ELSEVIER

Performance Evaluation 36–37 (1999) 333–358

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

Performance analysis of broadcast WDM networks under IP traffic

Martin W. McKinnon [a], Harry G. Perros [b,*], George N. Rouskas [b]

[a] Georgia Tech Research Institute, Georgia Tech, Atlanta, GA 30332-0850, USA
[b] Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534, USA

Abstract

We consider the problem of carrying IP packet traffic over a broadcast WDM network. The network operates under a schedule that masks the transceiver tuning latency. Variable length IP packets are segmented at the transmitting end, the fixed-size segments are transmitted in slots specified by the schedule and are then reassembled into the original packet at the receiving end. We develop and analyze approximately a queueing model of the network in order to obtain the queue-length distribution and loss probabilities at the transmitting and receiving side of nodes. The analysis is carried out assuming finite buffer sizes, non-uniform destination probabilities, and an appropriate arrival process model. Our work makes it possible to study the interactions among the various system parameters (such as load balancing and scheduling algorithms, the number of channels, and the buffer capacity) and to predict, explain, and fine tune the performance of the network. To the best of our knowledge, this is the first comprehensive performance study of optical local area networks under variable length packets.

Keywords: Optical networks; Wavelength division multiplexing (WDM); Discrete-time queueing networks; IP packets

1. Introduction

In recent years, optical broadcast-and-select local area network (LAN) architectures based on wavelength division multiplexing (WDM) have been a subject of research both theoretically [2,6,8–10,12,14–17,19,23,24,28,34] and experimentally [11,18,13,7]. The issues that have been addressed include the design of efficient media access control (MAC) protocols [6,8,31,34], the development of scheduling algorithms to mask the transceiver tuning latency [2,5,25,28,30], multicasting [4,25,29,35], and dynamic load balancing [3]. Because of the synchronization problems arising in a WDM environment, most of the studies make the assumption that the traffic offered to the network consists of fixed-size packets. Furthermore, the performance analysis of most architectures has been typically carried out under the unrealistic [26] assumptions of symmetric traffic and memoryless arrival processes. On the other hand, we have studied WDM broadcast star networks assuming bursty and correlated arrival processes, non-uniform destination probabilities, finite buffer capacities, and non-zero transceiver tuning delays in [22,31].

* Corresponding author. Tel.: +1 919 515 2858; Fax: +1 919 515 7925; E-mail: perros@csc.ncsu.edu

0166-5316/99/$ – see front matter
PII: S0166-5316(99)00031-0

The work in [31] derived a stability condition for a reservation-based MAC protocol, while in [22], we analyzed a fairly general queueing network model of the broadcast-and-select WDM architecture.

While it has been widely accepted [21] that current IP networks will naturally evolve to take advantage of WDM capabilities, it has also been recognized that there exists a gap between the optical network and Internet technologies [1]. We have taken a first step towards bridging this gap with our work in [25], where we designed a novel approach to provide support for LAN-wide MAC-layer multicast. This multicast capability can be useful to many link-layer Internet protocols, as well as protocols such as IP multicast and IGMP. In this paper, we address another important issue, namely the performance of IP traffic over WDM LANs. Specifically, we extend our analysis in [22] to the case where arriving packets are not fixed in size. Rather, we assume variable-size packets (such as IP datagrams) with an arbitrarily distributed length. An IP packet is fragmented at the MAC layer into fixed-sized segments. The segments are then individually transmitted over the broadcast star to the destination node, where they are reassembled into the original packet at the MAC layer before they are passed to the IP layer. All buffers in the MAC layer are finite in capacity. We construct and analyze a general queueing network model of a broadcast WDM LAN that captures the complex interaction among the various system parameters such as packet-length distribution, number of wavelengths (channels), the schedule itself, and buffer capacity. The analysis of the queueing network is approximate and it is based on a 'per channel' decomposition. To the best of our knowledge, such a performance study of a broadcast WDM architecture with variable-length packets has not been reported in the literature.

In the next section we present the salient features of a broadcast optical local area network and we provide some background information. We present the performance analysis of the network in Sections 3 and 4, we give numerical results in Section 5, and we conclude the paper in Section 6.

2. The network under study

In this section we introduce a model for the media access control (MAC) layer in a broadcast-and-select WDM LAN. The model consists of two parts, a queueing network and a transmission schedule. We also present a traffic model to characterize the arrival process of IP packets to the network.

2.1. The queueing model

The optical network architecture consists of N nodes communicating over a broadcast passive star that can support $C \le N$ wavelengths $\lambda_1, \ldots, \lambda_C$ (see Fig. 1). Each node is equipped with a laser that enables it to inject signals into the optical medium, and a filter capable of receiving optical signals. The laser at each node is assumed to be tunable over all available wavelengths. The optical filters, on the other hand, are fixed to a given wavelength. Let $\lambda(j)$ denote the receive wavelength of node j. Since $C \le N$, a set $\mathcal{R}_c$ of nodes may be sharing a single receive wavelength λ_c: $\mathcal{R}_c = \{j \mid \lambda(j) = \lambda_c\}, c = 1, \ldots, C$. Sets $\mathcal{R}_c$ will typically be obtained by running a load balancing algorithm [3].

Internally, the network operates by transmitting *fixed-size* units of data, hereafter referred to as *segments*. The nodes operate in a slotted mode, with a slot time equal to a segment transmission time. Since there are N nodes but $C \le N$ channels, each channel must run at a rate N/C times faster than the rate at which users at each node can generate or receive data (N/C need not be an integer). In other words, the MAC-to-network interface runs faster than the user-to-MAC interface. Thus, we

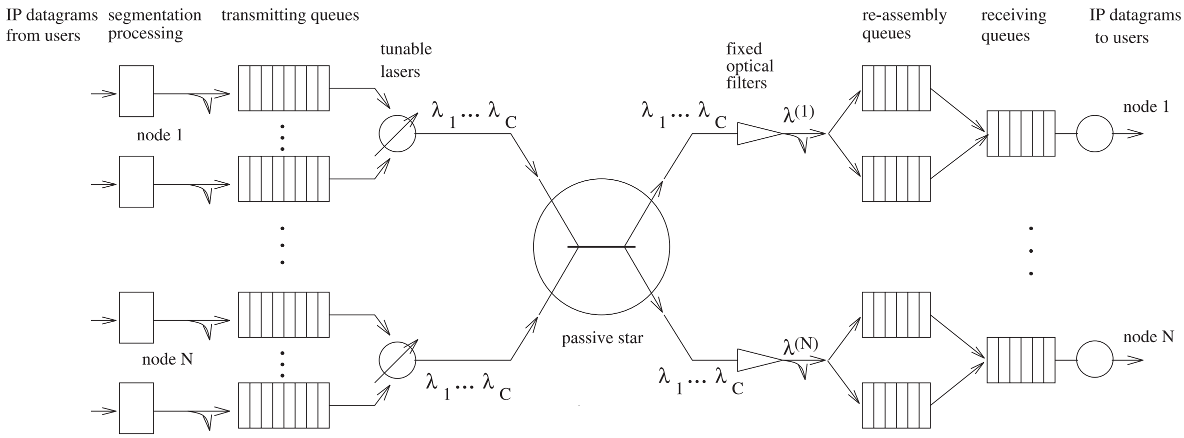


Fig. 1. Queueing model of the network architecture with variable-length IP packets.

distinguish between *arrival* slots (which correspond to the segment transmission time at the user rate) and *service* slots (which are equal to the segment transmission time at the channel rate within the network). Obviously, the duration of a service slot is equal to C/N times that of an arrival slot. Without loss of generality, we also assume that all N nodes are synchronized at service slot boundaries.

Each node consists of a transmitting side and a receiving side, as Fig. 1 illustrates. Network users generate *variable-length* IP packets, which arrive at the transmitting side of a node and undergo segmentation processing. As a result of this process, a packet is fragmented [1] into a number of segments, where the last segment is padded, if necessary [2]. These segments are buffered at a finite capacity queue, if the queue is not full. If, however, the queue is full, all segments of the packet are dropped upon arrival [3]. As Fig. 1 illustrates, the buffer space at the transmitting side of each node is assumed to be partitioned into C independent queues. Each queue $c, c = 1, \dots, C$, at the transmitting side of node $i, i = 1, \dots, N$, contains segments destined for receivers which listen to wavelength λ_c. This arrangement eliminates the head-of-line problem, and permits a node to send several segments back-to-back when its laser is tuned to a certain wavelength. We let $B_{ic}^{(\text{in})}$ denote the capacity, in segments, of the transmitting queue at node i that corresponds to wavelength λ_c.

Segments buffered at a transmitting queue are sent on a FIFO basis onto the optical medium by the node's laser. A schedule (discussed shortly) ensures that transmissions on a given channel will not collide, hence a transmitted segment will be correctly received by its destination node. Upon arrival at the receiving side of its destination node, the segments are buffered in a reassembly queue. At each node, there is one reassembly queue per source. Each reassembly queue has a finite capacity and it can accommodate a packet of maximum length. Segments accumulate in this queue until an entire packet has been completely received. At that instant, the packet is transferred instantaneously to the receiving queue (see Fig. 1), which also has a finite capacity. If adequate space is not available at the receiving queue, the entire packet is dropped.

[1] We emphasize that this segmentation process takes place at the MAC layer. Hence, it differs from IP-level fragmentation in that the segmentation and reassembly is transparent to the IP layer.

[2] We will omit implementation of specific issues of how segmentation is performed, and instead we will focus on an abstract queueing model that is applicable to a wide range of environments.

[3] This operation is reminiscent of the 'Partial Packet Discard' feature of some ATM switches.

We note that, due to the nature of the system, segments arrive at a reassembly queue in order. Furthermore, no losses can occur at a reassembly queue, since this queue can accommodate a packet of maximum size. We let $B_j^{(\mathrm{out})}$ denote the buffer capacity, in segments, of the receiving queue at node j. Packets in a receiving queue are also served on a FIFO basis.

2.2. Transmission schedules

One of the potentially difficult issues that arises in a WDM environment is that of coordinating the various transmitters/receivers. Some form of coordination is necessary because (a) a transmitter and a receiver must both be tuned to the same channel for the duration of a segment's transmission, and (b) a simultaneous transmission by one or more nodes on the same channel will result in a *collision*. The issue of coordination is further complicated by the fact that tunable transceivers need a non-negligible amount of time to switch between wavelengths.

The problem of scheduling packet transmissions in such an environment has received significant attention recently, and several scheduling algorithms have been proposed [2,5,25,28,30]. Although these algorithms differ in terms of their design and operation, surprisingly the resulting schedules are very similar. The underlying structure of these schedules is shown in Fig. 2. Typically, in a schedule, a node i is assigned a_{ic} *contiguous* service slots for transmitting segments on channel λ_c. These a_{ic} slots are followed by a *gap* of $g_{ic} \geq 0$ slots during which no node can transmit on λ_c. This gap may be necessary to ensure that the laser at node $i+1$ has sufficient time to tune from wavelength λ_{c-1} to λ_c before it starts transmission. Note that in Fig. 2 we have assumed that an arrival slot is an integer multiple of service slots. This may not be true in general, and it is not a necessary assumption for our model. Observe also that, although a schedule begins and ends on *arrival* slot boundaries, the beginning or end of transmissions by a node does not necessarily coincide with the beginning or end of an *arrival* slot (although they are, obviously, synchronized with *service* slots).

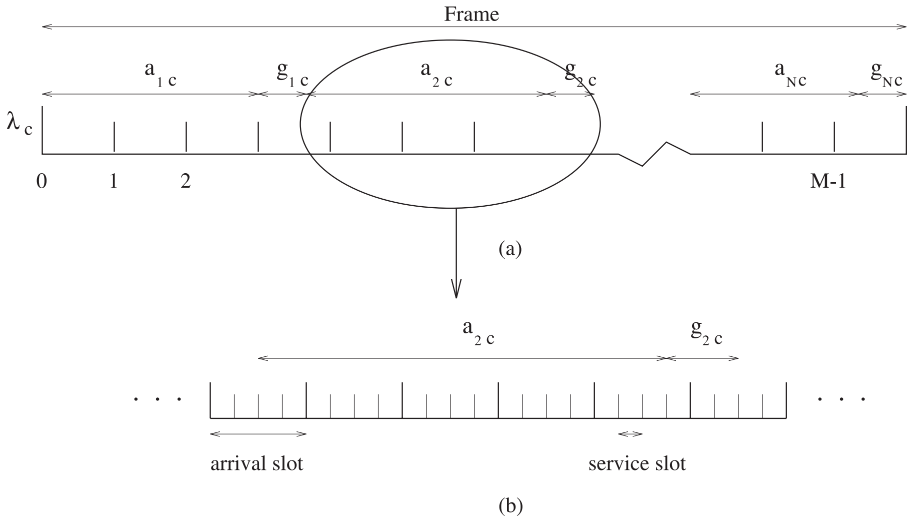


Fig. 2. (a) Schedule for channel λ_c. (b) Detail corresponding to transmitting queue 2.

We assume that transmissions by the transmitting queues onto wavelength λ_c follow a schedule as shown in Fig. 2. This schedule repeats over time. Each frame of the schedule consists of M arrival slots. Quantity $a_{ic}, i = 1, \ldots, N, c = 1, \ldots, C$, can be seen as the number of service slots per frame allocated to node i, so that the node can satisfy the required quality of service of its incoming traffic intended for wavelength λ_c. By fixing a_{ic}, we indirectly allocate a certain amount of the bandwidth of wavelength λ_c to node i. This bandwidth could, for instance, be equal to the effective bandwidth [27] of the total traffic carried by node i on wavelength λ_c. In general, the estimation of the quantities $a_{ic}, i = 1, \ldots, N, c = 1, \ldots, C$, is part of the connection admission algorithm [27], and it is beyond the scope of this paper. We note that as the traffic varies, a_{ic} may vary as well. In this paper, we assume that quantities a_{ic} are fixed, since this variation will more likely take place over larger scales in time.

2.3. The arrival process

Let T denote the maximum IP packet size, in segments, in the network. In other words, the size of a packet arriving to a transmitting queue will be anywhere between 1 and T segments. In order to capture the behavior of the various network users, a different arrival process is associated with each transmitting queue (i, c) (i.e., the queue at node i corresponding to wavelength $\lambda_c, i = 1, \ldots, N, c = 1, \ldots, C$). Specifically, we assume that a packet arriving at queue (i, c) consists of $s, s = 1, \ldots, T$, segments with probability $f_{ic}(s)$. These probabilities can be determined through careful analysis of traffic patterns (see, for instance [33, fig. 3]). Thus, our analysis is valid for arbitrary packet size distributions.

The rate diagram of the packet arrival process to a transmitting queue is shown in Fig. 3. There is a geometrically distributed idle period (state 0) during which no arrivals occur, followed by the arrival of a packet. The length s of the packet, expressed in segments, is distributed between 1 and T with probability $f_{ic}(s)$. Thus, from state 0, the process can jump to any state s between 1 and T that corresponds to the number of segments in the arriving packet. Subsequently, the process moves from state s to $s - 1$ until the last segment is received (when $s = 1$). At that moment, the state of the arrival process will change to either state $s = 0$ (if the process becomes idle), or to state $s, 1 \leq s \leq T$, if another packet arrives. The latter transition models a back-to-back transmission of packets by the user.

For the arrival process to queue $(i, c), i = 1, \ldots, N, c = 1, \ldots, C$, the transition probability matrix $\boldsymbol{Q}_{ic}$ is

$$\boldsymbol{Q}_{ic} = \begin{bmatrix} q_{ic}^{(0,0)} & q_{ic}^{(0,1)} & q_{ic}^{(0,2)} & \ldots & q_{ic}^{(0,T-1)} & q_{ic}^{(0,T)} \\ q_{ic}^{(1,0)} & q_{ic}^{(1,1)} & q_{ic}^{(1,2)} & \ldots & q_{ic}^{(1,T-1)} & q_{ic}^{(1,T)} \\ 0 & q_{ic}^{(2,1)} = 1.0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & q_{ic}^{(3,2)} = 1.0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & q_{ic}^{(T,T-1)} = 1.0 & 0 \end{bmatrix} \tag{1}$$

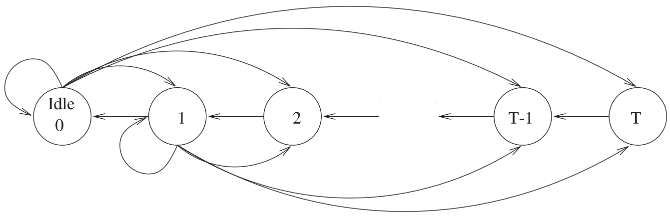


Fig. 3. State machine of the packet arrival process to a transmitting queue.

and the arrival probability matrix $\boldsymbol{A}_{ic}$

$$\boldsymbol{A}_{ic} = \begin{bmatrix} 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ \vdots & \vdots & \ddots & \ddots \\ 0 & \dots & 0 & 1 \end{bmatrix} \tag{2}$$

where $q_{ic}^{(k,l)}, k, l = 0, \dots, T$, is the probability that the process will make a transition to state l, given that it is currently at state k. Obviously, $\sum_l q_{ic}^{(k,l)} = 1, \forall\, k$. Transitions between states of $\boldsymbol{Q}_{ic}$ occur only at the boundaries of *arrival* slots (recall that the segment transmission time at the user rate is equal to an arrival slot).

The transition rates, $q_{i,c}^{(\cdot,\cdot)}$, can be derived from the known (or easily measurable) parameters of the arriving IP traffic stream as follows. Let ρ_{ic} denote the probability that an arrival slot will contain a segment for transmitting queue (i, c). Let also g denote the mean inter-packet gap. Therefore,

$$\rho_{ic} = \frac{E(\text{packet size})}{E(\text{packet size}) + E(\text{inter-packet gap})} = \frac{\sum_{s=1}^{T} s f_{ic}(s)}{\sum_{s=1}^{T} s f_{ic}(s) + g}. \tag{3}$$

Since the length of the inter-packet gap is assumed to be geometrically distributed with a mean of g, it can be easily seen that the transitions at each slot within the gap (and after the last slot of the preceding packet) are governed by a Bernoulli distribution. Denote the parameter for the Bernoulli distribution as q, such that $g = 1/(1-q)$. Based on our description of the arrival process, we have that $q = q_{ic}^{(0,0)} = q_{ic}^{(1,0)}$. The remainder of the probability mass $(1-q)$ may then be distributed according to the (known or measured) packet-length distribution, $f_{ic}(s)$; in other words, $q_{ic}^{(0,s)} = q_{ic}^{(1,s)} = (1-q)\, f_{ic}(s)\ \ \forall\, s > 0$.

3. Queueing analysis

In this section we analyze the queueing network described in Section 2 and shown in Fig. 1. We obtain the queue-length distribution in the transmitting queues (i, c), $i = 1, \dots, N$, $c = 1, \dots, C$, and the receiving queues j, $j = 1, \ldots, N$, from which performance measures such as the packet-loss and the segment-loss probabilities can be obtained.

3.1. Transmitting side analysis

Each transmitting queue (i, c) is served by a single wavelength λ_c. This wavelength, in fact, is shared by all transmitting queues $(l, c), l = 1, \ldots, N$. Within each frame, queue (i, c) is only served during a_{ic} service slots, and it is not served during the remaining service slots of the frame. The actual service slots allocated contiguously to queue (i, c) are determined by the transmission schedule. We define $v_{ic}(x)$ as the number of contiguous service slots allocated to transmitting queue (i, c) during arrival slot x. We then have that

$$\sum_{x=0}^{M-1} v_{ic}(x) = a_{ic}. \tag{4}$$

In view of the above service discipline, each transmitting queue (i, c) can be analyzed separately and independently of all other queues $(l, c), l \neq i$.

Let us now consider transmitting queue (i, c) in isolation. This queue can be analyzed numerically by solving its underlying Markov chain. The state of this chain will include the following: (a) the arrival slot number within a frame; (b) the number of packets and their sizes (in terms of segments) in the queue; (c) the state of the arrival process; and, (d) the state of the server, i.e., which segment of the packet is being served. Obviously, this Markov chain cannot be practically analyzed due to its large dimensionality.

In order to make the analysis of the transmitting queue more manageable, we reduce the most significant source of complexity, i.e., keeping track of the number of packets and their length. Instead, we only keep track of the number of segments in the queue. We also simplify the analysis by not tracking which segment of a packet is currently being served. We then analyze transmitting queue (i, c) by constructing an approximate Markov chain embedded at arrival slot boundaries. The state of this Markov chain consists of the tuple (x, y, z), where

- x represents the arrival slot number within a frame ($x = 0, 1, \ldots, M - 1$),
- y indicates the number of segments in the transmitting queue ($y = 0, 1, \ldots, B_{ic}^{(\mathrm{in})}$), and
- z indicates the state of the arrival process to this queue ($z = -(T - 1), \ldots, -1, 0, 1, \ldots, T$).

We note that the definition of z does not follow exactly the process shown in Fig. 3. Instead, we use a modified process in order to account for a discarded packet which is partially enqueued. Specifically, consider a segment arriving to find a full queue. In this case, the following take place: (a) the arriving segment is dropped, (b) other segments of the same packet which are already enqueued are discarded, and (c) the remaining segments of the packet which have not yet been received are also discarded. In order to account for the last possibility, the state description of the arrival process is augmented by an additional set of states, as shown in Fig. 4. The states identified by non-negative labels in Fig. 4 are identical to the corresponding states of Fig. 3. The states identified by negative labels, on the other hand, indicate that a segment from a discarded packet is being received (and, hence, the segment is discarded). Specifically, if the arrival process is in state z, $2 \leq z \leq T$, and the queue discards an arriving segment as described, the process will transition to state $-(z - 1)$, not state $z - 1$.

The order in which events occur in the Markov chain is as follows. The service (i.e., transmission) completion of a segment occurs at an instant just before the end of a service slot. An arrival may occur

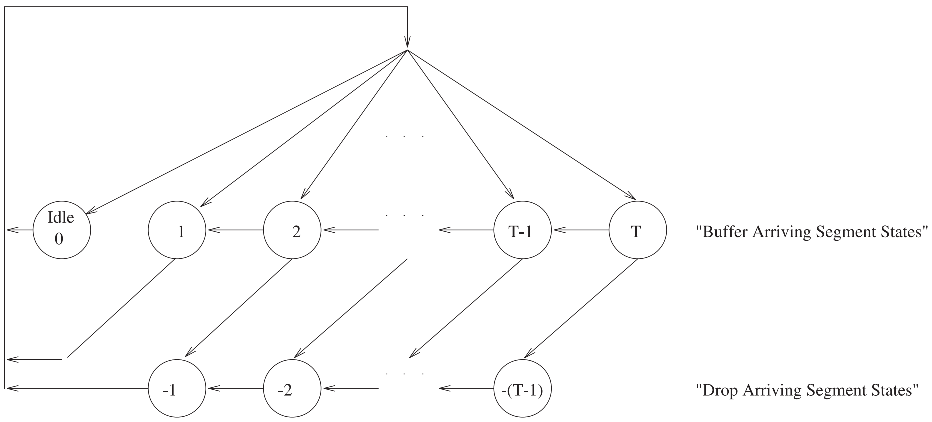


Fig. 4. State machine for arrival process accounting for buffering and dropping of packet segments.

at an instant just before the end of an arrival slot, but after the service completion instant of a service slot whose end is aligned with the end of an arrival slot. The arrival process may make a state transition immediately after the arrival instant. Finally, the Markov chain is observed at the boundary of each arrival slot, *after* the state transition of the arrival process. The order of these events is shown in Fig. 5b. The transition probabilities out of state (x, y, z) are given in Table 1. We note that $\oplus$ denotes modulo-M addition, where M is the number of arrival slots per frame; also, $\boldsymbol{I}_{f(x)}$ is an indicator function which is equal to 1 if the boolean condition $f(x)$ is true, and it is 0 otherwise.

From Table 1 we note that the next state after (x, y, z) always has an arrival slot number equal to $x \oplus 1$. In the first row of Table 1, we assume that the arrival process makes a transition from state z to state z' (from Eq. (1), this event has a probability $q_i^{(zz')}$ of occurring), and a segment arrives and is

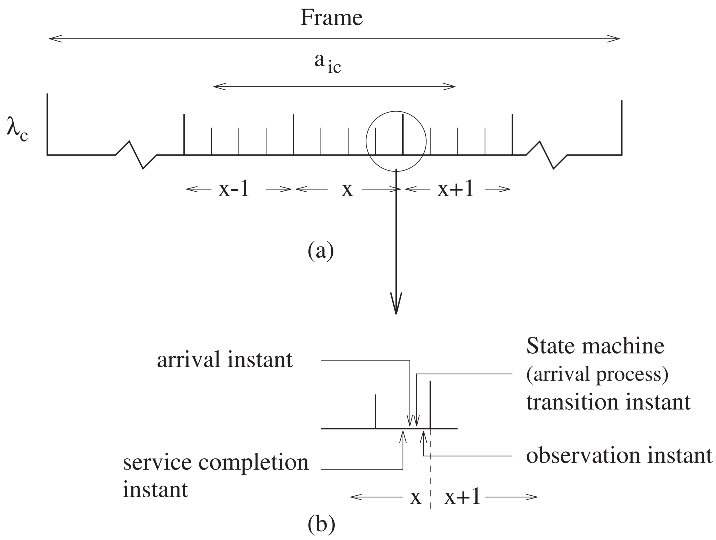


Fig. 5. (a) Service period of node i on channel λ_c. (b) Detail showing the relationship among service completion, arrival, arrival process state transition, and observation instants within a service slot and an arrival slot.

Table 1
Transition probabilities out of state (x, y, z) of the Markov chain

Current state	Next state	Transition probability
(x, y, z)	$(x \oplus 1, \max\{0, y - v_{ic}(x \oplus 1)\} + 1, z')$	$q_{ic}^{(zz')} \boldsymbol{I}_{v_{ic}(x)>0 \text{ or } y<B_{ic}} \times \boldsymbol{I}_{[z>1 \text{ and } z'=(z-1)] \text{ or } (z=1)}$
(x, y, z)	$(x \oplus 1, \max\{0, y - v_{ic}(x \oplus 1)\}, z')$	$q_{ic}^{(\lvert z\rvert\lvert z'\rvert)} \times \boldsymbol{I}_{(z=0,-1) \text{ or } [(z<-1) \text{ and } (z'=z+1)]}$
(x, y, z)	$(x \oplus 1, B_{ic} - s, z')$	$f_{ic}(s+z) q_{ic}^{(z\lvert z'\rvert)} \boldsymbol{I}_{v_{ic}(x)=0 \text{ and } y=B_{ic}} \times \boldsymbol{I}_{[z>1 \text{ and } z'=-(z-1)] \text{ or } (z=1 \text{ and } z'\geq 0)}$

buffered by the queue. This event can only occur if z' is positive (see Fig. 4) *and* either $v_{ic}(x) > 0$ or $y < B_{ic}$. The latter conditions are imposed to ensure that the new queue length will not exceed the capacity $B_{ic}^{(\text{in})}$ of the transmitting queue[4]. This arriving segment cannot be serviced during this slot, and has to be added to the queue. Since at most $v_{ic}(x \oplus 1)$ segments are serviced during arrival slot $x \oplus 1$, and since exactly one segment arrives, the queue length at the end of the slot is equal to $\max\{0, y - v_{ic}(x \oplus 1) + 1\}$.

In the second row of Table 1, we assume that the arrival process makes a transition from state z to state z' such that the arriving segment is not enqueued. This event will occur unconditionally only if the buffer has already overflowed or the source is idle (i.e., $z \leq 0$, also refer to the arrival probability matrix in Eq. (2)). Again, at most $v_{ic}(x \oplus 1)$ segments are serviced during arrival slot $x \oplus 1$, resulting in the queue length at the end of the slot being $\max\{0, y - v_{ic}(x \oplus 1)\}$.

Finally, the third row of Table 1 assumes that a segment arrives to the transmitting queue causing it to overflow. This event occurs if and only if the queue has not yet overflowed, the buffer is full, and the buffer receives no service during the arrival slot (i.e., $y = B_{ic}$, $z > 0$, and $v_{ic}(x) = 0$). In this case, the arrival process transitions to the appropriate state reflecting that future segments of this packet are to be dropped. Also, the queue will lose s segments of the arriving packet which have already been buffered, provided that the packet size was $s + z$ segments.

The probability transition matrix of this Markov chain has the following block form:

$$\boldsymbol{S}_{ic} = \begin{bmatrix} 0 & \boldsymbol{R}_{ic}(0) & 0 & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{R}_{ic}(1) & 0 & \dots & 0 \\ 0 & 0 & 0 & \boldsymbol{R}_{ic}(2) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \boldsymbol{R}_{ic}(M-2) \\ \boldsymbol{R}_{ic}(M-1) & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ M-2 \\ M-1 \end{matrix} \qquad (5)$$

This block form is due to the fact that at each transition instant (i.e., at each arrival slot boundary), the random variable x changes to $x \oplus 1$. Changes in the other two random variables, y and z, are governed by the matrices $\boldsymbol{R}_{ic}(x)$. There are M different $\boldsymbol{R}_{ic}$ matrices, one for each arrival slot x in the frame.

[4] Due to the nature of the system, segment loss can only occur if both of these conditions are not true and an arrival occurs.

Let us define matrices $\boldsymbol{X}_{ic}(\cdot \mid x, y)$ and $\boldsymbol{Y}_{ic}$ as:

$$\boldsymbol{X}_{ic}(\cdot\,|\,x, y) = \tilde{\boldsymbol{A}}_{ic}\,\tilde{\boldsymbol{Q}}_{ic}\left(\boldsymbol{I}_{v_{ic}(x)=0 \quad \text{and} \quad y=B_{ic}}\right) \text{ and } \boldsymbol{Y}_{ic} = (\boldsymbol{I} - \tilde{\boldsymbol{A}}_{ic})\,\tilde{\boldsymbol{Q}}_{ic}(0), \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix and $\tilde{\boldsymbol{A}}_{ic}$ and $\tilde{\boldsymbol{Q}}_{ic}(\cdot)$ are given in Eqs. (7)–(9). Matrix $\boldsymbol{X}_{ic}(\cdot \mid x, y)$ (respectively, $\boldsymbol{Y}_{ic}$) is the arrival process' state transition probability matrix given that an arrival is (respectively, is not) accepted by the transmitting queue. For the sake of clarity, $\boldsymbol{X}_{ic}(\cdot \mid x, y)$ will be denoted simply as $\boldsymbol{X}_{ic}(\cdot)$ for the remainder of the paper.

$$\tilde{\boldsymbol{A}}_{ic} = \begin{array}{c|ccc|ccc|} & -(T-1) & \cdots & -1 & 0 & \cdots & T \\ \hline -(T-1) & & & & & & \\ \vdots & & 0 & & & 0 & \\ -1 & & & & & & \\ \hline 0 & & & & & & \\ \vdots & & 0 & & & \boldsymbol{A}_{ic} & \\ T & & & & & & \end{array}, \tag{7}$$

$$\tilde{\boldsymbol{Q}}_{ic}(0) = \begin{array}{c|ccc|cccc|} \downarrow k \quad l \rightarrow & -(T-1) & \cdots & -1 & 0 & 1 & \cdots & T \\ \hline -(T-1) & & & & & & & \\ \vdots & & q_{ic}^{|k||l|} & & & & 0 & \\ -2 & & & & & & & \\ \hline -1 & & 0 & & & q_{ic}^{|k||l|} & & \\ \hline 0 & & & & & & & \\ \vdots & & 0 & & & \boldsymbol{Q}_{ic} & & \\ T & & & & & & & \end{array}, \tag{8}$$

and

$$\tilde{\boldsymbol{Q}}_{ic}(1) = \begin{array}{c|ccc|c|ccc|} \downarrow k \quad l \rightarrow & -(T-1) & \cdots & -1 & 0 & 1 & \cdots & T \\ \hline -(T-1) & & & & & & & \\ \vdots & & 0 & & 0 & & 0 & \\ 0 & & & & & & & \\ \hline 1 & & 0 & & q_{ic}^{|k||l|} & & q_{ic}^{|k||l|} & \\ \hline 2 & & & & & & & \\ \vdots & & q_{ic}^{|k||l|} & & q_{ic}^{|k||l|} & & 0 & \\ T & & & & & & & \end{array}. \tag{9}$$

Then, the transition matrix $\boldsymbol{R}_{ic}(x)$ associated with arrival slot x can be written as:

$$\boldsymbol{R}_{ic}(x \mid v_{ic}(x \oplus 1) > 0) = \begin{bmatrix} \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 & \dots \\ 0 & \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) & 0 & 0 & 0 & \dots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & \dots \\ 0 & \dots & 0 & \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) & 0 & \dots \end{bmatrix} \begin{matrix} 0 \\ \vdots \\ v_{ic}(x\oplus 1) \\ v_{ic}(x\oplus 1)+1 \\ \vdots \\ B_{ic}^{(\mathrm{in})} \end{matrix} \quad (10)$$

The structure of matrix $\boldsymbol{R}_{ic}(x)$ given in Eq. (10) can be explained as follows. Suppose that the number of segments y in the queue at the end of slot x is at most $v_{ic}(x \oplus 1)$. Since up to $v_{ic}(x \oplus 1)$ segments can be served within slot $x \oplus 1$, the number in the queue at the end of that slot will be 1 or 0, depending on whether an arrival occurred or not. This point is indicated by the transitions in rows 0 through $v_{ic}(x \oplus 1)$ of matrix $\boldsymbol{R}_{ic}(x)$. However, if at the end of slot x we have $y > v_{ic}(x \oplus 1)$, then the number in the queue at the next transition will be $y - v_{ic}(x \oplus 1)$ (plus one if an arrival occurred), as indicated by the transitions in rows $v_{ic}(x \oplus 1) + 1$ through B_{ic} of $\boldsymbol{R}_{ic}(x)$. Of course, y cannot exceed the queue capacity $B_{ic}^{(\mathrm{in})}$. Since the number of service slots $v_{ic}(x \oplus 1)$ depends on the particular slot $x \oplus 1$ within the frame, $\boldsymbol{R}_{ic}(x)$ is a function of x.

Matrix $\boldsymbol{R}_{ic}(x)$ is slightly different when $v_{ic}(x \oplus 1) = 0$; its structure is shown in Eq. (11). In this case, if the state of the transmitting queue is $y = B_{ic}^{(\mathrm{in})}$, not only will a new arrival be discarded, but some number of currently enqueued segments may also be discarded. Let matrix $\boldsymbol{P}_n(\cdot \mid x)$ represent the probability that segment n of an n'-segment packet causes the overflow of the buffer with the arrival process in a given state $z, z > 0$, during arrival slot x. The structure of $\boldsymbol{P}_n(\cdot \mid x)$ may be described simply: $\boldsymbol{P}_n$ is a square matrix with indices on both dimensions running from $-(T-1)$ to T. The matrix may have non-zero values only for rows $z, 0 < z \le (T - n + 1)$. The reason for this boundary is that for an arrival to occur, z must be greater than 0. Also, if exactly n segments (including the currently arriving segment) are to be lost, and the total packet size is bounded by T, z (before the arrival occurred) must be bounded by $T - (n-1)$. Given the arriving traffic description shown in Fig. 4, the complete packet size (in terms of segments) may be inferred exactly as $z + n - 1$, occurring with the probability shown in Eq. (12).

$$\boldsymbol{R}_{ic}(x \mid v_{ic}(x \oplus 1) = 0)$$
$$= \begin{bmatrix} \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 0 & 0 & \boldsymbol{Y}_{ic} & \boldsymbol{X}_{ic}(\cdot) \\ 0 & \dots & 0 & \boldsymbol{P}_T(\cdot)\boldsymbol{X}_{ic}(\cdot) & \dots & \boldsymbol{P}_2(\cdot)\boldsymbol{X}_{ic}(\cdot) & \boldsymbol{Y}_{ic} + \boldsymbol{P}_1(\cdot)\boldsymbol{X}_{ic}(\cdot) \end{bmatrix}, \quad (11)$$

$$\boldsymbol{P}_n(z,z) = \begin{cases} \sum_{k=0,1} \psi_{ic}(k \mid x \ominus n) q_{ic}^{(k,z+n-1)} & \forall z \text{ s.t. } (T-n+1) \geq z > 0, \\ 0.0, & \text{otherwise.} \end{cases} \tag{12}$$

We define $\psi_{ic}(s \mid x)$ as the probability that the arrival process occupies state s immediately before arrival slot x of the frame. The value $\psi_{ic}(s \mid 0)$ may be found by solving for the steady-state occupancy of the arrival process at the boundaries of the schedule. The matrix containing transition probabilities between the boundaries of the repeating schedule's frame (i.e., on either side of M arrival slots), is denoted $\boldsymbol{K}_{ic}$, and is obtained as:

$$\boldsymbol{K}_{ic} = \boldsymbol{Q}_{ic}^{M}. \tag{13}$$

The other values of $\psi(\cdot)$ may be found using Eq. (14) as follows:

$$\psi_{ic}(s \mid x) = \psi_{ic}(s \mid x \ominus 1)\, \boldsymbol{Q}_{ic}. \tag{14}$$

It can be verified that the Markov chain with probability transition matrix $\boldsymbol{S}_{ic}$ in Eq. (5) is irreducible, and therefore a steady-state distribution exists. Transition matrix $\boldsymbol{S}_{ic}$ defines a *p-cyclic* Markov chain, and therefore it can be solved using any of the techniques for p-cyclic Markov chains in [32, ch. 7]. We have used the LU decomposition method to obtain the steady-state probabilities $\pi_{ic}(x, y, z)$.

3.2. Receiving side analysis

As shown in Fig. 1, each receiver j consists of an optical filter, N reassembly queues, and one receiving queue. The receiver filters out segments from the passive star coupler, allowing only those segments transmitted on wavelength $\lambda(j)$, the receive wavelength of node j, to pass. As segments are received, from a particular transmitter, they are buffered in the appropriate reassembly queue until a complete packet is formed. At that instant, the packet is transferred to the receiving queue. If the receiving queue does not have enough space to accommodate the entire packet, the packet is lost. The reassembly queue is large enough to accommodate a packet of maximum length (i.e., it can hold T segments). Thus, no segment can be lost upon arrival at the reassembly queue.

In the following, we use the transmitting queue steady-state probabilities, $\pi_{ic}(x, y, z), i = 1, \ldots, N, c = 1, \ldots, C$, derived in the previous section, to analyze each receiver in isolation. We note that quantities $\pi_{ic}(x, y, z)$ do not provide any information about the size, in segments, of individual packets arriving at the reassembly part of the receivers. Therefore, one of the main tasks of our analysis is to reconstruct the packet sizes from the given information about segments transmitted by transmitting queues. This reconstruction must be performed in such a manner that the resulting packet sizes follow the original packet-length distribution as closely as possible. We also note that this approach is similar to the 'Kleinrock independence approximation' [20], whereby packets are assumed to take on new sizes as they travel through the nodes of a network. Just as the 'Kleinrock independence approximation' made the analysis of packet switched networks tractable, our assumption regarding the independence of packet sizes between transmitting and receiving queues significantly reduces the complexity of the analysis. Nevertheless, it does introduce some problems; these problems and our heuristic solution are discussed later in the section.

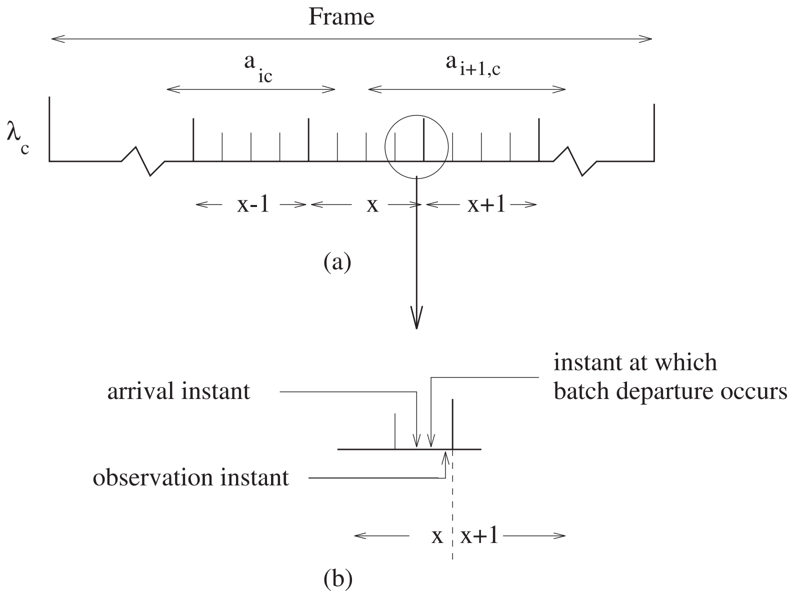


Fig. 6. (a) Arrivals to reassembly queue i of receiver j. (b) Detail showing the relationship of departure, arrival, and observation instants.

3.2.1. Analysis of the reassembly queue

Each of the N reassembly queues of receiver j can be considered in isolation. For each reassembly queue, we define a Markov chain, (x, ζ), where x indicates the arrival slot number within the frame ($x = 0, 1, \ldots, M-1$), and ζ indicates the occupancy of the reassembly queue in segments ($\zeta = 0, \ldots, T$). The ordering of the events is shown in Fig. 6, and the transition probabilities are shown in Table 2 [5]. As we can see in Fig. 6b, at the end of an arrival slot a batch departure may occur. This batch departure corresponds to the accumulation of a complete packet in the reassembly queue, which is then transferred to the receiving queue (refer also to Fig. 1).

The term $\mathcal{P}(\cdot)$ in Table 2 is the probability of receiving ζ_{add} segments from the appropriate

Table 2
Transition probabilities out of state (x, ζ) of the Markov chain for reassembly queue i of receiver j

Current state	Next state	Transition probability
(x, ζ)	$(x \oplus 1, \zeta')$	$\sum_{\zeta_{\text{add}}=\max(\zeta'-\zeta,0)}^{v_{ic}(x\oplus 1)} [\mathcal{P}(\zeta_{\text{add}} \mid x) \times \mathcal{F}(\zeta' \mid \zeta, \zeta_{\text{add}}, x)]$

[5] We note that, for this analysis, we impose the restriction that the minimum packet size must be greater than or equal to m segments, where m is the number of service slots that fully lie within one arrival slot. Thus, we assume at most one packet can be accumulated during an arrival slot. Allowing for multiple packets to be accumulated at a reassembly queue during an arrival slot would simply add a layer of complexity in the following discussion, which would only detract from our analysis.

transmitting queue during slot $x \oplus 1$. We have that

$$\mathcal{P}(\zeta_{\text{add}} \mid x) = \begin{cases} 1.0, & v_{ic}(x \oplus 1) = 0 \text{ and } \zeta_{\text{add}} = 0, \\ 0.0, & v_{ic}(x \oplus 1) = 0 \text{ and } \zeta_{\text{add}} \neq 0, \\ M \sum_z \pi_{ic}(x, \zeta_{\text{add}}, z), & \zeta_{\text{add}} < v_{ic}(x \oplus 1) \text{ and } v_{ic}(x \oplus 1) > 0, \\ M \sum_{y'=v_{ic}(x\oplus 1)}^{B_{ic}} \sum_z \pi_{ic}(x, y', z), & \zeta_{\text{add}} = v_{ic}(x \oplus 1). \text{ and } v_{ic}(x \oplus 1) > 0. \end{cases} \tag{15}$$

We note that $M \sum_z \pi_{ic}(x, y, z)$ is the conditional probability of having y segments in transmitting queue (i, c) given slot x regardless of the state of the arrival process. The first two expressions in Eq. (15) simply state that no segments can be received if transmitting queue (i, c) is not served during slot $x \oplus 1$. The third expression states that, if the transmitting queue is served for $v_{ic}(x \oplus 1)$ slots, then $\zeta_{\text{add}} < v_{ic}(x \oplus 1)$ segments will be received only if the transmitting queue had exactly ζ_{add} segments at the beginning of the slot. Finally, exactly $v_{ic}(x \oplus 1)$ segments will be received if the transmitting queue had at least that many segments, as the last expression indicates.

The term $\mathcal{F}(\cdot)$ in Table 2 is the probability that the system will be in state ζ' given that it had ζ segments in it and it received ζ_{add} segments from a transmitting queue. In other words, it is the conditional probability that a packet consisting of a number $\max(0, \zeta - \zeta' + \zeta_{\text{add}})$ of segments was transferred to the receiving queue. Then,

$$\mathcal{F}(\zeta' \mid \zeta, \zeta_{\text{add}}, x) = \begin{cases} F_{ic}(\zeta' \mid \zeta), & \zeta_{\text{add}} \leq \zeta' - \zeta \text{ and } \zeta_{\text{add}} = v_{ic}(x \oplus 1), \\ f_{ic}(\zeta - \zeta' + \zeta_{\text{add}} \mid \zeta), & \text{otherwise}. \end{cases} \tag{16}$$

In Eq. (16), $f_{ic}(s' \mid s)$ is defined to be the conditional probability that the packet size is equal to s' segments, given that it is greater than s segments:

$$f_{ic}(s' \mid s) = \begin{cases} \dfrac{f_{ic}(s')}{F_{ic}(s)}, & s' > s, \\ 0.0, & \text{otherwise}. \end{cases} \tag{17}$$

Also in Eq. (16), $F_{ic}(s \mid s')$ is the conditional cumulative probability distribution of the packet size in terms of segments, given that the packet size is greater than s', i.e.,

$$F_{ic}(s) = \sum_{\dot{s}=s+1}^{T} f_{ic}(\dot{s}), \tag{18}$$

$$F_{ic}(s \mid s') = \begin{cases} \dfrac{F_{ic}(s)}{F_{ic}(s')}, & s > s', \\ 0.0, & \text{otherwise}. \end{cases} \tag{19}$$

Let $\Phi_{ij}(x, \zeta)$ denote the steady-state probability that reassembly queue (i, j) has ζ segments at the end of slot x. The transition probability matrix for each reassembly queue (i, j) can be constructed using the above analysis, and the occupancy probability $\Phi_{ij}(x,\zeta)$ may be determined using the LU decomposition method [32].

We note, however, that our model does not keep track of the actual number of segments in each packet once the segments are buffered at a transmitting queue. Thus, in order to compute $\Phi_{ij}(x,\zeta)$, we used a probabilistic model for the reassembly of segments into packets which accounts for the original packet-size distribution through expression Eq. (16). Let us consider the following example. Let us assume that users generate packets which are always between four and seven segments long. Now, let us consider a packet, originally six segments long, that is forwarded one segment at a time to a reassembly queue. In a real system, the reassembly queue will wait for all six segments, and it will then reconstruct the original packet and forward it to the receiving queue. In our model, on the other hand, expression Eq. (16) will evaluate the probability that the arriving segments belong to a packet of size four, five, six, or seven segments long, and all these probabilities will contribute to the occupancy distribution $\Phi_{ij}(x,\zeta)$. Let us consider the scenario where, after the fifth segment has arrived, the reassembly queue forwards the five segments as a complete packet to the receiving queue. When the last segment arrives, it will remain in the reassembly queue until it becomes part of another packet. In a lightly loaded system, this packet will have to wait for a long time before more segments arrive and it can be considered as part of another packet. The fact that under our model this scenario has a non-zero probability of occurring, may affect the accuracy of the estimated buffer occupancy distribution $\Phi_{ij}(x,\zeta)$ under low loads. In the next section, we introduce a heuristic that significantly improves the accuracy of $\Phi_{ij}(x,\zeta)$.

3.2.2. Analysis of a receiving queue

As in the previous section, we obtain the queue-length distribution of receiving queue j at arrival slot boundaries. Let (x, w) be the state associated with receiving queue j, where x indicates the arrival slot number within the frame ($x = 0, 1, \ldots, M-1$), and w indicates the number of segments at the queue ($w = 0, 1, \ldots, B_j^{(\text{out})}$). The ordering of events is shown in Fig. 7.

Observe now that (a) at each state transition, x advances by one (modulo-M), (b) exactly one segment departs from the queue as long as the queue is not empty, (c) a number $0 \le s \le T$ of segments may be transmitted from each of the relevant reassembly queues to receiving queue j within arrival slot $x \oplus 1$, and (d) the queue capacity is $B_j^{(\text{out})}$. Based on the first item above, it can be easily seen that the transition matrix $\boldsymbol{T}_j$ of this Markov chain has the same structure as matrix $\boldsymbol{S}_{ic}$ given by Eq. (5). We have that:

$$\boldsymbol{T}_j = \begin{bmatrix} 0 & \boldsymbol{U}_j(0) & 0 & 0 & \dots & 0 \\ 0 & 0 & \boldsymbol{U}_j(1) & 0 & \dots & 0 \\ 0 & 0 & 0 & \boldsymbol{U}_j(2) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \boldsymbol{U}_j(M-2) \\ \boldsymbol{U}_j(M-1) & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{matrix} {\scriptstyle 0} \\ {\scriptstyle 1} \\ {\scriptstyle 2} \\ \vdots \\ {\scriptstyle M-2} \\ {\scriptstyle M-1} \end{matrix} \tag{20}$$

The construction of the matrices $\boldsymbol{U}_j(x)$ is somewhat complicated. This is mainly due to the fact that during an arrival slot x, more than one transmitting queue can transmit on the same channel. In this case, determining which packets are accepted by a receiving queue which is close to being full may be quite involved. The construction of a matrix $\boldsymbol{U}_j(x)$ is summarized in the algorithm given below. We note that the

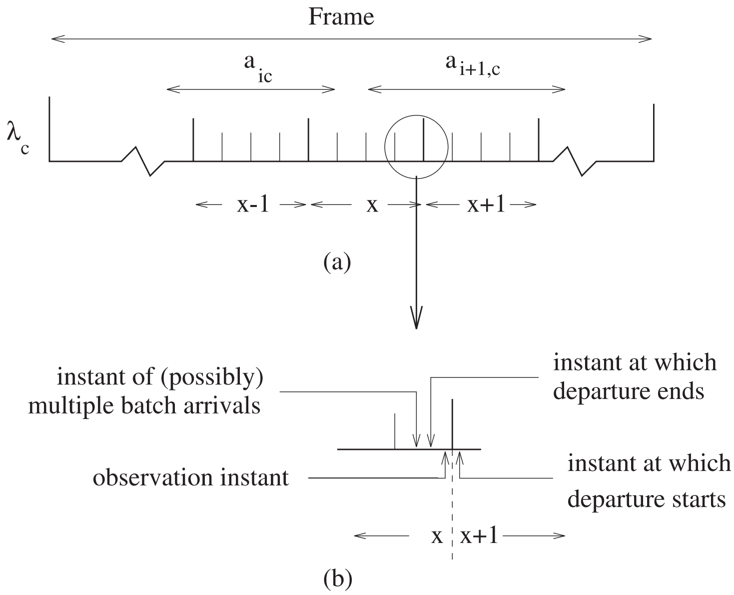


Fig. 7. (a) Arrivals to receiving queue j from reassembly queues i and $i+1$. (b) Detail showing the relationship of departure, arrival, and observation instants.

elements of matrix $\boldsymbol{U}_j(x)$ are denoted as $\boldsymbol{U}_j(x,k,l)$, where k is the row index and l is the column index.

$\boldsymbol{U}_j(x) \leftarrow [\mathbf{0}]$
loop: $\forall\, w \in \{0, 1, \dots, B_j\}$
  $\mathcal{I}_j(x) \leftarrow$ an ordered set of the transmitting queues serviced by channel λ_c
    during slot x with cardinality $|\mathcal{I}_j(x)|$
  loop: $\forall\, s \in \{\{\overbrace{0, 0, \dots, 0}^{|\mathcal{I}_j(x)|}\}, \{0, 0, \dots, 1\}, \dots, \{T, T, \dots, T\}\}$
    $w' \leftarrow w$
    loop: $\forall\, i \in \{1, \dots, |\mathcal{I}_j(x)|\}$
      if $\{w' + s_i \le B_j\} \Rightarrow \{w' \leftarrow w' + s_i\}$
    if $\{w > 0\} \Rightarrow \{w' \leftarrow w' - 1\}$
    $\boldsymbol{U}_j(x, w, w') \leftarrow \boldsymbol{U}_j(x, w, w') + \prod_{i=1}^{|\mathcal{I}_j(x)|} L_i(s_i \mid x)$

In the preceding algorithm, $L_i(s_i \mid x)$ is the probability that a packet consisting of s_i segments is transmitted during slot x from reassembly queue i to receiving queue j [6]. We have that:

$$L_i(s_i \mid x) = r_{cj}\, f_{ic}(s_i) \times p_{ij}(\text{packet generated} \mid x) \times \prod_{x' \in \mathcal{A}_{ij}^{(s_i)}(x)} (1.0 - p_{ij}(\text{packet generated} \mid x')) \tag{21}$$

[6] Since in most cases only one or two transmitting queues will transmit to the same channel within an arrival slot (refer also to Fig. 5), and since a packet can only be completed in a reassembly queue while the transmitting queue is transmitting over the given channel, the dimension of the vector $\boldsymbol{s}$ will generally be only one or two. Thus, this loop can be executed very fast, in spite of the exponential time implied by the general form presented.

where

$$p_{ij}(\text{packet generated} \mid x)$$
$$= M \sum_{\zeta=0}^{T-1} \Phi_{ij}(x,\zeta) \times \sum_{\zeta_{\text{add}}=1}^{v_{ic}(x\oplus 1)} \left\{ \sum_{\zeta'=0}^{\zeta_{\text{add}}-1} \mathcal{P}(\zeta_{\text{add}} \mid x)\mathcal{F}(\zeta - \zeta' + \zeta_{\text{add}} \mid \zeta, \zeta_{\text{add}}, x) \right\} \tag{22}$$

and r_{cj} is the probability that a packet transmitted on channel c is intended for node j.

Eq. (21) for calculating $L_i(s_i \mid x)$ is derived so as to reduce the error introduced when reconstructing a packet in the reassembly queue. In Eq. (21), $\mathcal{A}_{ij}^{(s_i)}(x)$ denotes the set of arrival slots during which reassembly queue (i, j) may receive segments. This set consists of the $s_i - 1$ service slots prior to arrival slot x. Using this set, we are able to take into account an appropriate minimal packet inter-arrival time, in service slots, given the size s_i of the packet and the time x of arrival within the schedule. We note that, depending on the values involved, the set $\mathcal{A}_{ij}^{(s_i)}(x)$ may contain more than M arrival slots. In this case, the set will span multiple frames, and a particular slot may be encountered in the set multiple times, each time in a different frame.

Finally, we solve for the steady-state occupancy probability of receiving queue j during a slot x, $\pi_j(x, w)$, using the LU decomposition method.

3.3. Summary of the decomposition algorithm

Below we summarize the approximation described above.

(1) Given each arrival process, defined by $\boldsymbol{A}_{ic}$ and $\boldsymbol{Q}_{ic}$, formulate $\tilde{\boldsymbol{A}}_{ic}$, $\tilde{\boldsymbol{Q}}_{ic}(0)$, and $\tilde{\boldsymbol{Q}}_{ic}(1)$ per Eqs. (7)–(9), respectively. Additionally, compose the matrices $\boldsymbol{P}_n$ corresponding to each arrival process per Eq. (12).
(2) For each arrival slot x, use the schedule and Eq. (4) to compute the quantities $v_{ic}(x), i = 1, \ldots, N, c = 1, \ldots, C$.
(3) For each transmitting queue (i, c), construct the transition probability matrix $\boldsymbol{S}_{ic}$ from Eqs. (1), (2), (5), (6) and (10). Solve this matrix for $\pi_{ic}(x, y, z)$.
(4) For each reassembly queue (i, j), use $\pi_{ic}(x, y, z)$ and Eqs. (15) and (16) to build its transition probability matrix. Solve the matrix to obtain $\Phi_{ij}(x, \zeta)$, the steady-state probability that reassembly queue (i, j) has ζ segments at the end of slot x.
(5) For each receiving queue $j \in \mathcal{R}_c$, use $\pi_{ic}(x, y, z)$, $\Phi_{ij}(x, \zeta)$, and Eq. (21) to construct the transition matrix $\boldsymbol{T}_j$ given by Eq. (20). Solve the matrix to obtain $\pi_j(x, w)$, the steady-state probability that receiving queue j has w cells at the end of slot x.

4. Loss probabilities

We now use the queue-length distributions $\pi_{ic}(x, y, z)$, $\Phi_{ij}(x, \zeta)$, and $\pi_j(x, w)$, derived in the previous section, to obtain the segment and packet loss probability at the transmitting and receiving queues.

4.1. Segment and packet loss probability at a transmitting queue

Let Ω_{ic} be the probability that a packet arriving to transmitting queue (i, c) will be lost. Ω_{ic} can be expressed as:

$$\Omega_{ic} = \frac{E[\text{number of packets lost per frame at transmitting queue } (i, c)]}{E[\text{number of arrivals per frame at transmitting queue } (i, c)]}. \tag{23}$$

For the denominator, we have that:

$$E[\text{number of arrivals per frame at transmitting queue } (i, c)] = \sum_{x=0}^{M-1} \left(\sum_{y} \pi_{ic}(x, y, z = 1) + \sum_{y} \pi_{ic}(x, y, z = -1) \right). \tag{24}$$

To calculate the numerator we observe that all packets must begin their transmission periods in one of the states for which $z \geq 1$. Thus, we have that:

$$E[\text{number of packets lost per frame at transmitting queue } (i, c)] = \sum_{x=0}^{M-1} \sum_{y} \pi_{ic}(x, y, z = -1) + \sum_{x:\, v_{ic}(x)=0} \pi_{ic}(x, y = B_{ic}, z = 1)\, f_{ic}(1). \tag{25}$$

Using this same approach, the segment loss probability, ω_{ic}, may also be calculated:

$$\omega_{ic} = \frac{E[\text{number of segments lost per frame at transmitting queue } (i, c)]}{E[\text{number of arriving segments per frame at transmitting queue } (i, c)]} = \frac{\sum_{\forall\, x,y;\, \forall\, z<0} \pi_{ic}(x, y, z) + \sum_{\forall\, x;\, \forall\, z>0} \pi_{ic}(x, y = B_{ic}, z)}{\sum_{\forall\, x,y;\, \forall\, z\neq 0} \pi_{ic}(x, y, z)}. \tag{26}$$

4.2. Segment and packet loss probability at a receiving queue

The packet and segment loss probabilities at a receiving port are more complicated to calculate, since we may have multiple packet arrivals (each from a different transmitting queue) to the given receiving queue within a single arrival slot. Additionally, the order of the arrivals must be accounted in order to determine which packets are lost. The packet and segment loss probabilities, $\Omega_j(x)$ and $\omega_j(x)$, are not easily expressed in closed form, but they can be calculated using a slightly modified version of the algorithm for calculating $\boldsymbol{U}_j(x)$ that was presented in Section 3.2.2.

5. Numerical results

We now discuss the accuracy of our analysis by applying the approximation algorithm to a network with eight nodes, and comparing the loss probabilities to simulation results. We consider four different

Table 3
Packet length distributions for each arrival process

Packet length	Percentage of all packets			
	Arrival Process 1	Arrival Process 2	Arrival Process 3	Arrival Process 4
5 segments	100%	25%	16.7%	0%
6 segments	0%	25%	16.7%	0%
7 segments	0%	25%	16.7%	0%
8 segments	0%	25%	50%	100%
Mean Sgmts/Pkt	5	6.5	7	8

Table 4
Channel sharing for $C = 2, 3$

	$C = 2$	$C = 3$
$\mathcal{R}_1$	$\{1, 3, 5, 7\}$	$\{1, 4, 7\}$
$\mathcal{R}_2$	$\{2, 4, 6, 8\}$	$\{2, 5, 8\}$
$\mathcal{R}_3$		$\{3, 6\}$

packet-length distributions (and, hence, arrival processes). In varying these four distributions, the mass of the packet-length distribution was shifted from favoring short packets to favoring long ones. The four packet-length distributions are shown in Table 3. The mean utilization of the arrival process to each transmitting queue remained fixed at 10% for all experiments.

In the numerical results presented below, the number of channels in the network is either two or three. The eight receivers were assigned to the channels using a round-robin assignment algorithm. The receiver assignments for each channel is shown in Table 4; $\mathcal{R}_c$ is a set that contains the receivers assigned to channel λ_c. Finally, for all transmitting and receiving queues we have assumed that $B_{ic}^{(\text{in})} = B_j^{(\text{out})} = B$. The buffer length B was varied from 10 to 20 segments.

The quantities a_{ic}, i.e., the number of service slots allocated to node i onto channel λ_c per frame, were fixed to be as close to 0.5 arrival slots as possible. Recall that, while the length of an arrival slot is independent of C and is taken as our unit of time, the length of a service slot depends on the number of channels. In cases in which this value was not an integral number of service slots, the value a_{ic} was rounded up to ensure that every queue was granted at least 0.5 arrival slots of service during each frame (i.e., $a_{ic} = \lceil N/2C \rceil\ \forall\, i, c$). We have assumed that the time it takes a laser to tune from one channel to another is equal to one arrival slot. The schedules which were used in these experiments are shown in Fig. 8.

We only present results for segment loss probability because the packet loss probability is identical to segment loss probability. The number of segments arriving to (respectively lost by) a transmitting queue is related to the number of packets arriving to (respectively lost by) the same queue by a factor of the average number of segments per packet. Since the average packet size is constant for the duration of a single experiment and packet arrivals (and their lengths) are independent of the occupancy level of the transmitting queue, packet and segment loss probabilities are essentially equal.

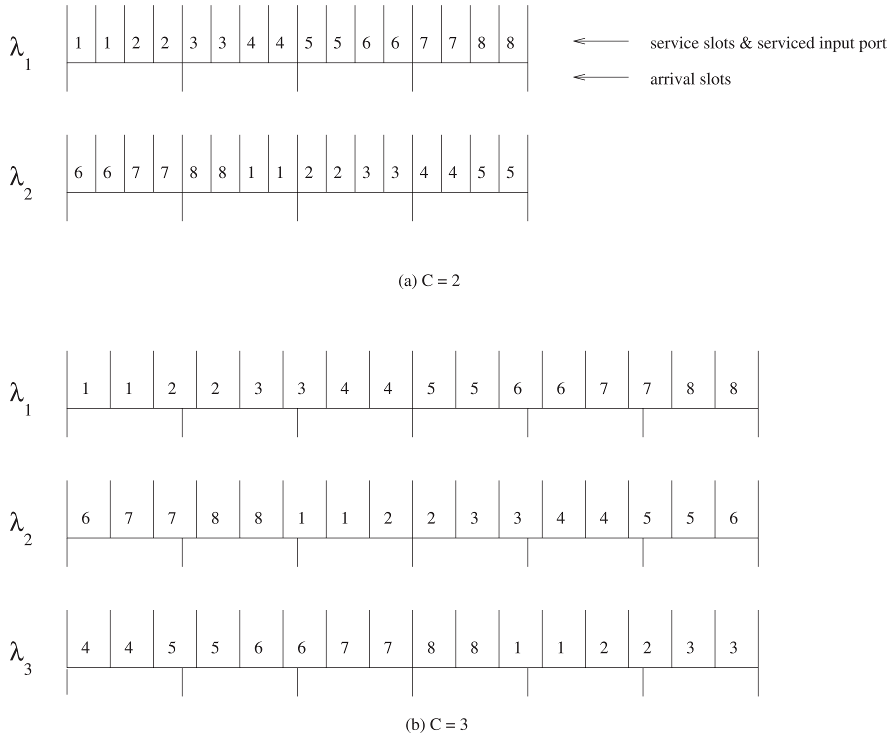


Fig. 8. Channel sharing schedules for $C = 2, 3$.

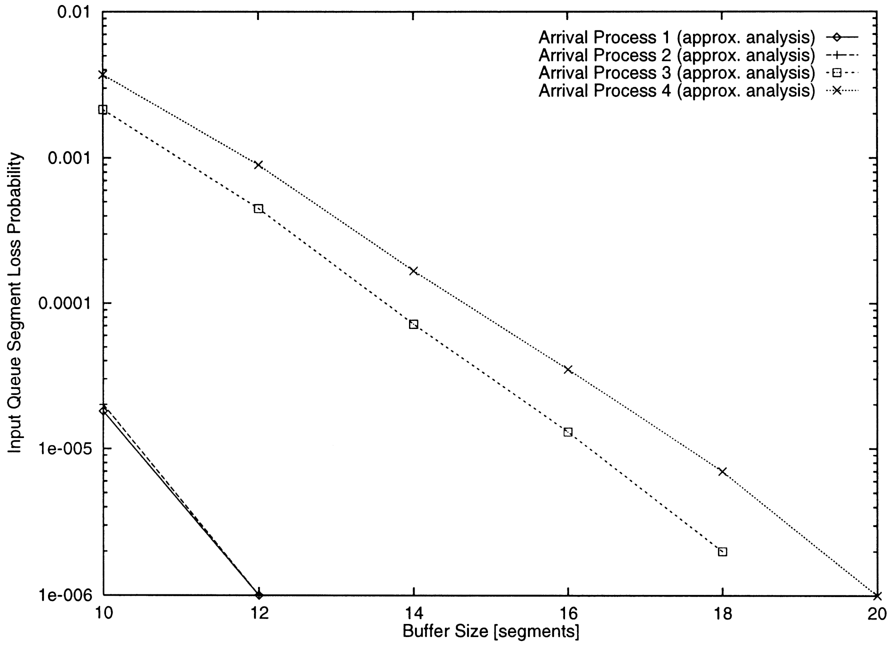


Fig. 9. Transmitting queue segment loss probability $\omega_{1,1}$ for $C = 2$.

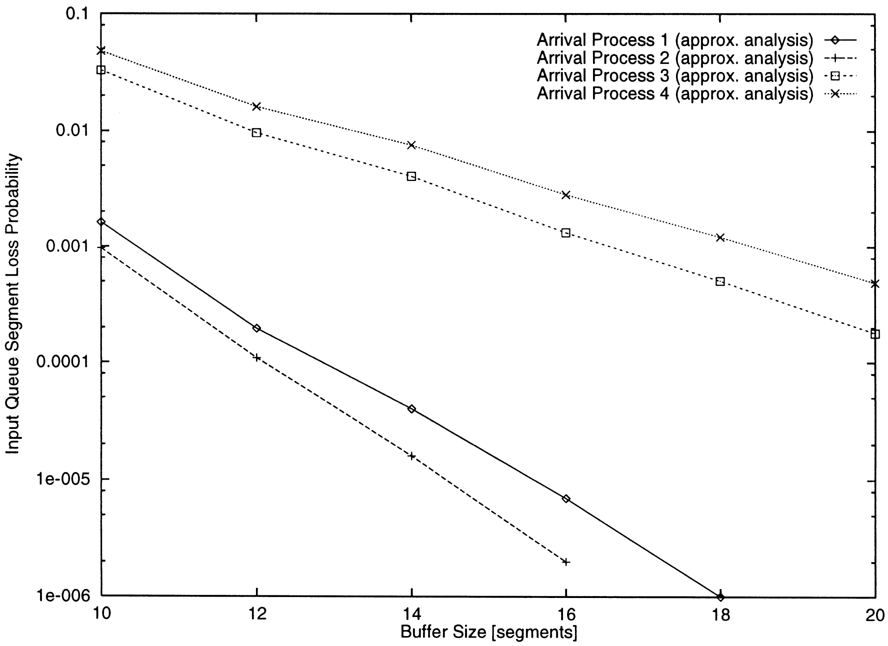


Fig. 10. Transmitting queue segment loss probability $\omega_{1,1}$ for $C = 3$.

Figs. 9 and 10 show the segment loss probabilities (derived analytically) for transmitting queue (1,1) and the four different arrival processes, as a function of buffer size B. Fig. 9 shows results for a two-wavelength switch, while Fig. 10 shows results for a three-wavelength switch. We present results only for the queue of node 1 which corresponds to the first wavelength, λ_1, as it is representative of the other transmitting queues. Note that, in Fig. 9 for Arrival Process 3 and $B = 20$, and in Fig. 10 for Arrival Process 2 and $B = 18$, the loss probabilities are not shown. For these parameters, our analyses reported loss probabilities significantly smaller than 1×10^{-6}; to improve the readability of the graphs, these values are omitted.

As one would intuitively expect, loss measures generally increased with average packet sizes and decreased as more buffer space was available. However, in Fig. 10, we observe that the loss measures for Arrival Process 2 are lower than those of Arrival Process 1. This behavior is the result of two factors which act synergistically: (a) the change in the inter-packet gap distribution between the two arrival processes, and (b) the worst-case instantaneous impact of a packet's arrival on buffer occupancy.

First, the fact that the mean segment arrival rates are constant at 10%, means that the occupancy probabilities for each arrival process are different. As Eq. (3) implies, the mean utilization (a known, fixed value in our experiments) affects parameters $q_{ic}^{(0,0)}$ and $q_{ic}^{(1,0)}$ and the mean inter-packet gaps (see Table 3). By increasing the mean inter-packet gap length, we are allowing for more service to occur between packet arrivals which, in turn, reduces the queues' occupancy levels.

Second, consider the case in which $C = 3$. Up to six segments per frame can arrive at a transmitting queue by the user, since the frame is six arrival slots long (see Fig. 8b). However, at most two segments can be transmitted from any of the transmitting queues in every frame, again due to the structure of our schedule. Packet and segment loss are dependent on a new packet arriving to a queue at a particular

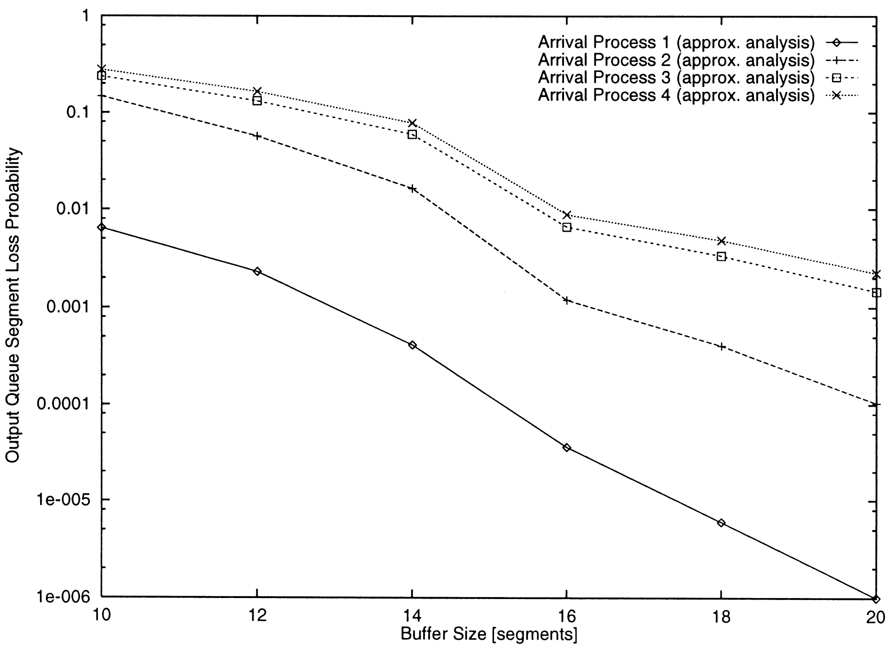


Fig. 11. Receiving queue segment loss probability ω_1 for $C = 2$.

point in the schedule and finding the queue vulnerable to overflow at some point while receiving the packet. Observe the effect of the arrival of a five-segment packet at various times during the frame on the occupancy of transmitting queue of say, node 8 for wavelength λ_1. If the packet arrives at the end of the first arrival slot of the frame, the buffer's occupancy will temporarily rise to as much as five segments above its occupancy level when the packet began to be received by the buffer. If the packet arrives at any other time during the frame, the buffer will peak only three segments above its initial point. Under the same conditions, observe that a six-segment packet may cause the buffer's occupancy to temporarily increase by as much as only four packets, a seven-segment packet may cause the buffer's occupancy to peak between three and five segments above the buffer's initial level, and an eight-segment packet between four and six segments above the buffer's initial level. With the interactions of these two effects, it seems reasonable that the loss measures for Arrival Process 2 are lower than those of Arrival Process 1 in Fig. 10.

Simulation was used to determine the error of our algorithm's results and 30 replications of 100,000 service slots each were executed; the simulation results were not plotted because they are extremely small when compared to the scales shown in the graphs. Instead, we present in Table 5 the mean errors associated with each curve shown in Figs. 9 and 10 [7]. Overall, we observed that simulation results closely matched those obtained through our analysis.

Figs. 11 and 12 show the segment loss probabilities for receiving queue 1, ω_1, for $C = 2$ and $C = 3$, respectively. Again, packet loss is not shown for the same reasons discussed earlier in this section. As one would expect, the loss probability decreases with buffer size and as the average packet size

[7] The mean absolute error is defined as the average of the absolute difference between all pairs of corresponding points (simulated and analysis) for a given experiment (i.e., for a given arrival process and number of channels).

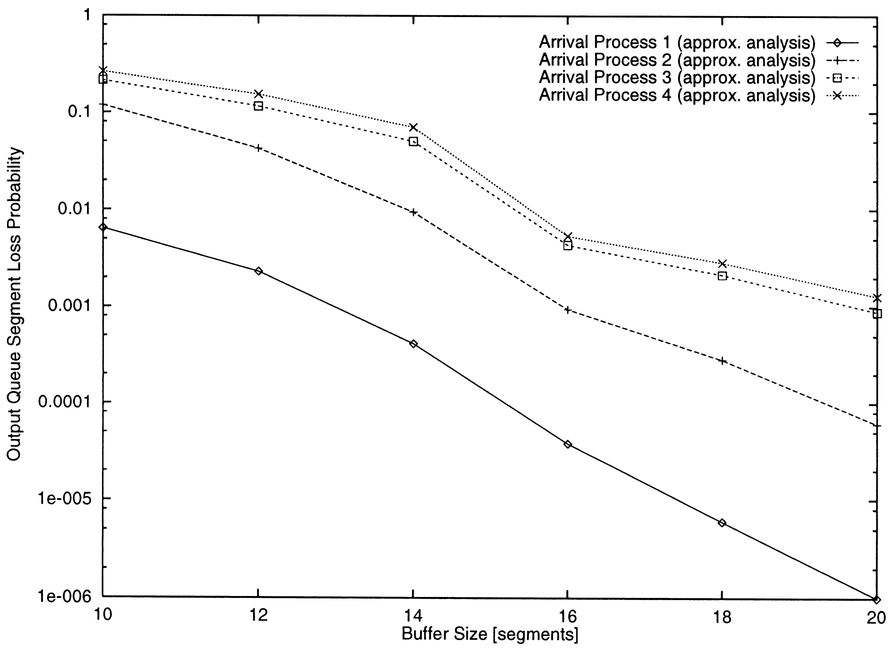


Fig. 12. Receiving queue segment loss probability ω_1 for $C = 3$.

Table 5
Mean absolute transmitting queue errors for $C = 2, 3$

	Arrival Process 1	Arrival Process 2	Arrival Process 3	Arrival Process 4
$C = 2$	3.2×10^{-6}	3.5×10^{-6}	7.8×10^{-5}	2.0×10^{-4}
$C = 3$	5.9×10^{-5}	3.2×10^{-6}	1.1×10^{-3}	1.0×10^{-3}

decreased. A significant drop in loss probabilities is observed as B increases from 14 to 16. This is mainly due to the fact that a buffer size of 16 can accommodate two complete eight-segment packets.

Confidence intervals derived from simulation results were, again, not shown as part of Figs. 11 and 12 as they were extremely small; instead, mean absolute errors are presented in Table 6. In comparing these errors to those in Table 5, it is immediately obvious that our receiving side analysis is not as accurate as the transmitting side analysis. This error is primarily attributable to the fact that our function $L_i(\cdot)$ is a heuristic and not an exact expression; this point was discussed in depth in Section 3.2.2. Comparing the

Table 6
Mean absolute receiving queue errors for $C = 2, 3$

	Arrival Process 1	Arrival Process 2	Arrival Process 3	Arrival Process 4
$C = 2$	7.4×10^{-4}	5.3×10^{-3}	4.7×10^{-3}	5.3×10^{-2}
$C = 3$	1.2×10^{-4}	2.4×10^{-2}	2.8×10^{-2}	3.2×10^{-2}

order of the error terms to the loss probability estimates shown, however, indicates that the average error associated with these curves is at least one order of magnitude less than the majority of the data points given.

6. Concluding remarks

We presented a discrete-time queueing network model of a broadcast WDM network architecture carrying IP packet traffic. Variable-length IP packets are segmented at the MAC layer of the source, and the fixed-size segments are transmitted in slots specified by a schedule that masks the transceiver tuning latency. Segments are then reassembled into the original packet at the receiving node. IP packets arriving at the network were assumed to follow an arbitrary length distribution. The queueing network model was analyzed approximately and the queue-length distribution and loss probabilities at the transmitting and receiving side of nodes were obtained. By comparing our analytical results to ones obtained through simulation, we established the accuracy of our approximation. Using our analysis it is possible to study the interactions among the various system parameters and to predict, explain, and fine tune the performance of the network. To the best of our knowledge, this is the first comprehensive performance study of optical local area networks under IP traffic, and it represents a first step towards bridging the gap between the optical and Internet technologies.

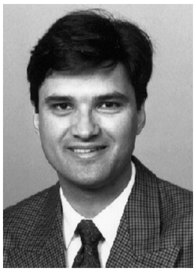
References

[1] Bridging the Gap Between Optical Networks and the Internet: a Mini-workshop, in: http://www.isi.edu/∼workshop/oi97, October 1997.
[2] M. Azizoglu, R.A. Barry, A. Mokhtar, Impact of tuning delay on the performance of bandwidth-limited optical broadcast networks with uniform traffic, IEEE J. Selected Areas Commun. 14 (5) (1996) 935–944.
[3] I. Baldine, G.N. Rouskas, Dynamic load balancing in broadcast WDM networks with tuning latencies, in: Proc. INFOCOM ’98, IEEE, March 1998.
[4] M. Borella, B. Mukherjee, A reservation-based multicasting protocol for WDM local lightwave networks, in: Proc. ICC ’95, IEEE, 1995, pp. 1277–1281.
[5] M.S. Borella, B. Mukherjee, Efficient scheduling of nonuniform packet traffic in a WDM/TDM local lightwave network with arbitrary transceiver tuning latencies, IEEE J. Selected Areas Commun. 14 (5) (1996) 923–934.
[6] Mon-Song, Chen, N.R. Dono, R. Ramaswami, A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks, IEEE J. Selected Areas Commun. 8 (6) (1990) 1048–1057.
[7] T.K. Chiang et al., Implementation of STARNET: a WDM computer communications network, IEEE J. Selected Areas Commun. 14 (5) (1996) 824–839.
[8] R. Chipalkatti, Z. Zhang, A.S. Acampora, Protocols for optical star-coupler network using WDM: performance and complexity study, IEEE J. Selected Areas Commun. 11 (4) (1993) 579–589.
[9] I. Chlamtac, A. Ganz, Channel allocation protocols in frequency–time controlled high speed networks, IEEE Trans. Commun. 36 (4) (1988) 430–440.
[10] R.L. Cruz, J.-T. Tsai, COD: alternative architectures for high speed packet switching, IEEE/ACM Trans. Networking 4 (1) (1996) 11–21.
[11] N.R. Dono, P.E. Green Jr., K. Liu, R. Ramaswami, F.F. Tong, A wavelength division multiple access network for computer communication, IEEE J. Selected Areas Commun. 8 (6) (1990) 983–994.
[12] P.W. Dowd, Random access protocols for high speed interprocessor communication based on an optical passive star topology, J. Lightwave Technol. 9 (1991) 799–808.
[13] P. Dowd et al., Lightning network and systems architecture, J. Lightwave Technol. 14 (6) (1996) 1371–1387.

[14] E.M. Foo, T.G. Robertazzi, A distributed global queue transmission strategy for a WDM optical fiber network, in: Proc. INFOCOM ’95, IEEE, 1995, pp. 154–161.
[15] A. Ganz, End-to-end protocols for WDM star networks, in: IFIP/WG6.1-WG6.4 Workshop on Protocols for High-Speed Networks, May 1989, pp. 219–235.
[16] A. Ganz, Z. Koren, WDM passive star — protocols and performance analysis, in: Proc. INFOCOM ’91, IEEE, 1991, pp. 991–1000.
[17] I.M.I. Habbab, M. Kavehrad, C.-E.W. Sundberg, Protocols for very high-speed optical fiber local area networks using a passive star topology, J. Lightwave Technol. 5 (12) (1987) 1782–1793.
[18] E. Hall et al., The Rainbow-II gigabit optical network, IEEE J. Selected Areas Commun. 14 (5) (1996) 814–823.
[19] J. Jue, M. Borella, B. Mukherjee, Performance analysis of the Rainbow WDM optical network prototype, IEEE J. Selected Areas Commun. 14 (5) (1996) 945–951.
[20] L. Kleinrock, Queueing Systems, Vol. 1, Computer Applications, Wiley, New York, 1976.
[21] J. Manchester, J. Andersen, B. Doshi, S. Dravida, IP over SONET, IEEE Commun. Mag. 36 (5) (1998) 136–142.
[22] M.W. McKinnon, G.N. Rouskas, H.G. Perros, Queueing-based analysis of broadcast optical networks, in: Proc. ACM SIGMETRICS/PERFORMANCE ’98, ACM, 1998, pp. 121–130.
[23] N. Mehravari, Performance and protocol improvements for very high-speed optical fiber local area networks using a passive star topology, J. Lightwave Technol. 8 (4) (1990) 520–530.
[24] A. Muir, J.J. Garcia-Luna-Aceves, Distributed queue packet scheduling algorithms for WDM-based networks, in: Proc. INFOCOM ’96, IEEE, March 1996, pp. 938–945.
[25] Z. Ortiz, G.N. Rouskas, H.G. Perros, Scheduling of multicast traffic in tunable-receiver WDM networks with non-negligible tuning latencies, in: Proc. SIGCOMM ’97, ACM, September 1997, pp. 301–310.
[26] V. Paxson, S. Floyd, Wide area traffic: the failure of poisson modeling, IEEE/ACM Trans. Networking 3 (3) (1995) 226–244.
[27] H.G. Perros, K.M. Elsayed, Call admission control schemes: a review, IEEE Commun. Mag. 34 (11) (1996) 82–91.
[28] G.R. Pieris, G.H. Sasaki, Scheduling transmissions in WDM broadcast-and-select networks, IEEE/ACM Trans. Networking 2 (2) (1994) 105–110.
[29] G.N. Rouskas, M.H. Ammar, Multi-destination communication over tunable-receiver single-hop WDM networks, IEEE J. Selected Areas Commun. 15 (3) (1997) 501–511.
[30] G.N. Rouskas, V. Sivaraman, Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies, IEEE/ACM Trans. Networking 5 (3) (1997) 359–370.
[31] V. Sivaraman, G.N. Rouskas, HiPeR-ℓ: A *Hi*gh *Pe*rformance *R*eservation protocol with ℓook-ahead for broadcast WDM networks, in: Proc. INFOCOM ’97, IEEE, April 1997, pp. 1272–1279.
[32] W. Stewart, Numerical Solutions of Markov Chains, Princeton University Press, Princeton, NJ, 1994.
[33] K. Thompson, G.J. Miller, R. Wilder, Wide-area internet traffic patterns and characteristics, IEEE Network Mag. 11 (6) (1997) 10–23.
[34] S. Tridandapani, J.S. Meditch, A.K. Somani, The MaTPi protocol: masking tuning times through pipelining in WDM optical networks, in: Proc. INFOCOM ’94, IEEE, June 1994, pp. 1528–1535.
[35] S.B. Tridandapani, B. Mukherjee, Channel sharing in multi-hop WDM networks: realization and performance of multicast traffic, IEEE J. Selected Areas Commun. 15 (3) (1997) 488–500.

Martin W. McKinnon received his doctoral degree from North Carolina State University in Computer Engineering in 1997, with his research focusing on the design and performance analysis of photonic device architectures. He has been employed by The MITRE Corporation, Nortel (formerly Bell Northern Research), and other telecommunication-related organizations. He is currently with the Communications and Networking Division of the Georgia Tech Research Institute. His current research is in photonic architectures, active networking, and queueing theory. He is a member of IEEE, ACM, and ORSA.

H.G. Perros received the B.Sc. degree in Mathematics in 1970 from Athens University, Greece, the M.Sc. degree in Operational Research with Computing from Leeds University, England, in 1971, and the Ph.D. degree in Operations Research from Trinity College Dublin, Ireland, in 1975. From 1976 to 1982 he was an Assistant Professor in the Department of Quantitative Methods, University of Illinois at Chicago. In 1979 he spent a sabbatical term at INRIA, Rocquencourt, France. In 1982 he joined the Department of Computer Science, North Carolina State University, as an Associate Professor, and since 1988 he is a Professor. During the academic year 1988–89 he was on a sabbatical leave of absence first at BNR, Research Triangle Park, North Carolina, and subsequently at the University of Paris 6, France. Also, during the academic year 1995–96 he was on a sabbatical leave of absence at NORTEL, Research Triangle Park, North Carolina. He has published extensively in the area of performance modelling of computer and communication systems, and he has organized several national and international conferences. He also published a monograph entitled 'Queueing Networks with Blocking: Exact and Approximate Solutions', Oxford Press, and currently he is writing a textbook on ATM networks. He is the chairman of the IFIP W.G. 6.3 on the Performance of Communication Systems. His current research interests are in the areas of optical networks and their performance, and software performance evaluation.



George N. Rouskas received the Diploma in Electrical Engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in Computer Science from the College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. He joined the Department of Computer Science, North Carolina State University in August 1994, where he has been an Associate Professor since July 1999. His research interests include high-speed and lightwave network architectures, multipoint-to-multipoint communication, and performance evaluation. He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award, and a co-author of a paper that received the *Best Paper* Award at the 1998 SPIE conference on All-Optical Networking. He also received the 1995 *Outstanding New Teacher* Award from the Department of Computer Science, North Carolina State University, and the 1994 *Graduate Research Assistant* Award from the College of Computing, Georgia Tech. He is a member of the IEEE, the ACM and of the Technical Chamber of Greece.