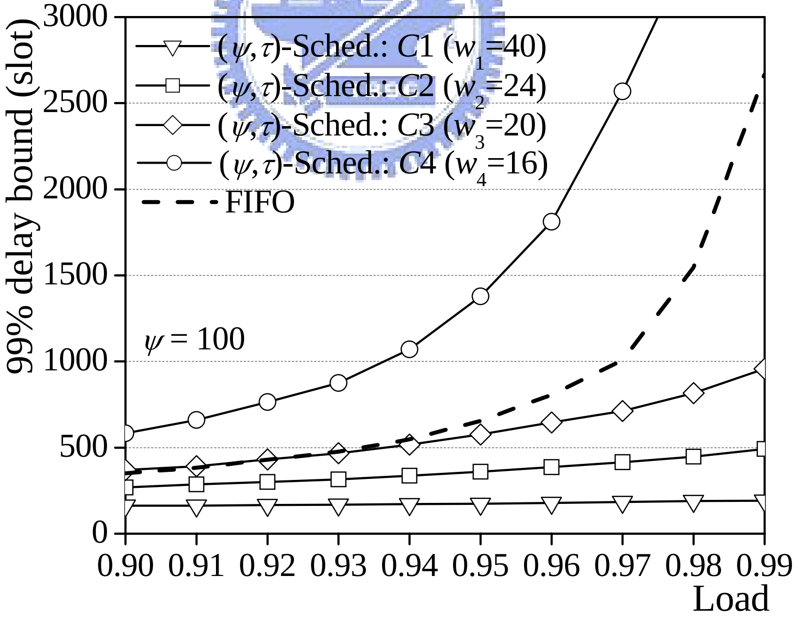
In the simulations, there are four delay classes (*C*1-*C*4), with the weights set as 10, 6, 5, and 4 (or 40, 24, 20, and 16, normalized with respect to $\psi = 100$). The system is served by a wavelength in a capacity of one 60-byte packet per slot time. Each of these four classes generate an equal amount of traffic based on a two-state (*H* and *L*) MMBP. In the MMBP, the probability of switching from state *H* to *L* is equal to 0.225, and from state *L* to *H* is equal to 0.025. The probability of having one packet arrival during state *H* is equal to $\bar{L}$ and during state *L* is equal to $\bar{L}/6$, under an offered load, $\bar{L}$, i.e., $\bar{L}$ /4 for each class. Accordingly, the burstiness of traffic is B = 4. To draw a comparison, a FIFO system was also experimented. Simulations are terminated after reaching 95% confidence interval. Simulation results are plotted in Figures 10 and 11.

We observe from Figure 10 that both mean delay and 99% delay bound of all classes increase with the offered load (from 0.9 to 0.99). Superior to the FIFO system that undergoes long delay/bound at high loads, (ψ, τ)-Scheduler invariably assures low delay/bound for high priority classes (e.g., *C*1 and *C*2) at a cost of increased delay/bound for low priority classes (e.g., *C*4). In Figure 11, we observe mean delay and 99% delay bound of *C*1-*C*4 under different weight of *C*1. In the setting that w_1 = 1, the normalized weights of C1, C2, C3 and C4 with respect to ψ = 100 are 6.25, 37.5, 31.25, and 25, respectively. The weight of a class can be adjusted to meet its delay/bound requirements. For example, as shown in Figure 11(b), to meet a 99% delay bound guarantee of 200 slots for class C1 under load=0.9, the weight of C1 must be greater than 7, given the weights of three other classes of 6, 5, and 4, respectively.

We also investigate the impact of (ψ,τ) setting on mean burstification delay. In this simulation, all flows are of the same delay class (w_1: w_2: w_3: w_4 = 1:1:1:1), and served by a single wavelength. Simulation results are shown in Figure 12. While τ is large (see Figure 12(a)), burstification delay is relevant to the number of arriving packets. At low load the system queue size is less than ψ under most condition, and the burstification delay is relevant to the arrival time of the ψ^{th} packet. The delay can be controlled by appropriate τ value (see Figure 12(b)). Under high ψ value and lower τ value (see Figure 12(c)), mean burstification delay is controlled by τ value. We conclude that the maximum burst assembly time (τ) serves the purpose of assuring bounded mean burst delay particularly under low to medium loads, despite the fact that a large ψ value is applied.

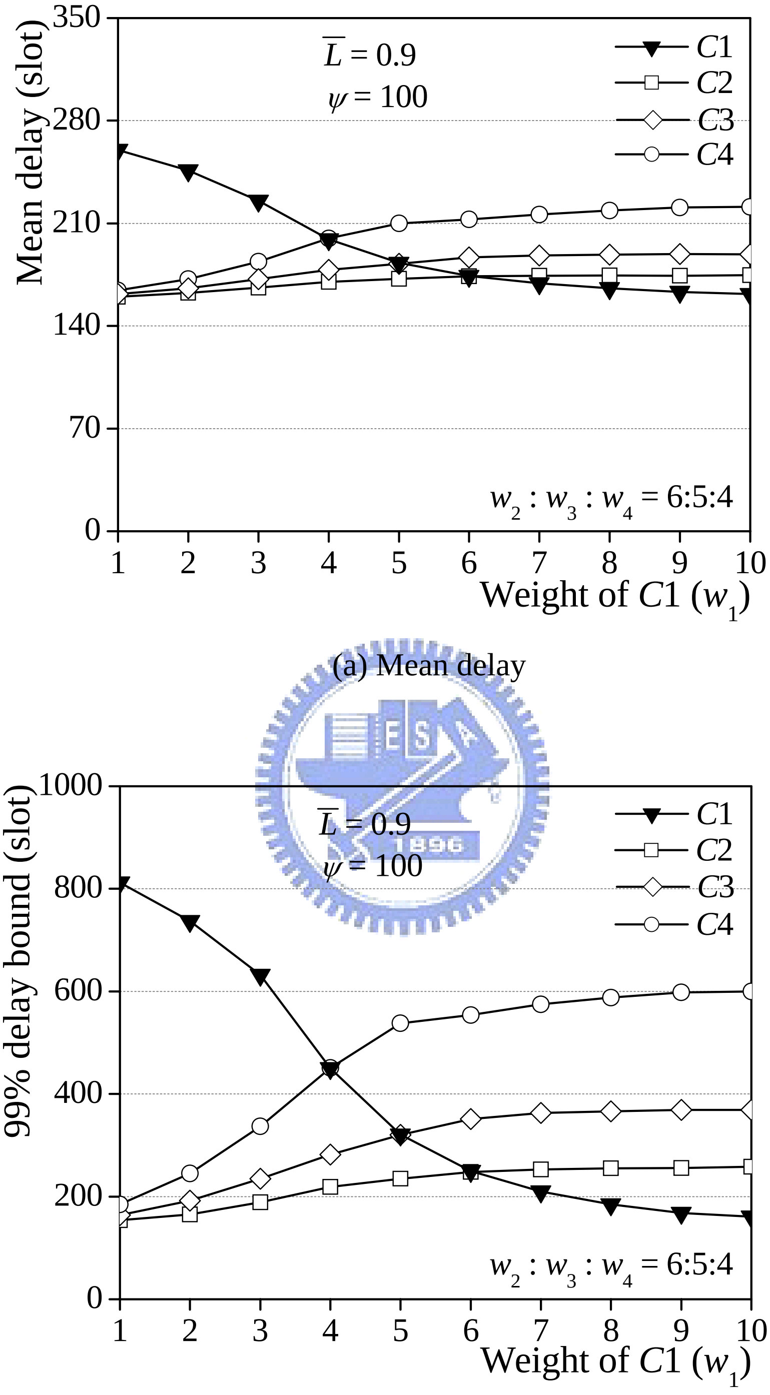


(a) Mean delay



(b) 99% delay bound

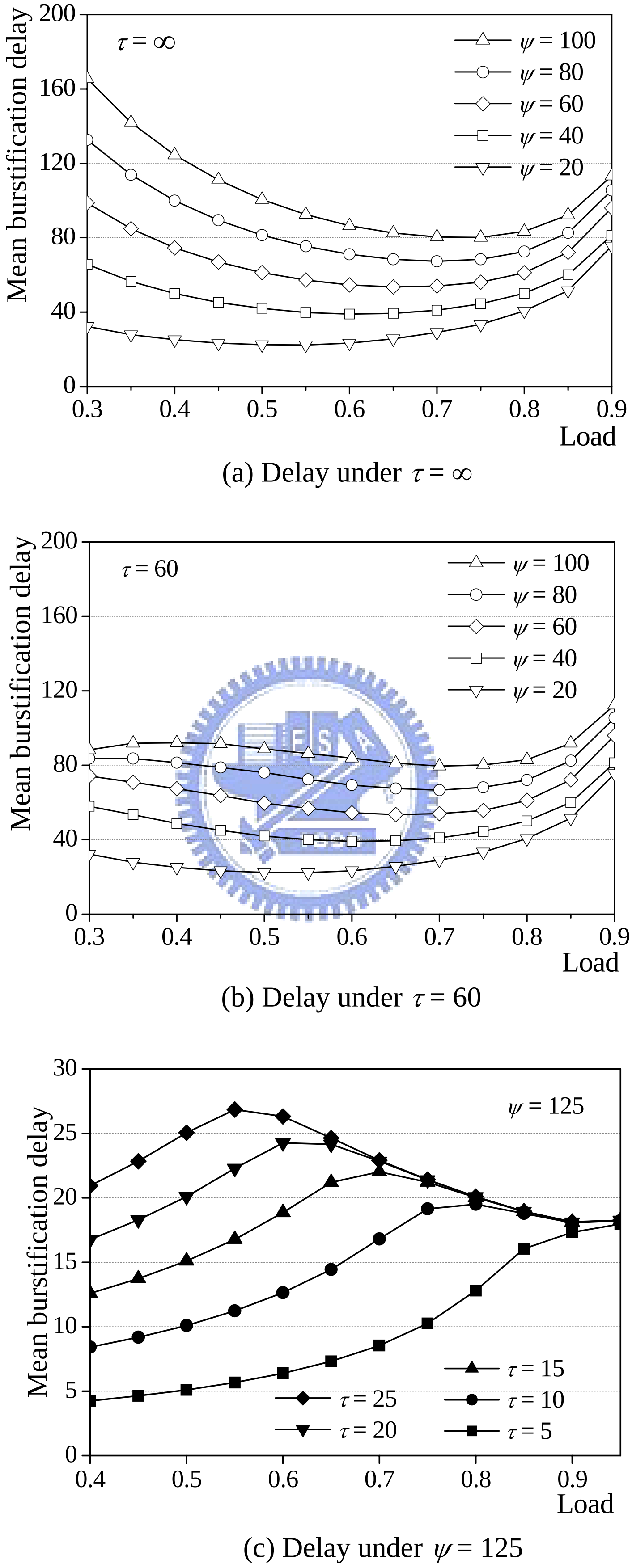
Figure 10. Delay QoS provision under various loads.



(a) Mean delay

(b) 99% delay bound

Figure 11. Delay QoS provision via the weight adjustment.



(a) Delay under $\tau = \infty$

(b) Delay under $\tau = 60$

(c) Delay under $\psi = 125$

Figure 12. Mean burstification delay under different ψ and τ.

Chapter 4. (ψ,τ)-Shaper and Departure Process Analysis

For clarity purposes, the operation of (ψ,τ)-Shaper is highlighted, particularly the BATr part of the system in the sequel. A burst of size ψ is generated and transmitted (see Figure 2(b)) if the total number of packets reaches ψ before the burst assembly time exceeds τ. Otherwise, a burst of size less than ψ is generated when BATr expires. The BATr is initialized as the τ value when it is *activated* or *reset*. The BATr is activated when the system is changed from being idle to busy due to new packet arrivals. The BATr is immediately *reset* when a burst departs leaving behind a non-empty queue.

In the sequel, we derive the departure process of a (ψ,τ)-Shaper system. The aggregate packet arrivals are modeled as a two-state Markov Modulated Bernoulli Process (MMBP) with batch arrivals. Then we carried out analytic computation and event-based simulation to validate the analysis and capture the departure process behavior under various parameter settings and traffic arrivals. Finally we observe the coefficient of variation between the burst size and the burst inter-departure time.

4.1 Departure Process Analysis

In a (ψ,τ)-Shaper system, bursts are served (transported) by one wavelength and forwarded via the same OLSP. In the analysis, (ψ,τ)-Shaper is considered on a discrete-time single-server queueing system, MMBP/G/1, in which a time slot is equal to the transmission of a fixed-length packet. The aggregate packet arrivals are assumed to follow a two-state MMBP that allows batch arrivals at each state. The two states are the *H* and *L* states, which correspond to high and low mean arrival rates, respectively. The MMBP is characterized by four parameters (α, β, λ_H, λ_L), where α is the probability of changing from state *H* to *L* in a slot, β is the probability of changing from state *L* to *H* in a slot, λ_H represents the probability of having a batch arrival at state *H*, and λ_L represents the probability of having a batch arrival at state *L*. For ease of description, the state change probability is denoted as $P_{i,j}$, $i, j \in \{H, L\}$. Namely, $P_{H,L} = 1 - P_{H,H} = \alpha$ and $P_{L,H} = 1 - P_{L,L} = \beta$. The batch sizes at state *H* and *L* possess distributions $b_H(m)$ and $b_L(m)$, with mean sizes $\bar{b}_H$ and $\bar{b}_L$, respectively. Let $\bar{L}$ represent the mean arrival rate (packets/slot) (i.e., the load), and *B* the burstiness of the arrival process, it thus have

$$B = \frac{\lambda_H \cdot \bar{b}_H}{\bar{L}} = \frac{\lambda_H \cdot \bar{b}_H}{\frac{\beta}{\alpha+\beta} \cdot \lambda_H \cdot \bar{b}_H + \frac{\alpha}{\alpha+\beta} \cdot \lambda_L \cdot \bar{b}_L}. \tag{4}$$

Figure 13 is drawn in aid of comprehension throughout the analysis. There are five possible events that sequentially occur in a slot as follows: (1) arrival process state change, (2) begin-of-burst departure, (3) packet arrivals, (4) end-of-burst departure, and (5) BATr activation/reset. While Events (1) and (2) occur at the beginning of a slot, Event (3) takes place at any time within a slot, and Events (4) and (5) occur at the end of a slot.

The departure process distribution consists of two parts: burst inter-departure time $(\tilde{t})$ and burst size $(\tilde{s})$ distributions. The burst inter-departure time takes values which are integer multiples of a slot. It is defined as the interval from the end of a previous burst to the beginning of the following burst. The goal is to

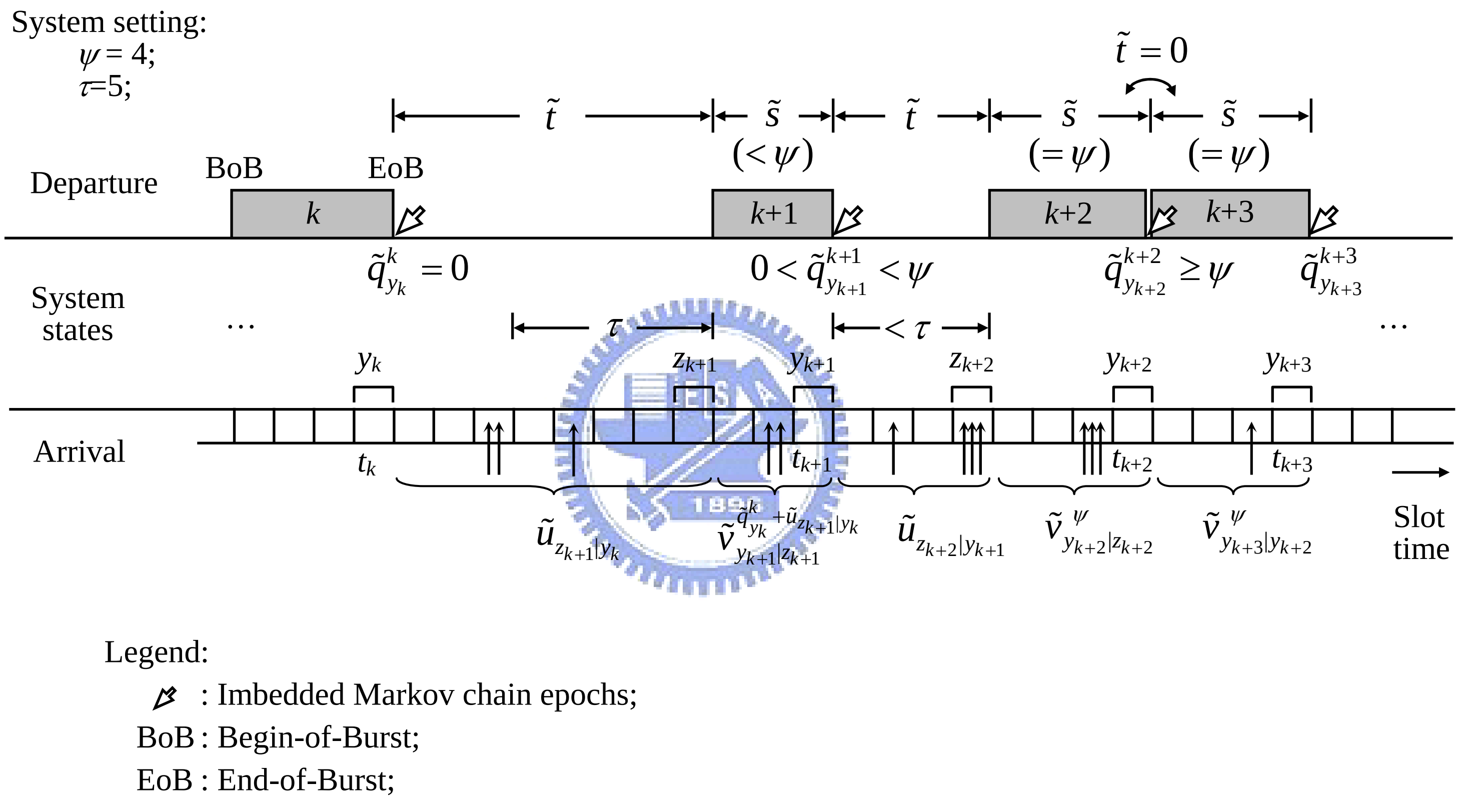

System setting:
$\psi = 4$;
τ=5;
$\tilde{t} = 0$
$\tilde{t}$
$\tilde{s}$
$(< \psi)$
$\tilde{t}$
$\tilde{s}$
$(= \psi)$
$\tilde{s}$
$(= \psi)$
Departure
BoB
EoB
k
k+1
k+2
k+3
System states
$\tilde{q}^{k}_{y_k} = 0$
$0 < \tilde{q}^{k+1}_{y_{k+1}} < \psi$
$\tilde{q}^{k+2}_{y_{k+2}} \geq \psi$
$\tilde{q}^{k+3}_{y_{k+3}}$
…
τ
$< \tau$
…
y_k
z_{k+1}
y_{k+1}
z_{k+2}
y_{k+2}
y_{k+3}
Arrival
t_k
t_{k+1}
t_{k+2}
t_{k+3}
$\tilde{u}_{z_{k+1}|y_k}$
$\tilde{v}^{\tilde{q}^{k}_{y_k}+\tilde{u}_{z_{k+1}|y_k}}_{y_{k+1}|z_{k+1}}$
$\tilde{u}_{z_{k+2}|y_{k+1}}$
$\tilde{v}^{\psi}_{y_{k+2}|z_{k+2}}$
$\tilde{v}^{\psi}_{y_{k+3}|y_{k+2}}$
Slot time
Legend:
: Imbedded Markov chain epochs;
BoB : Begin-of-Burst;
EoB : End-of-Burst;

Figure 13. (ψ, τ)-Shaper: departure process analysis.

find the joint distribution of $\tilde{t}$ and $\tilde{s}$, i.e., $P_{\tilde{t},\tilde{s}}(t,s),\ t \geq 0,\ 0 \leq s \leq \psi$. To approach it, we first obtain the queue length distribution seen by departing bursts, based on an imbedded Markov chain analysis placing the imbedded points at burst departure instants, as shown by the arrows in Figure 13.

Define random variable $\tilde{q}_{y_k}^{k}$ to be the number of packets left in queue behind by the k^{th} departing burst, say at time slot t_k, under the condition that the arrival process is in state y_k (=*H* or *L*) at t_k. Let random variable $\tilde{u}_{z|y}$ represent the number of packets that arrive during the burst inter-departure interval, under the condition that the arrival process changes from state *y* prior to the beginning of the interval, to state *z* at the end of the interval. Moreover, let random variable $\tilde{v}_{z|y}^{n}$ denote the number of packets that arrive during the transmission time of an *n*-packet burst, namely *n* slots, under the condition that the arrival process changes from state *y* prior to the beginning of the time interval, to state *z* at the end of the interval.

In Figure 13, the k^{th} burst depart at t_k, and there are no packet left in the queue. The next packet arrives at t_k +3. BATr is activated and set by τ. Since the traffic arrival is under low load, there are not enough packets arrival during t_k +3 and t_k +3+τ. At t_k +3+τ the BATr is expired, and the $(k+1)^{\text{st}}$ burst starts transmission at the next slot. The burst size is $\tilde{u}_{z_{k+1}|y_k}$. At the end of the $(k+1)^{\text{st}}$ burst transmission, there have some packets in the queue. BATr is reset at t_{k+1}. The $(k+2)^{\text{nd}}$ burst is generated while the queue size is more than ψ value. Finally at the end of the $(k+2)^{\text{nd}}$ burst transmitted, since the queue size $\tilde{q}_{y_{k+2}}^{k+2}$ is still more than ψ value, the $(k+3)^{\text{rd}}$ burst is immediately generated and transmits behind the $(k+2)^{\text{nd}}$ burst.

Accordingly, the next queue length $\tilde{q}^{k+1}_{y_{k+1}}$ is determined by the current queue length $\tilde{q}^{k}_{y_k}$, number of arrival during the inter-departure time $\tilde{u}_{z|y}$, number of departure packets, and number of arrival packets during transmission $\tilde{v}^{n}_{z|y}$. we find that

$$\tilde{q}^{k+1}_{y_{k+1}} = \left(\tilde{q}^{k}_{y_k} + \tilde{u}_{z_{k+1}|y_k} - \psi\right)^{+} + \tilde{v}^{\min\left\{\tilde{q}^{k}_{y_k}+\tilde{u}_{z_{k+1}|y_k},\ \psi\right\}}_{y_{k+1}|z_{k+1}}\ , \tag{5}$$

where $y_k,\ y_{k+1},\ z_{k+1} \in \{H, L\}$, and $(a)^{+} = \max\{a, 0\}$. In Equation (5), a non-negative term within the parentheses corresponds to the departure of a full-size (=ψ) burst; whereas a negative value corresponds to the departure of a burst due to BATr expiration. Significantly, since BATr is reset or activated after the k^{th} burst departure time, and $\tilde{u}_{z_{k+1}|y_k}$ and $\tilde{v}^{\min\left\{\tilde{q}^{k}_{y_k}+\tilde{u}_{z_{k+1}|y_k},\ \psi\right\}}_{y_{k+1}|z_{k+1}}$ are independent of any events that occur prior to time index *k*, $\left\{\tilde{q}^{k}_{y_k}, y_k \in \{H, L\},\ k \geq 1\right\}$ is hence an imbedded Markov chain.

Based on Equation (5), we can derive the limiting distributions of the queue length seen by departing bursts, rather than at all points in time. Notice that fortunately, such distribution is sufficient enough to determine the departure process distribution. Before we proceed, let us first derive the distribution for the number of packets that arrive in any given interval. Let $c^{t}_{r_t|r_0}(m)$ denote the probability that *m* packets have arrived in an interval of *t* slots, under the condition the arrival process changes from state r_0 (=*H* or *L*) prior to the beginning of the interval, to state r_t (=*H* or *L*) at the end of the interval. For $t = 0$, there is no packet arrived. we immediately have $c^{0}_{r_0|r_0}(m)$ =1 if *m*=0, and $c^{0}_{r_0|r_0}(m)$ =0, otherwise. For $t \geq 1$, $c^{t}_{r_t|r_0}(m)$ can be recursively computed as

$$c^t_{r_t|r_0}(m) = \sum_{x\in\{H,L\}} P_{x,r_t} \cdot \left[c^{t-1}_{x|r_0}(m)\cdot\left(1-\lambda_{r_t}\right) + \sum_{n=1}^{m} c^{t-1}_{x|r_0}(m-n)\cdot\lambda_{r_t}\cdot b_{r_t}(n) \right], \qquad (6)$$

where $r_0, r_t \in \{H, L\}$, P_{x,r_t} is the probability that the arrival process changes from state *x* to state r_t. The first term within the square bracket in Equation (6) corresponds to that all *m* packets arrive in the first *t*-1 slots and no packet arrives in the last slot. The second term represents that *m-n* packets arrived in the first *t*-1 slots and a batch of *n* $(n \le m)$ packets that arrive in the last slot with probability $\lambda_{r_t} \cdot b_{r_t}(n)$.

With the "$(\ )^+$" sign removed, Equation (5) can be expanded into three cases, as

$$\tilde{q}^{k+1}_{y_{k+1}} = \begin{cases} \tilde{q}^k_{y_k} - \psi + \tilde{v}^{\psi}_{y_{k+1}|y_k} & \text{, if } \tilde{q}^k_{y_k} \ge \psi \\ \tilde{q}^k_{y_k} + \tilde{u}_{z_{k+1}|y_k} - \psi + \tilde{v}^{\psi}_{y_{k+1}|z_{k+1}} & \text{, if } \tilde{q}^k_{y_k} < \psi,\ \tilde{q}^k_{y_k} + \tilde{u}_{z_{k+1}|y_k} \ge \psi \\ \tilde{v}^{\tilde{q}^k_{y_k} + \tilde{u}_{z_{k+1}|y_k}}_{y_{k+1}|z_{k+1}} & \text{, if } \tilde{q}^k_{y_k} + \tilde{u}_{z_{k+1}|y_k} < \psi \end{cases}. \qquad (7)$$

Notice that $\tilde{u}_{z_{k+1}|y_k}$ is absent from the first case of Equation (7) due to that the inter-departure time is zero if a departing burst leaves behind a system with ψ or more packets. we now compute the queue length distribution by first conditioning on the value of $\tilde{q}^k_{y_k}$ and separating case one from cases two and three in Equation (7), as

$$P\left[\tilde{q}^{k+1}_{y_{k+1}} = d\right] = \sum_{q=\psi}^{\psi+d} \sum_{y_k\in\{H,L\}} F_1 \cdot P\left[\tilde{q}^k_{y_k} = q\right] + \sum_{q=0}^{\psi-1} \sum_{y_k, z_{k+1}\in\{H,L\}} F_2 \cdot P\left[\tilde{q}^k_{y_k} = q\right], \qquad (8)$$

where

$$\begin{aligned} F_1 &\equiv P\left[\tilde{q}_{y_k}^{k} - \psi + \tilde{v}_{y_{k+1}|y_k}^{\psi} = d \mid \tilde{q}_{y_k}^{k} = q\right] \\ &= P\left[\tilde{v}_{y_{k+1}|y_k}^{\psi} = d - q + \psi\right] \\ &= c_{y_{k+1}|y_k}^{\psi}\left(d - q + \psi\right) \end{aligned} \tag{9}$$

and

$$\begin{aligned} F_2 &\equiv P\left[\left(\tilde{q}_{y_k}^{k} + \tilde{u}_{z_{k+1}|y_k} - \psi\right)^{+} + \tilde{v}_{y_{k+1}|z_{k+1}}^{\min\left\{\tilde{q}_{y_k}^{k} + \tilde{u}_{z_{k+1}|y_k},\psi\right\}} = d \mid \tilde{q}_{y_k}^{k} = q\right] \\ &= \sum_{u=0}^{\psi-q-1} P\left[\tilde{v}_{y_{k+1}|z_{k+1}}^{q+u} = d\right] \cdot P\left[\tilde{u}_{z_{k+1}|y_k} = u \mid \tilde{q}_{y_k}^{k} = q\right] \\ &\quad + \sum_{u=\psi-q}^{d+(\psi-q)} P\left[\tilde{v}_{y_{k+1}|z_{k+1}}^{\psi} = d - \left(q + u - \psi\right)\right] \cdot P\left[\tilde{u}_{z_{k+1}|y_k} = u \mid \tilde{q}_{y_k}^{k} = q\right]. \\ &= \sum_{u=0}^{\psi-q-1} c_{y_{k+1}|z_{k+1}}^{q+u}\left(d\right) \cdot P\left[\tilde{u}_{z_{k+1}|y_k} = u \mid \tilde{q}_{y_k}^{k} = q\right] \\ &\quad + \sum_{u=\psi-q}^{d+(\psi-q)} P\left[c_{y_{k+1}|z_{k+1}}^{\psi}\left(d - q - u + \psi\right)\right] \cdot P\left[\tilde{u}_{z_{k+1}|y_k} = u \mid \tilde{q}_{y_k}^{k} = q\right] \end{aligned} \tag{10}$$

To proceed, $P\left[\tilde{u}_{z_{k+1}|y_k} = u \mid \tilde{q}_{y_k}^{k} = q\right]$ in Equation (10) needs to solve first. It can be resolved by considering five cases depending on different ranges of u and q values as given in Equation (11) below. First of all, in case (1) when $q \geq \psi$ a full-size burst is immediately transmitted, yielding $\tilde{t} = 0$. Thus, the probability under $u = 0$ is one. In case (2), when $0 < q < \psi$ but $u + q \geq \psi$, the total number of packets in the queue must exceed ψ the first time at a particular slot before the BATr expires. Namely, within an interval t of less than or equal to τ, there arrives a total of m $\left(0 \leq m \leq \psi - q - 1\right)$ packets during t-1 slots, and exactly at this final slot, a batch of $u - m$ packets arrives, making $m + \left(u - m\right) + q \geq \psi$. The total number of packets exceed ψ the first time at the t^{th} slot. As opposed to case (2), in case (3) BATr expires.

That is, the total number of packets that arrive within an interval of τ is u $\left(u<\psi-q\right)$ and $u+q<\psi$. The probability is $c^{\tau}_{z_{k+1}|y_k}(u)$.

Case (4) in Equation (11) under $q=0$ corresponds to the termination of a busy period of the system. Notice that BATr is not activated until the arrival of the first batch with m $\left(0<m\le u\right)$ packets. This explains the term within the square bracket. Under such condition, this case becomes identical to that when a departing burst leaves behind a system with m packets, with the probability shown before the product sign. Notice that, this probability can be obtained by applying cases (1) to (3) once, depending on the m value. Combining the results from the cases discussed above, it has

$$P\left[\tilde{u}_{z_{k+1}|y_k}=u\,|\,\tilde{q}^{k}_{y_k}=q\right]$$
$$=\begin{cases} 1 & ,\text{ if } q\ge\psi,\ u=0,\ z_{k+1}=y_k \\ \displaystyle\sum_{t=1}^{\tau}\sum_{m=0}^{\psi-q-1}\sum_{x\in\{H,L\}} c^{t-1}_{x|y_k}\left(m\right)\cdot P_{x,z_{k+1}}\cdot\lambda_{z_{k+1}}\cdot b_{z_{k+1}}\left(u-m\right) & ,\text{ if } 0<q<\psi,\ u\ge\psi-q \\ c^{\tau}_{z_{k+1}|y_k}\left(u\right) & ,\text{ if } 0<q<\psi,\ u<\psi-q \\ \displaystyle\sum_{r\in\{H,L\}}\sum_{m=1}^{u}\left\{\begin{matrix} P\left[\tilde{u}_{z_{k+1}|r}=u-m\,|\,\tilde{q}^{k}_{r}=m\right]\cdot \\ \displaystyle\sum_{t=1}^{\infty}\sum_{x\in\{H,L\}} c^{t-1}_{x|y_k}\left(0\right)\cdot P_{x,r}\cdot\lambda_r\cdot b_r\left(m\right)\end{matrix}\right\} & ,\text{ if } q=0 \\ 0 & ,\text{ otherwise} \end{cases} \tag{11}$$

With Equations (6) and (8)-(11), the limiting queue length distribution under the arrival process being at state H or L, can be given by

$$P\left[\tilde{q}_y = d\right] = \lim_{k\to\infty} P\left[\tilde{q}_y^k = d\right],\ y \in \{H, L\}. \tag{12}$$

We are now in the position to determine the departure process distribution, $P_{\tilde{t},\tilde{s}}(t,s)$. There are four cases depending on different *t* and *s* values to be considered. First, in Case I when $t=0$, it is clear that the queue length is larger than ψ behind the burst departure. We get that

Case I: $t=0$

$$P_{\tilde{t},\tilde{s}}(t,s) = \begin{cases} \sum\limits_{y\in\{H,L\}} P\left[\tilde{q}_y \geq \psi\right] & \text{, if } s=\psi \\ 0 & \text{, if } s<\psi \end{cases}. \tag{13}$$

Second, Case II corresponds to the transmission of a full-size burst due to having a total of ψ or more packets before the BATr expires. Hence, we obtain that

Case II: $0<t<\tau$

$$P_{\tilde{t},\tilde{s}}(t,s) = \begin{cases} \sum\limits_{\substack{q+m<\psi; \\ y,i,j\in\{H,L\}}} \left[c_{i|y}^{t-1}(m) \cdot P_{i,j} \cdot \lambda_j \cdot \sum\limits_{n\geq\psi-q-m} b_j(n) \right] \cdot P\left[\tilde{q}_y = q\right] & \text{, if } s=\psi \\ 0 & \text{, if } s<\psi \end{cases}. \tag{14}$$

Third, in Case III when $t=\tau$ and $s=\psi$, the total number of packets in the system exceeds ψ exactly at the same time when the BATr expires. Otherwise, if $s<\psi$, a burst of size less than ψ is transmitted due to BATr time-out. That means,

Case III: $t=\tau$

$$P_{\tilde{t},\tilde{s}}\left(t,s\right)=\begin{cases}\sum\limits_{\substack{q+m<\psi;\\ y,i,j\in\{H,L\}}}\left[c_{i|y}^{t-1}\left(m\right)\cdot P_{i,j}\cdot\lambda_j\cdot\sum\limits_{n\geq\psi-q-m}b_j\left(n\right)\right]\cdot P\left[\tilde{q}_y=q\right] & \text{, if } s=\psi\\ \sum\limits_{\substack{0<q<\psi;\\ y,i\in\{H,L\}}}c_{i|y}^{\tau}\left(s-q\right)\cdot P\left[\tilde{q}_y=q\right] & \text{, if } s<\psi\end{cases}. \tag{15}$$

Finally, under the last case when $t>\tau$, the departing burst must have left an empty system $\left(P\left[\tilde{q}_y=0\right]\right)$ resulting in the deactivation of the BATr. The timer remains deactivated until the arrival of the first batch of packets. Then, whether the next departing burst is a full-size one or not depends on the total number of arriving packets, as

Case IV: $t>\tau$

$$P_{\tilde{t},\tilde{s}}\left(t,s\right)=\begin{cases}\sum\limits_{\substack{m<\psi;\\ y,i,j,h\in\{H,L\}}}\left[c_{i|y}^{t-\tau-1}\left(0\right)\cdot c_{j|i}^{\tau}\left(m\right)\cdot P_{j,h}\cdot\lambda_h\cdot\sum\limits_{n\geq\psi-m}b_h\left(n\right)\right]\cdot P\left[\tilde{q}_y=0\right]\\ \qquad\qquad\text{, if } s=\psi\\ \sum\limits_{y,i,j,h\in\{H,L\}}\left[c_{i|y}^{t-\tau-1}\left(0\right)\cdot P_{i,j}\cdot\lambda_j\cdot\sum\limits_{n\geq1}b_j\left(n\right)\cdot c_{h|j}^{\tau}\left(s-n\right)\right]\cdot P\left[\tilde{q}_y=0\right]\\ \qquad\qquad\text{, if } s<\psi\end{cases}. \tag{16}$$

Combining Equations (13)-(16), we achieve the joint-form departure process distribution.

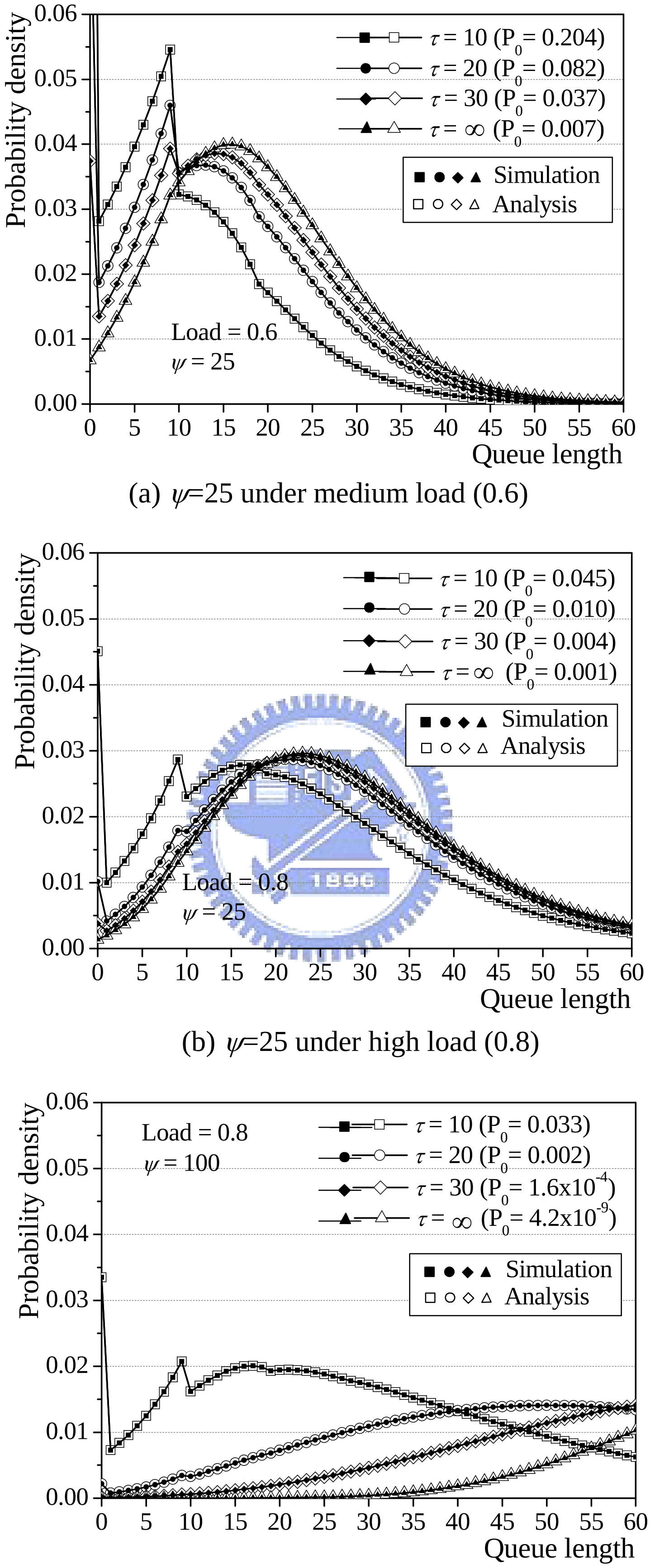
4.2 Numerical Results

We carried out analytic computation and event-based simulation to validate the analysis and capture the departure process behavior under various parameter settings and traffic arrivals. Analytical and simulation results of the queue length distribution and departure process distributions (inter-departure and burst size distributions) are shown in Figures 14, 15, 16, and 17 respectively. In the system setting, we adopt ψ = 25 or 100, and τ = 10, 20, 30, or ∞. In the MMBP, we adopt $\alpha = 0.225$, $\beta = 0.025$; $\lambda_H = 0.36$ and $\lambda_L = 0.0933$ at load 0.6; and $\lambda_H = 0.48$ and $\lambda_L = 0.1244$ at load 0.8. The batch size in any of states *H* and *L* was uniformly distributed between 1 and 9 ($\bar{b}_H = \bar{b}_L = 5$). Accordingly, the burstiness of traffic is *B* = 3 under both loads.

First, all analytical results are in profound agreement with simulation results. As shown in Figures 14(a) and 14(b), the queue length observed at burst departure time is small under lower load and lower τ value. It is caused by that most of the bursts are generated while the BATr is expired. If the τ value is large enough, the system accumulates ψ packets to assemble a burst. Since the large burst takes more transmission time than small burst, the number of packets arriving during burst transmission in large τ value is more than that in small τ value. The distribution of queue length in large τ value is to center on large queue size. As shown in Figure 14(c), the queue size distribution under large ψ value is expands to a greater scope. Finally, it is interesting that there are some spikes at queue-length=9 in the queue length distribution (see Figure 14). The phenomenon is caused by the maximum batch size of 9 in the arrival process.

In addition, we observe that the inter-departure time distribution is sensitive to ψ and τ. Under a medium load ($\bar{L}$=0.6) or high load ($\bar{L}$=0.8) condition, we observe the inter-departure time of zero (burst size=ψ=25 or 100) occurs with the larger probability under all τ values. It can be shown that during the burst transmission, there are enough packets (>ψ) accumulate in the queue under a high load. The other larger probability for different τ settings occurs at the inter-departure time being equal to the corresponding τ value. It is reasonable for that while system finish a burst transmission and queue is not empty, next burst wait at most τ slots. The results are shown by the spikes in Figures 15(a), 16(a) and 17(a).

The burst size distribution is also sensitive to ψ and τ. As shown in Figures 15(b), 16(b) and 17(b), if the τ value is small (=10), the system does not accumulate enough packets before BATr expired. The probability of burst size less than ψ is large under lower load (see Figure 15(b)) or under large ψ value. If the τ value is large enough (approached to limit), burst is always generated under the ψ^{th} packet arrived. The probability of burst size equal to ψ is one. Deciding an appropriate τ value can make the burst almost complete and the delay could be controlled. For example, if we want to get 95% complete burst under ψ=25 and load=0.8, we must set τ=30. In the last, it has similar result with Figure 14 that there is a turning point at burst size 9 duo to the maximum batch size is 9

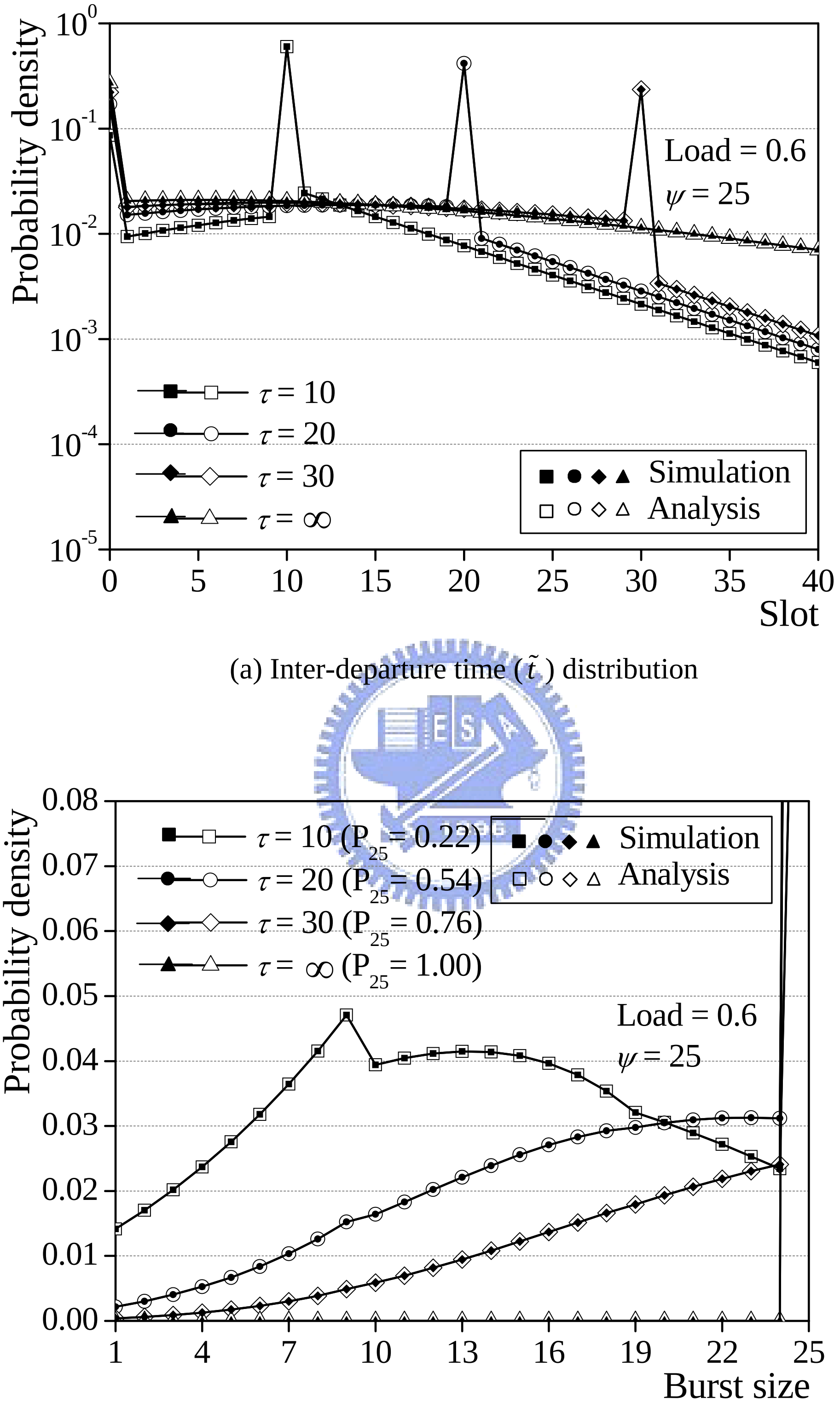


(a) ψ=25 under medium load (0.6)

(b) ψ=25 under high load (0.8)

(c) ψ=100 under high load (0.8)

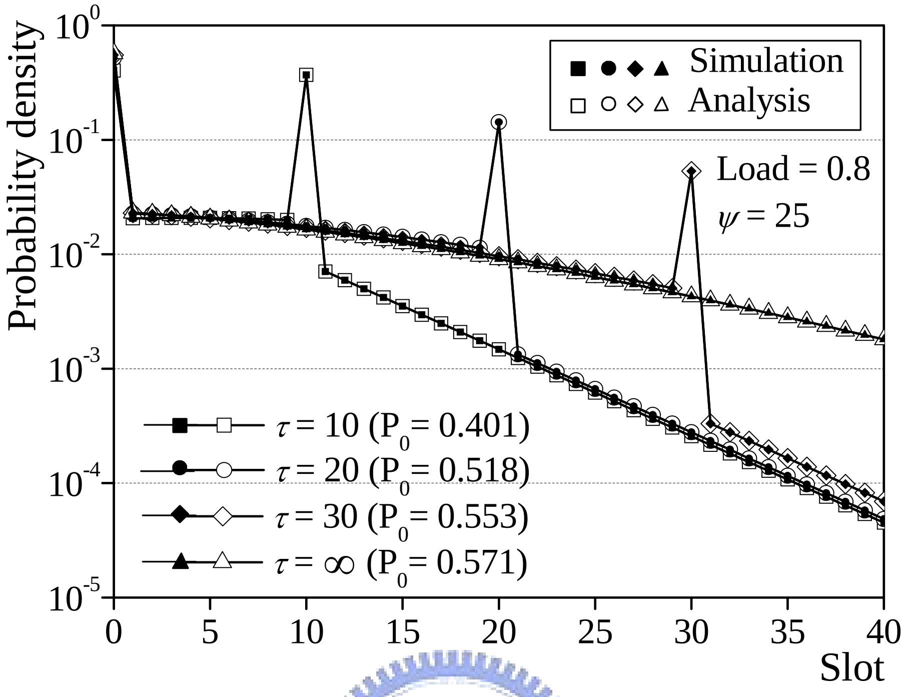
Figure 14. System queue length distribution.



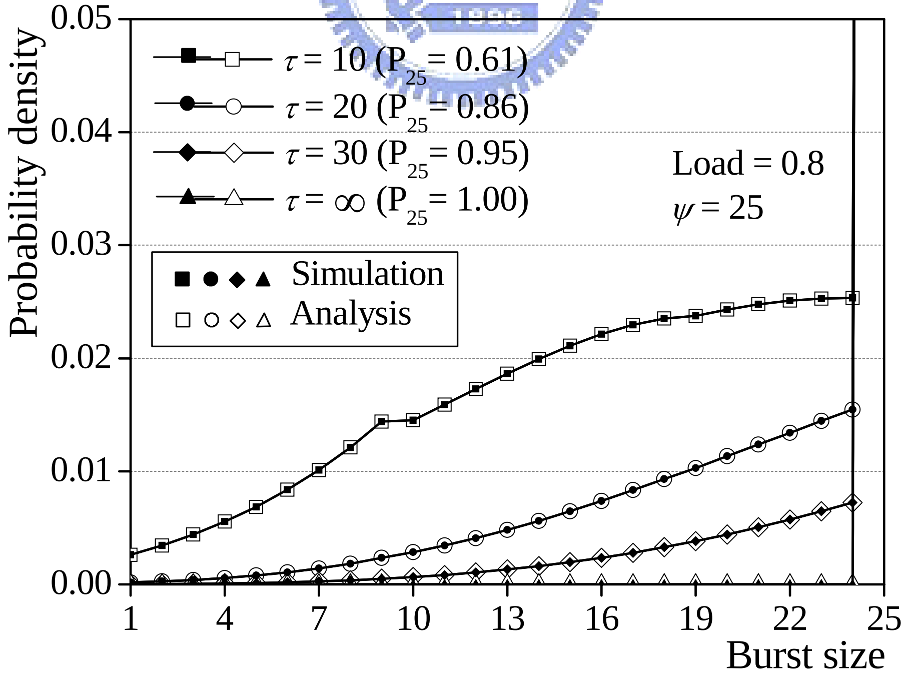
(a) Inter-departure time ($\tilde{t}$) distribution

(b) Burst size ($\tilde{s}$) distribution

Figure 15. Departure process distributions (ψ=25 under medium load).

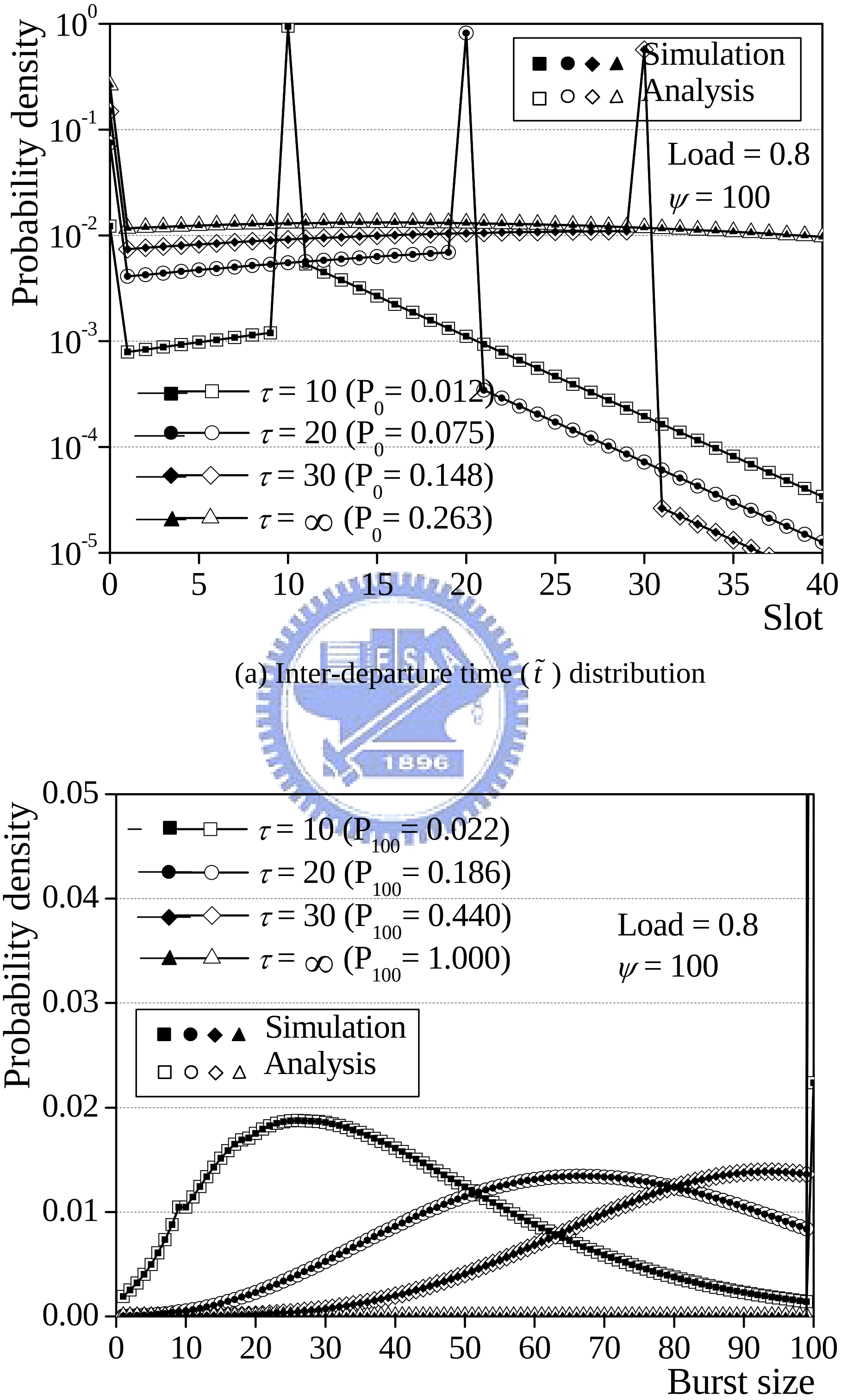


(a) Inter-departure time ($\tilde{t}$) distribution



(b) Burst size ($\tilde{s}$) distribution

Figure 16. Departure process distributions (ψ=25 under high load).



(a) Inter-departure time ($\tilde{t}$) distribution

(b) Burst size ($\tilde{s}$) distribution

Figure 17. Departure process distributions (ψ=100 under high load).

To examine the effectiveness of shaping, we further compute the Coefficient of Variation (CoV) for the inter-departure time and burst size, under three ψ values (ψ = 1, 10, and 100), three τ values (τ = 20, 40, and 80) and various MMBP arrivals (B = 1, 3, and 5; $\bar{b}_H = \bar{b}_L$ = 5, 7, and 9) under $\alpha = 0.225$, $\beta = 0.025$. The CoV is a measure of dispersion of a probability distribution. It is defined as

$$\mathrm{CoV}(\tilde{a}) = \frac{\text{Standard deviation of } \tilde{a}}{\text{Mean of } \tilde{a}} = \frac{\sqrt{\mathrm{Var}[\tilde{a}]}}{\mathrm{E}[\tilde{a}]},$$

where $\tilde{a}$ is a random variable. Distributions with CoV < 1 (such as an Erlang distribution) are considered low-variance, while those with CoV > 1 (such as a two-state MMBP distribution) are considered high-variance. Notice that the setting of ψ = 1 corresponds to a FIFO system with no shaping. Numerical results are plotted in Figures 18 and 19.

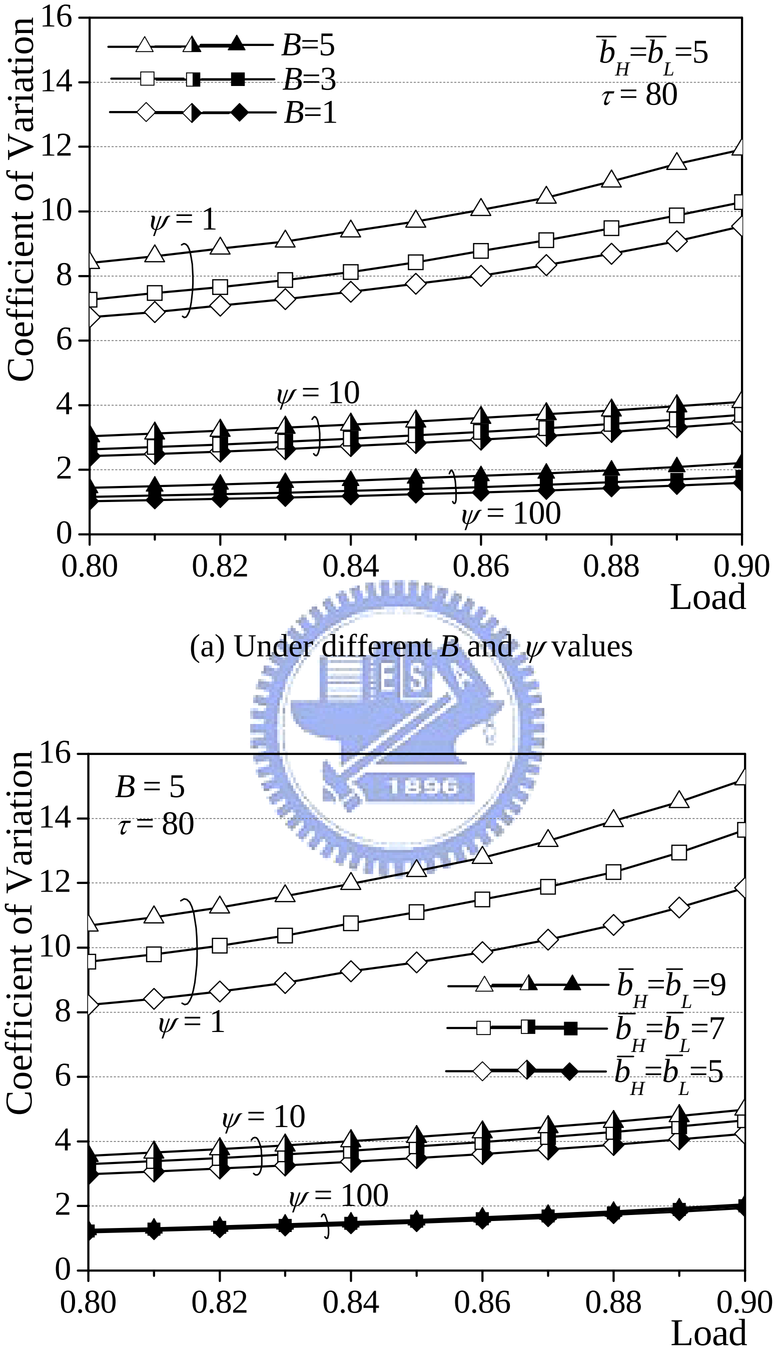
As shown in Figure 18, as expected, the CoV of the inter-departure time increases with the offered load. Crucially, under any MMBP arrival, we discover that CoV of inter-departure time is very large under ψ = 1. It means that departure process is of high variance. Then the CoV declines significantly with larger ψ values, yielding substantial reduction in burst loss probability. This fact will be again revealed in the network-wide simulation results presented in the Chapter 5.

Moreover, we observe from Figures 18 and 19 that the burstiness and batch size of the original MMBP arrival has impact on any of the CoVs- the higher the burstiness and batch size, the greater the CoV. Nevertheless, the impact is insignificant compared to the effect of using different ψ and τ values. As displayed in

Figure 19, the CoV of the burst size declines with larger τ values under any MMBP arrival. Notice that greater τ values imply larger burst sizes, namely better shaping effect.

We then investigate the impact of (ψ, τ) setting on mean burst size. In this simulation, we adopted the batch size being uniformly distributed between 1 and 9 (i.e., $\bar{b}_H = \bar{b}_L = 5$) for MMBP arrivals. By the result of Figures 18 and 19, we have learned that the larger ψ and τ values results in better shaping. To satisfy a given mean burst size, the results in Figure 20 can serve as a guideline for the selection of appropriate τ values under different traffic loads. For example, if $\psi = 200$, to achieve a mean burst size of 100 under load 0.6, the applicable values of τ are 40 and above.

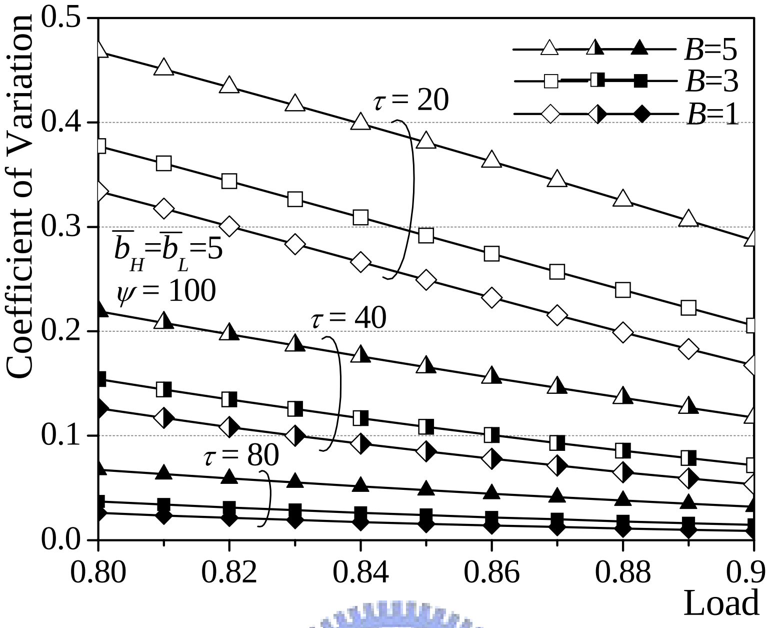
The results of the mean burst size associated with a set of (ψ, τ) pairs are indicated in Figure 21. Under medium load (see Figure 21(a)), τ value must be set large enough to achieve large burst size. Under high load, ψ has acted the main role to decide the mean burst size. For example (see Figure 21(b)), if $\tau > 20$, mean burst size is almost equal to the ψ value.



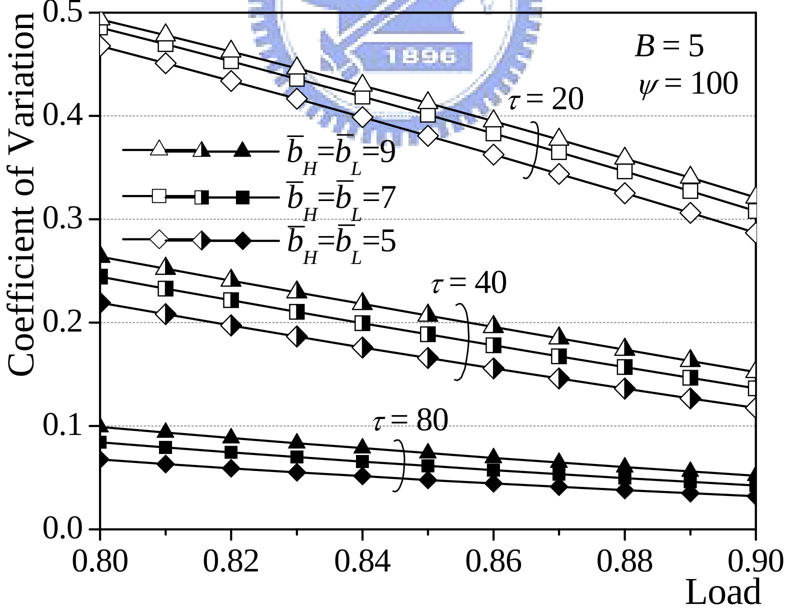
(a) Under different B and ψ values

(b) Under different batch sizes and ψ values

Figure 18. CoV of the inter-departure time.

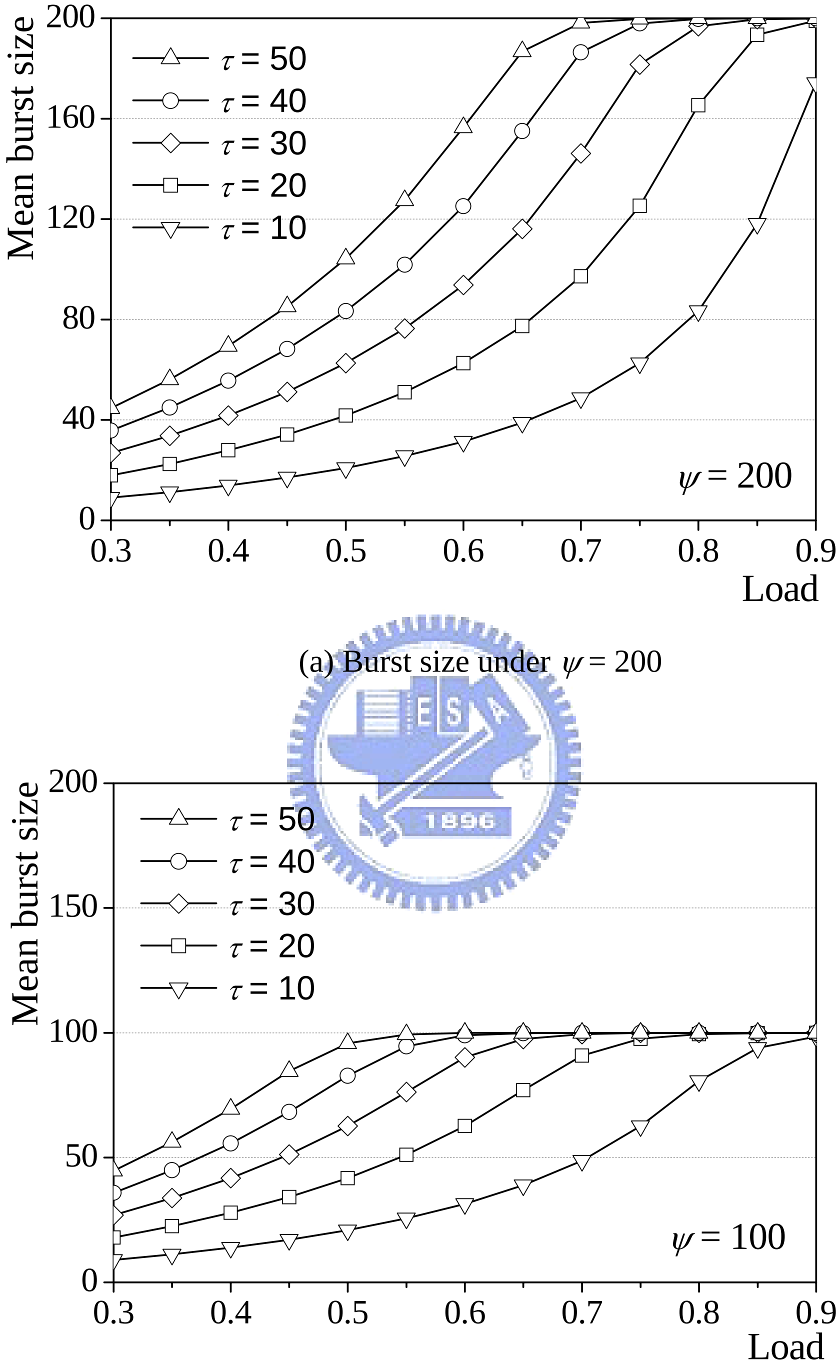


(a) Under different B and τ values



(b) Under different batch sizes and τ values

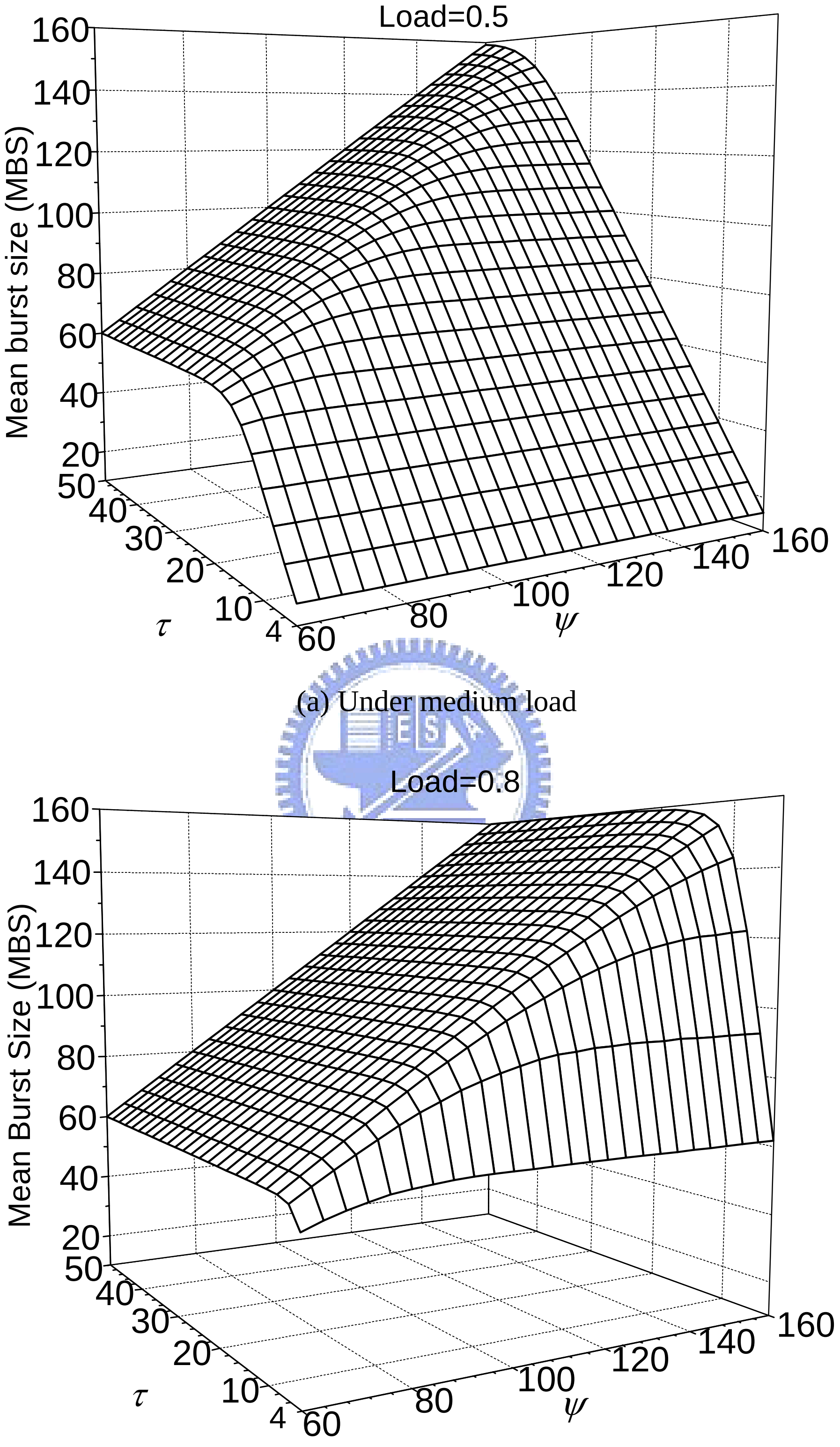
Figure 19. CoV of the burst size.



(a) Burst size under $\psi = 200$

(b) Burst size under $\psi = 100$

Figure 20. Mean burst size under different τ.



(a) Under medium load

(b) Under high load

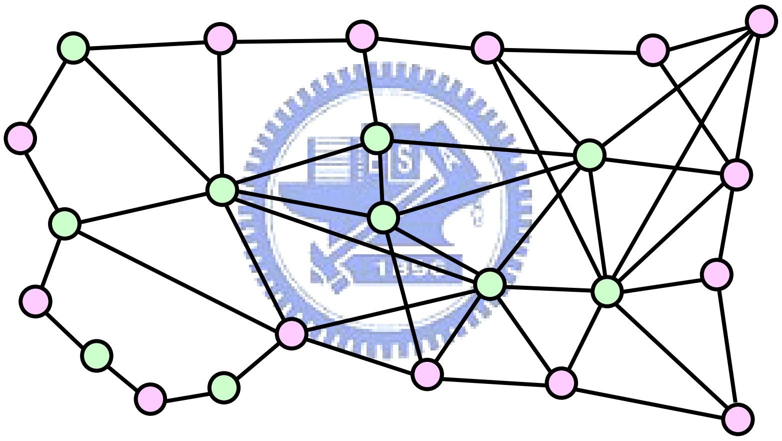
Figure 21. Mean burst size associated with ψ and τ.

Chapter 5. Loss QoS Provision and Performance Comparison

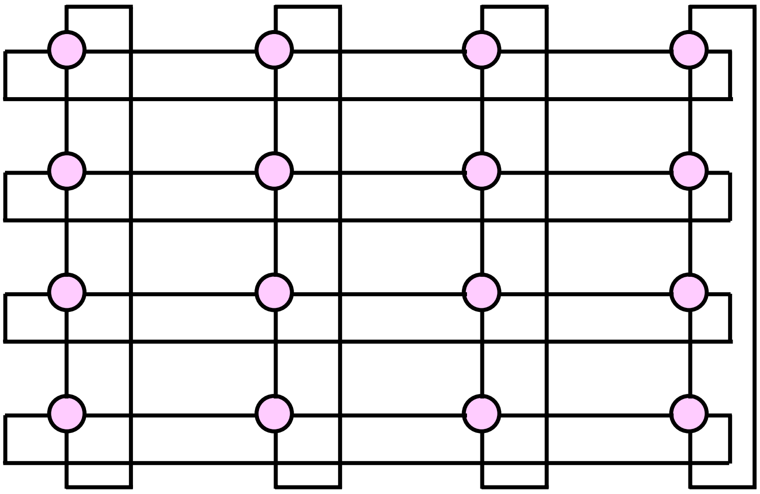
In this chapter, we demonstrate the performance of (ψ,τ)-Shaper from three aspects: (1) traffic shaping effect on loss performance; (2) loss QoS provisioning for OCPS networks; and (3) loss QoS performance comparison between the OCPS and the JET-based OBS [9] networks. For ease of description throughout the chapter, we refer to the three networks- OCPS without (ψ,τ)-Shaper, OCPS with (ψ,τ)-Shaper, and JET-based OBS, as the baseline, OCPS, and OBS networks, respectively.

Rather than considering one single switching node, we have simulated an entire optical network with QoS burst truncation and full wavelength conversion capabilities equipped in each switching node. The networks (see Figure 22) we used in the experiment are: (1) the well-known ARPANET-like network [30] with 24 nodes and 48 links, in which 14 nodes are randomly selected as edge nodes; and (2) a 4x4-torus network [30] with 16 nodes and 32 links, in which all nodes are edge nodes. For any two edge nodes (nodes *i*, *j*), there is a connection from node *i* to node *j* for each loss class. All connections from node *i* to node *j* are using the same OLSP routing. The routing is subject to load balance of the network. Each link has up to 100 wavelengths, transmitting at 1 Gb/s, or one 60-byte packet per slot of duration 0.48μs. In simulations, departing bursts from ingress nodes can be served by any free wavelength, though, only after the previous burst has been fully transmitted. we measure two performance metrics- burst and packet loss probabilities. The burst loss probability is measured when QoS burst truncation is disregarded, i.e., the entire burst is dropped as a result of no free wavelength. Otherwise, the packet loss probability is computed.

In simulations, we generate packets according to the MMBP with $\alpha = 0.225$, $\beta = 0.025$, and the batch size in both H and L states being uniformly distributed between 1 and 9 ($\overline{b}_H = \overline{b}_L = 5$). For a given load ($\overline{L}$), according to Equation (4), traffic burstiness (B) is then uniquely determined by λ_H. We adopt three different burstiness (B=1, 3, and 5) in simulations. For comparison, we also generate Binomial-distributed arrivals that have been used to model smooth traffic. The



(a) ARPANET-like network



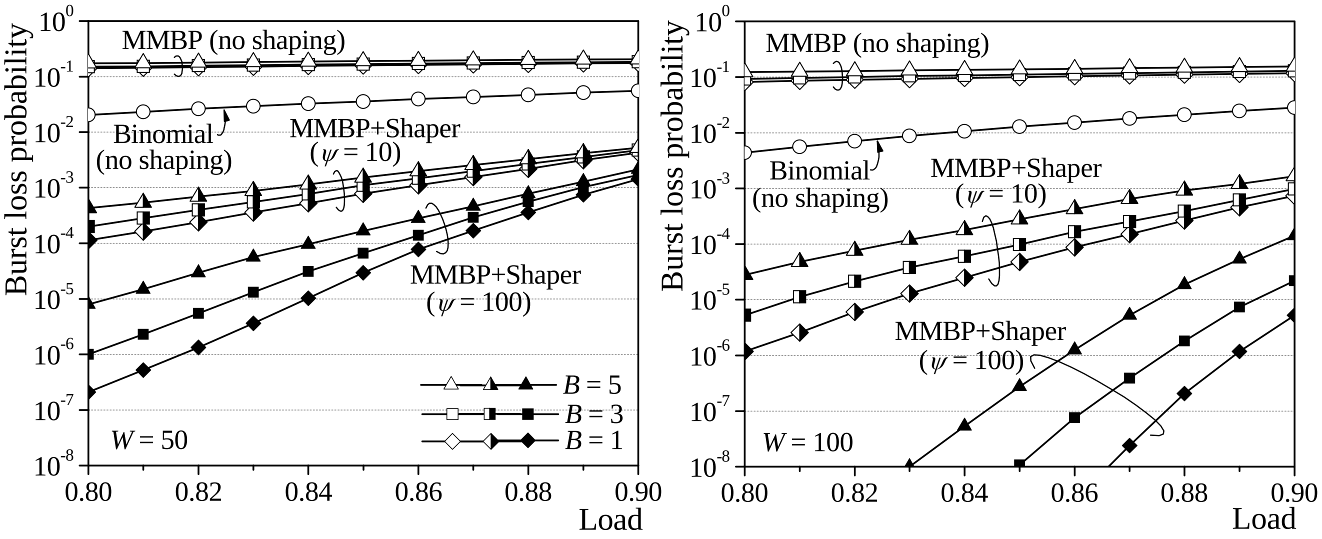
(b) 4x4-torus network

Figure 22. Network topology.

probability that a packet arrives at each slot is equal to the mean load $(\overline{L})$, yielding a total offered load of $W \cdot \overline{L}$, where W is the number of wavelengths. The utilization of ARPANET network is $0.4447 \cdot \overline{L}$, and of 4x4-torus network is $0.5375 \cdot \overline{L}$. Simulations are terminated after reaching 95% confidence interval. In the sequel, we explore these three aforementioned aspects in the three sections, respectively.

5.1 Traffic Shaping Effect

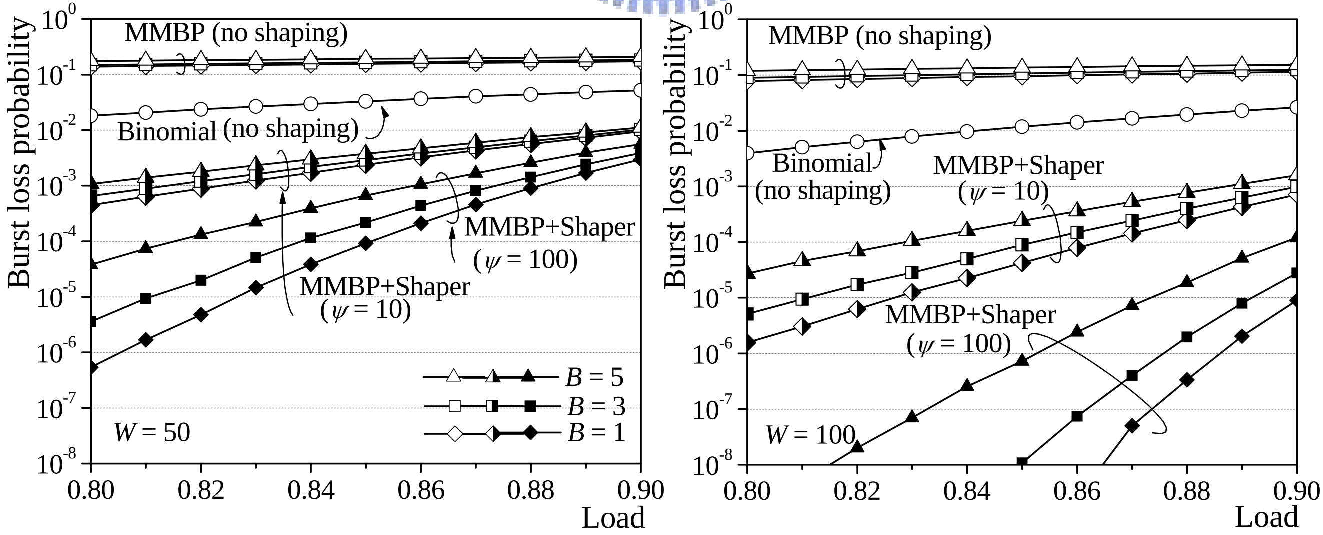
To examine the traffic shaping effect, we draw a comparison of burst loss probability between the baseline and OCPS networks under ARPANET network and 4x4-torus network. Simulation results are plotted in Figures 23 and 24. We first observe from the figure that increasing ψ value from 10 to 100, burst loss probability is decreasing several orders of magnitude. The results are consistent with my previous analytic CoV results- the greater the ψ value, the lower the CoV (see Figures 18 and 19) and the burst loss probability. As shown in Figures 23(a) and 24(a), compared with the baseline no-shaping network under MMBP arrivals, the OCPS network achieves more than five orders of magnitude reduction in burst loss probability under W=50, ψ =100, and $\overline{L}$ = 0.8 and below. Compared to smooth Binomial arrivals, the OCPS network with traffic shaping still yields several orders of magnitude improvement in burst loss probability. As shown in Figures 23(b) and 24(b), we discover that the improvement of loss probability is even more compelling in the presence of a large number of wavelengths (W=100) due to higher statistical multiplexing gain. Loss performance in ARPANET network and 4x4-torus network are similar.



(a) *W*=50

(b) *W*=100

Figure 23. Traffic shaping effect: a comparison between the OCPS and baseline networks under ARPANET network.



(a) *W*=50

(b) *W*=100

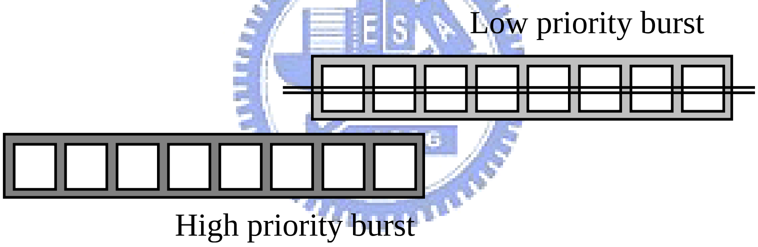
Figure 24. Traffic shaping effect: a comparison between the OCPS and baseline networks under 4x4-torus network.

5.2 Loss QoS Provision

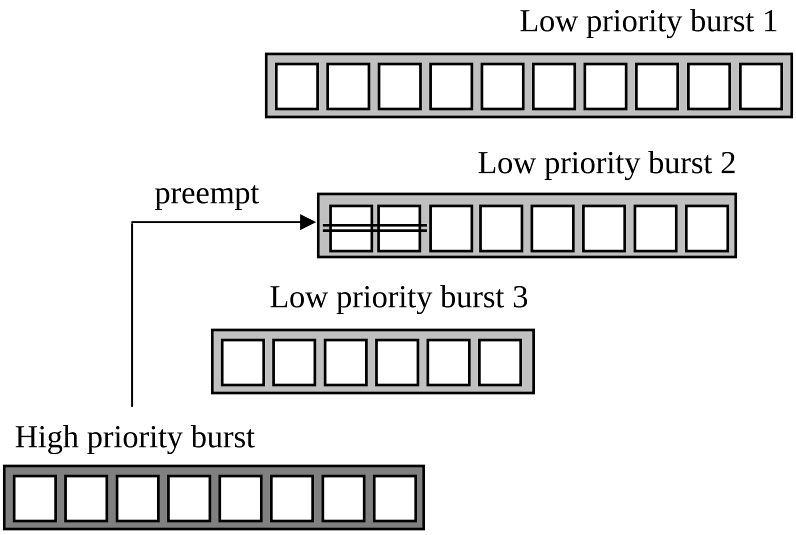
For OCPS networks, (ψ,τ)-Shaper facilitates loss QoS provisioning at edge nodes by means of burst size (ψ) adjustment. Higher priority classes are assigned larger burst sizes. Notice that in parallel, each switching node within the network performs QoS burst truncation in the absence of free wavelengths. Specifically, an arriving high priority burst that finds no free wavelength will preempt a burst that is of *lower* priority (than the arriving burst's priority), and that has the *least* amount of data left unsent. Namely, the preemption is made on a "least-harm" basis.

The least-harm preemption (see Figure 25) is proposed on QoS Control Processor (QCP) in OPSINET technology [22]. QCP is responsible for prioritized contention resolution and header integrity assurance. It is worth noting that, due to Arrayed Waveguide Grating design, any two bursts arriving from different input ports never contend. On the contrary, contention will occur for bursts arriving from the same input port but carried by different wavelengths, and destined for the same output port. Basically, to switch a burst to the destined output port, an idle wavelength is selected. If all wavelengths are busy, higher priority bursts receive absolute precedence over lower-priority bursts. That is, owing to buffer-less, one of the lower-priority bursts being served is preempted and discarded. It is worth noting that if partially destructed lower-priority bursts are still transmitted, the loss probability can be much improved. For example in Figure 25, if QCP doesn't support the partially collided burst, the high-priority burst randomly selected a lower-priority burst to preempt under busy system and the lower-priority burst will be totally discard. If QCP support the partially collided burst, due to the least-harm preemption, the high priority burst preempts the lower-priority burst with least overlap.

We now draw a comparison of the loss probability with multiple priority class between QCP and a prioritized queueing system described in the sequel. Notice that the traffic entering QCP has been previously shaped via the (ψ,τ)-Scheduler/Shaper at the ingress routers. The prioritized queueing system we analyze contains Y priority classes with K wavelengths under Poisson arrivals and exponentially distributed service, namely an ***M/M/K/K* loss system with *Y* priorities**. In such system, a high-priority burst preempts a randomly selected lower-priority burst if all wavelengths are found busy upon arrival. Let λ_i and μ_i denote the



(a) Without support of partially collided bursts



(b) With support of partially collided bursts

Figure 25. QCP with least-harm preemption.

arrival and service rates of class *i*, respectively. Class *i* has higher priority than class *j* if *i* < *j*. Let random variable $\tilde{n}_i$ (≥ 0) denote the total number of class-*i* bursts in the system. The system state is represented by *Y*-tuple $(\tilde{n}_1,\ \tilde{n}_2,\ \cdots,\ \tilde{n}_Y)$, where $\sum_{i=1}^{Y}\tilde{n}_i \leq K$. The loss probability for each class, say *i*, denoted as LP_i, can be derived from the limiting system distribution $\Pi = \left\{\pi_{n_1,\cdots,n_Y},\ \sum_{i=1}^{Y} n_i \leq K\right\}$, where $\pi_{n_1,\cdots,n_Y}$ is the joint distribution of the *Y*-tuple.

The limiting distribution is solved based on two sets of balance equations-one corresponds to a system with at least one available server ($\sum_{i=1}^{Y} n_i < K$), and the other one corresponds to a busy system ($\sum_{i=1}^{Y} n_i = K$). Through derivation, they can be given as:

Case I: $\sum_{i=1}^{Y} n_i < K$ **,**

$$\pi_{n_1,\cdots,n_Y}\cdot\sum_{i=1}^{Y}\left(\lambda_i + n_i\cdot\mu_i\right) = \sum_{i=1}^{Y}\left[\lambda_i\cdot\pi_{n_1,\cdots,n_i-1,\cdots,n_Y} + (n_i+1)\mu_i\pi_{n_1,\ldots,n_i+1,\ldots n_Y}\right]; \qquad (17)$$

Case II: $\sum_{i=1}^{Y} n_i = K$ **,**

$$\pi_{n_1,\cdots,n_Y}\cdot\sum_{i=1}^{Y}\left(\lambda_i^* + n_i\cdot\mu_i\right) = \sum_{i=1}^{Y}\left[\lambda_i\cdot\pi_{n_1,\cdots,n_i-1,\cdots,n_Y}\right] + \sum_{i=1}^{Y-1}\left[\lambda_i\cdot\sum_{j=i+1}^{Y}\frac{n_j\cdot\pi_{n_1,\cdots,n_i-1,\cdots,n_j+1,\cdots,n_Y}}{\max\left\{\sum_{l=i+1}^{Y} n_l,\ 1\right\}}\right], \qquad (18)$$

where $\lambda_i^* = \lambda_i$ if $\sum_{l=i+1}^{Y} n_l > 0$, otherwise $\lambda_i^* = 0$.

The left hand sides of Equations (17) and (18) differ in that the non-busy system allows any arrival of any class, whereas a busy system only permits a preemption of a lower-priority burst (if it exists) by a higher-priority burst. Moreover, at the right hand side of Equation (17) and Equation (18), the first term indicates a new burst arrival of class *i*. The second term of the right hand side of Equation (17) indicates a burst departure of class *i*, and the second term of Equation (18) indicates the preemption of class *j* by class *i*, making the size of class-*j* reduced by one and the size of class *i* incremented by 1. The probability of being preempted is proportional to the size of the class. Finally, a burst is lost if either the burst arrives at a busy system and there is no lower-priority burst that can be preempted, or the burst is later preempted by another newly arriving burst with higher priority. Accordingly, weobtain

$$LP_i = \sum_{n_1+\cdots+n_i=K} \pi_{n_1,\cdots,n_i,0,\cdots,0} + \frac{1}{\lambda_i}\cdot\sum_{j=1}^{i-1}\left[\lambda_j\cdot\sum_{n_1+\cdots+n_Y=K}\frac{n_i\cdot\pi_{n_1,\cdots,n_Y}}{\max\left\{\sum_{l=j+1}^{Y} n_l,\ 1\right\}}\right]. \qquad (19)$$

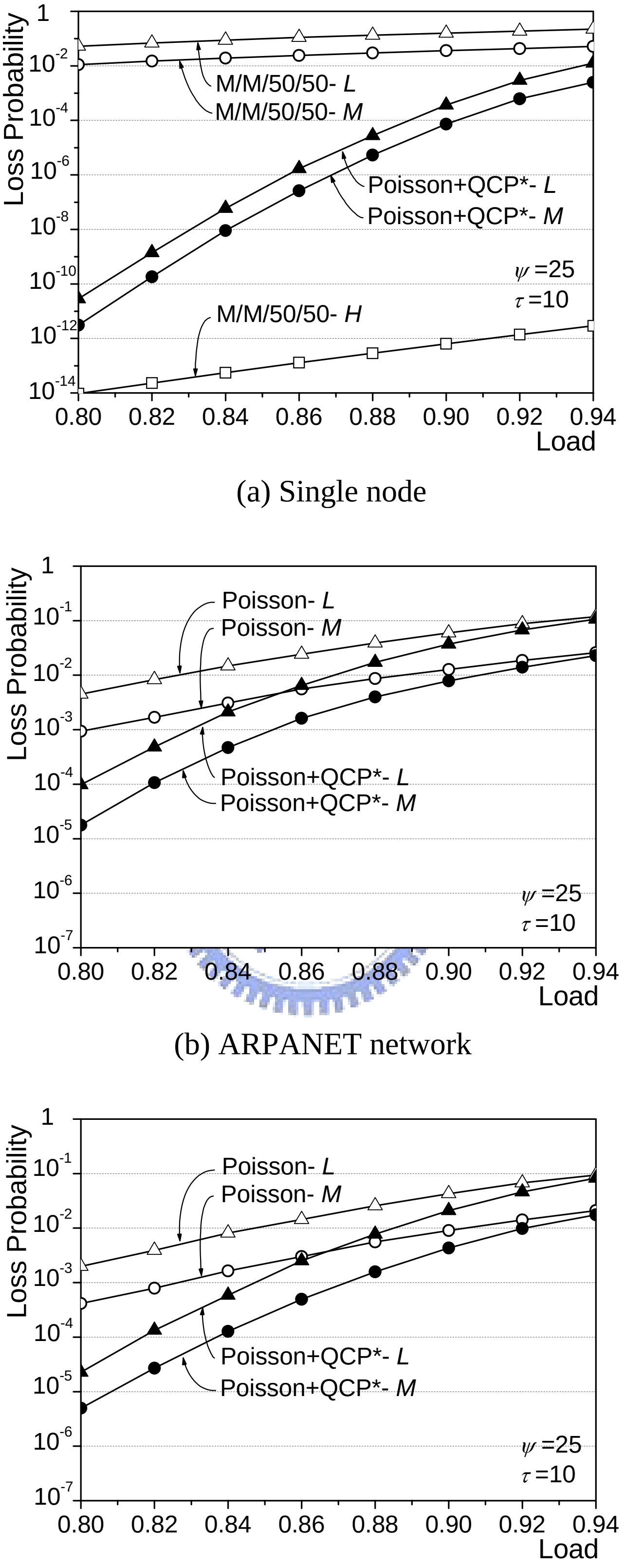
We draw comparisons of loss probability between the *M*/*M*/50/50 and QCP systems supporting three priorities. In the simulation, we computed the loss probability of the ARPANET network and 4x4-torus network. Analytical and simulation results are plotted in Figure 26. Under both cases as shown in Figure 26(a), compared to the *M*/*M*/50/50 system in single node, the QCP system yields superior performance for all three classes, due to traffic shaping. Notice that, due to super low loss probability for the *H* class (lower than 10^{-14} under load 0.94), the plotting is omitted in the figure. Since the burst traffic is associated from different edge nodes, the load of one switch is related by the network topology and routing algorithm. The burst loss performance under simulation results are shown in Figure 26(b) and 26(c). Load 0.8 means 40G traffic from edge node into network. Load of some switch behinds 0.8 but the other is over. It makes more loss under network.

Finally, we examine the performance of QCP with respect to packet loss probability with and without supporting partially collided bursts, under two different ψ values, and three different numbers of wavelengths (W=10, 30, and 90). In the simulation, there are two priorities (H and L) of traffic, both of which are Interrupt Poisson Processes (IPP) distributed with burstiness b=4 and shaped via (ψ,τ)-Shaper. At core switch router, a higher priority burst that finds no wavelength available upon arrival will preempt the lower-priority burst with the least remaining service time. Simulation results of packet loss probability for L-class traffic are displayed in Figures 27 and 28. As was expected, the loss performance is noticeably improved with QCP. Specifically, the loss probability declines by more than two orders of magnitude under loads of 0.9 and below, ψ=100 and W=90.

In the following simulations, other than the parameters described above, we employ three traffic classes. They are Classes H, M, and L, in the order of decreasing loss priorities. Each of these three classes generates an equal amount of MMBP traffic into the network. Notice that, to gain more insights into loss performance for networks with reasonable wavelength-based statistical multiplexing gain, we adopt 50 wavelengths in this simulation. As a result, the packet loss probability for Class H becomes too low to be measured within affordable time periods. Though, it is sufficient to show the packet loss behavior for both Classes M and L. Simulation results are shown in Figures 29-32.

In Figures 29 and 30, we show the packet loss probabilities of both Classes *M* and *L*, as a function of offered load under three different burst sizes of Class *H* (in Figure 29) and Class *M* (in Figure 30). As expected, the packet loss probability drastically increases with the load. Class *M* traffic receives a higher grade of loss performance than Class *L* traffic. Focusing on burst size adjustment, in Figures 31 and 32 we plot the packet loss probabilities of Classes *M* and *L* as a function of the burst size of Class *H* (in Figure 31) and Class *M* (in Figure 32). we discover a win-win phenomenon from the figure that, by increasing the burst size of Class *H*, the packet loss probabilities for both Classes *M* and *L* (and Class *H*) decline noticeably. This is because since Class *H* experiences better loss performance due to the use of a larger burst size (better shaping effect), Class *H* makes less preemption toward Classes *M* and *L* traffic.

As shown in Figure 31, due to the “least-harm” preemption guideline, Class *M* with a larger size (ψ_M=20) becomes less likely to be truncated than Class *L* with a smaller size (ψ_L=5), and thus results in greater reduction in packet loss probability. In contrast, suffering from preemption, Class *L* undergoes invariably poor packet loss probability particularly at high load 0.9. By furthermore increasing the burst size of Class *M*, as shown in Figure 32, we observe more reduction in packet loss probabilities for both Classes *M* and *L*. In this case, Class *L* benefits from being less frequently preempted by Class *M*, and thus experiences more performance improvement than that in the previous case.



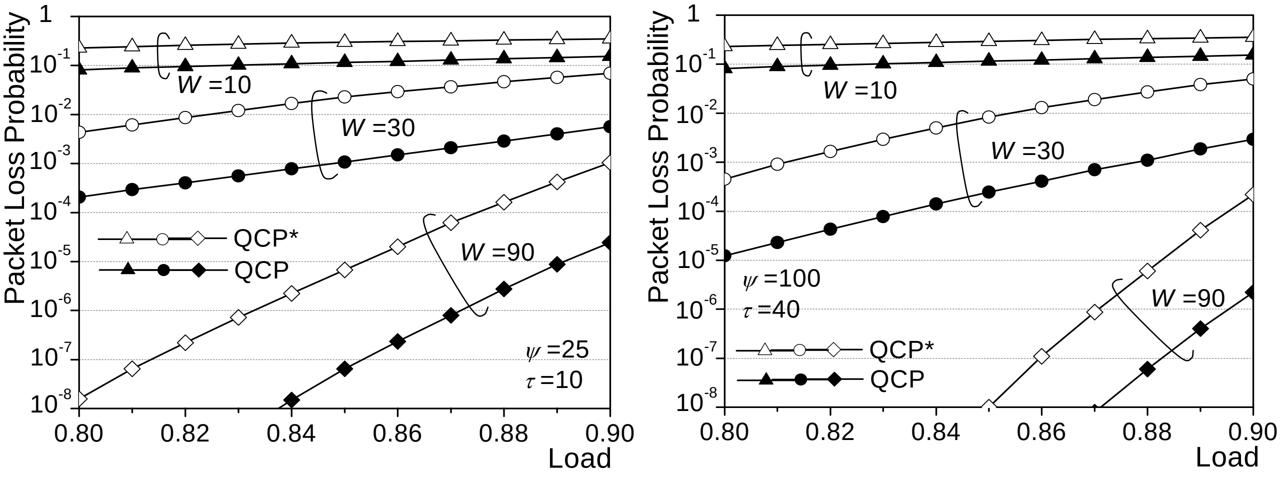
(a) Single node

(b) ARPANET network

(c) 4x4-torus network

QCP*: QCP without support of partially collided bursts.

Figure 26. Performance of QCP (without support of partially collided bursts).

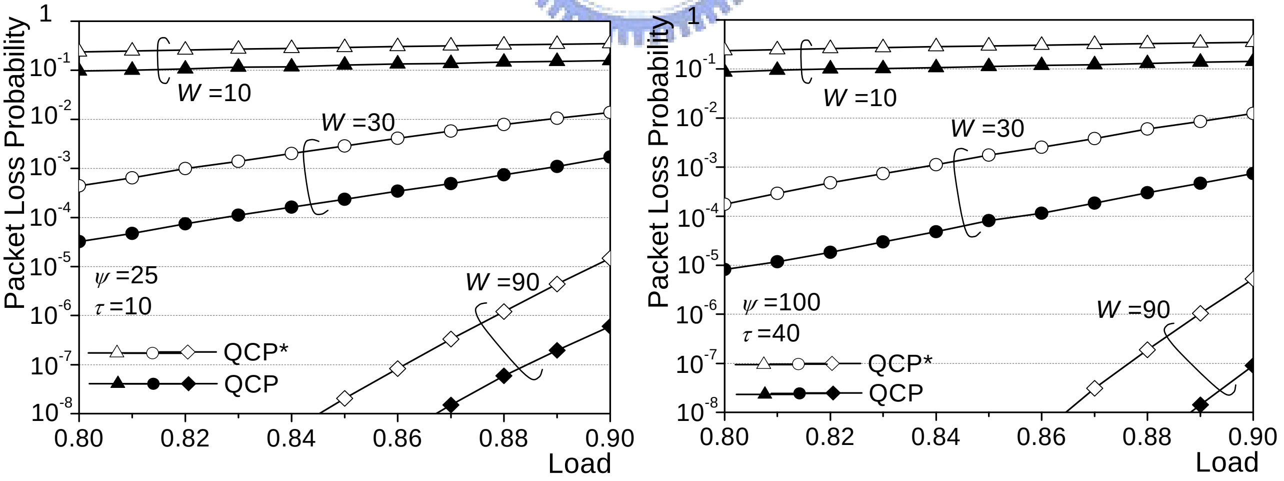


(a) Smaller burst size (ψ=25)

(b) Greater burst size (ψ=100)

QCP*: QCP without support of partially collided bursts.

Figure 27. Performance of QCP under ARPANET network (with and without support of partially collided bursts).

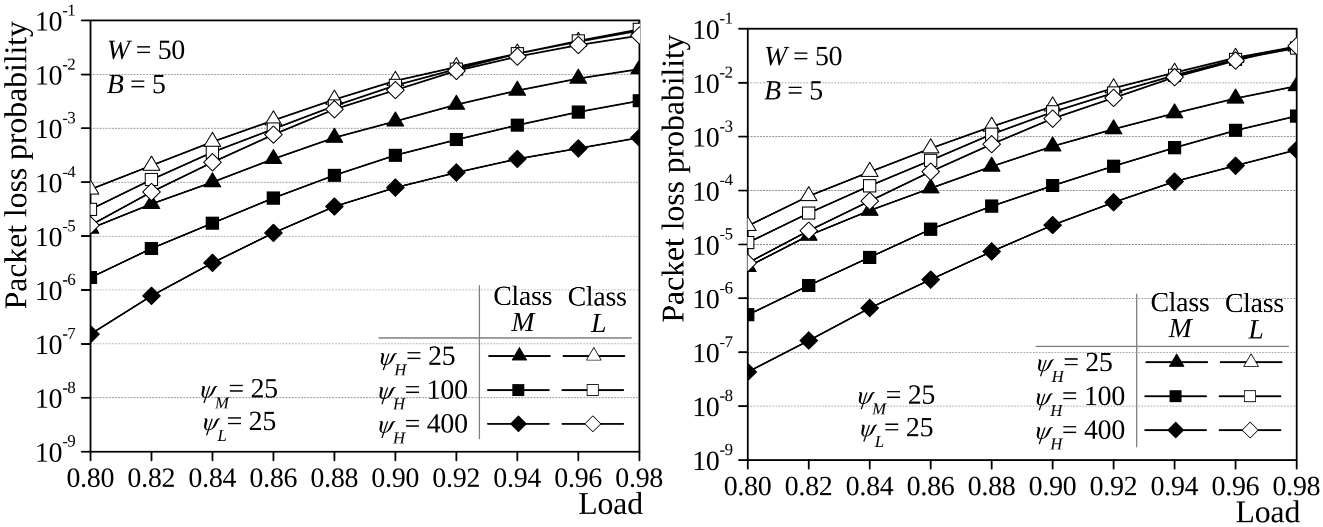


(a) Smaller burst size (ψ=25)

(b) Greater burst size (ψ=100)

QCP*: QCP without support of partially collided bursts.

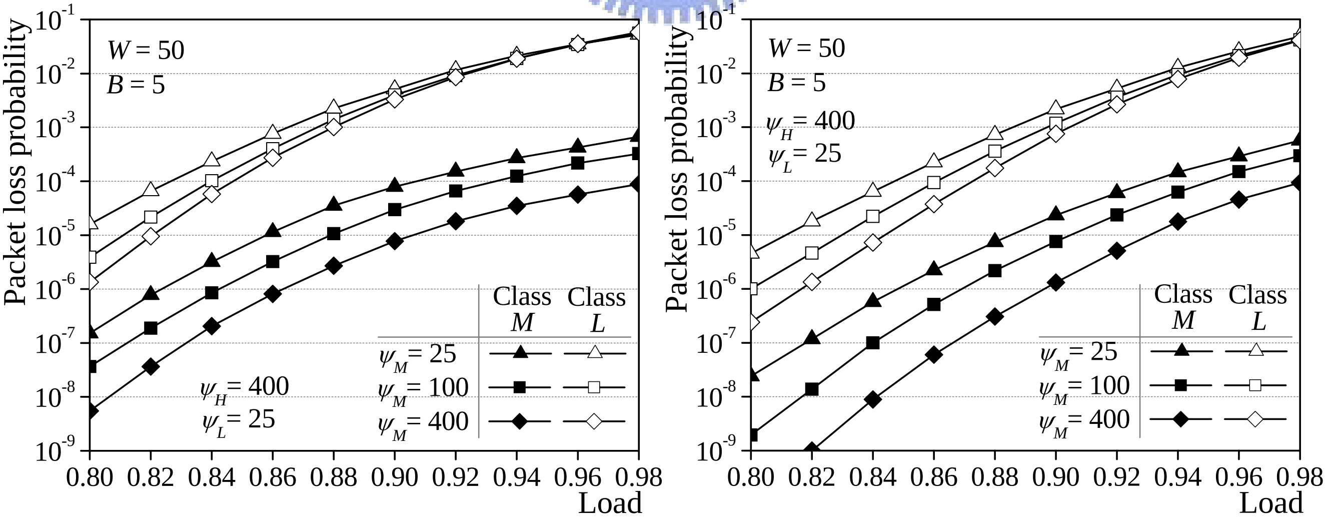
Figure 28. Performance of QCP under 4x4-torus network (with and without support of partially collided bursts).



(a) ARPANET network

(b) 4x4-torus network

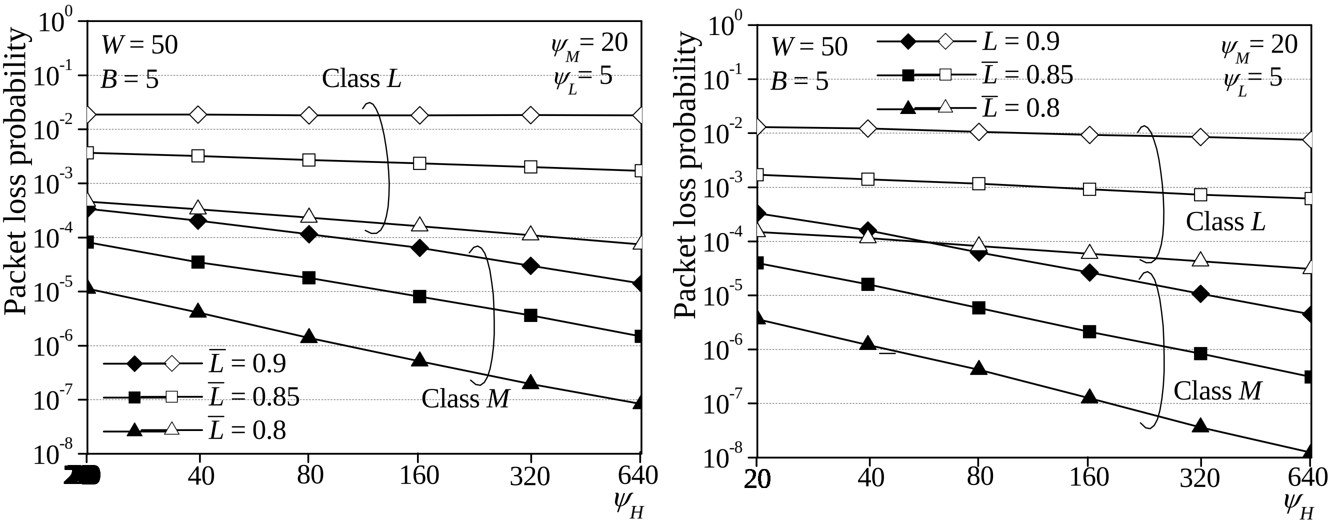
Figure 28. (ψ, τ)-Shaper: loss performance under various burst size of Class *H*.



(a) ARPANET network

(b) 4x4-torus network

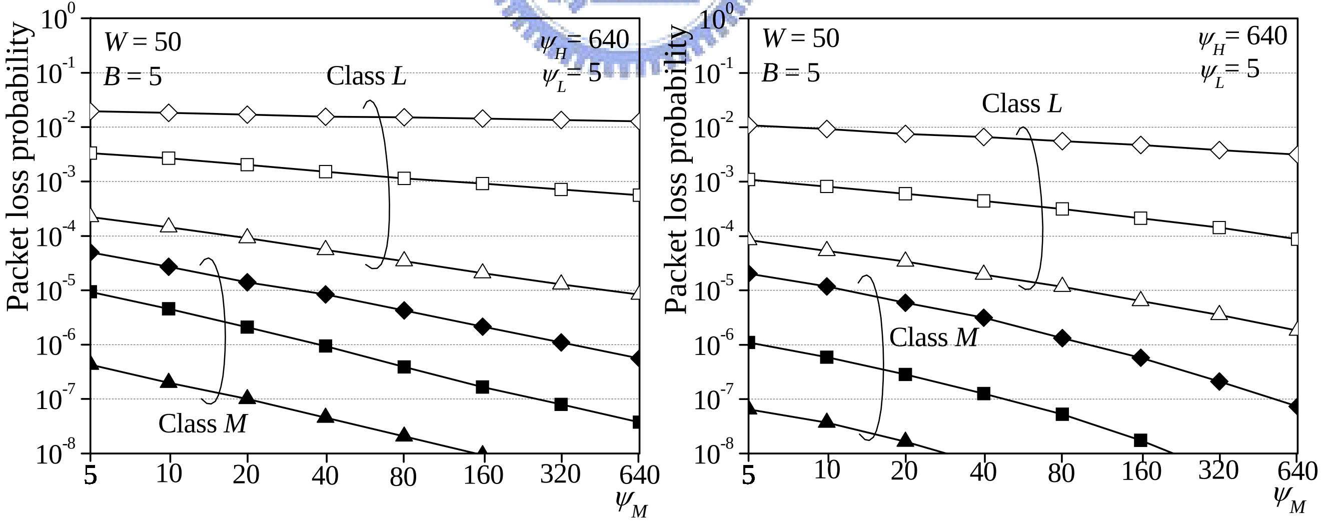
Figure 30. (ψ, τ)-Shaper: loss performance under various burst size of Class *M*.



(a) ARPANET network

(b) 4x4-torus network

Figure 31. (ψ,τ)-Shaper: loss QoS provision via burst size of Class *H* adjustment.



(a) ARPANET network

(b) 4x4-torus network

Figure 32. (ψ,τ)-Shaper: loss QoS provision via burst size of Class *M* adjustment.

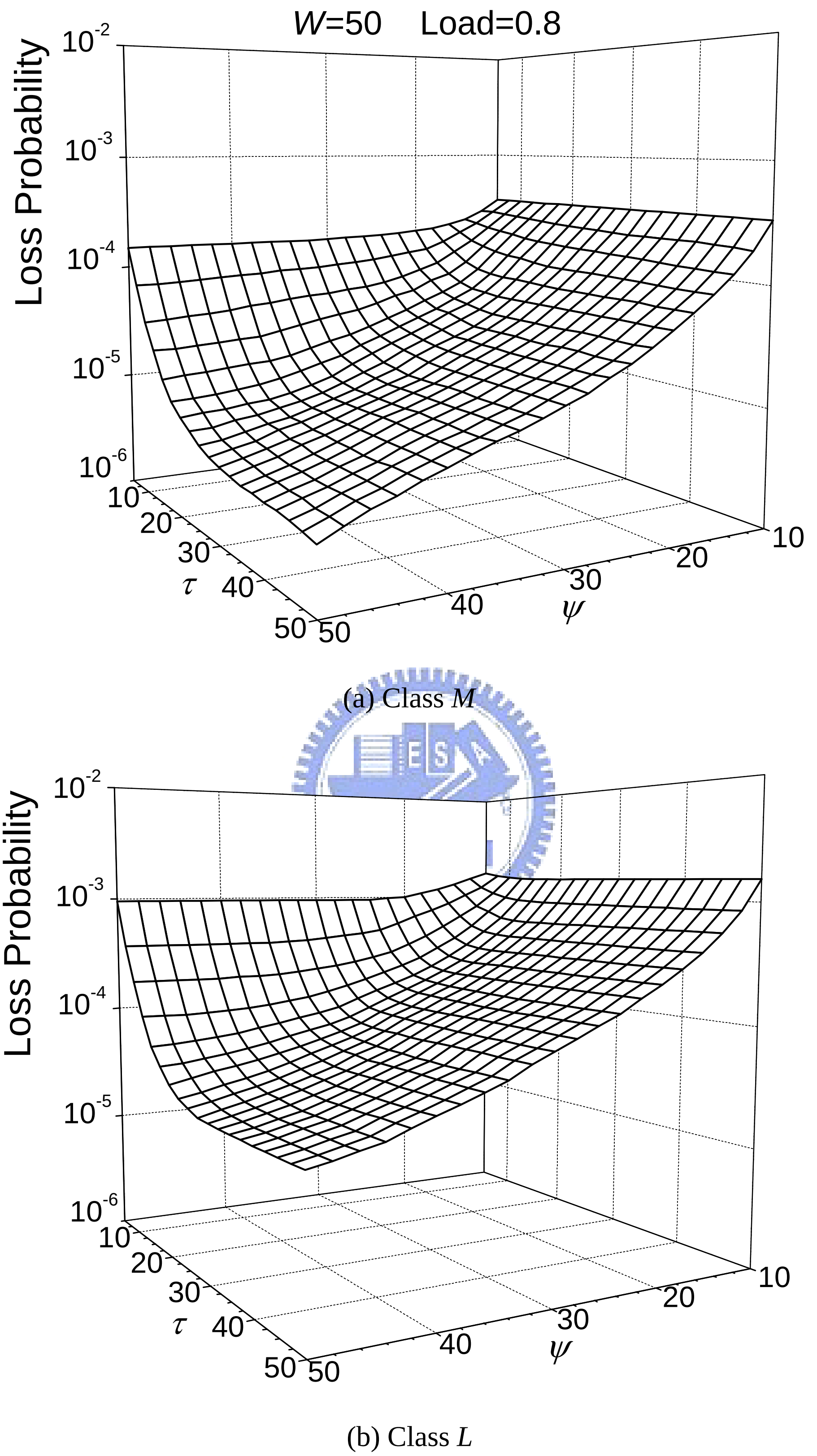
Now we are at the stage to determine two parameters, ψ and τ, for all connections, satisfying both delay and loss QoSs. ψ and τ decide the mean burst size. First of all, we discover from simulation results (see Figures 33 and 34) that under any given load and wavelength number, the loss probability of any class is a function of ψ and τ. Assume there are N loss classes. Thus, all $\{(\psi_1,\tau_1),...,(\psi_N,\tau_N)\}$ satisfying the loss probability of all classes, form the *Loss*-set. Second, all $\{(\psi_1,\tau_1),...,(\psi_N,\tau_N)\}$ simultaneously satisfying the delay bound requirements of all flows within the classes, form another set, called *Delay*-set. Namely,

$$\textit{Delay}\text{-set} = \left\{ \{(\psi_1,\tau_1),...,(\psi_N,\tau_N)\} \middle| \begin{array}{l} \theta_{\min}^{i,j}(\psi_j,\tau_j) \le \text{Delay-QoS}_{i,j}, \\ \forall \text{ loss class } i, \text{ delay class } j \end{array} \right\},$$

where $\theta_{\min}^{i,j}$ is $\theta_{\min}$ given in Theorem 1 for loss class i and delay class j. Then, accordingly to Theorem 1, $\{(\psi_1,\tau_1),...,(\psi_N,\tau_N)\}$ can be selected from the intersection of *Loss*-set and *Delay*-set, such that $\tau+\dfrac{\psi}{R}$ is minimized. Formally, defining $\Psi \equiv \{\{(\psi_1,\tau_1),...,(\psi_N,\tau_N)\} \mid \{(\psi_1,\tau_1),...,(\psi_N,\tau_N)\} \in \textit{Loss}\text{-set} \cap \textit{Delay}\text{-set}\}$, and $f(\Psi) \equiv \left\{\tau_1+\dfrac{\psi_1}{R}+...+\tau_N+\dfrac{\psi_N}{R}\right\}$, the feasible solution can be formally expressed as

$$\Psi^* = \{(\psi_1^*,\tau_1^*),...,(\psi_N^*,\tau_N^*)\} \quad \text{s.t. } f(\Psi^*) = \min_{\Psi} f(\Psi). \tag{20}$$

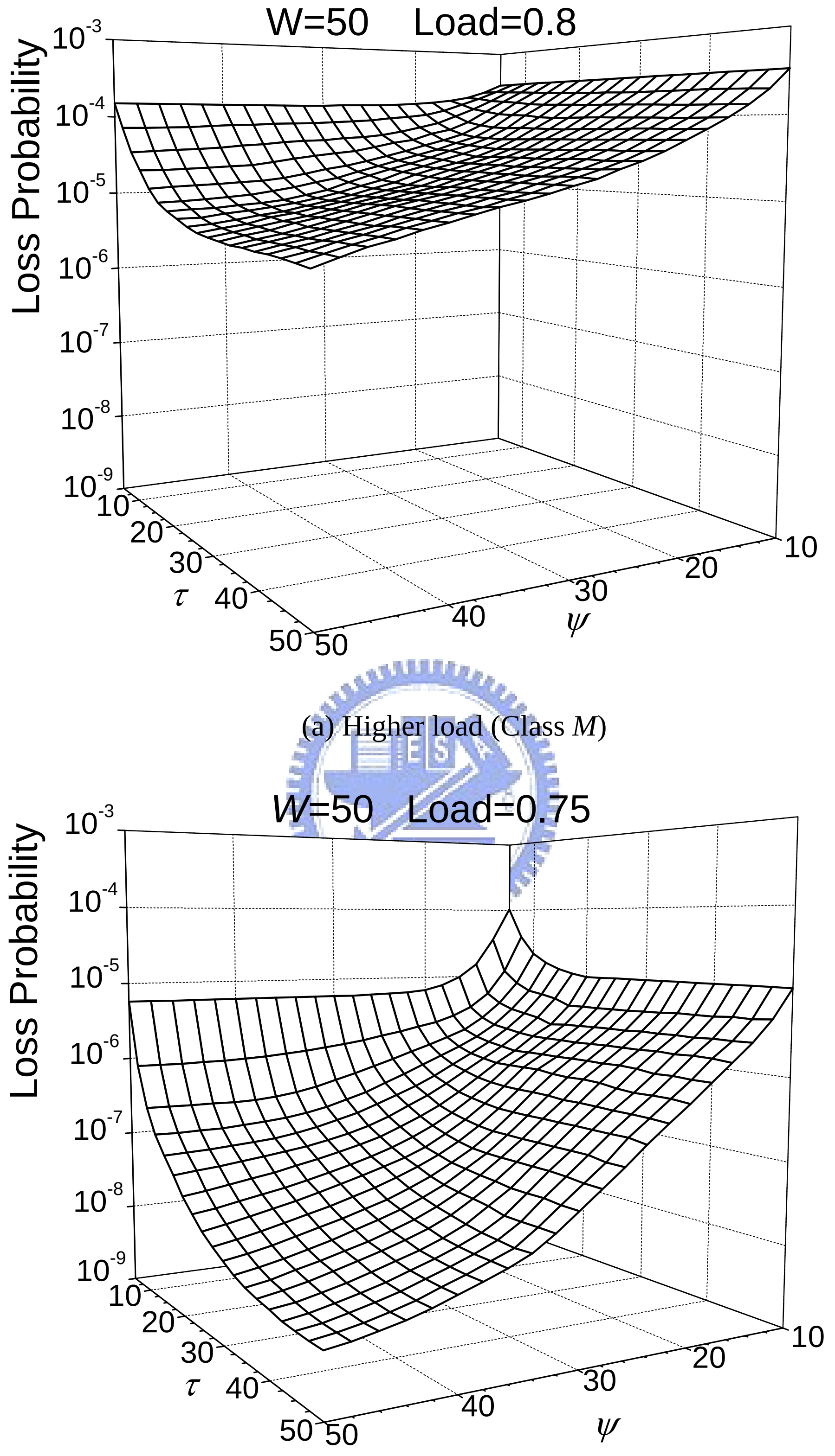
Notice that non-existence of a feasible solution corresponds to the failure of simultaneous satisfaction of delay and loss QoS requirements. Such information can be used to determine the acceptance or rejection of a call during admission control.



(a) Class *M*

(b) Class *L*

Figure 33. Loss probability as a function of (ψ, τ).



(a) Higher load (Class *M*)

(b) Lower load (Class *M*)

Figure 34. Loss probability of Class *M* under different load.

5.3 OCPS and OBS Performance Comparison

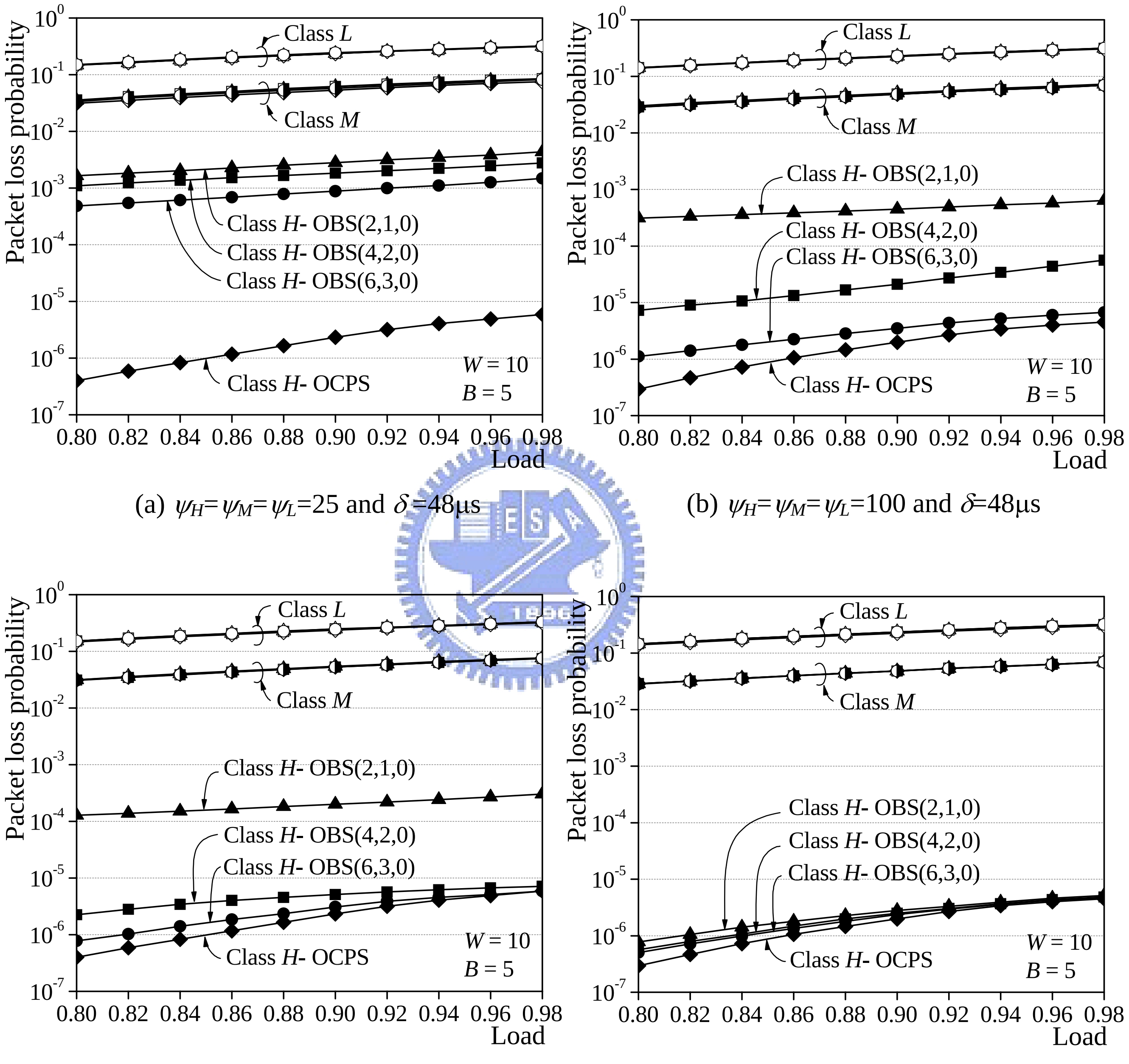
As was mentioned, owing to the near-far problem and header-payload decoupling design, a JET-based OBS network supports *restricted* QoS burst truncation, resulting in loss performance degradation for high priority traffic classes. In this section, we focus on this issue by making a comparison of packet loss probability between the OCPS and JET-based OBS networks. Simulations are carried out on the same 24-node ARPANET network and 4x4-torus network in which three traffic classes (Classes *H*, *M*, and *L*) were adopted. In simulations, each ingress node generates a total of 39 connections (3 classes for each of 13 destination nodes) under ARPANET network and 45 connections (3 classes for each of 15 destination nodes) under 4x4-torus network that follow different load-balancing OLSPs. For ease of comparison, the burst size for all three classes are used the same during burstification, namely $\psi_H = \psi_M = \psi_L$.

For OCPS networks, we conduct QoS burst truncation in switching nodes on priority plus least-harm-preemption bases. For OBS networks, the offset time assigned to a burst is the total control packet processing time (path-dependent) plus the extra delay $x \cdot T$, where *T* is the maximum burst transmission time (e.g., 12μs for $\psi = 25 = 1500$ bytes, and 48μs for $\psi = 100 = 6000$ bytes), and *x* is (6,3,0), (4,2,0), or (2,1,0) for Classes (*H*, *M*, *L*), respectively. Notice that, in the OBS work reported in [9], the burst length is assumed exponentially distributed, and *T* is assigned as the mean burst length. It thus requires a large offset time difference between any two adjacent classes, such as (6,3,0), to meet 95 percent of traffic isolation degree. In these following simulations, we apply the same timer (τ) and threshold (ψ) combined scheme to packet burstification for the OBS network. As a result, with *T* given as the maximum burst transmission time, all three above extra-delay settings, namely (6,3,0), (4,2,0), and (2,1,0), achieves 100 percent of traffic isolation degree. In addition, the header processing time (δ) at each switching node is assumed fixed. Finally, we employ restricted QoS burst truncation during contention for the OBS network. Specifically, truncation of bursts is also accomplished on priority plus least-harm-preemption bases, but restricted to those bursts whose control packets have not yet departed from the switch. Simulations results are displayed in Figures 35 (ARPANET network) and 36 (4x4-torus network).

In Figures 35 and 36, we draw comparisons of packet loss probabilities of all three traffic classes between the OCPS and three variants of OBS networks using three extra-delay settings, respectively, under four cases set by two burst sizes (ψ = 25, 100) and two header processing times (δ = 9.6μs = 20slots, 48μs = 100slots). First, we observe from the figure that the OCPS and OBS networks provide typically the same grade of loss performance for Classes *M* and *L* under all four cases. Significantly, we discover that, compared to OCPS in ARPANET network (as shown in Figures 35(a) and (c)), OBS undergoes several orders of magnitude deterioration in packet loss performance for Class *H* traffic particularly under a smaller burst size, i.e., ψ_H=ψ_M=ψ_L=25. Among the three OBS variants, OBS(2,1,0) using the smallest extra offset time difference (=*T*) invariably suffers from the poorest packet loss probability. Such performance degradation is caused by the near-far problem that exacerbates under a smaller burst size, a larger header processing time, and/or a smaller extra offset time difference. Under any of the conditions, the offset time of a Class-*H* burst is more likely to be smaller than that of a Class-*M* or Class-*L* burst, resulting in failing to make earlier wavelength reservation for the burst. This fact accounts for the poorest performance for Class-*H* taking place under ψ_H=ψ_M=ψ_L=25 and δ=48μs, as shown in Figure 35(a). As the burst size increases and the processing time decreases, as shown in Figures 35(b), (c) and (d), the near-far problem is relaxed, yielding noticeable performance improvement for Class *H* in OBS networks.

In Figures 35 and 36, compared with OCPS and OBS networks under ARPANET network and 4x4-torus network, the near-far problem of OBS is unobvious under 4x4-torus network, except for Figure 36(a). Such performance is caused by the number of hops in the 4x4-torus network less than or equal to 3. For example, set the condition under ψ_H=ψ_M=ψ_L=100 and δ=48μs (see Figure 36(b)), the smallest offset time of Class-*H* under OBS(4,2,0) is $4 \cdot T + 1 \cdot \delta$=240, and the largest offset time of Class-*M* and Calss-*L* are 288 and 192. While the contention takes place, the Class-*H* burst could first reserve the resource under most conditions.

As opposed to OBS, the in-band-controlled-based OCPS networks are shown to provide invariably superior packet loss probability for Class *H* traffic, enabling effective facilitation of loss class differentiation.



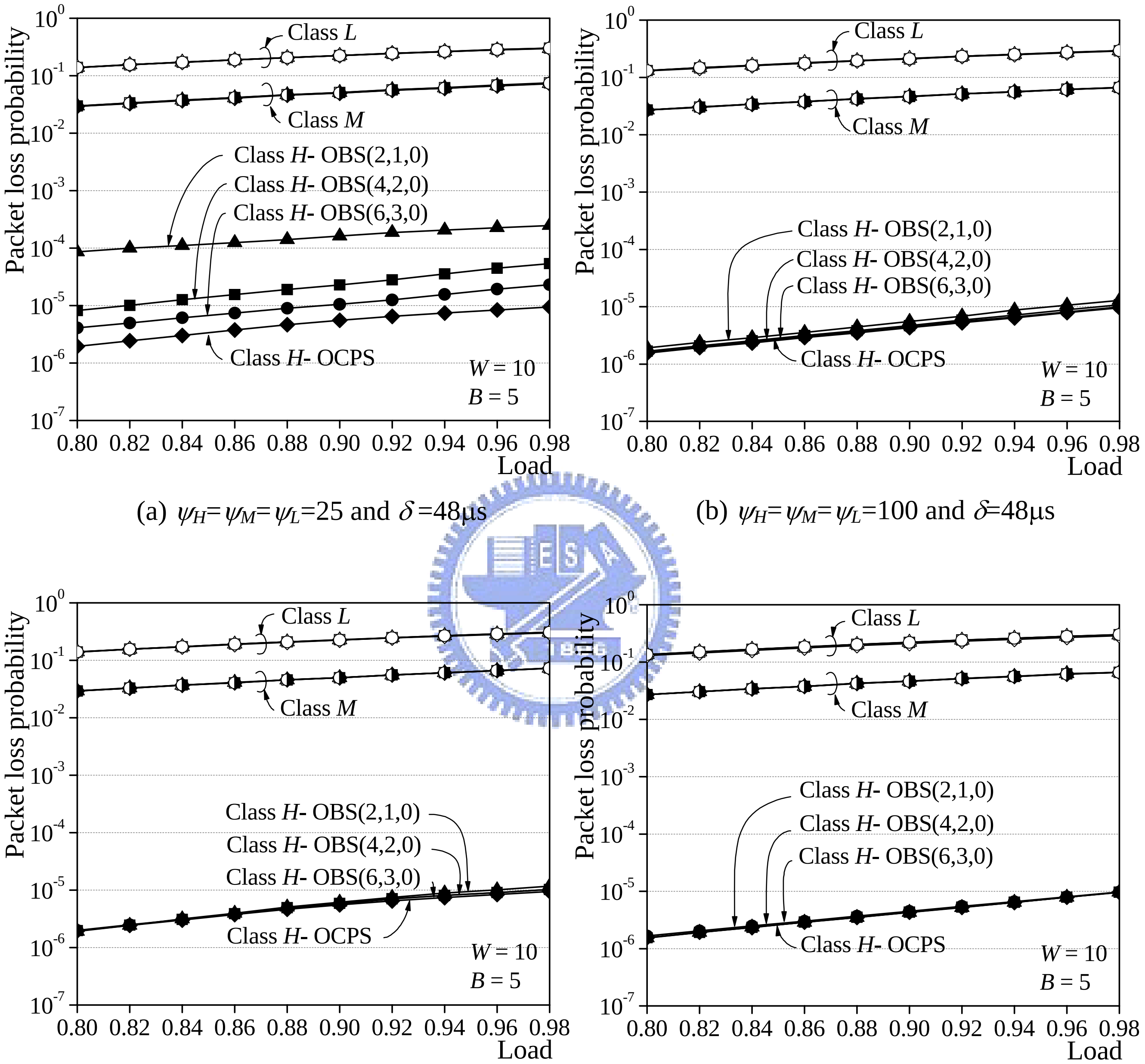
(a) $\psi_H = \psi_M = \psi_L$=25 and δ=48μs

(b) $\psi_H = \psi_M = \psi_L$=100 and δ=48μs

(c) $\psi_H = \psi_M = \psi_L$=25 and δ=9.6μs

(d) $\psi_H = \psi_M = \psi_L$=100 and δ=9.6μs

Figure 35. OCPS and OBS loss performance comparison under ARPANET network.



(a) ψ_H=ψ_M=ψ_L=25 and δ=48μs

(b) ψ_H=ψ_M=ψ_L=100 and δ=48μs

(c) ψ_H=ψ_M=ψ_L=25 and δ=9.6μs

(d) ψ_H=ψ_M=ψ_L=100 and δ=9.6μs

Figure 36. OCPS and OBS loss performance comparison under 4x4-torus network.

Chapter 6. Conclusions and Future Work

6.1 Summary

The thesis provides a dual-purpose, delay and loss QoS-enhanced traffic control scheme, called (ψ,τ)-Scheduler/Shaper, exerted at ingress nodes for OCPS IP-over-WDM networks. Providing delay class differentiation, (ψ,τ)-Scheduler assures each weight-based delay class a worst delay bound derived from the corresponding stepwise service curve; and a stochastic 99% delay bound obtained from simulation results. In addition, (ψ,τ)-Shaper provides loss class differentiation by means of assigning larger burst sizes to higher priority classes. Through a precise departure process analysis of an MMBP/G/1 system, we have delineated that (ψ,τ)-Shaper effectively reduces the CoV of the burst inter-departure time, resulting a substantial reduction in burst loss probability.

We have performed simulations on an ARPANET network and 4x4-torus network to make loss performance comparisons between the OCPS with (ψ,τ)-Shaper and the Just-Enough-Time (JET)-based OBS networks. Simulation results demonstrated that, due to the near-far problem, OBS undergoes several orders of magnitude increase in packet loss probability for Class *H* traffic particularly under a smaller burst size. As opposed to OBS, the in-band-controlled-based OCPS network was shown to provide invariably superior packet loss performance for a high priority traffic class, enabling effective facilitation of loss class differentiation.

Finally, this thesis proposed a mechanism to decide the ψ value and τ value. Based on the delay QoS requirement, the worst delay bound and 99% delay bound of all delay classes will limit the maximum ψ and τ value. According to the loss QoS requirement, ψ and τ value must limit by minimum values. ψ and τ can be determined by these constraints.

6.2 Future work

This thesis emphasis assumes the optical switches inside the network are buffer-less and all wavelengths are shared using wavelength converters. However, the ability of the tunable laser source (TLS) will limit the tuning distance, i.e., TLS only converts the old wavelength to the nearby wavelength. We can not take the advantage in the fully wavelength sharing. In the future work, to achieve the loss QoS requirement, we could consider to use a small buffer (fiber delay line) to replace the large tunable distance demand.

In addition, this thesis assumes all the packets are fixed size. It reduces the complexity during burstification. However, the IP packets inside the network are variable size. In the future work, we could consider to assembly the burst by packets of variable size, and use stepwise service curve to show the worst delay bounds for each delay class.

Appendix

A

Proof of Lemma 1: First, we denote stepwise function $\delta\left(t,\theta^{*}_{\min}\right)$ as δ'. Given time t between interval $\left[T_k^{\delta'}, T_{k+1}^{\delta'}\right)$, by Definition 1 and the definition of T_k^{Π}, we get the first inequality:

$$\delta\left(t,\theta^{*}_{\min}\right) = k \cdot G \le \Pi\left(T_k^{\Pi}\right).$$

Since $\Pi(t)$ is monotonically increasing and $T_k^{\Pi} \le T_k^{\delta'} \le t$, we have the second inequality:

$$\Pi\left(T_k^{\Pi}\right) \le \Pi(t).$$

Combining the two inequalities, we obtain

$$\delta\left(t,\theta^{*}_{\min}\right) \le \Pi(t).$$

According to Definition 2, since there exists only one minimum θ, namely $\theta_{\min}$, $\theta^{*}_{\min}$ is thus lower bounded by $\theta_{\min}$, namely

$$\theta^{*}_{\min} \ge \theta_{\min}.$$

Moreover, Equation (2) leads to a fact that inequality $T_k^{\Pi} \le T_k^{\Delta}$ holds at $\theta = \theta_{\min}$ for all $k \ge 0$. From the definition of $\theta^{*}_{\min}$ in the lemma, which indicates that $\theta^{*}_{\min}$ is the minimum θ making $T_k^{\Pi} \le T_k^{\delta}$ satisfied for all $k \ge 0$, for all stepwise functions including $\Delta(t)$, we imply that $\theta^{*}_{\min}$ is upper bounded by $\theta_{\min}$, namely

$$\theta^{*}_{\min} \le \theta_{\min} .$$

Accordingly, the lemma is proved. ■

B

Proof of Theorem 1: With the focus placed on an observed busy period of class *i*, let P_i^1 be the first packet initiating the busy period, and P_i^j represent the j_{th} ($j \ge 1$) packet of the observed busy period. Let $d_1(t)$ denote the index of the window being served at time *t* from the beginning of the busy period, and $d_2\left(P_i^j\right)$ denote the index of the window in which P_i^j is placed. We immediately have the boundary condition, $d_1(0) = d_2\left(P_i^1\right) = 1$. According to the virtual-window service policy of (ψ,τ)-Scheduler, we get the following inequality:

$$d_2\left(P_i^j\right) - d_2\left(P_i^{j-1}\right) \le \left\lceil \frac{1}{w_i} \right\rceil \le 1,\ \forall j > 1. \tag{21}$$

Suppose after packet P_i^j has been served, the total service amount has first exceeded $k \cdot w_i$. We get $d_2\left(P_i^j\right) = d_1\left(T_k^{\Pi}\right)$ and $j-1 < k \cdot w_i \le j$. Since $k \cdot w_i$ is greater than $j-1$, packet P_i^{j-1} must have been served no later than the k_{th} window. In other words, one gets

$$d_2\left(P_i^{j-1}\right) \le k . \tag{22}$$

By summing Equations (21) and (22), we arrive at

$$d_1\left(T_k^{\Pi}\right) \le k + 1 . \tag{23}$$

Equation (23) can be described in words as that, in order to finish $k \cdot w_i$ service amount, the total number of windows elapsed $d_1\left(T_k^{\Pi}\right)$ is bounded by k+1.

Moreover, due to the fact that the normalized weight of any class can be a non-integer value, the actual number of packets in a virtual window can be less, equal to, or greater than the window size, ψ. Under the worst case, the maximum offered service in a total of k+1 windows can be easily computed as $(k+1)\cdot\psi + N$. In other words, with the maximum offered service divided by ψ, we reach that P_i^j will be placed at worst in the $\left(k+1+\left\lceil \frac{N}{\psi} \right\rceil\right)_{\text{th}}$ burst. Considering the worst case, each burst is generated when the BATr expires. The maximum delay from the beginning of the busy period to the time service amount $k \cdot w_i$ has been offered is bounded as

$$T_k^{\Pi} \le \left(k+1+\left\lceil \frac{N}{\psi} \right\rceil\right)\cdot\left(\tau+\frac{\psi}{R}\right). \tag{24}$$

By assigning the least upper bound of T_k^{Π} to T_k^{Δ}, we have

$$T_k^{\Pi} \le \left(k+1+\left\lceil \frac{N}{\psi} \right\rceil\right)\cdot\left(\tau+\frac{\psi}{R}\right) \equiv T_k^{\Delta}.$$

Subtracting T_k^{Δ} by $k \cdot I$, where $I = \tau + \frac{\psi}{R}$, we obtain $\theta_{\min}^{*} = \left(1+\left\lceil \frac{N}{\psi} \right\rceil\right)\cdot\left(\tau+\frac{\psi}{R}\right)$. By Lemma 1, the theorem is proved. ■

References

[1] B. Mukherjee, “WDM Optical Communication Networks: Progress and Challenges,” *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 10, Oct. 2000, pp. 1810-1824.

[2] T. El-Bawab, and J. Shin, “Optical Packet Switching in Core Networks: Between Vision and Reality,” *IEEE Communications Magazine*, vol. 40, no. 9, Sep. 2002, pp. 60-65.

[3] L. Xu, H. Perros, and G. Rouskas, “Techniques for Optical Packet Switching and Optical Burst Switching,” *IEEE Communications Magazine*, vol. 39, no. 1, Jan. 2001, pp. 136-142.

[4] F. Callegati, G. Corazza, and C. Raffaelli, “Exploitation of DWDM for Optical Packet Switching with Quality of Service Guarantees,” *IEEE Journal on Selected Areas in Communication*, vol. 20, no. 1, Jan. 2002, pp. 190-201.

[5] L. Xu, H. Perros, and G. Rouskas, “The Perspective of Optical Packet Switching in IP Dominant Backbone and Metropolitan Networks,” *IEEE Communications Magazine*, vol. 39, no. 3, March 2001, pp. 136-141.

[6] H. Dorren, M. Hill, Y Liu, N. Calabretta, A. srivatsa, F. Huijskens, H. Waardt, and G. Khoe, “Optical Packet Switching and Buffering By Using All-Optical Signal Processing Methods,” *IEEE Journal of Lightwave Technology*, vol. 21, no. 1, Jan. 2003, pp. 2-12.

[7] D. Hunter, W. cornwell, T. Gilfedder, A. Franzen, and I. Andonovic, “SLOB: A switch With Large Optical Buffers for Packet Switching,” *IEEE Journal of Lightwave Technology*, vol. 16, no. 10, Oct. 1998, pp. 1725-1736.

[8] T. Battestilli, and H. Perros, “An Introduction to Optical Burst Switching,” *IEEE Communications Magazine*, vol. 41, no. 8, Aug. 2003, pp. S10-S15.

[9] M. Yoo, C. Qiao, and S. Dixit, “Optical Burst Switching for Service Differentiation in the Next Generation Optical Internet,” *IEEE Communications Magazine*, vol. 39, no. 2, Feb. 2001, pp. 98-104.

[10] V. Vokkarane, and J. Jue, “Prioritized Burst Segmentation and Composite Burst-Assembly Techniques for QoS Support in Optical Burst-Switched Networks,” *IEEE Journal on Selected Areas in Communication*, vol. 21, no. 7, Sep. 2003, pp. 1198-1209.

[11] J. Wei, and R. McFarland, “Just-In-Time Signaling for WDM Optical Burst Switching Networks,” *IEEE Journal of Lightwave Technology*, vol. 18, no. 12, Dec. 2000, pp. 2019-2037..

[12] C. Qiao, “Labeled Optical Burst Switching for IP-over-WDM Integration,” *IEEE Communications Magazine*, vol. 38, no. 9, Sep. 2000, pp. 104-114.

[13] Y. Xiong, M. Vandenhoute, and H. Cankaya, “Control Architecture in Optical Burst-Switched WDM Networks,” *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 10, Oct. 2000, pp. 1838-1851.

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

[15] J. White, M. Zukerman, and H. Vu, “A Framework for Optical Burst Switching Network Design,” *IEEE Communications Letters*, vol. 6, no. 6, June 2002, pp. 268-270.

[16] N. Barakat, and E. H. Sargent, “Dual-header Optical Burst Switching: a New Architecture for WDM Burst-Switched Networks,” in *Proc. IEEE INFOCOM*, 2005, pp. 685-693.

[17] Z. Rosberg, Hai Le Vu, and M. Zukerman, J. White, “Performance Analyses of Optical Burst-Sswitching Networks,” *IEEE Journal on Selected Areas in Communication*, vol. 21, no. 7, Sep. 2003, pp. 1187-1197.

[18] A. Ge, F. Callegati, and L. Tamil, “On Optical Burst Switching and Self-Similar Traffic,” *IEEE Communications Letters*, vol. 4, no. 3, March 2000, pp. 98-100.

[19] M. Yuang, J. Shih, and P. Tien, “QoS Burstification for Optical Burst Switched WDM Networks,” in *Proc. IEEE OFC*, 2002, pp. 781-783.

[20] L. Yang, Y. Jiang, and S. Jiang, “A Probabilistic Preemptive Scheme for Providing Service Differentiation in OBS Networks,” in *Proc. IEEE GLOBECOM*, 2003.

[21] A. Detti, V. Eramo, and M. Listanti, “Performance Evaluation of a New Technique for IP Support in a WDM Optical Network: Optical Composite Burst Switching (OCBS),” *IEEE Journal of Lightwave Technology*, vol. 20, no. 2, Feb. 2002, pp. 154-165.

[22] M. Yuang, P. Tien, J. Shih, S. Lee, Y. Lin, F. Tsai, and A. Chen, “Optical Coarse Packet-Switched IP-over-WDM Network (OPSINET): Technologies and Experiments,” *IEEE Journal on Selected Areas in Communication*, vol. 24, no. 8, Aug. 2006.

[23] M. Yuang, P. Tien, J. Shih, and A. Chen, “QoS Scheduler/Shaper for Optical Coarse Packet Switching IP-over-WDM Networks,” *IEEE Journal on Selected Areas in Communication*, vol. 22, no. 9, Nov. 2004.

[24] M. Yuang, J. Shih, and P. Tien, “Traffic Shaping for IP-over-WDM Networks based on Optical Coarse Packet Switching Paradigm,” in *Proc. IEEE ECOC*, 2003.

[25] Y. Lin, M. Yuang, S. Lee, and W. Way, “Using Superimposed ASK Label in a 10 Gbps Multi-Hop All-Optical Label Swapping System,” *IEEE Journal of Lightwave Technology*, vol. 22, no. 2, Feb. 2004.

[26] S. Danielsen, P. Hansen, and K. Stubkjaer, “Wavelength Conversion in Optical Packet Switching,” *IEEE Journal of Lightwave Technology*, vol. 16, no. 12 , Dec. 1998, pp. 2095-2108.

[27] A. Parekh, and R. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: the single-node case,” *IEEE/ACM Transcations on Networking*, vol. 1, no. 3, June 1993, pp. 344-357.

[28] J. Bennett, and H. Zhang, “WF^{2}Q: Worst-case Fair Weighted Fair Queueing,” in *Proc. IEEE INFOCOM*, 1996, pp. 120-128.

[29] D. Stiliadis, and A. Varma, “Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms,” *IEEE/ACM Transcations on Networking*, vol. 6, no. 5, Oct. 1998, pp. 611-624.

[30] H. Harai, M. Murata, and H. Miyahara, “Performance analysis of wavelength assignment policies in all-optical networks with limited-range wavelength conversion,” *IEEE Journal on Selected Areas in Communication*, vol. 16, no. 7, Sep. 1998, pp. 1051-1060.

Biography

Julin Shih was born in Taipei, Taiwan, in 1976. He received the B.S. degree in management and information system from the National Central University, Chung-li, Taiwan, in 1999 and he received the M.S. degree in 2001 and currently a Ph.D. candidate, in computer science and information engineering from the National Chiao Tung University, Hsin-chu, Taiwan. His currently research interests include high speed networking, optical networking, and performance modeling and analysis.