QoS Scheduler/Shaper for Optical Coarse Packet Switching IP-Over-WDM Networks

Maria C. Yuang, Po-Lung Tien, Julin Shih, and Alice Chen

***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 paper, 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 interdeparture time. Furthermore, we conduct extensive simulations on a 24-node ARPANET 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.**

***Index Terms*—Departure process, IP-over-WDM networks, Markov modulated Bernoulli process (MMBP), optical burst switching (OBS), optical packet switching (OPS), quality-of-service (QoS), traffic scheduling, traffic shaping.**

I. 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]–[14] has received the 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 [15], packet-count threshold [10], and a combination of both [10], [13], [16]. 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 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 JIT 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 intranodal 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

Manuscript received August 11, 2003; revised February 3, 2004. This work was supported in part by the Phase-II Program for Promoting Academic Excellence of Universities, Taiwan, under Contract NSC93-2752-E009-004-PAE, in part by the NCTU/CCL Joint Research Center, and in part by the National Science Council (NSC), Taiwan, under Grant NSC93-2213-E-009-049.

M. C. Yuang, P.-L. Tien, and J. Shih are with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

A. Chen is with the Optical Communications and Networking Technologies Department, Computer and Communications Research Labs, Industrial Technology Research Institute (ITRI), Hsinchu 310, Taiwan, R.O.C.

Digital Object Identifier 10.1109/JSAC.2004.833837

[17], 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 [18] in switching nodes. The prioritized burst segmentation approach proposed in [10], which is 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 [19]. 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. We refer to this type of operation 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 [20]. 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 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 our newly designed superimposed amplitude shift keying (SASK) technique [21]. Besides, they are time-aligned during modulation via necessary padding added to the header. They are realigned 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 preestablished optical label switched path (OLSP). At each switching node, the header and payload are first SASK-based demodulated [21]. 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 the paper 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 our work, we assume optical switches are buffer-less and all wavelengths are shared using wavelength converters [3], [22]. 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 [23]–[25] 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 [17], in this work we focus on the design of traffic shaping with QoS provisioning.

In this paper, 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 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 [23]. We also demonstrate the mean delay and 99% delay bound for each delay class 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

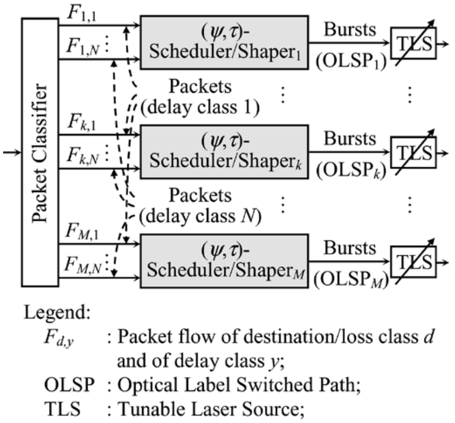

Packet Classifier
$F_{1,1}$
$F_{1,N}$
(ψ,τ)-Scheduler/Shaper$_1$
Bursts
(OLSP$_1$)
TLS
Packets (delay class 1)
$F_{k,1}$
$F_{k,N}$
(ψ,τ)-Scheduler/Shaper$_k$
Bursts
(OLSP$_k$)
TLS
Packets (delay class N)
$F_{M,1}$
$F_{M,N}$
(ψ,τ)-Scheduler/Shaper$_M$
Bursts
(OLSP$_M$)
TLS
Legend:
$F_{d,y}$: Packet flow of destination/loss class d and of delay class y;
OLSP : Optical Label Switched Path;
TLS : Tunable Laser Source;

Fig. 1. (ψ,τ)-Scheduler/Shaper system architecture.

yields substantial reduction in the coefficient of variation (CoV) of the burst interdeparture time. The greater the burst size, the more reduction in the CoV. Furthermore, we conduct extensive simulations on a 24-node ARPANET 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 paper is organized as follows. In Section II, we introduce the (ψ,τ)-Scheduler/Shaper system architecture. In Section III, we describe the (ψ,τ)-Scheduler design, the stepwise service curve, and show the worst and 99% delay bounds for each delay class. In Section IV, we present a precise departure process analysis for (ψ,τ)-Shaper to analytically delineate the shaping effect on departing traffic characteristics. In Section V, 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 Section VI.

II. (ψ,τ)-Scheduler/Shaper System Architecture

In any ingress node, incoming packets (see Fig. 1) are first classified on the basis of their destination, loss, and delay classes. Packets belonging to the same destination and loss class are assembled into a burst. Thus, a burst may contain packets belonging to different 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 preassigned delay-associated weights. Departing bursts from any (ψ,τ)-Scheduler/Shaper are optically transmitted, and forwarded via their corresponding, preestablished OLSP.

Essentially, (ψ,τ)-Scheduler/Shaper is a dual-purpose scheme. It is a scheduler for packets, abbreviated as $(\boldsymbol{\psi},\boldsymbol{\tau})$-**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 $(\boldsymbol{\psi},\boldsymbol{\tau})$-**Shaper**, which determines the sizes and departure times of bursts. They are discussed in Sections III and IV, respectively.

III. (ψ,τ)-Scheduler and Delay QoS

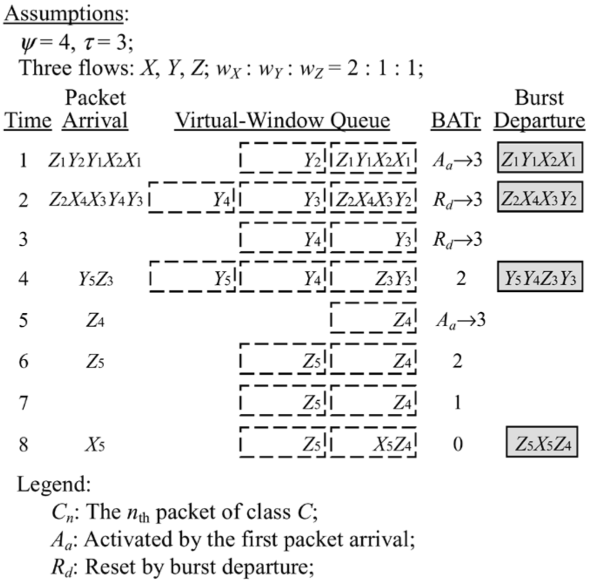
In the (ψ,τ)-Scheduler system, each delay class is associated with a predetermined weight [23]. A higher delay priority class is given a greater weight, which corresponds to a more stringent delay bound requirement. In addition, we assume all packets are of fixed size of one unit. Generally, (ψ,τ)-Scheduler 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.

A. 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 Fig. 2. 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 Fig. 3. 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



Assumptions:

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

Three flows: 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 | $Z_1Y_1X_2X_1$	$A_a \rightarrow 3$	$Z_1Y_1X_2X_1$
2	$Z_2X_4X_3Y_4Y_3$	Y_4 | Y_3 | $Z_2X_4X_3Y_2$	$R_d \rightarrow 3$	$Z_2X_4X_3Y_2$
3		Y_4 | Y_3	$R_d \rightarrow 3$	
4	Y_5Z_3	Y_5 | Y_4 | Z_3Y_3	2	$Y_5Y_4Z_3Y_3$
5	Z_4	Z_4	$A_a \rightarrow 3$	
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:

C_n: The n_{th} packet of class C;

A_a: Activated by the first packet arrival;

R_d: Reset by burst departure;

Fig. 2. (ψ, τ)-Scheduler: an example.

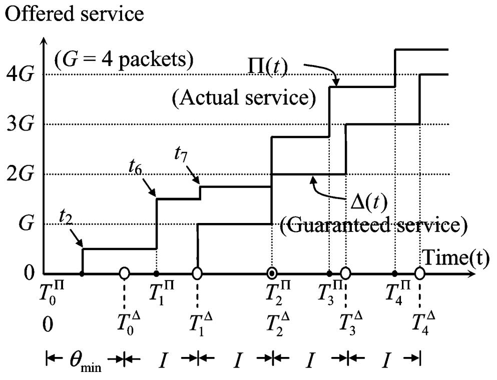
```
Variable
  $w_i$ : normalized weight of class $i$ ($\Sigma w_i = \psi$);
  $cw$ : index of currently served window;
  $lw_i$ : index of window containing the last class $i$'s
        packet;
  $q_i$ : net quota for class $i$;
  $P_i$ : newly arriving packet from class $i$;
  $Bu$ : the generated burst;
  BATr: burst assembly timer;
Initialization()   /* idle to busy */
  1. $cw \leftarrow 1$;
  2. for (each class $i$) do
  3.     $lw_i \leftarrow 1$;   $q_i \leftarrow w_i$;   endfor
  4. BATr $\leftarrow \tau$;
Arrival($P_i$) /*a newly arriving packet from class $i$*/
     Determine the window $P_i$ can be placed;
  1. if ($lw_i < cw$)   $lw_i \leftarrow cw$;   $q_i \leftarrow w_i$;   endif
  2. while ($q_i < 1$) do
  3.     $lw_i \leftarrow lw_i + 1$;   $q_i \leftarrow q_i + w_i$;   endwhile
     Place packet in window $lw_i$ and update quota;
  4. 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;
  1. Dequeue($Bu$);
     Update information;
  2. $cw \leftarrow$ index of the next window with packets;
  3. if (queue is not empty)   BATr $\leftarrow \tau$;   endif
```

Fig. 3. (ψ, τ)-Scheduler: the algorithm.

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 nonempty, the BATr is reset to the τ value. It is worth noting that the algorithm works under noninteger normalized weights which are practically the case in real systems.

B. Worst Delay Bound Guarantee-Stepwise Service Curve

The service curve specification [23], [25] 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 nonwork-conserving server, in which packets do not depart from the system before the burst is generated. Our objective is to characterize the stepwise nature of the service curve for the nonwork-conserving system, (ψ, τ)-Scheduler.



Legend:

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_x: Time when x-packets are actually served;

Fig. 4. Concept of stepwise service curve.

In the sequel, we first define the stepwise function and introduce the stepwise service curve guaranteed by a general server, S. We then specify the stepwise service curve guaranteed for a delay class by (ψ, τ)-Scheduler in Theorem 1. We finally provide the worst delay bound in two different forms based on the theorem. Throughout this section, we assume that there are N 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

$$\delta(t, \theta) = \begin{cases} kG, & T_k^{\delta} \leq t < T_{k+1}^{\delta}, \quad \text{and} \quad k \geq 0 \\ 0, & 0 \leq t < T_0^{\delta}, \end{cases} \tag{1}$$

where T_k^{δ} is the kth 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 quasiconstant-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 Fig. 4, 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 Fig. 4, a G amount of service corresponds to the finishing transmission of four packets. Due to batch service, server S actually finishes a two-packet transmission at t_2, and a total of six-packet transmission at t_6. Thus, T_1^{Π} is equal to t_6 which is the earliest time upon which $1G$ (four-packet) service has been received.

The problem of seeking guaranteed service becomes the determination of a stepwise function which is the greatest lower bound of all possible scenarios of $\Pi(t)$. We call such function the *stepwise service curve*, $\Delta(t)$ 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

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

where $\mathrm{E} = \{\theta\colon \delta(t,\theta) \leq \Pi(t), \forall t \geq 0\}$. The supremum of (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. Our 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

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

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

The Proof of Lemma 1 is in Appendix A. To find the stepwise service curve for class i, we are to determine three parameters, G, I, and $\theta_{\min}$. 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, we 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 $G = w_i, I = \tau + \psi/R$, and $\theta_{\min} = (1 + \lceil N/\psi \rceil) \cdot (\tau + \psi/R)$.

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 [23] provided an absolute delay bound, subject to the constraint that arriving packets are leaky-bucket regulated. In our work, due to the lack of traffic regulation, a time-independent delay bound is unachievable. In lieu, we provide the worst delay bound for each class 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 Fig. 5, 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 (class 2), the worst delay bound until packet P^4 is served is $\theta_{\min} + 2I - t_4(\theta_{\min} + 4I - t_4)$. Accordingly, for the jth packet P^j_i of class i arriving at time t_j from the beginning of a busy period, the worst delay bound is $\theta_{\min} + \lceil j/w_i \rceil \cdot I - t_j$, where $\theta_{\min}$ and I are given in Theorem 1.

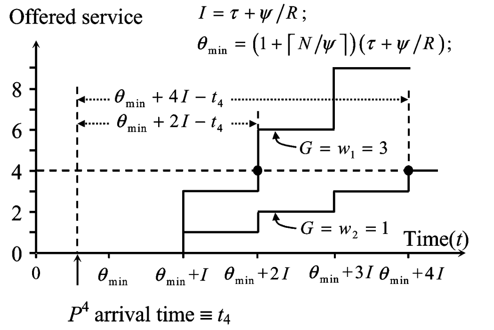
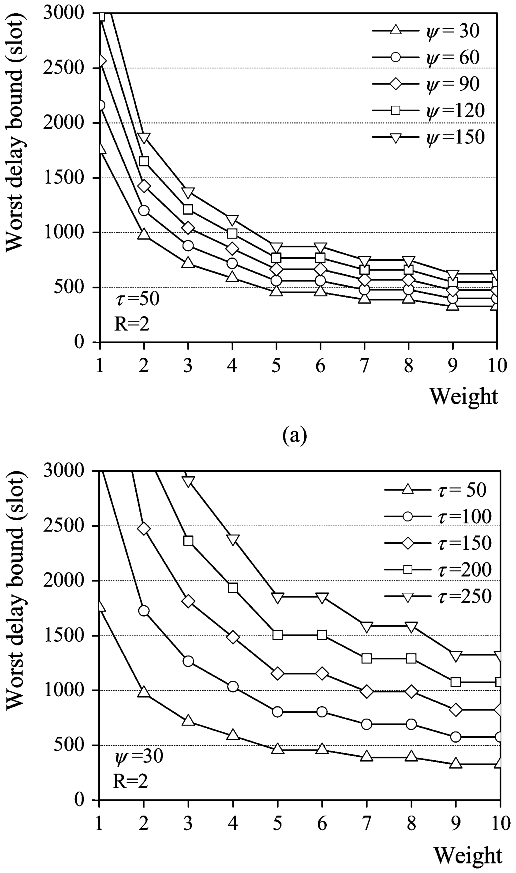


Fig. 5. (ψ,τ)-Scheduler's stepwise service curves for two classes.

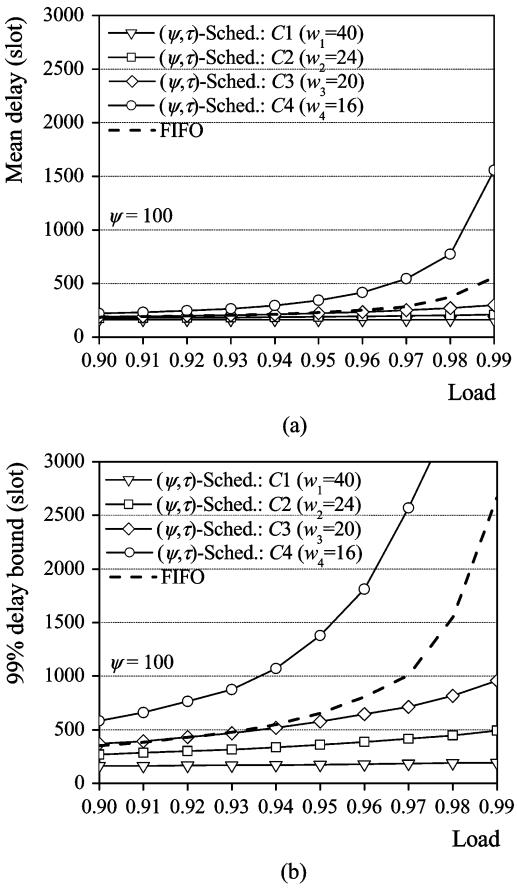


(a)

(b)

Fig. 6. Worst delay bound of an observed packet in bulk arrival. (a) Under different ψ values. (b) Under different τ values.

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 Fig. 6 the worst delay bound as a function of the normalized weight for the observed packet, under a bulk arrival of 25 packets (including the observed packet). We reveal from the figure that the worst delay bound dramatically declines as the class weight increases under all (ψ,τ) settings. 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)

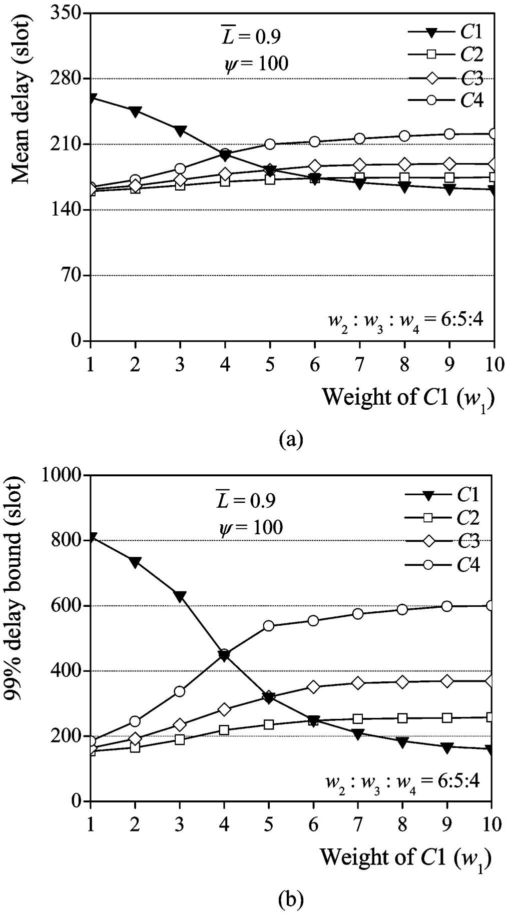
(b)

Fig. 7. Delay QoS provision under various loads. (a) Mean delay. (b) 99% delay bound.

C. 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, we have four delay classes ($C1$–$C4$), 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 $H\ (L)$ to $L\ (H)$ is equal to $\alpha = 0.225$ ($\beta = 0.025$), and the probability of having one packet arrival during state $H\ (L)$ is equal to $\bar{L}\ (\bar{L}/6)$, under an offered load, $\bar{L}$. 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 Figs. 7 and 8.

We observe from Fig. 7 that both mean delay and 99% delay bound of all classes increase with the offered load. 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., $C1$ and $C2$) at a cost of increased delay/bound for low-priority classes (e.g., $C4$). In Fig. 8, we illustrate how the weight of a class can be adjusted to meet its delay/bound requirements. For example, as shown in Fig. 8(b), to meet a 99% delay bound guarantee of 200 slots for class $C1$, the weight of $C1$ must be greater than 7, given the weights of three other classes of 6, 5, and 4, respectively.



(a)

(b)

Fig. 8. Delay QoS provision via the weight adjustment. (a) Mean delay. (b) 99% delay bound.

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

For clarity purposes, we highlight the operation of (ψ, τ)-Shaper, particularly the BATr part of the system in the sequel. A burst of size ψ is generated and transmitted 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 nonempty queue.

A. 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, we consider (ψ, τ)-Shaper 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

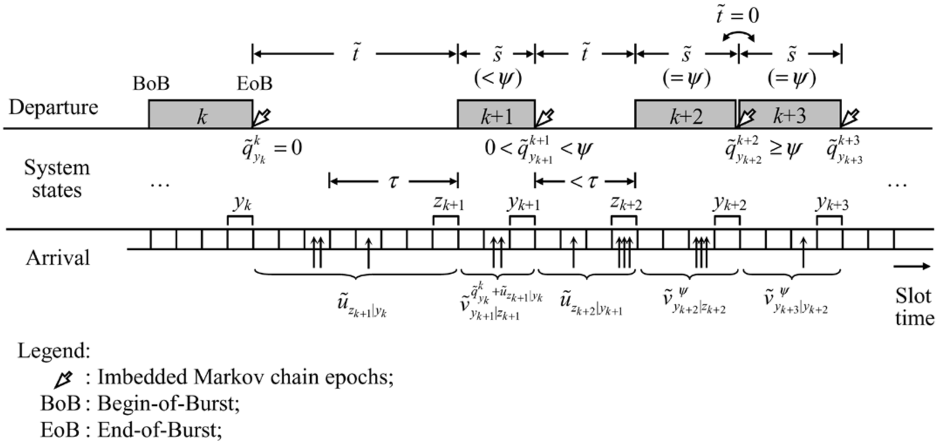

$\tilde{t}=0$
$\tilde{t}$
$\tilde{s}$ $(<\psi)$
$\tilde{t}$
$\tilde{s}$ $(=\psi)$
$\tilde{s}$ $(=\psi)$
BoB
EoB
Departure
k
$k+1$
$k+2$
$k+3$
System states
$\tilde{q}^{k}_{y_k}=0$
$0<\tilde{q}^{k+1}_{y_{k+1}}<\psi$
$\tilde{q}^{k+2}_{y_{k+2}}\geq\psi$
$\tilde{q}^{k+3}_{y_{k+3}}$
τ
$<\tau$
y_k
z_{k+1}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

Fig. 9. (ψ,τ)-Shaper: departure process analysis.

states, which correspond to high and low mean arrival rates, respectively. The MMBP is characterized by four parameters $(\alpha,\beta,\lambda_H,\lambda_L)$, where $\alpha(\beta)$ is the probability of changing from state $H(L)$ to $L(H)$ in a slot, and $\lambda_H(\lambda_L)$ represents the probability of having a batch arrival at state $H(L)$. 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, we thus have

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

Fig. 9 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 interdeparture time $(\tilde{t})$, and burst size $(\tilde{s})$ distributions. The burst interdeparture 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. Our goal is to find the joint distribution of $\tilde{t}$ and $\tilde{s}$, i.e., $P_{\tilde{t},\tilde{s}}(t,s), t\geq 0, s\leq\psi$. 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 Fig. 9.

Define random variable $\tilde{q}^{k}_{y_k}$ to be the number of packets left behind by the kth 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 interdeparture 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 $\tilde{v}^{n}_{z\,|\,y}$ 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.

Accordingly, we find that

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

where $y_k, y_{k+1}, z_{k+1}\in\{H,L\}$, and $(a)^{+}=\max\{a,0\}$. In (5), a nonnegative term within the parentheses corresponds to the departure of a full-size $(=\psi)$ 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 kth burst departure time, and $\tilde{u}_{z_{k+1}\,|\,y_k}$ and $\tilde{v}^{\min\{\tilde{q}^{k}_{y_k}+\tilde{u}_{z_{k+1}\,|\,y_k},\psi\}}_{y_{k+1}\,|\,z_{k+1}}$ are independent of any events that occur prior to time index $k, \{\tilde{q}^{k}_{y_k}, y_k\in\{H,L\}, k\geq 1\}$ is hence an imbedded Markov chain.

Based on (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 prior to the beginning of the interval, to state r_t at the end of the interval. For $t=0$, we immediately have $c^{0}_{r_0\,|\,r_0}(0)=1$. For $t\geq 1, c^{t}_{r_t\,|\,r_0}(m)$ can be recursively computed as

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

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 (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 a batch of $n(n\leq m)$ packets that arrive in the last slot with probability $\lambda_{r_t}b_{r_t}(n)$.

With the "$(\,)^+$" sign removed, (5) can be expanded into three cases, as seen in (7) at the bottom of the page. Notice that $\tilde{u}_{z_{k+1}\,|\,y_k}$ is absent from the first case of (7) due to the fact that the interdeparture 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 (7), as

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

where

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

and

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

To proceed, we need to solve $P[\tilde{u}_{z_{k+1}\,|\,y_k} = u \,|\, \tilde{q}^k_{y_k} = q]$ in (10). It can be resolved by considering five cases depending on different ranges of u and q values as given in (11). 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 $q < \psi$ but $u + q \geq \psi$, the total number of packets must exceed ψ the first time at a particular slot before the BATr expires. Namely, within an interval of less than τ, there arrives a total of $m (0 \leq m \leq \psi - q - 1)$ packets, and exactly at this slot, a batch of $u - m$ packets arrives, making $m + (u - m) + q \geq \psi$. 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 $u (u < \psi - q)$, and $u + q < \psi$. Case (4) in (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 $m (0 < m \leq u)$ 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, we have (11) at the bottom of the page.

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

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

We are now in the position to determine the departure process distribution, $P_{\tilde{t},\tilde{s}}(t, s)$. We consider four cases depending on different t and s values. First, in Case I) when $t = 0$, it is clear that

Case I) $t = 0$

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

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

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

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

Case II) $0 < t < \tau$ (see (14) at the bottom of the page).

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) $t = \tau$ [see (15) at the bottom of the page].

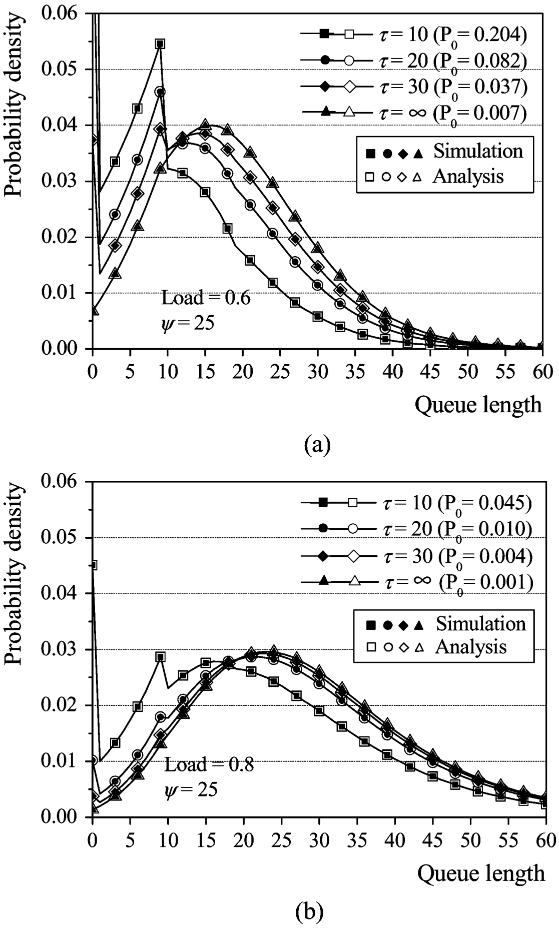
Finally, under the last case when $t > \tau$, the departing burst must have left an empty system ($P[\tilde{q}_y = 0]$), resulting in the deactivation of the BATr. The timer remains deactivated until the arrival of the first batch of packets. Then, whether the next departing burst is a full-size one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

Case IV) $t > \tau$ [see (16) at the bottom of the page].

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

B. 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 (interdeparture and burst size distributions) are shown in Figs. 10 and 11, respectively. 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. Accordingly, the burstiness of traffic is $B = 3$ under both loads.



(a)

(b)

Fig. 10. System queue length distribution. (a) Medium load (0.6). (b) High load (0.8).

First, all analytical results are in profound agreement with simulation results. Interestingly, we discover from Fig. 10 that there are some spikes at queue-length = 9 in the queue length distribution. The phenomenon is caused by the maximum batch size of 9 in the arrival process. In addition, we observe that the interdeparture time distribution is sensitive to ψ and τ. Under a high load ($\bar{L} = 0.8$) condition, we observe the interdeparture time of zero (burst size $= \psi = 25$) occurs with the largest probability under all τ values. The second largest probability for different τ settings occurs at the interdeparture time being equal to the corresponding τ value, as shown by the spikes in Fig. 11(a).

To examine the effectiveness of shaping, we further compute the coefficient of variation (CoV) for the interdeparture time and burst size, under three ψ values ($\psi = 1, 10$, and 100) and various MMBP arrivals ($B = 1, 3$, and 5; $\bar{b}_H = \bar{b}_L = 5, 7$, and 9). Notice that the setting of $\psi = 1$ corresponds to a FIFO system

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

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

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

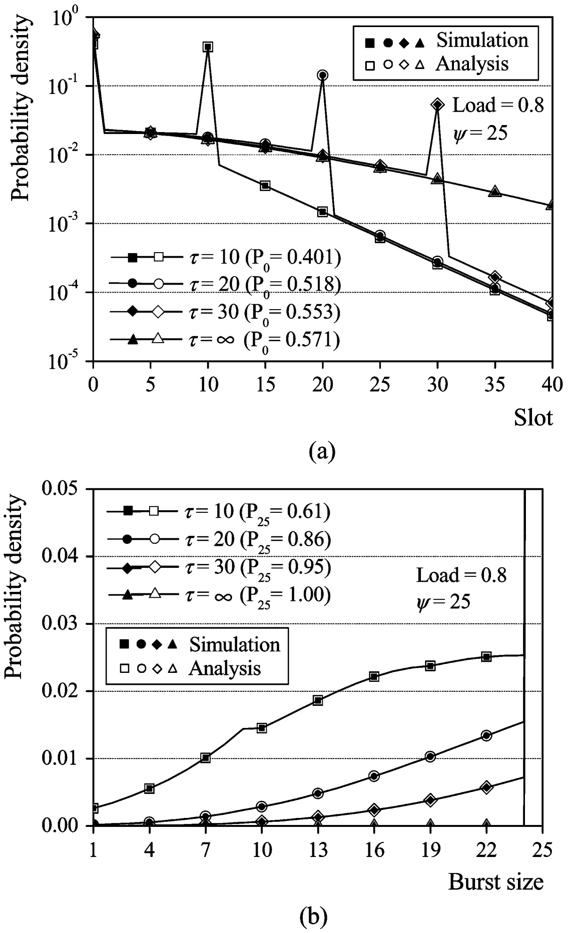


Fig. 11. Departure process distributions. (a) Interdeparture time ($\tilde{t}$) distribution. (b) Burst size ($\tilde{s}$) distribution.

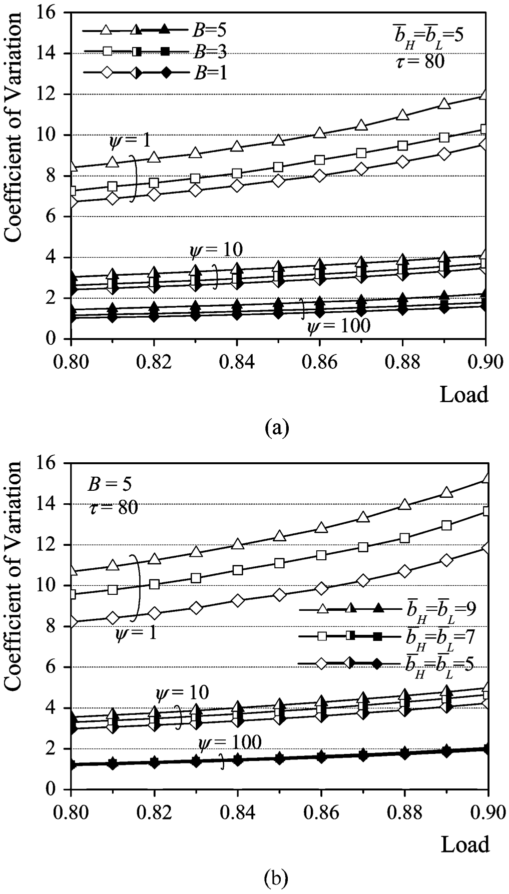


Fig. 12. CoV of the interdeparture time. (a) Under different B and ψ values. (b) Under different batch sizes and ψ values.

with no shaping. Numerical results are plotted in Figs. 12 and 13.

As shown in Fig. 12, as expected, the CoV of the interdeparture time increases with the offered load. Crucially, under any MMBP arrival, we discover that 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 next section. Moreover, we observe from Figs. 12 and 13 that the burstiness and batch size of the original MMBP arrival has an 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 Fig. 13, 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.

V. Loss QoS Provision and Performance Comparison

In this section, 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 section, 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 network we used in the experiment is the well-known ARPANET network [26] with 24 nodes and 48 links, in which 14 nodes are randomly selected as edge nodes. OLSP 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 0.48 μs. 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 ($\bar{b}_H = \bar{b}_L = 5$). For a given load ($\bar{L}$) according to (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 probability that a packet arrives at

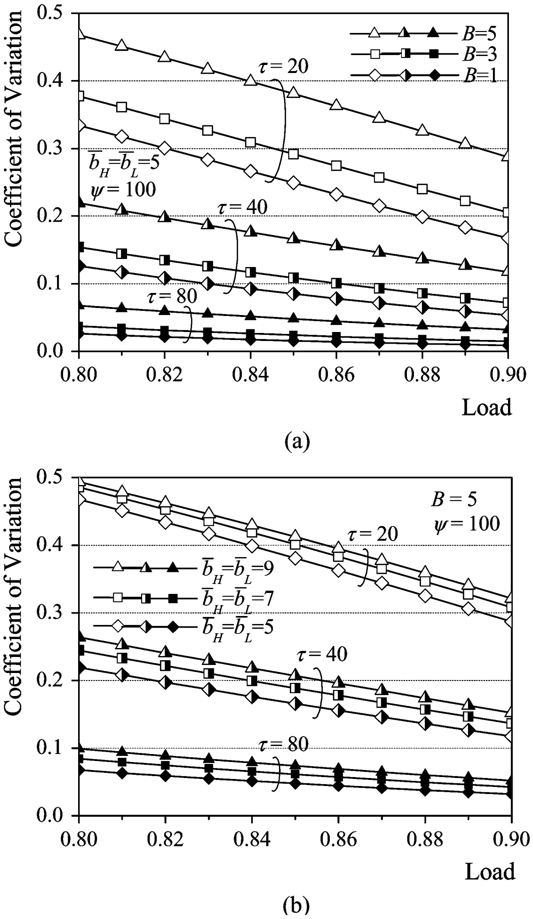


Fig. 13. CoV of the burst size. (a) Under different B and τ values. (b) Under different batch sizes and τ values.

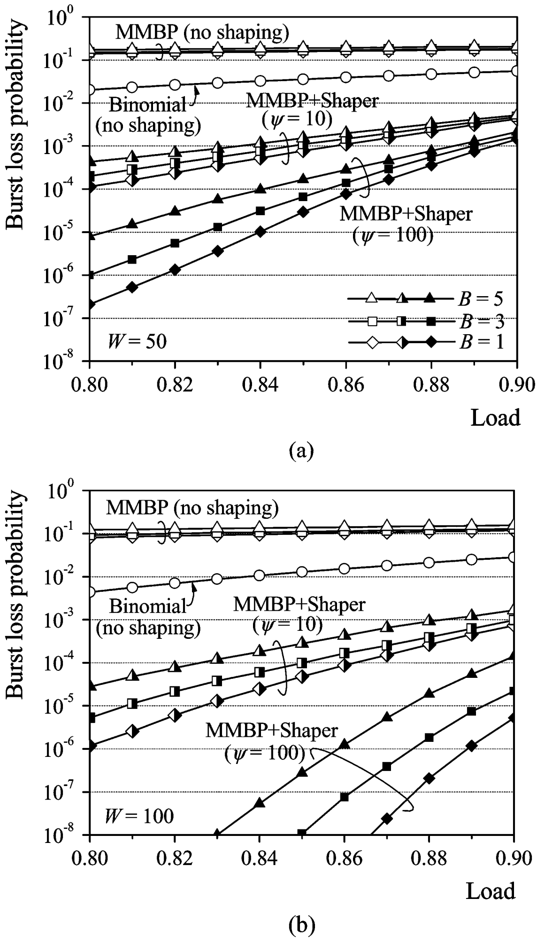


Fig. 14. Traffic shaping effect: a comparison between the OCPS and baseline networks. (a) $W = 50$. (b) $W = 100$.

each slot is equal to the mean load $\bar{L}$, yielding a total offered load of $W \cdot \bar{L}$, where W is the number of wavelengths. Simulations are terminated after reaching 95% confidence interval. In the sequel, we explore these three aforementioned aspects in the three subsections, respectively.

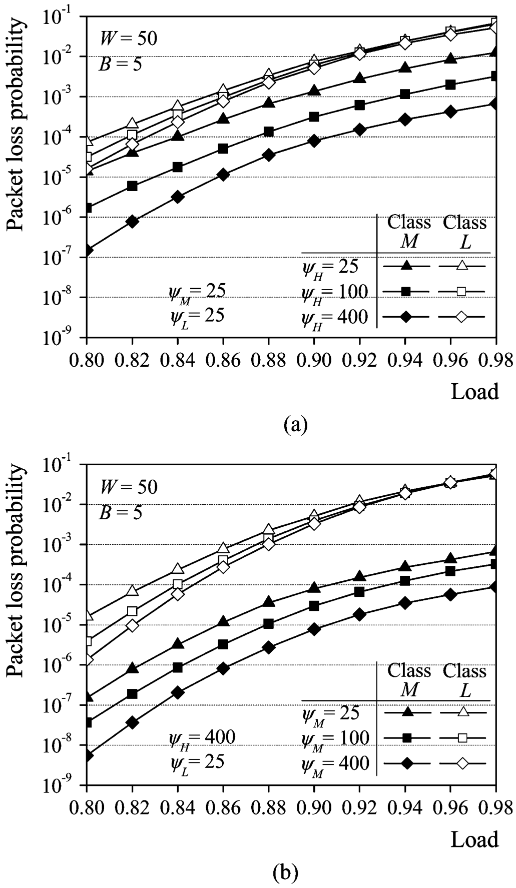
A. *Traffic Shaping Effect*

To examine the traffic shaping effect, we draw a comparison of burst loss probability between the baseline and OCPS networks. Simulation results are plotted in Fig. 14. We first observe from the figure that the results are consistent with our previous analytic CoV results—the greater the ψ value, the lower the CoV and the burst loss probability. As shown in Fig. 14(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 $W = 50, \psi = 100$, and $\bar{L} = 0.8$ 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 Fig. 14(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.

B. *Loss QoS Provisioning*

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.

In 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 Figs. 15 and 16.

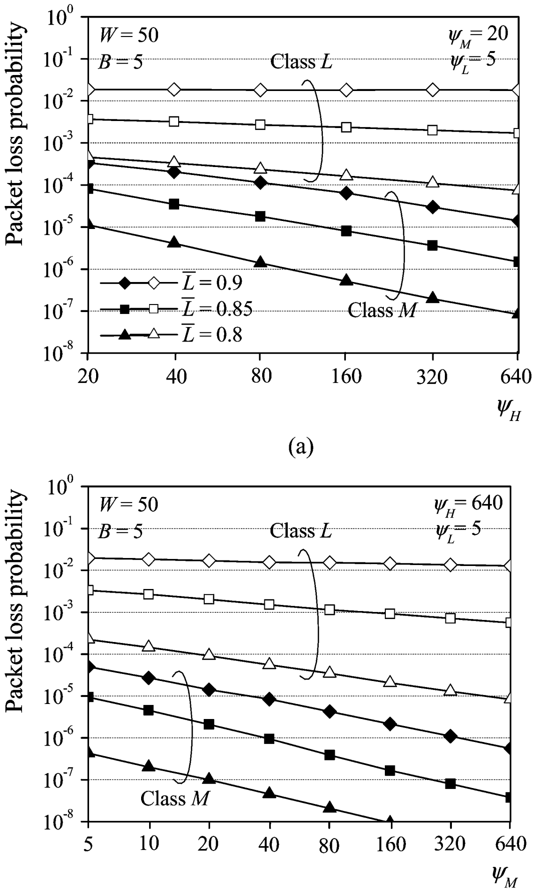


(a)

(b)

Fig. 15. (ψ, τ)-Shaper: loss performance under different loads. (a) Changing of the burst size of Class H. (b) Changing of the burst size of Class M.

In Fig. 15, 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 Fig. 15(a)] and Class M [in Fig. 15(b)]. 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 Fig. 16 we plot the packet loss probabilities of Classes M and L as a function of the burst size of Class H [in Fig. 16(a)] and Class M [in Fig. 16(b)]. 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 Fig. 16(a), due to the "least-harm" preemption guideline, Class M with a larger size $(\psi_M = 20)$ becomes less likely to be truncated than Class L with a smaller size $(\psi_L = 5)$, 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 Fig. 16(b), 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)

(b)

Fig. 16. (ψ, τ)-Shaper: loss QoS provision via burst size adjustment. (a) Changing of the burst size of Class H. (b) Changing of the burst size of Class M.

C. 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. We carried out simulations on the same 24-node ARPANET network in which three traffic classes (Classes H, M, and L) were adopted. In simulations, each ingress node generates a total of 39 connections (three classes for each of 13 destination nodes) that follow different load-balancing OLSPs. For ease of comparison, we use the same burst size for all three classes 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., 48 μs for $\psi = 100$), 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

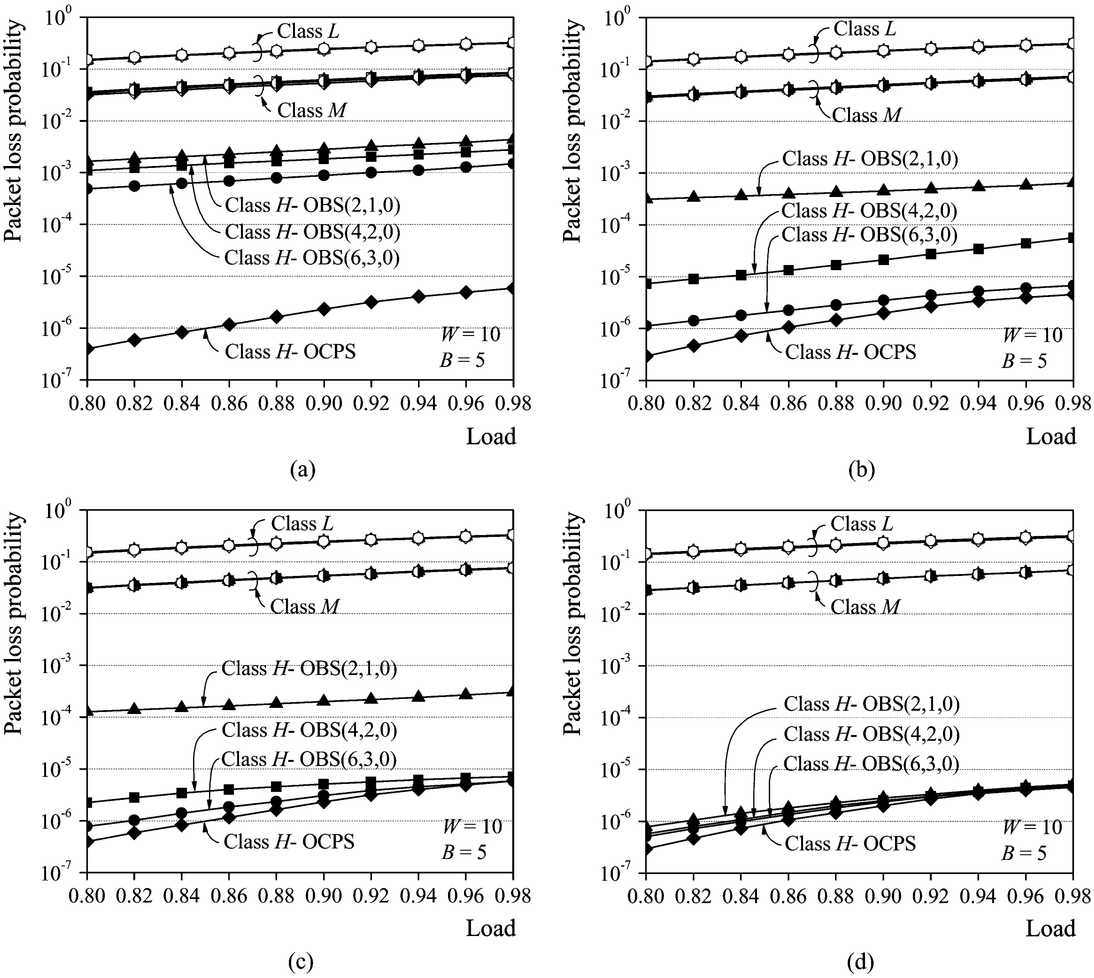


Fig. 17. OCPS and OBS loss performance comparison. (a) $\psi_H = \psi_M = \psi_L = 25$ and $\delta = 48$ μs. (b) $\psi_H = \psi_M = \psi_L = 100$ and $\delta = 48$ μs. (c) $\psi_H = \psi_M = \psi_L = 25$ and $\delta = 9.6$ μs. (d) $\psi_H = \psi_M = \psi_L = 100$ and $\delta = 9.6$ μs.

length. It thus requires a large offset time difference between any two adjacent classes, such as (6, 3, 0), to meet 95% of traffic isolation degree. In our 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% 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 Fig. 17.

In Fig. 17, 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 $(\psi = 25, 100)$ and two header processing times ($\delta = 9.6$ μs, 48 μs). 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 as shown in Fig. 17(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., $\psi_H = \psi_M = \psi_L = 25$. Among the three OBS variants, $\text{OBS}(2, 1, 0)$ 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 $\psi_H = \psi_M = \psi_L = 25$ and $\delta = 48$ μs, as shown in Fig. 17(a). As the burst size increases and the processing time decreases, as shown in Fig. 17(b)–(d), the near-far problem is relaxed, yielding noticeable performance improvement for Class H in OBS networks. 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.

VI. Conclusion

In this paper, we have proposed 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 interdeparture time, resulting a substantial reduction in burst loss probability. We have performed simulations on an ARPANET network to make loss performance comparisons between the OCPS with (ψ, τ)-Shaper and the 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.

Appendix A

Proof of Lemma 1: First, we denote stepwise function $\delta(t, \theta^*_{\min})$ as δ'. Given time t between interval $[T^{\delta'}_k, T^{\delta'}_{k+1})$, by Definition 1 and the definition of T^Π_k, we get the first inequality: $\delta(t, \theta^*_{\min}) = k \cdot G \leq \Pi(T^\Pi_k)$. Since $\Pi(T)$ is monotonically increasing and $T^\Pi_k \leq T^{\delta'}_k \leq t$, we have the second inequality: $\Pi(T^\Pi_k) \leq \Pi(t)$. Combining the two inequalities, we obtain $\delta(t, \theta^*_{\min}) \leq \Pi(t)$. According to Definition 2, since there exists only one minimum θ, namely $\theta_{\min}, \theta^*_{\min}$ is thus lower bounded by $\theta_{\min}$, namely $\theta^*_{\min} \geq \theta_{\min}$.

Moreover, (2) leads to a fact that inequality $T^\Pi_k \leq T^\Delta_k$ holds at $\theta = \theta_{\min}$, for all $k \geq 0$. From the definition of $\theta^*_{\min}$ in the lemma, which indicates that $\theta^*_{\min}$ is the minimum θ making $T^\Pi_k \leq T^\delta_k$ satisfied for all $k \geq 0$, for all stepwise functions including $\Delta(t)$, we imply that $\theta^*_{\min}$ is upper bounded by $\theta_{\min}$ namely $\theta^*_{\min} \leq \theta_{\min}$. Accordingly, the lemma is proved. ■

Appendix B

Proof of Theorem 1: With the focus placed on an observed busy period of class i, let P^1_i be the first packet initiating the busy period, and P^j_i represent the jth $(j \geq 1)$ 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 $d_2(P^j_i)$ denote the index of the window in which P^j_i is placed. We immediately have the boundary condition, $d_1(0) = d_2(P^1_i) = 1$. According to the virtual-window service policy of (ψ, τ)-Scheduler, we get the following inequality:

$$d_2\left(P^j_i\right) - d_2\left(P^{j-1}_i\right) \leq \lceil 1/w_i \rceil \leq 1, \quad \forall j > 1. \tag{17}$$

Suppose after packet P^j_i has been served, the total service amount has first exceeded $k \cdot w_i$. We get $d_2(P^j_i) = d_1(T^\Pi_k)$, and $j - 1 < k \cdot w_i \leq j$. Since $k \cdot w_i$ is greater than $j - 1$, packet P^{j-1}_i must have been served no later than the kth window. In other words, one gets

$$d_2\left(P^{j-1}_i\right) \leq k. \tag{18}$$

By summing (17) and (18), we arrive at

$$d_1\left(T^\Pi_k\right) \leq k + 1. \tag{19}$$

Equation (19) can be described in words as that, in order to finish service amount $k \cdot w_i$, the total number of windows elapsed $d_1(T^\Pi_k)$ is bounded by $k + 1$.

Moreover, due to the fact that the normalized weight of any class can be a noninteger 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^j_i will be placed at worst in the $(k + 1 + \lceil N/\psi \rceil)$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

$$T^\Pi_k \leq (k + 1 + \lceil N/\psi \rceil) \cdot (\tau + \psi/R). \tag{20}$$

By assigning the least upper bound of T^Π_k to T^Δ_k, we have

$$T^\Pi_k \leq (k + 1 + \lceil N/\psi \rceil) \cdot (\tau + \psi/R) \equiv T^\Delta_k.$$

Subtracting T^Δ_k by $k \cdot I$ where $I = \tau + \psi/R$, we obtain $\theta^*_{\min} = (1 + \lceil N/\psi \rceil)(\tau + \psi/R)$. By Lemma 1, the theorem is proved. ■

Acknowledgment

The authors deeply thank the anonymous reviewers for the valuable comments and suggestions that greatly improved the presentation of the paper.

References

[1] B. Mukherjee, "WDM optical communication networks: Progress and challenges," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1810–1824, Oct. 2000.

[2] T. El-Bawab and J.-D. Shin, "Optical packet switching in core networks: Between vision and reality," *IEEE Commun. Mag.*, vol. 40, pp. 60–65, Sept. 2002.

[3] L. Xu, H. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Commun. Mag.*, vol. 39, pp. 136–142, Jan. 2001.

[4] F. Callegati, G. Corazza, and C. Raffaelli, "Exploitation of DWDM for optical packet switching with quality of service guarantees," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 190–201, Jan. 2002.

[5] L. Xu, H. Perros, and G. Rouskas, "The perspective of optical packet switching in IP dominant backbone and metropolitan networks," *IEEE Commun. Mag.*, vol. 39, pp. 136–141, Mar. 2001.

[6] H. Dorren *et al.*, "Optical packet switching and buffering by using all-optical signal processing methods," *J. Lightwave Technol.*, vol. 21, pp. 2–12, Jan. 2003.

[7] D. Hunter *et al.*, "SLOB: A switch with large optical buffers for packet switching," *J. Lightwave Technol.*, vol. 16, pp. 1725–1736, Oct. 1998.

[8] T. Battestilli and H. Perros, "An introduction to optical burst switching," *IEEE Commun. Mag.*, vol. 41, pp. S10–S15, Aug. 2003.

[9] M. Yoo, C. Qiao, and S. Dixit, "Optical burst switching for service differentiation in the next generation optical internet," *IEEE Commun. Mag.*, vol. 39, pp. 98–104, Feb. 2001.

[10] V. Vokkarane and J. Jue, "Prioritized burst segmentation and composite burst-assembly techniques for qos support in optical burst-switched networks," *IEEE J. Select. Areas Commun.*, vol. 21, pp. 1198–1209, Sept. 2003.

[11] J. Wei and R. McFarland, "Just-in-time signaling for WDM optical burst switching networks," *J. Lightwave Technol.*, vol. 18, pp. 2019–2037, Dec. 2000.

[12] C. Qiao, "Labeled optical burst switching for IP-over-WDM integration," *IEEE Commun. Mag.*, vol. 38, pp. 104–114, Sept. 2000.

[13] Y. Xiong, M. Vandenhoute, and H. Cankaya, "Control architecture in optical burst-switched WDM networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1838–1851, Oct. 2000.

[14] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 2062–2071, Oct. 2000.

[15] A. Ge, F. Callegati, and L. Tamil, "On optical burst switching and self-similar traffic," *IEEE Commun. Lett.*, vol. 4, pp. 98–100, Mar. 2000.

[16] M. Yuang, J. Shih, and P. Tien, "QoS burstification for optical burst switched WDM networks," in *Proc. IEEE OFC*, 2002, pp. 781–783.

[17] L. Yang, Y. Jiang, and S. Jiang, "A probabilistic preemptive scheme for providing service differentiation in OBS networks," in *Proc. IEEE GLOBECOM*, 2003.

[18] 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)," *J. Lightwave Technol.*, vol. 20, pp. 154–165, Feb. 2002.

[19] J. White, M. Zukerman, and H. Vu, "A framework for optical burst switching network design," *IEEE Commun. Lett.*, vol. 6, pp. 268–270, June 2002.

[20] M. Yuang, J. Shih, and P. Tien, "Traffic shaping for IP-over-WDM networks based on optical coarse packet switching paradigm," in *Proc. Eur. Conf. Optical Communication (ECOC)*, 2003.

[21] Y. Lin, M. Yuang, S. Lee, and W. Way, "Using superimposed ASK label in a 10 Gbps multi-hop all-optical label swapping system," *J. Lightwave Technol.*, vol. 22, pp. 351–361, Feb. 2004.

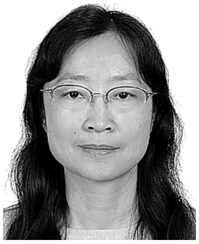
[22] S. Danielsen, P. Hansen, and K. Stubkjaer, "Wavelength conversion in optical packet switching," *J. Lightwave Technol.*, vol. 16, pp. 2095–2108, Dec. 1998.

[23] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Trans. Networking*, vol. 1, pp. 344–357, June 1993.

[24] J. Bennett and H. Zhang, "WF^{2}Q: Worst-case fair weighted fair queueing," in *Proc. IEEE INFOCOM*, 1996, pp. 120–128.

[25] D. Stiliadis and A. Varma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," *IEEE/ACM Trans. Networking*, vol. 6, pp. 611–624, Oct. 1998.

[26] H. Harai, M. Murata, and H. Miyahara, "Performance analysis of wavelength assignment policies in all-optical networks with limited-range wavelength conversion," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1051–1060, Sept. 1998.



Maria C. Yuang received the B.S. degree in applied mathematics from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1978, the M.S. degree in computer science from the University of Maryland, College Park, in 1981, and the Ph.D. degree in electrical engineering and computer science from the Polytechnic University, Brooklyn, NY, in 1989.

From 1981 to 1990, she was with AT&T Bell Laboratories and Bell Communications Research (Bellcore), where she was a Member Of Technical Staff working on high-speed networking and protocol engineering. She was also an Adjunct Professor in the Department of Electrical Engineering, Polytechnic University, during 1989–1990. In 1990, she joined National Chiao Tung University, where she is currently a Professor of the Department of Computer Science and Information Engineering. Her current research interests include optical and broad-band networking, wireless local/access networking, multimedia communications, and performance modeling and analysis.

Po-Lung Tien received the B.S. degree in applied mathematics, the M.S. degree in computer and information science, and the Ph.D. degree in computer and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1992, 1995, and 2000, respectively.

In 2000, he joined National Chiao Tung University, where he is currently a Research Assistant Professor of the Department of Computer Science and Information Engineering. His current research interests include optical networking, wireless local networking, multimedia communications, performance modeling and analysis, and applications of soft computing.

Julin Shih was born in Taipei, Taiwan, R.O.C., in 1976. He received the B.S. degree in management and information systems from the National Central University, Chung-li, Taiwan, in 1999 and the M.S. degree in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2001, where he is currently pursuing the Ph.D. degree.

His currently research interests include high speed networking, optical networking, and performance modeling and analysis.

Alice Chen received the B.S. degree in electronics engineering and the M.S. degree in computer science and information engineering from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1984 and 1992, respectively.

Currently, she is a Senior Engineer at Computer and Communications Research Laboratories of Industrial Technology Research Institute, Hsinchu, where she works on network control and management of optical networks.