Performance Analysis of a Photonic Switch Architecture Supporting Variable Length Packets

Martin W. McKinnon Harry G. Perros George N. Rouskas

TR-97-11

November 4, 1997

Abstract

We consider a photonic switch based on a broadcast WDM architecture with tunable transmitters and fixed frequency receivers. The switch supports variable length packets which are internally segmented and reassembled. The switch operates under a schedule that masks the transceiver tuning latency. We analyze approximately a queueing model of the switch in order to obtain the queue-length distribution and loss probabilities at the input and output ports. Numerical results for loss probabilities are presented and compared against simulation results.

Keywords: Optical networks, Wavelength Division Multiplexing (WDM), Discrete-time queueing networks, Variable Length Packets

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534

1 Introduction

In recent years there has been a rapid progress towards using optical networks for packet switching. Specifically, a significant amount of research has been carried out on the design of wavelength division multiplexing (WDM) based switching architectures. The performance analysis of these architectures has been typically carried out under the unrealistic assumption of symmetric traffic, that is memoryless arrival processes and uniform destination probabilities (see [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], among others). Two studies of optical networks that use non-Poisson traffic models appeared recently in [16, 17]. The work in [16] derives a stability condition for the HiPer-ℓ reservation protocol, while [17] studies the effects of wavelength conversion in wavelength routing networks. Furthermore, in [18], we analyzed a queueing network model of the broadcast-and-select WDM architecture, assuming bursty and correlated arrival processes, non-uniform destination probabilities, finite buffer capacities, and non-zero transceiver tuning delays.

In this paper, we extend our analysis described in [18] to the case where the arriving packets are not fixed in size. Rather, they have an arbitrarily distributed length. A packet upon arrival at the switch is fragmented to fixed-sized segments. The segments are then transmitted individually to the destination output port, where they are re-assembled into the original packet. The packet is finally transmitted out of the output port. All input and output buffers are finite in capacity. We construct and analyze a queueing network model 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.

The next section presents our system model and provides some background information. The performance analysis of the switch is presented in Sections 3 and 4, numerical results are given in Section 5, and we conclude the paper in Section 6.

2 The Switch Under Study

The optical switch architecture consists of N input ports and N output ports interconnected through a broadcast passive star (the switch fabric) supporting $C \leq N$ wavelengths $\lambda_1, \cdots, \lambda_C$. Each input port is equipped with a laser that enables it to inject signals into the optical medium. Similarly, each output port is capable of receiving optical signals through an optical filter. The laser

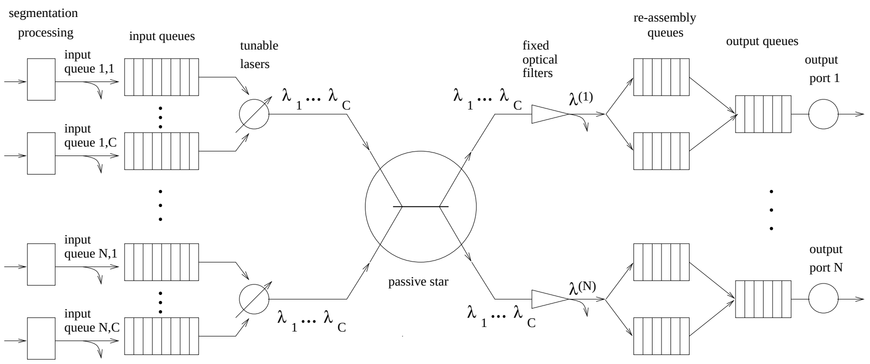

Figure 1: Queueing model of the switch architecture for variable length packets

at each input port 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 output port j. Since $C \leq N$, a set $\mathcal{R}_c$ of output ports may be sharing a single receive wavelength λ_c:

$$\mathcal{R}_c \quad = \quad \{j \ : \ \lambda(j) = \lambda_c\}, \qquad c = 1, \cdots, C \tag{1}$$

Sets $\mathcal{R}_c$ will typically be obtained by running a load balancing algorithm [19].

The switch accommodates packets of varying length. Specifically, a packet is assumed to consist of an arbitrary number of fixed length segments (the last segment is padded if necessary).

The switch operates in a slotted mode. Since there are N ports but $C \leq N$ channels, each channel must run at a rate $\frac{N}{C}$ times faster than the rate of the input links ($\frac{N}{C}$ need not be an integer). The rate of an output link is equal to the rate of an input link. Thus, *arrival* slots (which correspond to the segment transmission time at the input/output link rate) and *service* slots (which are equal to the segment transmission time at the channel rate within the switch) are distinguished as different units of time. Obviously, the duration of a service slot is equal to $\frac{C}{N}$ times that of an arrival slot. Without loss of generality, assume that all input links are synchronized at arrival slot boundaries; similarly for output links. On the other hand, all C channels internal to the switch are synchronized at service slot boundaries.

The queueing network model depicting the above switch architecture is shown in Figure 1. Each input port i consists of C finite capacity queues. Each queue c at input port i contains packets destined for the output ports with receivers listening on wavelength λ_c, $c = 1, \cdots, C$. This arrangement eliminates the head-of-line problem. We let $B_{ic}^{(in)}$ denote the capacity, in segments, of queue c at input port i, $c = 1, \cdots, c$ and $i = 1, \cdots, N$.

Since an arriving packet consists of a number of segments, it requires several arrival slots in order to be completely received at an input queue. Therefore, it is possible for some of the segments of a packet to be in the input queue while the remainder of the packet has not as yet been received. An arriving segment is buffered if the input queue is not full. If, however, the queue is full, the segment is dropped along with all other segments of the packet already in the input queue. Those segments of the packet that have not arrived yet are also dropped upon arrival [1].

Segments buffered in the input queues of the N input ports are transmitted onto the optical medium by the port's laser transmitter according to a transmission schedule (see Section 2.1). The schedule guarantees that a segment will be correctly received by its destination port. Segments in an input queue are transmitted one at a time on a FIFO basis.

Upon arrival at the output port, the segments are buffered in a re-assembly queue. There is one re-assembly queue per input port. Each re-assembly queue has a finite capacity and it can accommodate a packet of the maximum length. Each queue accumulates segments of packets until the entire packet has been completely received. At that instant, the entire packet is transferred instantaneously to the output queue, which has a finite capacity. If adequate space is not available in the output queue, the entire packet is lost. It should be noted that, due to the nature of the system, segments arrive at a re-assembly queue in order. Furthermore, no losses can occur from a re-assembly queue, since a reassembly queue can accommodate a packet of maximum size. We let $B_j^{(out)}$ denote the buffer capacity in segments of output port j, $j = 1, \cdots, N$. Packets in an output queue are transmitted onto the outgoing link on a FIFO basis.

2.1 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 input ports 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. For the Gigabit per second rates envisioned here, the tuning latency of state-of-the-art tunable lasers or filters can be as long as several times the size of a service slot [20]. Consequently, approaches that require each tunable transmitter to send a single segment and then switch to a new wavelength, will suffer a high tuning overhead and will result in a very low throughput.

In a recent paper [21], it was shown that careful scheduling can mask the effects of arbitrarily long tuning latencies, making it possible to build high-throughput photonic ATM switches using

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

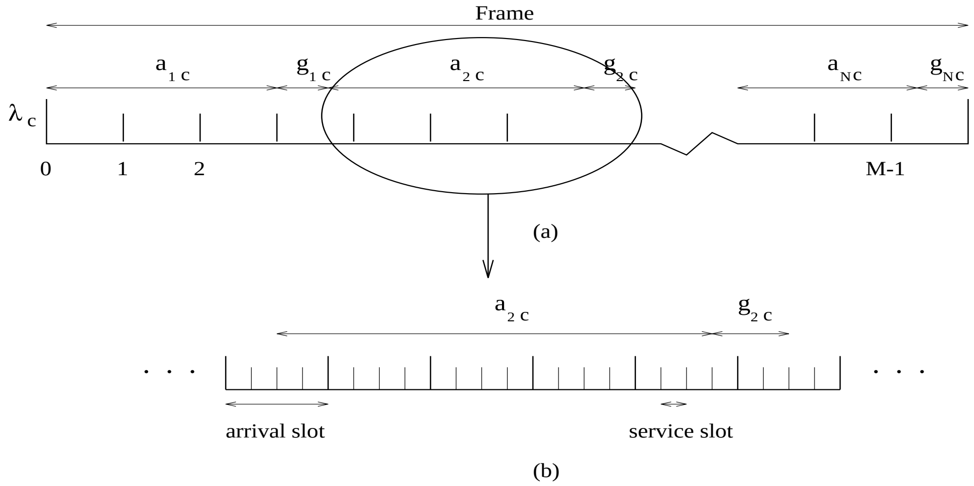

Figure 2: (a) Schedule for channel λ_c, and (b) detail corresponding to input port 2

currently available lightwave technology. The key idea is to have each tunable transmitter send a *block* of segments on a wavelength before switching to another one. The main result of [21] was a set of new algorithms for constructing near-optimal (and, under certain conditions, optimal) schedules for transmitting a set of traffic demands $\{a_{ic}\}$. Quantity a_{ic} represents the number of segments to be transmitted by input port i onto channel λ_c per frame. The schedules are such that no collisions occur. Furthermore, they are easy to implement in a high speed environment, since the order in which the various input ports transmit is the same for all channels.

Quantity $a_{ic},\ i = 1, \cdots, N,\ c = 1, \cdots, C$, can be seen as the number of service slots per frame allocated to input port i, so that the port can satisfy the required quality of service of its incoming traffic intended for wavelength λ_c. By fixing a_{ic}, a certain amount of the bandwidth of wavelength λ_c is indirectly allocated to port i. This bandwidth could be equal to the effective bandwidth of the total traffic carried by input port i on wavelength λ_c. In general, the estimation of the quantities $a_{ic},\ i = 1, \cdots, N,\ c = 1, \cdots, C$, is part of the call admission algorithm, and it is beyond the scope of this paper. Notice 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.

We assume that transmissions by the input ports onto wavelength λ_c follow a schedule as shown in Figure 2. This schedule repeats over time. Each frame of the schedule consists of M arrival slots. Within each frame, input port 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 port can transmit on λ_c. This gap may be necessary to ensure that input port $i+1$ has sufficient time to tune from wavelength λ_{c-1} to λ_c before it starts transmission. The algorithms in [21] are such that

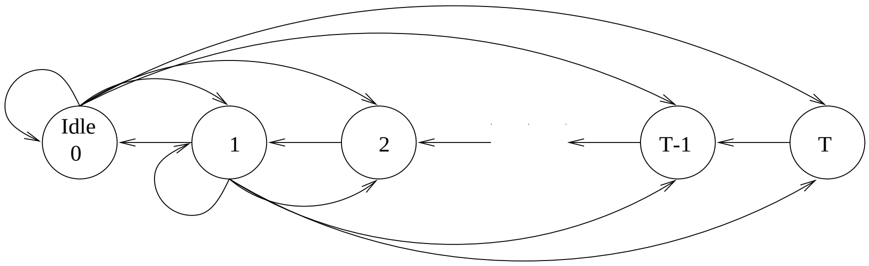

Figure 3: State machine for packet segment arrival process

the number of slots in most of the gaps is equal to either zero or a small integer. Thus, the length of the schedule is very close to the lower bound $\max_i\{\sum_{c=1}^{C} a_{ic}\}$. Note that in Figure 2 it is 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 the frame begins and ends on *arrival* slot boundaries, the beginning or end of transmissions by a port does not necessarily coincide with the beginning or end of an *arrival* slot (although it is, obviously, synchronized with *service* slots).

2.2 The Arrival Process

In our model, a different arrival process is associated with each queue $(i, c), i = 1, \cdots, N, \ c = 1, \cdots, C$. Let T denote the maximum packet size. An arriving packet at queue (i, c) consists of s segments with probability $f_{i,c}(s)$. Thus, our analysis is valid for arbitrary packet size destributions.

The rate diagram of the packet arrival process is shown in Figure 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 arrives (when $s = 1$). At that moment, the arrival process' state will change to another state $s, 1 \leq s \leq T$, if another packet arrives, or to state 0 if it becomes idle.

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

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

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

$$\mathbf{A}_{ic} = \begin{bmatrix} 0 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{3}$$

where $q_{ic}^{(k,l)}, k, l = 0, \cdots, T$, 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 $\mathbf{Q}_{ic}$ occur only at the boundaries of *arrival* slots.

3 Queueing Analysis

In this section we analyze the queueing network described above and shown in Figure 1. We obtain the queue-length distribution in the input queues $(i, c), i = 1, \cdots, N, c = 1, \cdots, C$, and output queues $j, j = 1, \cdots, N$, from which performance measures such as the packet-loss and the segment-loss probabilities can be obtained.

3.1 Analysis of an Input Queue

Each input queue (i, c) is served by a single wavelength λ_c. This wavelength, in fact, is shared by all input queues $(i, c), \ i = 1, \cdots, N$. 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 input 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 input queue (i, c) can be analyzed separately and independently of all other queues $(l, c), l \neq i$.

Let us now consider input queue (i, c) in isolation. This queue can be analyzed numerically by solving its underlying Markov chain. The state of the system should 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 input queue's analysis 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 the analyze input queue i by constructing its underlying Markov chain embedded at arrival slot boundaries. The Markov chain consists of the tuple (x, y, z), where

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

It should be noted that the definition of z does not follow exactly the process shown in Figure 3. The process is modified in order to account for a discarded packet which is partially enqueued. Specifically, consider the a segment arriving to find a full queue. In this case, not only are the segments of the packet which are already enqueued discarded, but the remaining segments of the packet which have not yet been received must also be discarded. In order to preserve the inter-packet arrival distribution, an additional set of states is added to the arrival process' state description to indicate when a segment from a discarded packet is being received (and, hence, itself discarded). These states are identified by the negative of the states shown in Figure 3. Therefore, if the arrival process is in state $z, 2 \leq z \leq T$, and the queue discards the segment as described, the process will transition to state $-(z-1)$, not state $z-1$. Figure 4 gives the rate diagram of this augmented process.

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 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

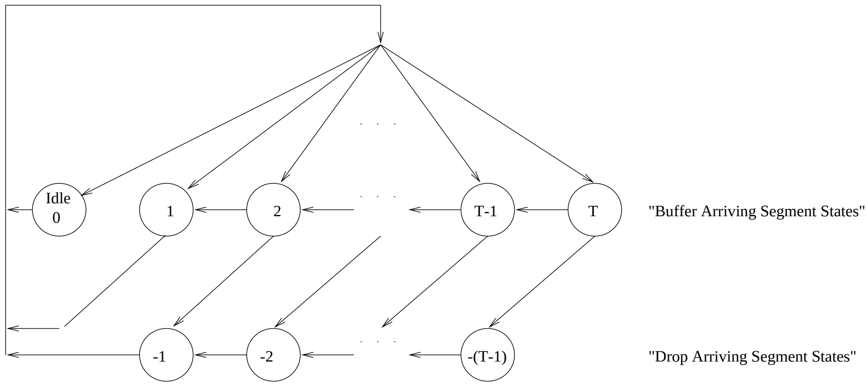

Figure 4: State machine for arrival state accounting for buffering and dropping of packet segments

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

Current State	**Next State**	**Transition Probability**
(x, y, z)	$(x \oplus 1,$ $\max\{0, y - v_{ic}(x \oplus 1)\} + 1, z')$	$q_{ic}^{(zz')}\mathbf{I}_{v_{ic}(x)>0 \text{ or } y<B_{ic}} \times$ $\mathbf{I}_{(z>1 \ \& \ z'=(z-1)) \text{ or } (z=1)}$
(x, y, z)	$(x \oplus 1,$ $\max\{0, y - v_{ic}(x \oplus 1)\}, z')$	$q_{ic}^{(\lvert z\rvert\lvert z'\rvert)} \times$ $\mathbf{I}_{(z=0,-1) \text{ or } ((z<-1) \ \& \ (z'=z+1))}$
(x, y, z)	$(x \oplus 1, B_{ic} - s, z')$	$f_{ic}(s+z)q_{ic}^{(z\lvert z'\rvert)}\mathbf{I}_{v_{ic}(x)=0 \ \& \ y=B_{ic}} \times$ $\mathbf{I}_{(z>1 \ \& \ z'=-(z-1)) \text{ or } (z=1 \ \& \ z'\geq 0)}$

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 Figure 5(b). 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, $\mathbf{I}_{f(x)}$ 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 (2), this event has a probability $q_i^{(zz')}$ of occurring), and a segment arrives and is buffered by the queue. This event can only occur if z' is positive (see Figure 4) *and* either $v_{ic}(x) > 0$ or $y < B_{ic}$. The latter conditions are imposed to ensure that the new queue length

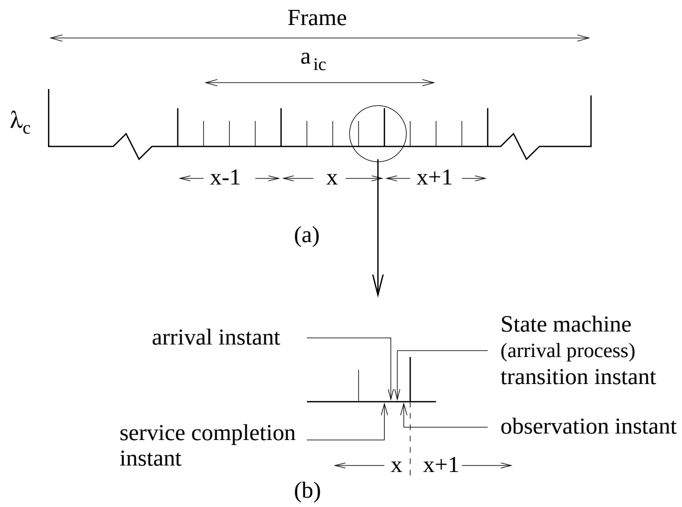

Figure 5: (a) Service period of input port i on channel λ_c, and (b) detail showing the relationship among service completion, arrival, arrival process state transition, and observation instants within an service slot and an arrival slot

will not exceed the capacity $B_{ic}^{(in)}$ of the input queue [2]. 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 (3)). 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 input 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.

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

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

$$
\mathbf{S}_{ic} \quad = \quad \begin{array}{c} \left[\begin{array}{cccccc} 0 & \mathbf{R}_{ic}(0) & 0 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{R}_{ic}(1) & 0 & \cdots & 0 \\ 0 & 0 & 0 & \mathbf{R}_{ic}(2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \mathbf{R}_{ic}(M-2) \\ \mathbf{R}_{ic}(M-1) & 0 & 0 & 0 & \cdots & 0 \end{array} \right] \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ M-2 \\ M-1 \end{array} \qquad (5)
$$

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 $\mathbf{R}_{ic}(x)$. There are M different $\mathbf{R}_{ic}$ matrices, one for each arrival slot x in the frame.

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

$$
\mathbf{X}_{ic}(\cdot \mid x, y) \quad = \quad \tilde{\mathbf{A}}_{ic}\, \tilde{\mathbf{Q}}_{ic}(\mathbf{I}_{v_{ic}(x)=0 \text{ and } y=B_{ic}}) \qquad \text{and} \qquad \mathbf{Y}_{ic} \quad = \quad (\mathbf{I} - \tilde{\mathbf{A}}_{ic})\, \tilde{\mathbf{Q}}_{ic}(0), \qquad (6)
$$

where $\mathbf{I}$ is the identity matrix and $\tilde{\mathbf{A}}_{ic}$ and $\tilde{\mathbf{Q}}_{ic}(\cdot)$ are given in (7), (8), and (9) Matrix $\mathbf{X}_{ic}(\cdot \mid x, y)$ (respectively, $\mathbf{Y}_{ic}$) is the arrival process' state transition probability matrix given that an arrival is (resp., is not) accepted by the input queue. For the sake of clarity, $\mathbf{X}_{ic}(\cdot \mid x, y)$ will be denoted simply as $\mathbf{X}_{ic}(\cdot)$ for the remainder of the paper.

$$
\tilde{\mathbf{A}}_{ic} \;=\; \begin{array}{c} \\ -(T-1) \\ \vdots \\ -1 \\ 0 \\ \vdots \\ T \end{array} \begin{array}{c} \begin{array}{ccc|ccc} -(T-1) & \cdots & -1 & 0 & \cdots & T \end{array} \\ \left(\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{A}_{ic} \end{array} \right) \end{array} \qquad (7)
$$

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

and

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

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

$$\mathbf{R}_{ic}(x|v_{ic}(x\oplus 1)>0) = \begin{array}{c} \left[\begin{array}{ccccccc} \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 & \cdots \\ 0 & \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) & 0 & 0 & 0 & \cdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & \cdots \\ 0 & \cdots & 0 & \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) & 0 & \cdots \end{array}\right] \end{array} \begin{array}{c} 0 \\ \vdots \\ v_{ic}(x\oplus 1) \\ v_{ic}(x\oplus 1)+1 \\ \vdots \\ B_{ic}^{(in)} \end{array} \tag{10}$$

The structure of matrix $\mathbf{R}_{ic}(x)$ given in (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 $\mathbf{R}_{ic}(x)$. 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 $\mathbf{R}_{ic}(x)$. Of course, y cannot exceed the queue capacity $B_{ic}^{(in)}$. Since the number of service slots $v_{ic}(x \oplus 1)$ depends on the particular slot $x \oplus 1$ within the frame, $\mathbf{R}_{ic}(x)$ is a function of x.

Matrix $\mathbf{R}_{ic}(x)$ is slightly different when $v_{ic}(x \oplus 1) = 0$; its structure is shown in (11). In this case, if the state of the input queue is $y = B_{ic}^{(in)}$, not only will a new arrival be discarded, but some number of currently enqueued segments may also be discarded. Let matrix $\mathbf{P}_n(\cdot \mid x)$ 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 $\mathbf{P}_n(\cdot \mid x)$ may be described simply: $\mathbf{P}_n$ 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, $0 < z \leq (T-n+1)$. 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 Figure 4, the complete packet size (in terms of segments) may be inferred exactly as $z + n - 1$, occurring with the probability shown in (12).

$$\mathbf{R}_{ic(x|v_{ic}(x\oplus 1)=0)} = \begin{bmatrix} \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \mathbf{Y}_{ic} & \mathbf{X}_{ic}(\cdot) \\ 0 & \cdots & 0 & \mathbf{P}_T(\cdot)\mathbf{X}_{ic}(\cdot) & \cdots & \mathbf{P}_2(\cdot)\mathbf{X}_{ic}(\cdot) & \mathbf{Y}_{ic}+\mathbf{P}_1(\cdot)\mathbf{X}_{ic}(\cdot) \end{bmatrix} \tag{11}$$

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

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 $\mathbf{K}_{ic}$, and is obtained as:

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

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

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

It can be verified that the Markov chain with probability transition matrix $\mathbf{S}_{ic}$ in (5) is irreducible, and therefore a steady-state distribution exists. Transition matrix $\mathbf{S}_{ic}$ defines a *p-cyclic* Markov chain, and therefore it can be solved using any of the techniques for p-cyclic Markov chains in [22, ch. 7]. We have used the LU decomposition method to obtain the steady state probability $\pi_{ic}(x, y, z)$.

3.2 Output Side Analysis

As shown in Figure 1, each output port j consists of a receiver, N re-assembly queues, and an output queue. The receiver filters out segments from the passive star coupler, allowing only those segments transmitted on a given frequency to pass. As segments are received, from a particular input port, they are buffered in the appropriate re-assembly queue until a complete packet is formed. At that instant, the packet is transferred to the output queue. If the output queue does not have enough space to accommodate the entire packet, the packet is lost. The re-assembly queue is large enough to accommodate at least T segments. Thus, no segment can be lost upon arrival at the re-assembly queue.

Using the steady-state probabilities, $\pi_{ic}(x, y, z)$, $i = 1, \cdots, N$, $c = 1, \cdots, C$, we analyze each output port in isolation.

3.2.1 Analysis of the Re-assembly Queue

Let us consider the N re-assembly queues of output port j. For each re-assembly queue, we define a Markov chain, (x, ζ), where x indicates the arrival slot number within the frame ($x = 0, 1, \cdots, M-1$), and ζ indicates the occupancy of the reassembly queue in segments ($\zeta = 0, \cdots, T$). The ordering of the events is shown in Figure 6, and the transition probabilities are shown in Table 2.

The term $\mathcal{P}(\cdot)$ is the probability of receiving ζ_{add} segments from the appropriate input queue during slot x. We have that

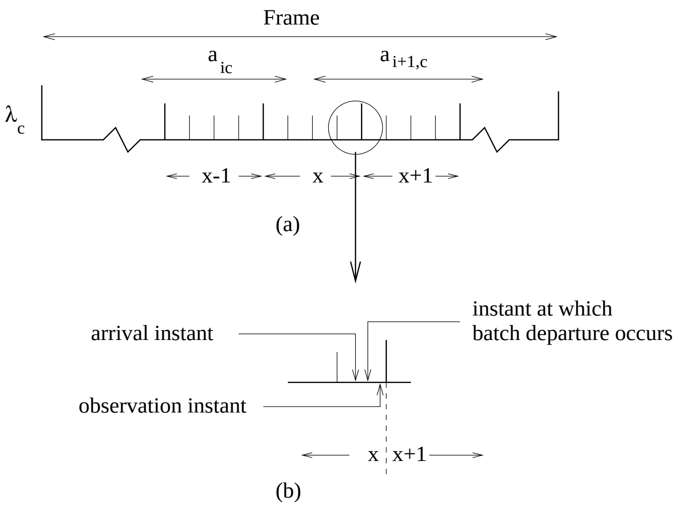

Figure 6: (a) Arrivals to reassembly queue i of output port j, and (b) detail showing the relationship of departure, arrival, and observation instants

Table 2: Transition probabilities out of state (x, ζ) of the Markov chain for re-assembly queue i of output port j

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

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

We note that $M \sum_z \pi_{ic}(x, y, z)$ is the conditional probability of having y customers in input queue (i, c) given slot x and without regard for the state of the input process. The term $\mathcal{F}(\cdot)$ 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 arrive from an input queue (i.e., it discharges $\max(0, \zeta - \zeta' + \zeta_{add})$ segments), where

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

We note that $f_{ic}(s' \mid s)$ is defined to be the probability that the packet size is equal to s' segments given that it is greater than s segments.

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

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

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

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

It should be noted that the number of departures at the end of an arrival slot, if any, must exceed the occupancy of the system at the end of the previous arrival slot, ζ. Also, we assumed that only one packet may be completed during an arrival slot, we impose the restriction that the minimum packet size must be greater than or equal to m segments,where m is the ratio of ports to channels. This restriction is imposed only to simplify the analysis by eliminating the complexity which would be added by considering the possibility of multiple packets being discharged from a re-assembly queue during one arrival slot.

The transition probability matrix for each reassembly queue (i, j) is then constructed and the occupancy probability, $\Phi_{ij}(x, \zeta)$, may be is determined using the LU decomposition method [22]. $\Phi_{ij}(x, \zeta)$ is the steady-state probability that reassembly queue (i, j) has ζ segments at the end of slot x. Since the actual number of segments in each packet is lost from our model once the segments are buffered at an input queue, the re-assembly of the segments at this point will most likely not result in the exact reconstruction of the packets. Let us consider the following example. Let us assume a packet length distribution which allows for packets between four and seven segments long. Now, let us consider that a packet, originally six segments long is forwarded one segment at a time to a reassembly queue. It is possible that, after the fifth segment has arrived, the re-assembly queue forwards the five segments as a packet to the output queue. When the last segment arrives, it will remain in the re-assembly queue until it becomes part of another packet. In a lightly loaded system, this type of situation can affect the accuracy of the estimated buffer occupancy distribution, $\Phi_{ij}(x, \zeta)$. In the next section, we introduce a heuristic that significantly improves the accuracy of $\Phi_{ij}(x, \zeta)$.

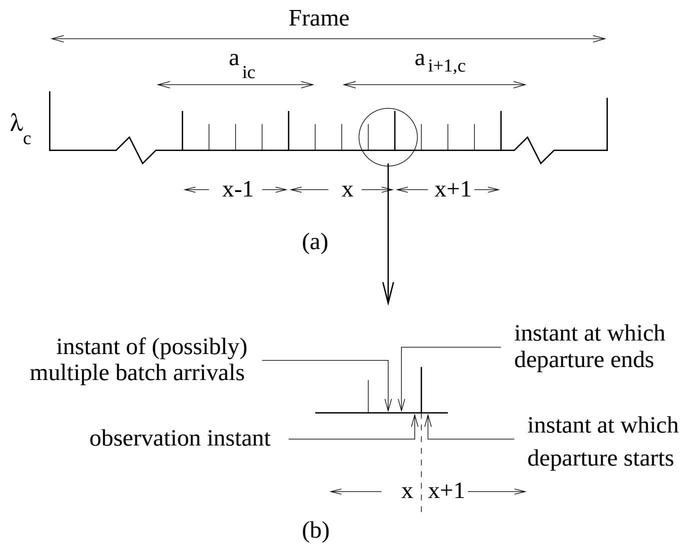

Figure 7: (a) Arrivals to output port j from re-assembly queues i and $i+1$, and (b) detail showing the relationship of departure, arrival, and observation instants

3.2.2 Analysis of an Output Queue

As in the previous section, we obtain the queue-length distribution of output port j at arrival slot boundaries. Let (x, w) be the state associated with output port j, where x indicates the arrival slot number within the frame ($x = 0, 1, \cdots, M-1$), and w indicates the number of segments at the output queue ($w = 0, 1, \cdots, B_j^{(out)}$). The ordering of events is shown in Figure 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 $0 \leq s \leq T$ of segments may be transmitted from each of the relevant reassembly queues to output port j within arrival slot $x \oplus 1$, and (d) the queue capacity is $B_j^{(out)}$. Based on the first item above, it can be easily seen that the transition matrix $\mathbf{T}_j$ of this Markov chain has the same structure as matrix $\mathbf{S}_{ic}$ given by (5). We have

$$\mathbf{T}_j = \begin{array}{c} \left[\begin{array}{cccccc} 0 & \mathbf{U}_j(0) & 0 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{U}_j(1) & 0 & \cdots & 0 \\ 0 & 0 & 0 & \mathbf{U}_j(2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \mathbf{U}_j(M-2) \\ \mathbf{U}_j(M-1) & 0 & 0 & 0 & \cdots & 0 \end{array} \right] \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ M-2 \\ M-1 \end{array} \qquad (20)$$

The construction of the matrices $\mathbf{U}_j(x)$ is somewhat complicated. This is mainly due to the fact that during an arrival slot x, more than one input port can transmit on the same channel. In this case, determining which packets are accepted by an output queue which is close to being full may be quite involved. The construction of a matrix $\mathbf{U}_j(x)$ is summarized in the algorithm given below. We note that the elements of matrix $\mathbf{U}_j(x)$ are denoted as $\mathbf{U}_j(x,k,l)$, where k is the row index and l is the column index.

```
U_j(x) ← [0]
loop: ∀ w ∈ {0, 1, ···, B_j}
    I_j(x) ← an ordered set of the input ports serviced by channel c
        during slot x with arity | I_j(x) |
    loop: ∀ s ∈ {{0, 0, ···, 0}, {0, 0, ···, 1}, ···, {T, T, ···, T}}   (each set has |I_j(x)| elements)
        w' ← w
        loop: ∀ i ∈ {1, ···, | I_j(x) |}
            if {w' + s_i ≤ B_j} ⇒ {w' ← w' + s_i}
        if {w > 0} ⇒ {w' ← w' − 1}
        U_j(x, w, w') ← U_j(x, w, w') + ∏_{i=1}^{|I_j(x)|} L_i(s_i | x)
```

$\mathbf{U}_j(x) \leftarrow [\mathbf{0}]$

loop: $\forall\ w \in \{0,1,\cdots,B_j\}$

$\mathcal{I}_j(x) \leftarrow$ an ordered set of the input ports serviced by channel c during slot x with arity $\mid \mathcal{I}_j(x) \mid$

loop: $\forall\ s \in \{\{\overbrace{0,0,\cdots,0}^{|\mathcal{I}_j(x)|}\},\{0,0,\cdots,1\},\cdots,\{T,T,\cdots,T\}\}$

$w' \leftarrow w$

loop: $\forall\ i \in \{1,\cdots,\mid \mathcal{I}_j(x) \mid\}$

if $\{w'+s_i \le B_j\} \Rightarrow \{w' \leftarrow w'+s_i\}$

if $\{w>0\} \Rightarrow \{w' \leftarrow w'-1\}$

$\mathbf{U}_j(x,w,w') \leftarrow \mathbf{U}_j(x,w,w') + \prod_{i=1}^{|\mathcal{I}_j(x)|} L_i(s_i \mid x)$

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 output port j [3]. We have:

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

where

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

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

[3] Since in most cases only one or two input ports will transmit to the same channel within an arrival slot (refer also to Figure 5), and since a packet can only be completed in a reassembly queue while the input port is transmitting over the given channel, the dimension of the vector 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.

We calculate $L_i(s_i \mid x)$ so as to reduce the error introduced when reconstructing a packet in the re-assembly queue. Specifically, by using the set $\mathcal{A}_{ij}^{(s_i)}(x)$ (the set of arrival slots during which re-assembly queue (i, j) may receive segments and which encompasses exactly $s_i - 1$ service slots prior to arrival slot x), we are able to take into account an appropriate minimal packet inter-arrival time, in terms of service slots, given the size of the packet s_i and the time of arrival within the schedule x. It should be node that, depending on the values involved, the same arrival slot may need to be considered (and, hence, found in $\mathcal{A}_{ij}^{(s_i)}(x)$) multiple times, based on its being encountered in different frames.

We solve for the steady state occupancy probability of output port j during a slot x, $\pi_j(x, w)$ using an LU decomposition.

3.3 Summary of the Decomposition Algorithm

Below we summarize the approximation described above.

1. Given each arrival process, defined by $\mathbf{A}_{ic}$ and $\mathbf{Q}_{ic}$, formulate $\tilde{\mathbf{A}}_{ic}$, $\tilde{\mathbf{Q}}_{ic}(0)$, and $\tilde{\mathbf{Q}}_{ic}(1)$ per equations (7), (8), and (9), respectively. Additionally, compose the matrices $\mathbf{P}_n$ corresponding to each arrival process per equation (12).

2. For each arrival slot x, use the schedule and expressions (4) to compute the quantities $v_{ic}(x)$, $i = 1, \cdots, N,\ c = 1, \cdots, C$.

3. For each input queue i, construct the transition probability matrix $\mathbf{S}_{ic}$ from (2), (3), (5), (6), and (10). Solve this matrix for $\pi_{ic}(x, y, z)$.

4. For each reassembly queue (i, j), use $\pi_{ic}(x, y, z)$, (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 in its queue at the end of slot x.

5. For each output port $j \in \mathcal{R}_c$, use $\pi_{ic}(x, y, z)$, $\Phi_{ij}(x, \zeta)$, and (21) to construct the transition matrix $\mathbf{T}_j$ given by (20). Solve the matrix to obtain $\pi_j(x, w)$, the steady-state probability that port j has w cells in its queue 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 input and output ports.

4.1 Segment and Packet Loss Probability at an Input Port

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

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

Therefore, we have:

$$\begin{aligned} E[\text{number of arrivals per frame at queue } c \text{ of port } i] &= \\ \textstyle\sum_{x=0}^{M-1}(\sum_y \pi_{ic}(x,y,z=1) + \sum_y \pi_{ic}(x,y,z=-1)) \end{aligned} \tag{24}$$

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:

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

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

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

4.2 Segment and Packet Loss Probability at an Output Port

The packet and segment loss probabilities at an output port is more complicated to calculate, since we may have multiple packet arrivals to the given output port 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 a closed form expressions but they can be easily calculated using a slightly modified version of the algorithm for calculating $\mathbf{U}_j(x)$ presented in Section 3.2.2.

5 Numerical Results

We now discuss the accuracy of our analysis by applying the approximation algorithm to a 8×8 switch and comparing the loss probabilities to simulation results. Four different packet length

Table 3: Packet length distributions for considered arrival processes

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 $C = 2, 3$

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

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

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

The quantities a_{ic}, i.e., the number of service slots allocated to input port 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., $a_{ic} = \lceil \frac{N}{2C} \rceil \;\; \forall\, i, c$). We have assumed that the time it takes a laser to tune from one channel to another is equal to one arrival slot.

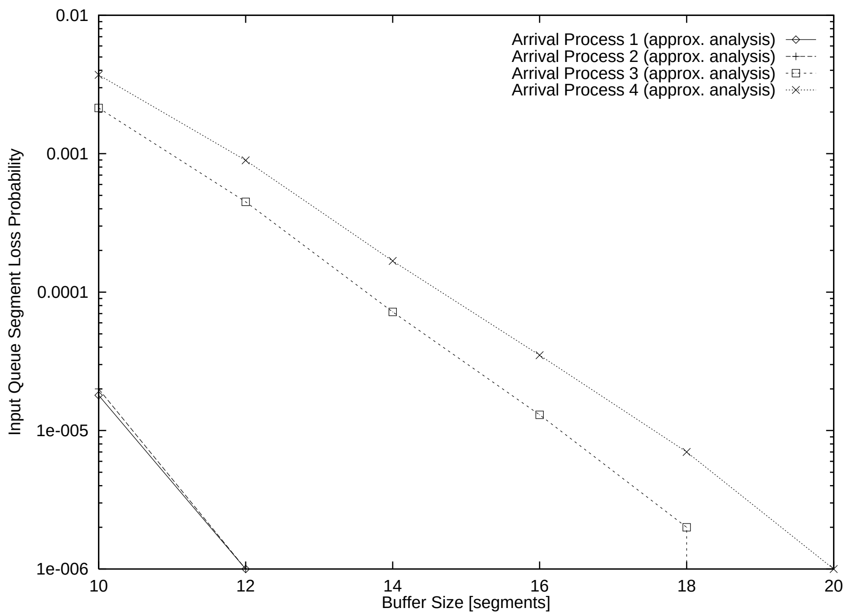

Figure 8: Input Queue Segment Loss Probability $\omega_{1,1}$ for $C = 2$

Figures 8 and 9 show the segment loss probabilities for input queue $(1, 1)$ for the four different arrival processes as a function of buffer size B in the two and three wavelength switches. We present only the input queue in input port 1 which corresponds to the the first wavelength, λ_1, as it is representative of the other input queues. Notice that, in 8 for Arrival Process 3 and $B = 20$ and in 9 for Arrival Process 2 and $B = 18$, the loss probabilities are rounded down from being below 1×10^{-6} to exactly 0.0; on the logarithmic scale, the curve therefore appears as a vertical line from the previous non-zero data point.

Results are not presented for packet loss probability since they are identical to segment loss probability. The number of segments arriving to (lost by) an input queue is related to the number of packets arriving to (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 input queue, packet and segment loss probabilities are essentially equal.

We observe that packet and segment loss are, therefore, dependent on a new packet arriving to a queue at a particular point in the schedule and finding the queue vulnerable to overflow at some point in receiving the packet. Consider the case in which $C = 2$. Packets are received at any of the queues at a rate of 4 segments per frame. Further, 2 segments are serviced in every frame due to the structure of our schedule. Observe the effect of the arrival of a 5 segment packet at

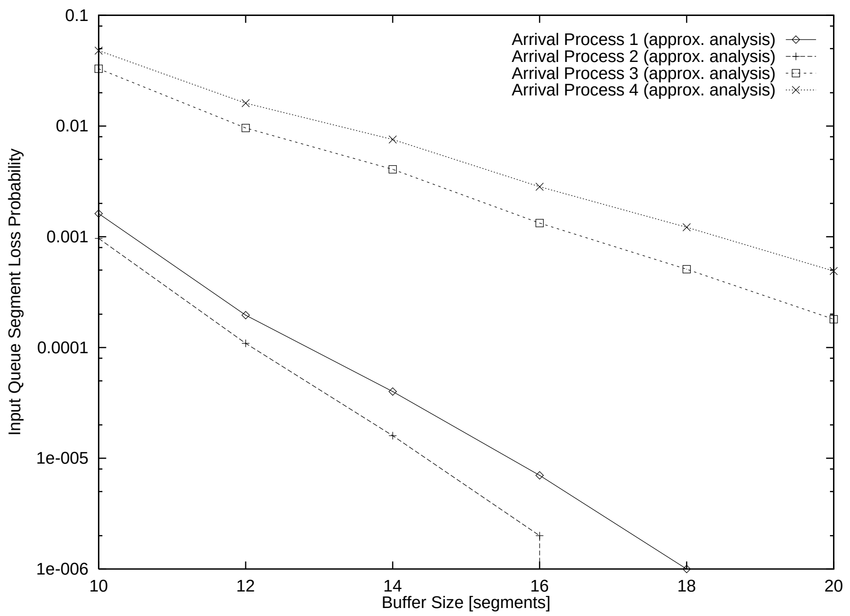

Figure 9: Input Queue Segment Loss Probability $\omega_{1,1}$ for $C = 3$

various times during the frame on the receiving queue's occupancy. If the packet arrives 1, 2, or 3 arrival slots before the queue's service period, the buffer's occupancy will temporarily rise as much as three segments above its occupancy level when the packet began to be received by the buffer. However, if the packet arrives when the queue's service period begins, the buffer's occupancy will temporarily rise only two segments above the buffer's initial occupancy level. Under the same conditions, observe that a six segment packet may similarly cause an increase in occupancy by as much as either two or four segments, depending on when during the frame the packet's arrival began. This point accounts for the fact that it is possible that, for instance, a six segment packet may be able to be received by a buffer which has a higher occupancy than that required by a five segment packet which is presented at a different point in the schedule [4]. It is for this reason that the loss measures for Arrival Process 2 are lower than those of Arrival Process 1 in Figure 9: given the schedules used in these experiments, it is more likely that the five segment packets will cause the buffer occupancies to momentarily rise to higher levels than six and seven segment packets. In the remainder of the cases, however, loss measures generally increased with average packet sizes.

A 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

[4] Note that both seven and eight segment packets may cause an increase of between three and five segments.

Table 5: Mean Absolute Input 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}

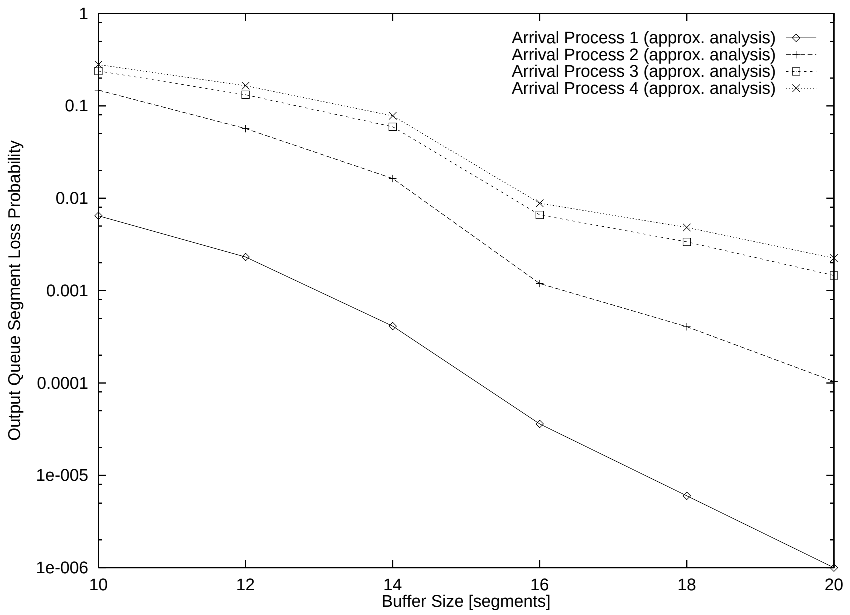

Figure 10: Output Port Segment Loss Probability ω_1 for $C = 2$

the mean errors associated with each curve shown in Figures 8 and 9[5].

The errors which are observed are generally felt to be reasonable given the possible variability of the simulation. The authors feel that Arrival Process 2 is the most unrealistic of the four as it is highly unlikely for packet sizes to be completely uniformly distributed. Instead, it is generally considered that the vast majority of packets are sized according to a maximum transmission unit depending on the protocol being used, hence our interest in Arrival Processes 1, 3, and 4.

Figures 10 and 11 show the output port segment loss 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, losses tended to decrease with buffer size and as the average packet size decreased. A

[5]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).

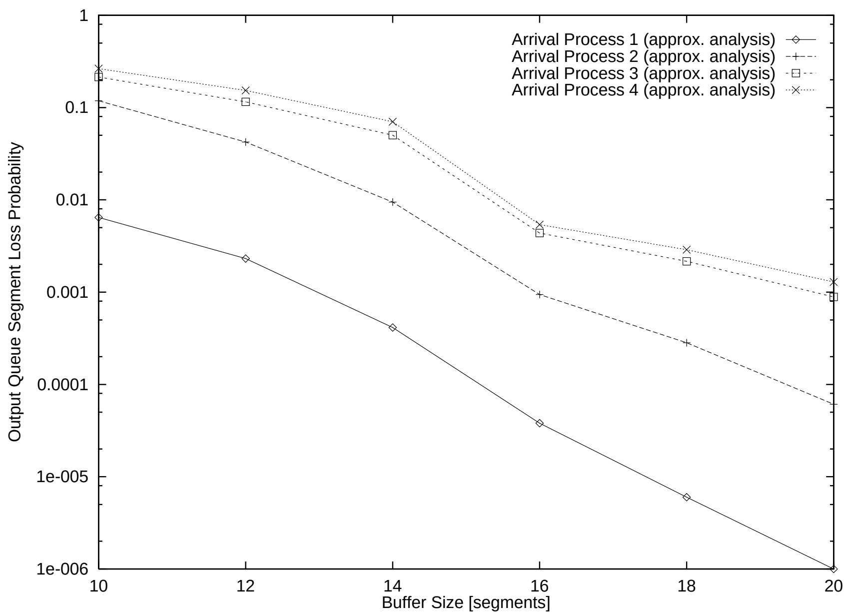

Figure 11: Output Port Segment Loss Probability ω_1 for $C = 3$

Table 6: Mean Absolute Output 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}

significant drop in loss probabilities is obvious as B increased from 14 to 16. This is due to the use of probabilities based on steady state values (in $\Phi_{ij}(x, \zeta)$) in our heuristic for $L_i(\cdot)$ in 21 along with the fact that the arrival processes are heavily weighted for 8 segment packets. It is reasonable, therefore, that our calculations would show a significant drop in loss when the buffer size increased to a point that it could accommodate two complete 8 segment packets.

Confidence intervals derived from simulation results were, again, not shown as part of Figures 10 and 11 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 output analysis is much less accurate than the input 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 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 have presented an extension of the basic TTFR switch architecture to allow for variable length transmission units (i.e., packets). Using this architecture, we have significantly extended previous performance analyses of single-hop photonic architectures. In this work, we saw the significant impact that traffic patterns, in addition to the previously mentioned parameters, would have on this type of photonic switch. Issues associated with using steady-state values under highly correlated traffic patterns are highlighted, justifying the use of a heuristic for calculating loss measures on the output side of the switch in Sections 3 and 4. Finally, we have presented numerical results for comparison with our approximation. These examples emphasize both (1) the complex interaction between the traffic patterns, switch parameters, and the schedule characteristics and (2) the impact that the independence assumptions which were made in this analysis can have on applications of this work.

References

[1] I. M. I. Habbab, M. Kavehrad, and C.-E. W. Sundberg. Protocols for very high-speed optical fiber local area networks using a passive star topology. *Journal of Lightwave Technology*, LT-5(12):1782–1793, December 1987.

[2] N. Mehravari. Performance and protocol improvements for very high-speed optical fiber local area networks using a passive star topology. *Journal of Lightwave Technology*, 8(4):520–530, April 1990.

[3] I. Chlamtac and A. Ganz. Channel allocation protocols in frequency-time controlled high speed networks. *IEEE Transactions on Communications*, 36(4):430–440, April 1988.

[4] P. W. Dowd. Random access protocols for high speed interprocessor communication based on an optical passive star topology. *Journal of Lightwave Technology*, LT-9:799–808, June 1991.

[5] A. Ganz. End-to-end protocols for WDM star networks. In *IFIP/WG6.1-WG6.4 Workshop on Protocols for High-Speed Networks*, pages 219–235, May 1989.

[6] A. Ganz and Z. Koren. WDM passive star - protocols and performance analysis. In *Proceedings of INFOCOM '91*, pages 991–1000. IEEE, April 1991.

[7] S. Tridandapani, J. S. Meditch, and A. K. Somani. The MaTPi protocol: Masking tuning times through pipelining in WDM optical networks. In *Proceedings of INFOCOM '94*, pages 1528–1535. IEEE, June 1994.

[8] Mon-Song Chen, N. R. Dono, and R. Ramaswami. A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks. *IEEE Journal on Selected Areas in Communications*, 8(6):1048–1057, August 1990.

[9] R. Chipalkatti, Z. Zhang, and A. S. Acampora. Protocols for optical star-coupler network using WDM: Performance and complexity study. *IEEE Journal on Selected Areas in Communications*, 11(4):579–589, May 1993.

[10] E. M. Foo and T. G. Robertazzi. A distributed global queue transmission strategy for a WDM optical fiber network. In *Proceedings of INFOCOM '95*, pages 154–161. IEEE, April 1995.

[11] A. Muir and J. J. Garcia-Luna-Aceves. Distributed queue packet scheduling algorithms for WDM-based networks. In *Proceedings of INFOCOM '96*, pages 938–945. IEEE, March 1996.

[12] G. R. Pieris and G. H. Sasaki. Scheduling transmissions in WDM broadcast-and-select networks. *IEEE/ACM Transactions on Networking*, 2(2):105–110, April 1994.

[13] M. Azizoglu, R. A. Barry, and A. Mokhtar. Impact of tuning delay on the performance of bandwidth-limited optical broadcast networks with uniform traffic. *IEEE Journal on Selected Areas in Communications*, 14(5):935–944, June 1996.

[14] R. L. Cruz and J-T. Tsai. COD: Alternative architectures for high speed packet switching. *IEEE/ACM Transactions on Networking*, 4(1):11–21, February 1996.

[15] J. Jue, M. Borella, and B. Mukherjee. Performance analysis of the Rainbow WDM optical network prototype. *IEEE Journal Selected Areas in Communications*, 14(5):945–951, June 1996.

[16] V. Sivaraman and G. N. Rouskas. HiPeR-ℓ: A High Performance Reservation protocol with ℓook-ahead for broadcast WDM networks. In *Proceedings of INFOCOM '97*, pages 1272–1279. IEEE, April 1997.

[17] S. Subramanian, A. K. Somani, M. Azizoglu, and R. A. Barry. A performance model for wavelength conversion with non-poisson traffic. In *Proceedings of INFOCOM '97*, pages 500–507. IEEE, April 1997.

[18] M. W. McKinnon, G. N. Rouskas, and H. G. Perros. Performance analysis of a photonic single-hop ATM switch architecture with tunable transmitters and fixed frequency receivers. *Performance Evaluation*, 1998. (To appear).

[19] I. Baldine and G. N. Rouskas. Dynamic load balancing in broadcast WDM networks with tuning latencies. In *Proceedings of INFOCOM '97*. IEEE, March 1998.

[20] P. E. Green. *Fiber Optic Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

[21] G. N. Rouskas and V. Sivaraman. Packet scheduling in broadcast WDM networks with arbitrary transceiver tuning latencies. *IEEE/ACM Transactions on Networking*, 5(3):359–370, June 1997.

[22] W. Stewart. *Numerical Solutions of Markov Chains*. Princeton University Press, Princeton, New Jersey, 1994.