

Capacity Allocation and Contention Resolution in a Photonic Slot Routing All-Optical WDM Mesh Network

Hui Zang, Jason P. Jue, and Biswanath Mukherjee

Abstract—Photonic slot routing (PSR) has been proposed as an approach to implement an all-optical packet-switched network in a manner which is scalable and not overly complex. In PSR, packets are transmitted within a basic transport unit referred to as a photonic slot. The photonic slot is fixed in length and spans multiple wavelengths. Each photonic slot is routed through the network as a single entity; thus, individual wavelengths do not need to be multiplexed or demultiplexed at intermediate nodes through which the photonic slot is traversing. When implementing PSR in a mesh environment, a number of significant issues must be addressed. Two such issues are fairness and contention resolution. In this study, we propose a novel approach for allocating capacity on each link in a fair manner, and we investigate various approaches, such as buffering and deflection, for handling contention. We develop an analytical model to evaluate the performance of such networks, and validate the analysis through simulation. It is shown that the proposed capacity allocation approach can significantly reduce contention in the network and provide a fair allocation of bandwidth to each source–destination pair.

Index Terms—Contention resolution, deflection routing, optical buffering, packet-switching, photonic slot routing, wavelength division multiplexing.

I. INTRODUCTION

WAVELENGTH-DIVISION multiplexing (WDM) has been rapidly gaining acceptance as a means to handle the ever-increasing bandwidth demands of network users [1]. In addition to providing a huge amount of bandwidth, an all-optical WDM network also offers the benefit of high-speed data transmissions without electronic conversions at intermediate nodes. By transmitting the signal entirely in the optical domain, data transparency can be achieved and electronic processing costs may be reduced. One such type of all-optical network is the wavelength-routed WDM network, in which all-optical lightpaths are set up on specific wavelengths between pairs of nodes [2]–[6]. Some of the challenges in designing wavelength-routed WDM networks include the complexity and scalability issues which arise from the need to demultiplex and individually route each wavelength at a node.

Manuscript received April 4, 2000; revised August 14, 2000. This work was supported in part by NSF Grants NCR-95-08239 and ANI-98-05285.

H. Zang and B. Mukherjee are with the Department of Computer Science, University of California, Davis, CA 95616 USA (e-mail: zang@ucdavis.edu; mukherje@ucdavis.edu).

J. P. Jue is with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080 USA (e-mail: jjue@utdallas.edu).

Publisher Item Identifier S 0733-8724(00)10341-X.

An emerging alternative to the wavelength-routed WDM network is a WDM network based on optical packet switching. One recently proposed approach to optical packet switching is Photonic Slot Routing (PSR) [7], [8]. In PSR, time is slotted, and data is transmitted in the form of photonic slots which are fixed in length and span across all wavelengths in the network. Each wavelength in the photonic slot may contain a single packet, and all packets in the photonic slot are destined to the same node. By requiring the packets to have the same destination, the photonic slot may be routed as a single integrated unit without the need for demultiplexing individual wavelengths at intermediate nodes. Thus, wavelength-insensitive components may be used at each node, resulting in less complexity, faster routing, and lower network cost [7], [9].

A number of alternative approaches to optical packet switching have been proposed in which packets on each wavelength are routed independently, or in which wavelengths are used for contention resolution [10], [11]; however, these approaches tend to be more expensive in terms of hardware, requiring tunable wavelength converters, larger switch fabrics, and additional multiplexing and demultiplexing elements at each node.

Although PSR has been studied in bus and ring networks [7], [9], it has not been considered in mesh networks until recently [8]. In [8], three operations, namely, slot switching, slot merging, and slot copying are allowed at a switching node. Packets are transmitted by following a TDM schedule, which is constructed with a graph coloring approach. In our study, we also consider the application of PSR to an arbitrary mesh network. In a mesh network, nodes may have multiple input and output ports, resulting in a higher degree of contention than in ring-based networks. We propose a novel PSR protocol which utilizes simple capacity-allocation and slot-assignment algorithms, and we develop an analytical model to measure the performance of the protocol in a network environment. We investigate various approaches for reducing the amount of contention, and for resolving contentions when they occur. We also outline approaches for dealing with issues such as synchronization.

In Section II, we describe the proposed PSR network architecture for arbitrary mesh networks. The protocols for transmitting packets and for resolving contentions are discussed in Section III. Section IV provides a detailed analytical model for measuring the performance of the proposed network. We validate the analysis through simulation and present numerical examples in

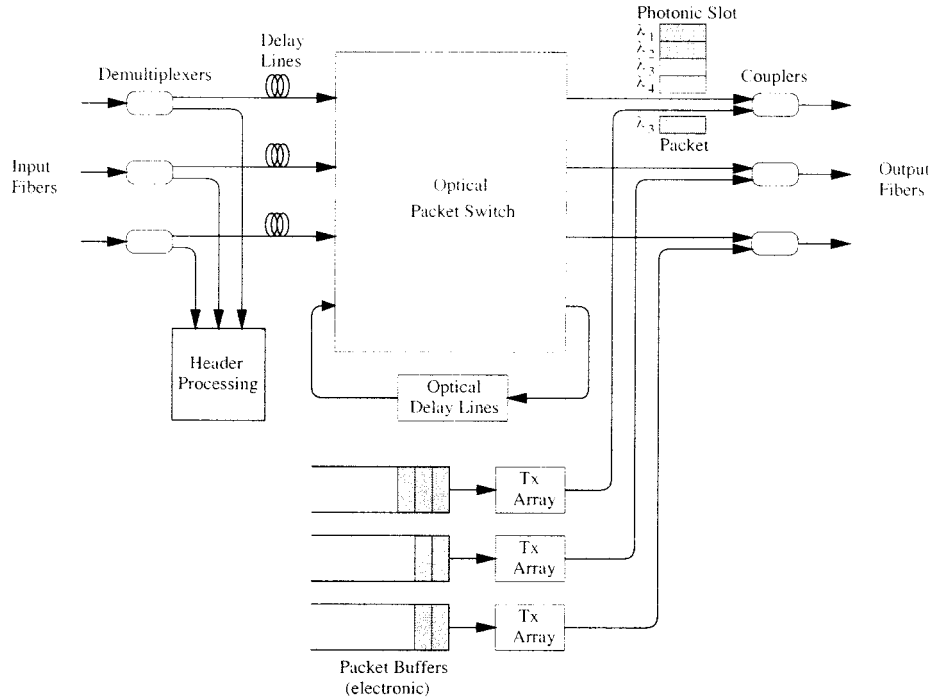


Fig. 1. PSR node architecture.

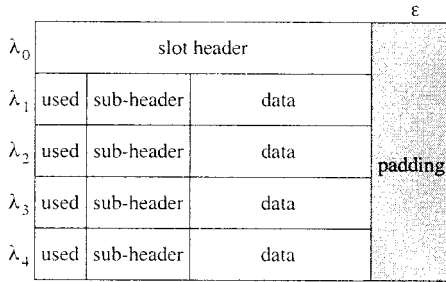


Fig. 2. Format of a photonic slot.

Section V. Section VI concludes the paper and discusses areas for future research.

II. NETWORK ARCHITECTURE

In this work, we consider a mesh interconnected network. Each node consists of a wavelength-insensitive optical packet switch, optical buffers consisting of delay lines (to hold photonic slots at intermediate nodes), and electronic packet buffers (to hold packet arrivals at each source node). Each node (while acting as a source) maintains a separate packet buffer for each destination node. A diagram of the node architecture is shown in Fig. 1.

A photonic slot (Fig. 2) spans all of the WDM channels (wavelengths) in the network, and each channel can carry a single packet. At each input fiber link, an optical splitter (or demultiplexer) is used to extract the header of each photonic slot. The header, which may be carried on a separate wavelength (see Fig. 2), contains information such as the destination of the slot, and which wavelengths in the slot are occupied by packets. The destination information is used to determine/configure the switch setting so that the slot can be appropriately routed

toward its destination using a standard routing algorithm (e.g., shortest-path routing). The slot-occupancy information determines the extra allowable packet transmissions for each outgoing photonic slot. Because it takes some time to process the slot header and to configure the optical packet switch, delay lines may be required on each input fiber link, as shown in Fig. 1. Alternatively, the slot header may precede the data payload of the slot by some fixed duration.

On a given output fiber link, the node may insert packets into existing photonic slots which are headed for the same destination, or the node may transmit newly created photonic slots if no other slots are contending for the link. The packet insertion may be performed by an optical coupler. For example, in Fig. 1, a photonic slot departing on the top output fiber link contains packets on wavelengths λ_1 and λ_2 . The node may then insert a packet into the photonic slot on wavelength λ_3 .

In the given network architecture, there exists the possibility of contention between two or more photonic slots at each of the output fiber links. One approach for resolving contention is to buffer the photonic slots optically when they lose a contention. In this case, optical delay line buffers, consisting of lengths of fiber, are required at each node [12]. Alternative approaches for dealing with contention include deflection routing and dropping photonic slots. We will discuss these approaches in Section III, and we will also investigate approaches for minimizing contention.

One of the challenges in implementing PSR in a mesh environment is maintaining the synchronization of photonic slots at each node. Each of the incoming photonic slots must arrive simultaneously on all input fibers in order to be routed through the optical packet switch at the same time. We address this problem by requiring the length of each fiber link (including nodal processing delays) to be an integer multiple of the photonic-slot

size. One approach for adjusting the fiber length is to include adjustable delay elements at the input ports of each node. An example of a variable-delay line device for optical packet synchronization is presented in [13].

Another physical issue in PSR is dispersion, which causes each wavelength to travel at a different speed along a fiber link, thus resulting in the “dilation” or spreading of a photonic slot. In order to compensate for slot dilation, padding can be added to extend the photonic slot length by some amount ϵ (see Fig. 2). A limitation of this approach is that it will result in reduced network utilization. The value of ϵ will increase with the diameter of the network; thus, in order to maintain reasonable utilization, we may need to limit the size of the network.

III. PSR PROTOCOLS

Transmitter Protocol: When a photonic slot is received by a node, the node can add packets to the slot if not all of the wavelengths are in use. The transmitter protocol can be partitioned into two steps: hunt and add.

- Hunt: When a node has a packet to transmit to destination Node i , it hunts for a passing slot which is either:
 - 1) empty, and has been assigned to destination Node i according to some slot-assignment policy (such slot-assignment policies will be discussed shortly), or
 - 2) destined to Node i and is not full.
- Add: When an appropriate slot has been found, the node inserts its packets into the slot following some packet-insertion protocol. If the slot is empty and does not have a destination assigned to it, then the node sets the slot destination to the destination of its packets.

We discuss two types of slot-assignment policies and their performance in Section III-A, and packet-insertion policies, which determine the number of packets to be inserted into a slot, are discussed in Section III-B.

Routing and Contention Protocols: Nodes which receive forwarded slots from upstream nodes are referred to as *intermediate nodes*. A basic task for an intermediate node is to route nonempty slots to the appropriate output link. Ideally, the intermediate node should put the slot onto the proper link as determined by the routing algorithm.¹ If two or more slots arrive at the same time and require the same outgoing fiber link, then contention occurs. In the case of contention, only one of the contending slots can be routed properly. A slot that does not win the contention may be handled in a number of different ways:

- the slot may be buffered in an optical buffer;
- it may be deflected to another outgoing link with the hope that the slot will eventually make it to the destination; or
- the slot may be dropped.

In practice, a combination of these approaches may be used. When the slot buffer is full or a slot has been deflected up to a certain number of times (a predetermined threshold), the slot will be dropped. Buffering can also be combined with deflection

in a hybrid scheme. We will explain and discuss the performance of these schemes in Section III-C.

A. Slot-Assignment and Capacity-Allocation Algorithms

We study two types of slot-assignment policies in this section. First, we consider a slot-assignment algorithm based on packet arrivals. We then consider a slot-assignment algorithm based on capacity allocation.

In both schemes, a node which has packets to transmit can always add its packets to slots which are headed to the same destination node. However, when a node receives an empty slot, two decisions must be made:

- whether or not to transmit packets within this slot; and
- which destination should be assigned to this slot (if a destination has not already been assigned).

The order of these two decisions depends on which of the slot-assignment schemes is being implemented. If we have slot-assignment based on packet arrivals (Section III-A1), the former should be decided first; if we have slot-assignment based on capacity allocation (Section III-A2), the latter should be decided first.

1) *Slot Assignment Based on Packet Arrivals:* In the slot-assignment algorithm based on packet arrivals, no knowledge of the network traffic pattern is required. Upon receiving an empty slot, a node randomly chooses one queue from among its packet queues which are not empty. It then inserts a number of packets into the slot. The number of packets inserted into the slot will be determined by a packet-insertion policy. (We will discuss packet-insertion policies in Section III-B. Let us assume for now that we are using *Greedy Insertion* in which the node places as many packets as possible, which are headed to the same destination, into the slot.) Thus, if the queues for a given link² are not all empty at the beginning of a time slot, then the slot departing on this link will always have a destination assigned to it, and the slot will also be nonempty.

This algorithm works well under low load, as we shall shortly observe; however, under high load, it results in a high probability of contention as well as unfairness in resource allocation. Nodes located toward the edge of the network, which have little or no chance of being intermediate nodes, always receive empty slots into which they can transmit their packets. For a (tagged) node located in the internal region of the network, the passing slots may be empty with lower probability; thus, the node may need to add its packets more often to nonempty passing slots. Hence, this node has fewer opportunities to transmit its packets than nodes at the edge of the network. (Of course, when a slot which is destined to the tagged node is received by the tagged node, the tagged node may then generate a new empty slot in which it can transmit its packets.)

In addition to this unfairness issue, the slots transmitted by the nodes at the edge of the network may also result in increased contention at the tagged node to such a degree that the tagged node may be unable to promptly handle all of the slots that it receives. For example, consider the six-node network with appropriate link lengths shown in Fig. 3. Fixed shortest-path routing

¹For example, shortest-path routing, which is what we use throughout this study.

²If fixed routing is used, a packet queue always requires the same outgoing link.

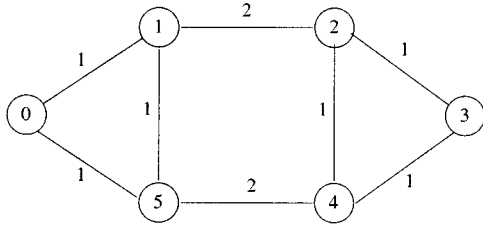


Fig. 3. Simulation Network 1.

		Destination Node					
		0	1	2	3	4	5
Current Node	0	-	1	1	1	5	5
	1	0	-	2	2	5	5
	2	1	1	-	3	4	4
	3	4	2	2	-	4	4
	4	5	2	2	3	-	5
	5	0	1	1	4	4	-

Fig. 4. Routing matrix for Network 1.

is used and the routing matrix is shown in Fig. 4. Note that the numbers in the routing matrix are identifiers of the next node for the routing. Also, since the routing tables are assumed to be static, all packets from a particular source node to a particular destination node will follow the same route. We can see from the routing matrix that, from Node 0, the traffic to Nodes 2 and 3 are routed over the link from Node 1 to 2, which we will denote as Link $1 \rightarrow 2$. The traffic from Node 5 to Node 2 also traverses Link $1 \rightarrow 2$. Since Node 0 is at the edge of the network, it doesn't have to handle forwarded traffic. Thus, all of the traffic on Link $0 \rightarrow 1$ is traffic which has originated at Node 0, and Node 0 has full control over the destinations assigned to each of the slots on Link $0 \rightarrow 1$. Similarly, by examining the routing matrix, we observe that the traffic on Link $5 \rightarrow 1$ consists only of slots which have originated at Node 5. If Nodes 0 and 5 both have a high volume of traffic to Node 2, and if Nodes 0 and 5 are allowed to assign all of the slots on Links $0 \rightarrow 1$ and $5 \rightarrow 1$ to destination Node 2, then Node 1 may be unable to handle all of the slots headed for Link $1 \rightarrow 2$, in which case the system may become unstable (if slot buffer size is infinite), or slots may have to be dropped at Node 1 (if slot buffer size is finite). Also, since all of the slots on Link $1 \rightarrow 2$ are destined for Node 2, Node 1 may not be able to transmit packets to Node 3, resulting in some degree of unfairness.

Although the packet-arrival based slot-assignment scheme leads to a high degree of contention and may be of limited practical utility, we have included it in order to provide a baseline for comparison. Also, because of its simplicity and contention-intensive characteristics, it can be used to study the effectiveness of contention-resolution schemes.

As an alternative to packet-arrival-based slot allocation, we can assign destinations to the empty slots in such a way that the

upstream nodes will never overwhelm the downstream nodes. This approach is based on capacity allocation.

2) *Slot Assignment Based on Capacity Allocation*: Chlamtac *et al.* [9] proposed a transmission control protocol based on slot preassignment for PSR in optical networks with a ring topology. In that approach, a TDM frame consists of L slots, and the source and destination of each slot in the TDM frame is determined by a network-wide TDM schedule. The number of slots in the TDM frame assigned to each source–destination pair is determined in such a way that fair bandwidth distribution among the source–destination pairs, as well as contention-free slot routing at intermediate nodes, are achieved. A similar protocol which applies to mesh networks is proposed later in [8].

In a PSR-based mesh network, the objectives of our slot-assignment policy are:

- to maximize throughput;
- to minimize contention;
- to provide fair allocation of bandwidth.

The TDM frame construction problem is NP-hard [8]; thus it may be difficult to find a TDM frame which meets all of these objectives. An alternative to generating TDM frames in a mesh network is to assign destinations probabilistically to arriving slots based on the capacity allocation results. The slot assignment consists of two steps:

- 1) Capacity-Allocation step which determines the fraction a_{jl} of capacity (or slots) on each link l that a source node i should assign to destination j ;
- 2) Slot-Assignment step in which a node assigns a destination to a slot based on the capacity allocation and a randomly-generated number.

Note that the results obtained from the capacity-allocation and slot-assignment algorithms may not be optimal. Contention may still occur and not all source–destination pairs will be allocated an equal amount of network capacity. However, as we will see, with very low computation cost, these algorithms can achieve a significant improvement over a greedy approach in terms of contention and fairness. The algorithms also work fairly well combined with different contention-resolution schemes, which we will present later.

3) *The Capacity-Allocation Algorithm*: The capacity-allocation algorithm is based on the Max-Min Flow Control algorithm [14]. Inputs to the algorithm are the traffic matrix and the routing matrix of the network. We briefly describe the algorithm below. Given traffic matrix T , with t_{ij} denoting the arrival rate to node i of packets destined to node j , and the capacity-allocation matrix C , with c_{ijl} denoting the fraction of slots going out on link l which are allocated to source–destination pair (i, j) , c_{ijl} should satisfy

$$c_{ijl_1} = c_{ijl_2} = c_{ij} > 0, c_{ijl_3} = 0 \quad (1)$$

if both link l_1 and link l_2 are on path $i \xrightarrow{*} j$ and link l_3 is not.³ Equation (1) is called the *flow conservation constraint*.

³The notation $i \xrightarrow{*} j$ indicates the path from i to j which may contain multiple links, i.e., $i \xrightarrow{*} j \equiv i \rightarrow \dots \rightarrow j$.

For a node i , the probability with which it assigns destination j to an empty slot going out on link l is given by

$$Pr_{jl} = \frac{c_{ij}}{\sum_k c_{ik}}. \quad (2)$$

The sum in the denominator is over all nodes k such that path $i \xrightarrow{*} k$ goes through link l . That is, the probability equals the capacity allocated to path $i \xrightarrow{*} j$ divided by the sum of all capacities allocated to the paths starting at link l .

The algorithm to calculate c_{ij} for all sources i and destinations j is as follows:

- **Step 1.** For each link l , let A_l denote the available capacity on link l . Set $A_l = 1$.
- **Step 2.** For each link l , the fraction of capacity available to path $i \xrightarrow{*} j$ which contains link l is

$$\frac{t_{ij}}{\sum_{xy} t_{xy}}, \quad (3)$$

with the sum in the denominator being over all paths containing link l and which have *not* been allocated yet. Calculate the minimum available capacity over all the paths $i \xrightarrow{*} j$ containing link l , and assign this value, denoted by MC_l , to link l as the minimum available capacity to a path on link l that has not been allocated, i.e.,

$$MC_l = \min_{i,j} A_l \times \frac{t_{ij}}{\sum_{xy} t_{xy}}. \quad (4)$$

- **Step 3.** Let

$$c_{i'j'} = \min_l MC_l, \quad (5)$$

where $i' \xrightarrow{*} j'$ is the path containing link l' with the minimum MC_l over all links l , and path $i' \xrightarrow{*} j'$'s available capacity is the minimum on link l . Assign $c_{i'j'}$ to path $i' \xrightarrow{*} j'$.

For all links l'' contained in path $i' \xrightarrow{*} j'$, $A_{l''} \leftarrow A_{l''} - c_{i'j'}$, according to the *flow conservation constraint*.

If all paths are allocated, stop. Otherwise go to **Step 2**.

In the above algorithm, a path can never be allocated a capacity which causes the capacity allocated on any given link to exceed 1. Also, in this approach, a path consisting of only a single link is allocated the remaining capacity on that link after all of the other paths containing this link have been allocated. In this way, the capacity of each link is fully utilized and the capacity allocated to the one-link path is not limited by the traffic load.

Given the capacity allocation for all paths, we can then calculate the probability that a node i will assign destination j to an empty slot going out on link l by using (2). Upon generating a new empty slot on a link, a node generates a random number uniformly distributed over $(0, 1)$ and decides which destination to assign to that empty slot based on the probabilities from (2). If

no packets are available for this destination, the node can put the empty slot onto the link either with the destination unassigned, or with the destination assigned.

Under very low load, this approach results in a larger delay than for the packet-arrival-based scheme because a node may fail to place packets into an empty slot on an outgoing link even if the node has packets for that link. This situation occurs if the slot has been assigned a destination for which the node doesn't have any packets, but which utilizes the same outgoing link. Also, as we shall show, this capacity-allocation algorithm provides more transmission opportunities for internal nodes than for nodes at the edge of the network. Fairness is not completely achieved. We have compared the fairness of this approach with the slot-assignment policy based on packet arrivals. Our results show that the capacity-allocation approach is fair up to a high load, while the fairness of the packet-arrival-based approach decreases rapidly when load increases. Another advantage of the capacity-allocation scheme is that it decreases contention remarkably. With capacity allocation, it is possible to use a small optical buffer for slots which cannot be routed properly, and still maintain a very low slot-drop rate. Overall, the capacity-allocation approach has better performance under moderate and higher load.

For the network to be stable, the maximum arrival rate of packets to source i which are destined for node j should be less than the capacity allocated for this path, c_{ij} , times the number of packets that a node can fill in a slot. Usually, this threshold is higher for the capacity-allocation approach than for the packet-arrival-based approach.

This probabilistic slot-assignment scheme has some similarities to the probabilistic feature of the *Distributed Queue Dual Bus (DQDB)* architecture [14]; however, in our approach, each node has a different probability of using an empty slot which is determined by the overall network traffic demands, while in DQDB, each node has the same probability $1 - f$, $0 \leq f \leq 1$ of using an empty slot.

B. Policies for Inserting Packets into Slots

Policies are needed to determine how many packets should be placed into a slot, and when to fill a slot. There are several variations of policies which, together with the transmitter protocols, have different effects on fairness, packet delay, network throughput, and contention probability.

Here, we discuss three policies:

- **Greedy Insertion.** A transmitter fills a slot whenever it has packets to transmit, and it fills the slot with as many packets as the slot can hold.
- **Lower-Bound Insertion.** A transmitter cannot fill an empty slot with packets unless it has *at least* a certain number of packets for the same destination. The transmitter follows the Greedy-Insertion approach for placing packets in nonempty slots.
- **Upper-Bound Insertion.** A transmitter can only fill a slot with *up to* a certain number of packets. The upper bound for the number of packets may differ depending on whether the slot is empty or not, and on the position of the node in the path to the destination.

Greedy Insertion performs better under low load by fully utilizing the network resources. Under higher load, it will result in increased contention at intermediate nodes and also result in fewer transmission opportunities for intermediate nodes.

Lower-Bound Insertion will avoid contention up to some load limit. Every node waits for a certain number of packets to arrive before it starts to transmit. Hence, the number of nonempty slots generated from a source node is less than the case in which there is no lower bound. When the load limit is exceeded, a node will always receive enough packets to exceed the lower bound in the time between slot arrivals. Therefore, this policy becomes Greedy Insertion under high load.

Upper-Bound Insertion, which limits the number of packets that a node may transmit in a slot, provides a greater number of transmission opportunities for downstream nodes, thereby improving fairness. In order to determine the degree to which a node can fill a slot, it may be necessary to determine the position of a node in the path to the destination. However, this approach would require too much information for a node to collect and remember. Also, even with full knowledge of a node's location, network resources may be wasted under low load. In the remainder of this study, we will assume a Greedy-Insertion policy.

C. Schemes to Resolve Contention

When multiple slots contend for the same outgoing link, a node must not only decide which slot to transmit, but must also decide what to do with the remaining slots. We consider three schemes:

- **Scheme 1—Buffering.** Each node has a finite optical buffer (as in [12]), which consists of parallel optical delay lines, each of which has a propagation delay of 1 slot time. In the case of contention, slots which cannot be routed properly are buffered if the buffer is not full; otherwise, the slots are dropped. Whenever a slot is buffered at an intermediate node, one of the outgoing links will be idle. The node is then responsible for either routing a previously-buffered slot on this idle link, or generating an empty slot on this link. In this way, the total number of outgoing slots will be equal to the number of incoming slots at this node. If a slot buffered at a previous time is contending with a new incoming slot, then the node must decide whether to route the buffered slot or the new slot. If a first-come, first-served scheme is followed, then the previously-buffered slot should be routed, and the new slot should be buffered.
- **Scheme 2—Deflection.** By *deflection*, we mean that the slot is routed to some nonbusy link other than the link that is specified by the routing algorithm. Non-busy links are always available when contention occurs, because the number of incoming links and the number of outgoing links at a node are equal. Thus, if two slots are contending, then there will be at least one outgoing link which is free. Therefore, a slot can always be deflected if it cannot be routed properly. A problem which may arise is that a slot may be deflected multiple times and may remain in the network for a significant amount of time before being

routed to its correct destination. To prevent this slot from using up too much network bandwidth and thereby reducing network throughput, the slot will be dropped if it has been deflected up to some maximum number of times (a predetermined threshold). A deflection counter is added into the header field of a slot. Each time a slot is deflected, the counter is updated.

- **Scheme 3—Hybrid scheme.** In the hybrid scheme, each node has a finite optical buffer. In the case of contention at a node, a slot will be buffered if the buffer at the node is not full; otherwise, the slot will be deflected. If the slot has been in the network for some maximum amount of time, it will be dropped. In this scheme, the probability of dropping a slot is naturally lower than in the buffering scheme.

We have implemented Scheme 1 and Scheme 2 in our simulation studies, and we shall compare their performances in Section V.

IV. ANALYTICAL MODEL

In order to study the performance of the proposed PSR protocols in a WDM mesh network, an analytical model is developed for a network which utilizes a slot-assignment scheme based on capacity allocation and a greedy approach for filling slots. Following are the model parameters:

- N Number of nodes in the network.
- W Number of wavelengths for data packet transmission (excluding any separate wavelength for the header). W is also equal to the maximum number of packets that a slot can hold.
- Λ Traffic matrix.
- R Routing matrix. R_{ij} indicates the next node in the route from a Node i to reach destination Node j . The routing matrix along with the traffic matrix is used to calculate the fraction of slots on each link for a given destination.
- L Link distance matrix. L_{ij} is the length of the link between Node i and Node j if such a link exists, otherwise it is ∞ . The link distance matrix is used to find the shortest path between any pair of nodes. This information is then incorporated into the routing matrix.

The slot transmission rate on each link is assumed to be normalized to one slot per unit time. The packet buffers at each node are modeled as Poisson arrival/batch service queues. Each queue is identified by its source node s and destination node d , and the input to each queue (s, d) is assumed to be Poisson with rate λ_{sd} . The service discipline for a queue at a given node is determined by the rate of incoming slots which are headed for the same destination, the number of packets already in these slots, and the rate of new slots which are created by the node and addressed to the appropriate destination. For tractability, the packet queue size is limited to hold at most B_p packets. This assumption of packet queue size may be different in some of the simulation models.

Note that, for a given destination d , the combined routes from each node to this destination form a reverse shortest-path tree rooted at d (see Fig. 5). The leaves of this (reverse) tree will be

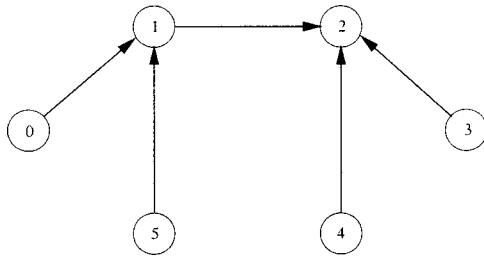


Fig. 5. The reverse shortest-path tree for destination Node 2. Nodes 0, 3, 4, and 5 are source nodes, and Node 1 is an intermediate node.

called *source nodes* for destination d , since they do not receive any upstream slots for d and must create new slots. Nodes other than the source and destination nodes will be referred to as *intermediate nodes*. Intermediate nodes may place packets in slots received from upstream nodes, or may create new slots. Thus, the slots on a given link l which are destined for node d may be from a number of different source nodes.

We define the following terms:

r_l^d The probability that a photonic slot destined for node d is transmitted on link l in a given time slot. This probability is assumed to be the same as the fraction of capacity allocated on link l to destination d , and can be calculated using the approach described in Section III.

$r_{l_{new}}^d$ The probability that a new photonic slot for destination d is generated at the node that sources link l in a given time slot.

$P_l^d(n)$ The probability that a slot on link l destined for node d contains n packets, where $n = 0, 1, 2, \dots, W$.

$P_l^{d'}(n)$ The probability that an incoming slot or a new slot destined for node d contains n packets before any packets are added at the node that is sourcing link l .

$Q_l^d(n)$ The probability that the queue for packets destined to Node d located at the node sourcing link l contains n packets immediately after a slot has departed.

For each link l , $P_l^d(n)$ can be calculated in terms of the packet occupancy of arriving or new slots, the number of packets queued for destination d after the previous slot's departure, and the number of arrivals since the last slot departure, i.e.,

$$P_l^d(n) = \sum_{i=0}^n P_l^{d'}(i) \cdot \sum_{j=0}^{n-i} Q_l^d(j) \cdot \sum_{t=1}^{\infty} e^{-\lambda t} \frac{(\lambda t)^{n-i-j}}{(n-i-j)!} \cdot r_l^d (1 - r_l^d)^{t-1} \quad \text{for } n < W, \quad (6)$$

and

$$P_l^d(W) = P_l^{d'}(W) + \sum_{i=0}^{W-1} P_l^{d'}(i) \cdot \left[\sum_{j=0}^{W-i-1} Q_l^d(j) \cdot \sum_{k=W-i-j}^{\infty} \sum_{t=1}^{\infty} e^{-\lambda t} \frac{(\lambda t)^k}{k!} \cdot r_l^d (1 - r_l^d)^{t-1} \right]. \quad (7)$$

Equation (6) is based on the fact that an outgoing slot on link l destined for node d will contain n packets if: 1) the slot arrived to the current node containing i packets, 2) the current node had j packets remaining in its packet buffer following the departure of the previous slot to destination d , and 3) $n-i-j$ packets arrived to the current node since the departure of the previous slot to destination d . The summation term over t gives the probability that $n-i-j$ packets have arrived to the queue since the last slot departure, where the last slot departure occurred t time slots ago. The term $r_l^d (1 - r_l^d)^{t-1}$ is the probability that the last slot departure for destination d occurred t time slots ago.

Equation (7) yields the probability that the number of packets already in the photonic slot plus the number of packets waiting in the packet buffer at the current node is greater than or equal to W , the total number of wavelengths in the slot. In this case, the photonic slot will depart the node with its maximum capacity of W packets.

$P_l^{d'}(n)$ is calculated by averaging the distribution of packets over all incoming links which are forwarding slots to link l . Let \mathbf{K} be the set of such links. Then,

$$P_l^{d'}(n) = \frac{1}{r_l^d} \left[r_{l_{new}}^d \cdot u(n) + \sum_{k \in \mathbf{K}} r_k^d \cdot P_k^d(n) \right] \quad (8)$$

where

$$u(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Note that, because of the way the r 's are generated, the following relation holds:

$$r_l^d = r_{l_{new}}^d + \sum_{k \in \mathbf{K}} r_k^d. \quad (9)$$

Source nodes which do not receive any upstream slots destined for d only have new slots. For these nodes, we can write

$$P_l^{d'}(n) = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{if } n > 0. \end{cases}$$

Therefore, the above equations can be solved by starting from the source nodes and working toward the destination node along the destination's reverse shortest-path tree (Fig. 5).

The $Q_l^d(n)$ are found by considering the embedded discrete-time Markov chain for the queue size at the slot-departure instants. It is assumed (as an approximation for analytical tractability) that the slot-interdeparture times are geometrically distributed with parameter r_l^d . This assumption of geometrically distributed slot-interdeparture times is reasonable since each node assigns destinations to empty slots randomly based on the capacity allocation for a given destination on a given link. For example, if 10% of the capacity on link l is to be allocated for destination d according to the capacity-allocation algorithm, then the node will assign destination d to each empty slot with probability 0.10. Packet departures at the slot-departure instants are assumed to be in batches with the batch size being equal to either the number of spaces available in the slot, or the number of packets in the queue, whichever is smaller. Let N_E be the number of empty spaces available in the slot, and N_A be the number of arrivals since the last slot departure. We define $p_E(n) = \Pr(N_E = n)$ as the probability of n empty spaces

in a slot, and $p_A(n) = \Pr(N_A = n)$ as the probability of n arrivals since the last slot departure. Then, we can write

$$p_E(n) = P_l^d(W - n) \quad (10)$$

and

$$p_A(n) = \sum_{t=1}^{\infty} e^{-\lambda t} \frac{(\lambda t)^n}{n!} r_l^d (1 - r_l^d)^{t-1}. \quad (11)$$

Following are the transition probabilities for the Markov chain:

For $i = j = 0$,

$$p_{ij} = \sum_{n=0}^W \Pr(N_E \geq n) p_A(n).$$

For $0 \leq i \leq j$, $0 < j < B_p - W$,

$$p_{ij} = \sum_{n=0}^W p_E(n) p_A(n - i + j).$$

For $0 \leq i \leq j$, $B_p - W \leq j < B_p$,

$$p_{ij} = p_E(B_p - j) \Pr(N_A \geq B_p - i) + \sum_{n=0}^{B_p - j - 1} p_E(n) p_A(n - i + j).$$

For $0 \leq i \leq j$, $j = B_p$,

$$p_{ij} = p_E(0) \Pr(N_A \geq B_p - i).$$

For $j < i \leq j + W$, $j = 0$,

$$p_{ij} = \sum_{n=i}^W \Pr(N_E \geq n) p_A(n - i).$$

For $j < i \leq j + W$, $0 < j < B_p - W$,

$$p_{ij} = \sum_{n=i-j}^W p_E(n) p_A(n - i + j).$$

For $j < i \leq j + W$, $B_p - W \leq j < B_p$,

$$p_{ij} = p_E(B_p - j) \Pr(N_A \geq B_p - i) + \sum_{n=i-j}^{B_p - j - 1} p_E(n) p_A(n - i + j).$$

For $i > j + W$,

$$p_{ij} = 0. \quad (12)$$

From the transition probability matrix $\mathbf{P} = \{p_{ij}\}$, we can solve for the steady-state probabilities $Q_l^d(n)$. The expected number of packets in the queue at a slot departure is then given by

$$N_{qd} = \sum_{n=0}^{B_p} n \cdot Q_l^d(n), \quad (13)$$

and using Little's result, the waiting time in the queue is:

$$W_{ld} = N_{qd} / \lambda' + \frac{1}{r_l^d}, \quad (14)$$

where $\lambda' = \lambda \cdot (1 - Q_l^d(B_p))$, and $1/r_l^d$ is the expected amount of time until the next slot departure.

The total delay is the sum of the amount of time spent in the queue, the transmission time at the source node and intermediate nodes, and the propagation delay from the source node to the destination node

$$D_{ld} = W_{ld} + T_{tx} + T_{prop}. \quad (15)$$

The transmission time T_{tx} is the time required to transmit a slot multiplied by the number of links that the slot traverses. The number of links traversed can be determined from the routing matrix \mathbf{R} . The propagation delay T_{prop} can be found from the routing matrix and the link distance matrix \mathbf{L} . To find the average delay for the entire system, we average the delays of all the queues

$$D_{avg} = \sum_l \sum_{d=1}^N D_{ld} \cdot \frac{1}{N(N-1)}. \quad (16)$$

Utilization is calculated by averaging over the distribution of packets in each slot

$$U = \frac{1}{N_{links}} \sum_l \sum_d r_l^d \sum_{n=0}^W \frac{n}{W} \cdot P_l^d(n), \quad (17)$$

where N_{links} is the total number of links in the network.

The analytical model doesn't consider the delay due to photonic slot queueing. However, this delay is expected to be small when utilizing the appropriate slot-allocation policies.

V. ILLUSTRATIVE NUMERICAL EXAMPLES

We consider two networks whose performance we illustrate via our analytical model and simulation. The first (shown in Fig. 3) is a small network with six nodes and eight bidirectional links. The length of each link is either one or two slots as shown in the figure. The second (Fig. 6) is a larger network with 15 nodes and 21 bidirectional links. The lengths of the links range from ten slots to 32 slots. Packets arrive at each node according to a Poisson process, and the packet rate is assumed to be λ for each source-destination pair.

Following are the default parameters used in the simulation:

- Packet size: 1500 bytes.
- Bandwidth of a link: OC-48, i.e., 2.5 Gbps. The propagation delay on a 1 km link is 1 slot time.
- Number of wavelengths: $W = 4$ or 8.
- Slot buffer size when buffering is used to resolve contention: $B = 10$, which means that we have ten parallel delay lines and each is of length 1 km.
- Packet buffer size at each node to each destination $B_p = \infty$ (unless otherwise stated).
- Maximum number of times a slot can be deflected before being dropped when deflection is used to resolve contention: $M = 10$.

An event-driven simulation developed in C++ is deployed, and 500 000 packets are simulated for each data point. We consider the following performance metrics:

- Average packet delay.
- Network throughput: defined as the average number of packets received by their destinations per time slot.

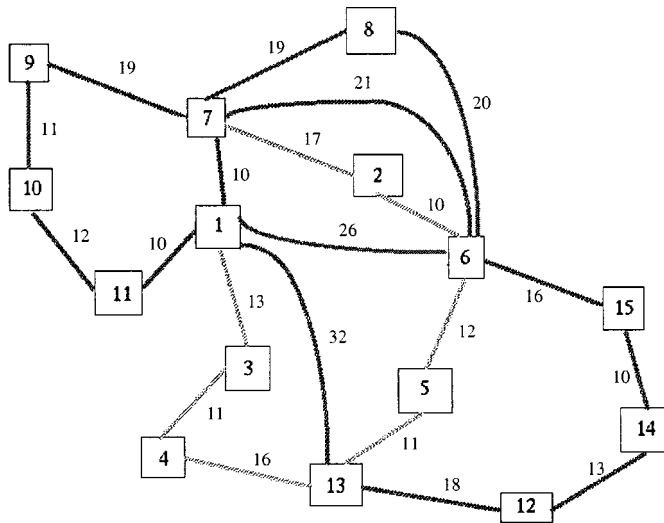


Fig. 6. Simulation Network 2, a typical telecommunication network consisting of interconnected rings.

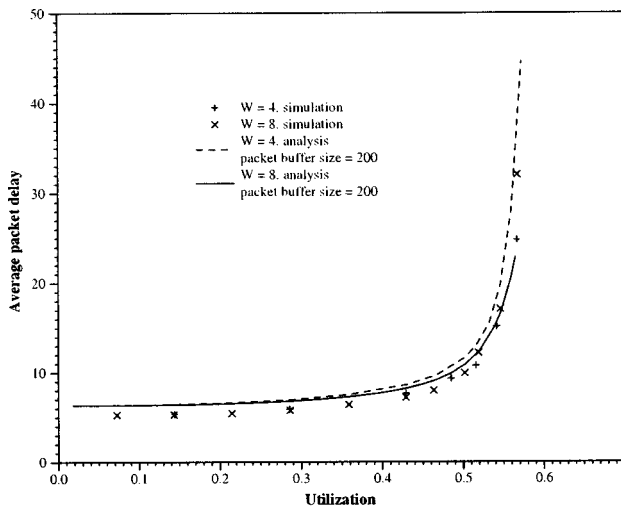


Fig. 7. Average delay versus link utilization in Network 1, for varying number of wavelengths ($W = 4, 8$).

- Link utilization: average percentage of link capacity used.
- Fairness index:

$$FI = \frac{\left(\sum x_{ij}\right)^2}{N \times (N - 1) \times \sum x_{ij}^2} \quad (18)$$

where x_{ij} can be the throughput for source–destination pair (i, j) . If the throughput for all the source–destination pairs are equal, $FI = 1$. Note that, due to finite buffering, the network throughput may be lower than the offered load and the fairness index may be lower than unity. By substituting other metrics into x_{ij} , this index can also be used to measure fairness with respect to metrics such as average packet delay, average queueing delay, and average queue length for each source–destination pair. Fairness with respect to average queue length will also be investigated.

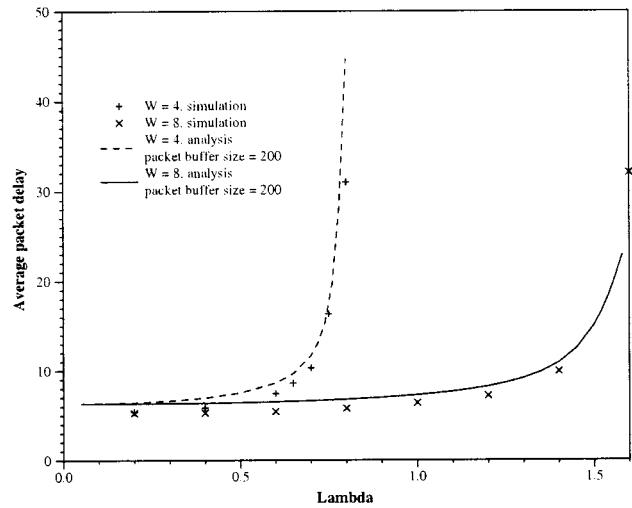


Fig. 8. Average delay versus packet arrival rate (λ) in Network 1, for varying number of wavelengths ($W = 4, 8$).

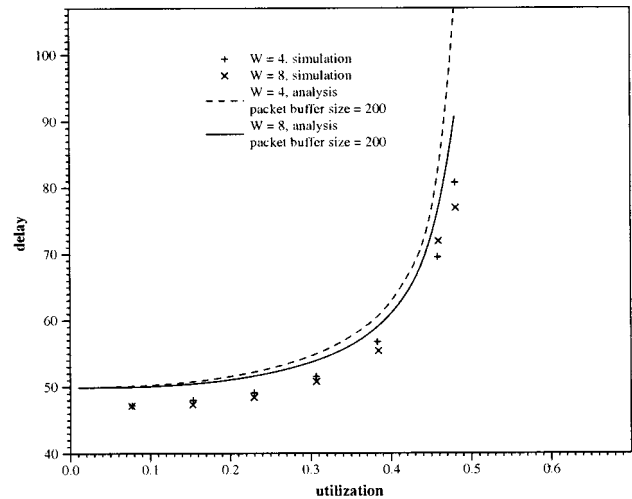


Fig. 9. Average delay versus link utilization in Network 2, for varying number of wavelengths ($W = 4, 8$).

A. Performance of Slot Assignment Based on Capacity Allocation

Figs. 7 and 8 show analytical and simulation results for Network 1 (Fig. 3). Figs. 9 and 10 show analytical and simulation results for Network 2 (Fig. 6). Fig. 7 plots the average packet delay versus link utilization for a varying number of wavelengths ($W = 4, 8$). Note that the average packet delay is normalized to number of time slots, as is the case for the other plots. For the analytical results, the packet buffer size is $B_p = 200$. The simulation results assume infinite packet buffers and infinite slot buffers (these assumptions may seem impractical but are useful to show how the networks perform under the protocols we are examining). The analytical model assumes finite packet buffer for tractability, which is reasonable since the number of packets in a packet buffer seldom exceeds $B_p = 200$ when the network is in a stable state. Note the good agreement between the performance analysis and the simulation. Also, for different number of wavelengths, there is little difference in the network utilization. Fig. 9 shows similar results for Network 2. Although the utilizations in Figs. 7 and 9 are limited

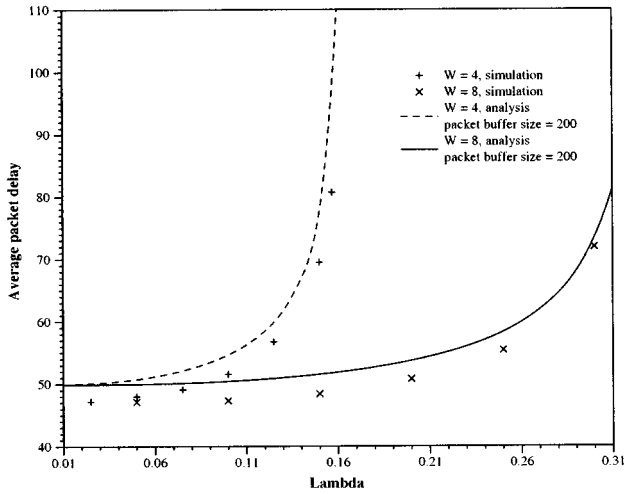


Fig. 10. Average delay versus packet arrival rate (λ) in Network 2, for varying number of wavelengths ($W = 4, 8$).

to around 0.6 and 0.5, respectively, this performance is not a limitation of PSR, but rather it is a limitation of the routing approach. By using fixed shortest-path routing, certain links in the network become more heavily loaded than other links. When these loaded links become saturated, the delays on these links approach infinity, while the remaining links in the network remain under-utilized. By implementing better routing algorithms which more evenly balance the load across all links in the network, higher utilizations may be achieved.

Figs. 8 and 10 plot the average packet delay versus packet arrival rate λ for varying number of wavelengths ($W = 4, 8$). Both figures show results from the analysis and the simulation. As expected packet delays are higher and the maximum sustainable λ is much lower for $W = 4$ than for $W = 8$. It would be of interest to calculate the average packet delay when λ approaches 0. There is no queueing delay in this case. The only factors which contribute to the packet delay will be the time a packet spends waiting for a slot to be assigned the desired destination and the propagation delay to the destination. We show the calculation of delay for Network 1 below. The calculation for Network 2 is similar.

Let \overline{T}_0 denote the average packet delay when $\lambda \rightarrow 0$, \overline{T}_1 denote the average waiting time for a packet to find a slot with the same destination, and \overline{T}_2 denote the average propagation delay for a packet to reach its destination. We have

$$\overline{T}_0 = \overline{T}_1 + \overline{T}_2.$$

\overline{T}_2 is easily computed for a given network, and turns out to be 2.13 time slots for Network 1.

When $\lambda \rightarrow 0$, most slots travelling in the network are empty so we can approximately assume that all packets are transmitted in newly-assigned slots. For a given source-destination pair (i, j) , if the probability that destination j is assigned to a slot on the first link along path $i \rightarrow j$ is p_{ij} , then the expected waiting time of a packet for destination j at source i is $1/p_{ij}$, because the waiting time is approximately geometrically distributed with parameter p_{ij} . We can then find \overline{T}_1 by taking the average waiting delay over all source-destination pairs.

		Destination Node					
		0	1	2	3	4	5
Source Node	0	0	.60	.20	.20	.25	.75
	1	.75	0	.20	.20	.25	.75
	2	.25	.25	0	.60	.80	.20
	3	.20	.25	.75	0	.60	.20
	4	.20	.25	.75	.75	0	.20
	5	.60	.80	.20	.25	.25	0

Fig. 11. Capacity-allocation matrix for Network 1.

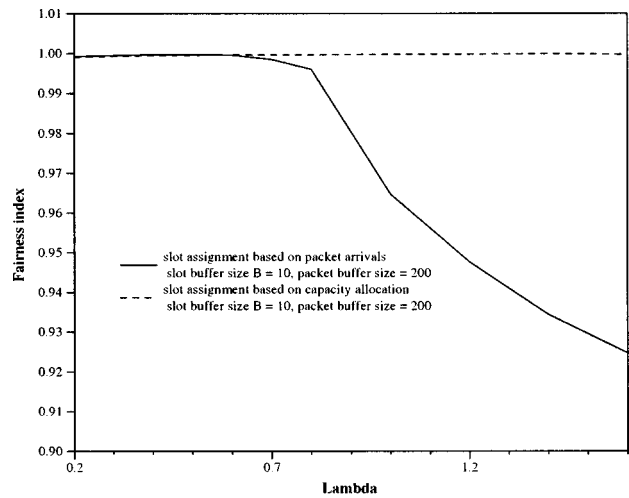


Fig. 12. Fairness index based on through-put versus packet arrival rate (λ) in Network 1, for $W = 8$ and different slot-assignment schemes based on packet arrivals and capacity allocation.

Given the routing matrix shown in Fig. 4, the probability matrix of all p_{ij} 's, shown in Fig. 11, is found by applying the algorithm introduced in Section III-A3. We then obtain

$$\overline{T}_1 = \frac{\sum_{1 \leq i, j \leq N, i \neq j} 1/p_{ij}}{N \times (N - 1)} = 3.31.$$

Finally, for Network 1, $\overline{T}_0 = 3.31 + 2.13 = 5.44$ times slots, which is very close to what we have in Fig. 8.

Similarly for Network 2, we find that the average packet delay when $\lambda \rightarrow 0$ is 51.93 time slots, which also matches with the results in Fig. 10.

B. Comparison of Slot-Assignment Schemes

Figs. 12–14 compare the performance of the two slot-assignment schemes discussed in Section III-A. These results are obtained from simulations. For each scheme, we evaluate the performance of Network 1 with $W = 8$.

Fig. 12 plots the fairness index versus λ for the two slot-assignment algorithms. In both schemes, each node has a finite packet buffer of size 200 for each destination, and a finite slot buffer of size 10. Therefore, when the load increases, both

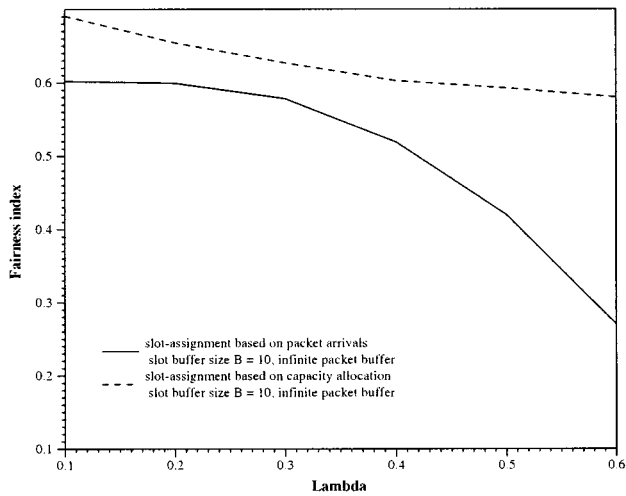


Fig. 13. Fairness index based on average packet queue length versus packet arrival rate (λ) in Network 1, for $W = 8$ and different slot-assignment schemes based on packet arrivals and capacity allocation.

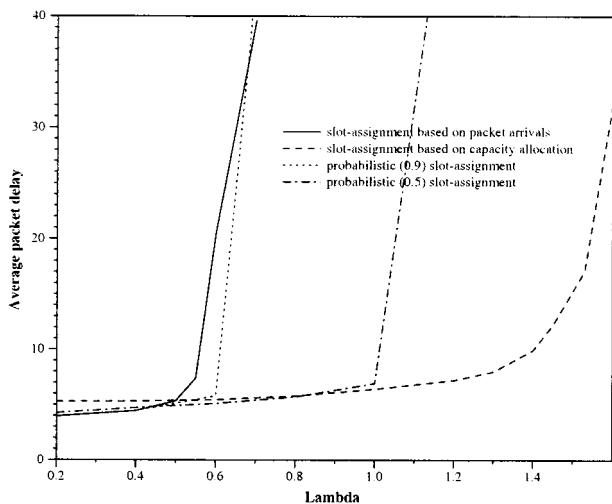


Fig. 14. Average delay versus packet arrival rate (λ) of difference slot-assignment schemes in Network 1, for varying number of wavelengths ($W = 4, 8$).

packets and slots may be dropped. We observe that, when the load increases, the fairness index decreases faster for the packet-arrival-based slot-assignment scheme than for the capacity-allocation-based slot-assignment scheme. Under the packet-arrival-based scheme, source nodes may allocate slots in a manner which may prevent intermediate nodes from receiving enough slots for a given destination. Also, source nodes may completely fill each slot, leaving very little capacity for downstream nodes. By allocating slots according to the capacity-allocation scheme, each node is guaranteed to have a certain number of empty slots available for each destination, resulting in a higher degree of fairness.

Fig. 13 plots the fairness index derived from average packet queue length for all source–destination pairs. The same assumptions are applied as in the previous plot. We can see that both schemes yield a fairness index which is below 1 and that this index is more sensitive to the different schemes.

Fig. 14 plots the average packet delay versus λ for the different slot-assignment algorithms: slot-assignment based on

packet arrivals, slot-assignment based on capacity allocation, and probabilistic assignment in which a node which has a packet to transmit will only use an empty slot with a certain probability (e.g., a probability of 0.9 means that a node with a packet to transmit will only utilize an empty slot 90% of the time). Infinite packet buffers and infinite slot buffers are assumed. Note that this model is different from the simulation model we assumed to obtain the results in Fig. 12. We use different models in these simulations because we are interested in different aspects of the protocols. In Figs. 12 and 13, the fairness issue is studied. Unfairness comes from the fact that packets for certain source–destination pairs are transmitted more often than for others. This fact is made clearer if we limit the packet buffer size and slot buffer size so that some packets are dropped. However, when we study the packet delay in Fig. 14, we keep both the packet buffer size and the slot buffer size infinite so that large delays can be observed for a certain load.

For low load, the algorithm based on packet arrivals performs better than the others. In the packet-arrival-based scheme, an arriving packet is immediately transmitted in the next empty slot. Under low loads, empty slots are fairly plentiful, and a packet will not have to wait long before being transmitted. On the other hand, in the capacity-allocation scheme, even when there are empty slots, there is a certain probability that a packet will not be transmitted if these slots are assigned to some other destinations. Thus, a packet may have to wait a little longer for a slot to the correct destination while letting empty slots pass by. This is true for all probabilistic assignment schemes, which under-utilizes resources when load is low. Under high loads, the capacity-allocation-based scheme offers better performance than the packet-arrival-based scheme. In the packet-arrival-based scheme, since no restrictions are placed on the source nodes, there is the potential for a significant load imbalance. This load imbalance is created when the source nodes generate too many slots for a particular destination, and these slots overload some link in the network. In the capacity-allocation-based scheme, by limiting the rate at which source nodes generate slots for each destination, the problem of load imbalance is eliminated and a higher throughput can be achieved. For the same reason, the other two probabilistic assignment schemes both perform better than the packet-arrival based approach. We also observe that when the assignment probability equals 0.5, the network can perform better than when the probability is 0.9. For a certain network, we can carefully choose this value to optimize the network performance.

C. Comparison of Contention-Resolution Schemes

Figs. 15–20 compare the performance of the two contention-resolution schemes: *Buffering* and *Deflection*. These results are obtained through simulation. Fig. 15 compares the average packet delay versus load in Network 1 with $W = 8$, infinite packet buffers, varying B (slot buffer size for the buffering scheme), and varying M (maximum number of deflections per slot for the deflection scheme). Fig. 16 shows throughput versus delay for Network 1. Since we want to study the effectiveness of each scheme in dealing with con-

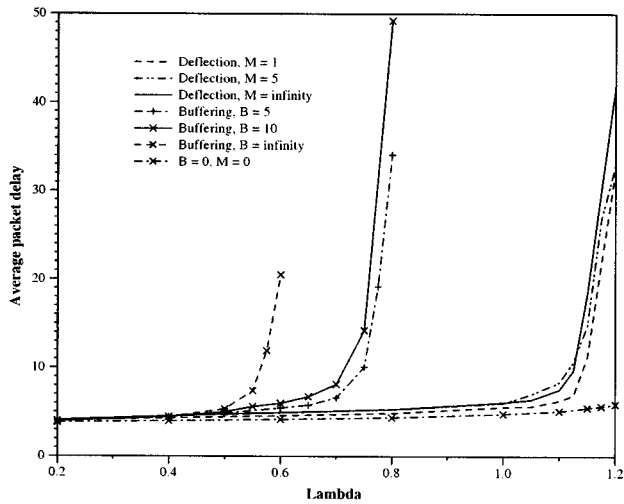


Fig. 15. Average delay versus packet arrival rate (λ) in Network 1, for $W = 8$ and different policies to resolve contention.

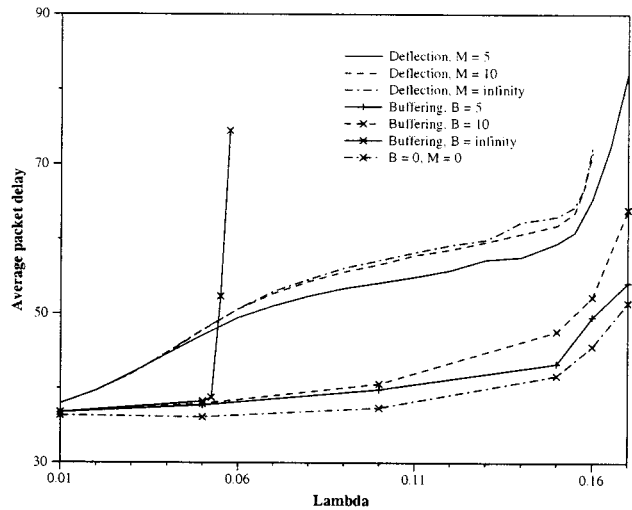


Fig. 17. Average delay versus packet arrival rate (λ) in Network 2, for $W = 8$ and different policies to resolve contention.

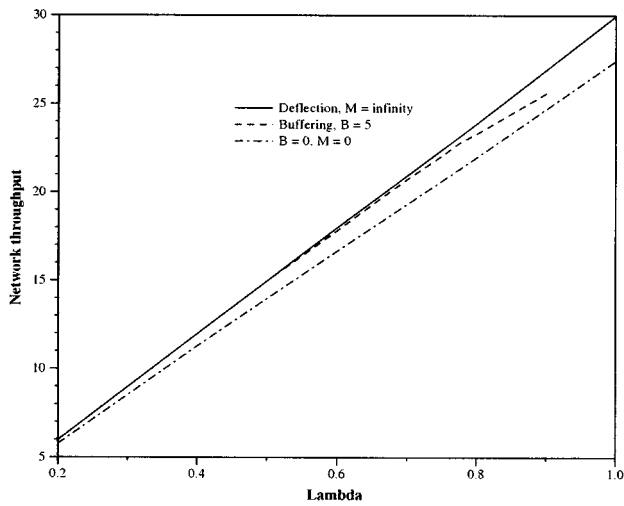


Fig. 16. Throughput versus packet arrival rate (λ) in Network 1, for $W = 8$ and different policies to resolve contention.

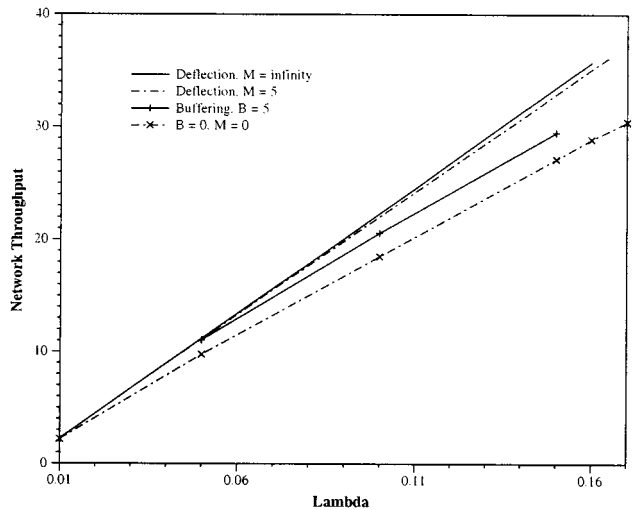


Fig. 18. Throughput versus packet arrival rate (λ) in Network 2, for $W = 8$ and different policies to resolve contention.

tention, both schemes are implemented in conjunction with the packet-arrival-based slot-assignment scheme.

Under low load, the performance of the two schemes is very similar. Since there is little contention, no deflection or buffering is necessary. When the load increases, we see that the deflection scheme offers better performance. The reason for this behavior is that, if the fixed shortest-path routing matrix results in an imbalance in link loads, then the deflection scheme may be able to balance the load better on the links by providing alternate paths to each destination. With the buffering scheme, once the most heavily loaded link becomes saturated, the buffers associated with that link will become unstable if the buffer size is infinite, or become filled if the buffer size is finite, even if there is spare capacity elsewhere in the network. By implementing more intelligent and adaptive routing schemes instead of fixed shortest-path routing, the load can be better balanced across the network links, and contention will be reduced. Deflection routing, which can be viewed as a form of adaptive routing, is relatively easy and inexpensive to implement when compared to other adaptive routing approaches.

In the deflection scheme we see that, in most cases, when M is increased, the delay also increases. Increasing M will allow deflected slots to remain in the system longer, and in some cases, a slot may be deflected back and forth among the same nodes without ever reaching its destination. These increased deflections result in a higher consumption of network resources, and consequently, a higher degree of contention. Note that there is a crossover of $M = \infty$ and $M = 5$, this may be because the network is becoming unstable when the crossover occurs.

We also note that, as the slot-buffer size increases, the delay for the buffering scheme increases. Packets are remaining in the slot buffers longer before reaching their destinations. Also, since there are more slots maintained in the system, the amount of contention will be higher. The decrease in delay for lower buffer sizes comes at the cost of higher slot-drop rates and lower network throughput. An extreme situation is one in which we have no buffers and no deflection ($B = 0, M = 0$ in Fig. 15). In this case, if a contending slot cannot be transmitted, it is simply dropped. Although dropping slots may lead to very low average

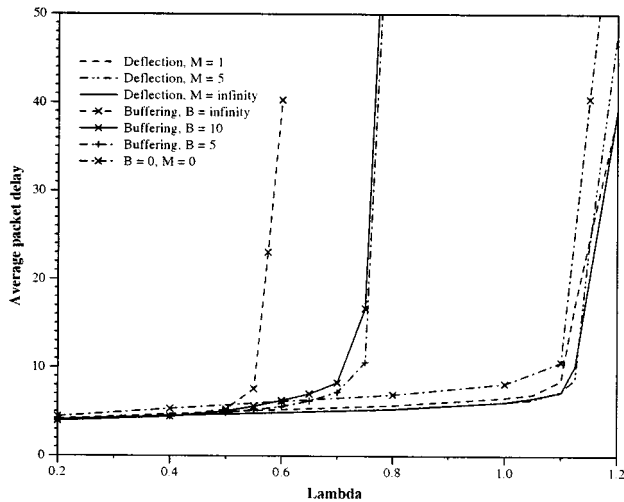


Fig. 19. Average delay versus packet arrival rate (λ) in Network 1 with retransmission of dropped packets, for $W = 8$ and different policies to resolve contention.

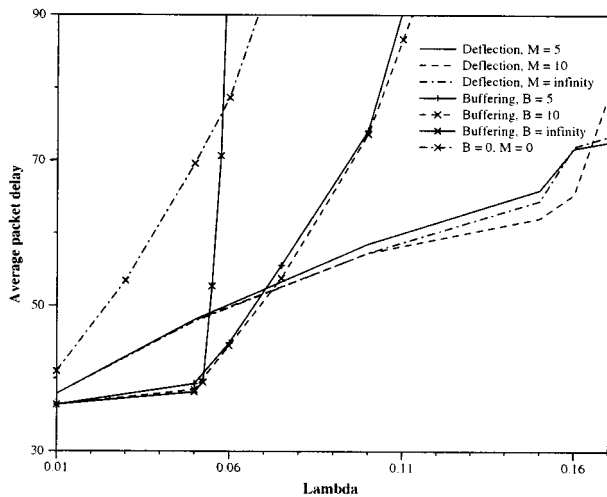


Fig. 20. Average delay versus packet arrival rate (λ) in Network 2 with retransmission of dropped packets, for $W = 8$ and different policies to resolve contention.

delays (Fig. 15), we see that it also results in lower throughput (Fig. 16).

Fig. 17 compares the average packet delay versus load for Network 2, and Fig. 18 shows throughput versus delay for the same network. We observe that the buffering scheme yields lower delays than the deflection scheme, except for the case in which the slot buffer size is infinite. It can be shown that deflection does not give better performance because the lower delay in the buffering scheme comes at the cost of dropping more slots. We simulated the case in which dropped packets (including both the packets being dropped because the packet buffer is full and the packets being dropped because the slots containing them are dropped) are injected again into the network and retransmitted. The results are shown in Figs. 19 and 20. It turns out that deflection yields lower overall delays among all packets. In Network 1, the performance of both schemes is similar to the case without retransmissions of dropped packets, except that the no-buffering-no-deflection

protocol gives higher delay than the cases where there are deflections. In Network 2, the delays of buffering schemes increase quickly as load increases while delays of deflections increase slowly and almost linearly.

We also find an interesting fact that, for Network 2, under low load, the buffering scheme achieves lower delay than the deflection scheme even with retransmission of dropped packets. It appears that, for this network, the benefit of providing alternate paths to the destination is outweighed by the increase in network load caused by deflections.

The benefits of deflection routing seem to depend heavily on the network topology and the routing matrix. Deflection may yield higher benefits if, for a given slot, there is an alternate path to the destination which is less loaded than the path specified by the routing matrix. Also, in order to find this alternate path, a slot may have to be deflected multiple times. An interesting problem would be to find the optimal value of M which yields the best network performance.

VI. CONCLUSION

PSR has been proposed as a solution to fast routing in all-optical packet-switched WDM networks. In this approach, packets which have the same destination and which are transmitted on different wavelengths by different source nodes may form a photonic slot which is routed through the network as an integrated unit. This study demonstrated how the concept of PSR can be applied to a mesh network, introduced an analytical model to evaluate the performance of such networks, and discussed different slot-assignment algorithms, contention-resolution schemes, and slot-filling policies.

It was found that the slot-assignment algorithm based on capacity allocation can lead to better performance in terms of average packet delay, network throughput, and fairness in the allocation of network resources. It was also shown that the deflection approach to contention resolution provides better performance than a buffering approach in some cases, while in other cases, deflection can lead to worse performance in the network due to the excess consumption of network resources by deflected slots, or due to lack of less-loaded alternate paths to a given destination.

One important observation is that, since packets must wait for a slot which is headed to the same destination, there is some degree of inefficiency in PSR, particularly at low loads. As the size of the network grows, this inefficiency will increase, since slots destined for a given destination node will become less frequent. One approach to improve the scalability of such a network is to allow a packet to traverse multiple hops to its destination. Rather than requiring a packet to wait for a slot with the same destination, it could instead join a slot which is headed to a different destination, but is in the direction of the original destination. At the intermediate node, the packet would be converted to electronics and would incur some additional queuing delays before joining a slot to its final destination. In this approach, a protocol would be required to determine which slots a packet would be allowed to join. Such a protocol is a topic of future research.

Some additional possible areas for future research include the development of analytical models for networks in which de-

flexion is used, and the study of capacity allocation algorithms which result in a higher degree of fairness.

REFERENCES

- [1] B. Mukherjee, *Optical Communication Networks*. New York: McGraw-Hill, 1997.
- [2] I. Chlamtac, A. Farago, and T. Zhang, "Lightpath (wavelength) routing in large WDM networks," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 909–913, June 1996.
- [3] R. Ramaswami and K. N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 5, pp. 489–500, Oct. 1995.
- [4] D. Banerjee, "Design and analysis of wavelength-routed optical networks," Ph.D. dissertation, Univ. of California, Department of Computer Science, Davis, CA, 1996.
- [5] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Trans. Networking*, vol. 4, no. 5, pp. 684–696, Oct. 1996.
- [6] A. Mokhtar and M. Azizoglu, "Adaptive wavelength routing in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 197–206, April 1998.
- [7] I. Chlamtac, V. Elek, A. Fumagalli, and C. Szabo, "Scalable WDM access network architecture based on photonic slot routing," *IEEE/ACM Trans. Networking*, vol. 7, no. 1, pp. 1–9, Feb. 1999.
- [8] I. Chlamtac, A. Fumagalli, and G. Wedzinga, "Slot routing as a solution for optically transparent scalable WDM wide area networks," *Photon. Network Commun.*, vol. 1, no. 1, pp. 9–21, June 1999.
- [9] I. Chlamtac, V. Elek, and A. Fumagalli, "A fair slot routing solution for scalability in all-optical packet-switched networks," *J. High Speed Networks*, vol. 6, pp. 181–196, 1997.
- [10] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *J. Lightwave Technol.*, vol. 16, no. 12, pp. 2081–2094, Dec. 1998.
- [11] C. Guillemot, M. Renaud, and P. Gambini *et al.*, "Transparent optical packet switching: The European ACTS KEOPS project approach," *J. Lightwave Technol.*, vol. 16, no. 12, pp. 2117–2134, Dec. 1998.
- [12] I. Chlamtac, A. Fumagalli, and L. G. Kazovsky *et al.*, "CORD: Contention resolution by delay lines," *IEEE J. Select. Areas Commun.*, vol. 14, no. 5, pp. 1014–1029, June 1996.
- [13] L. Zucchelli, M. Burzio, and P. Gambini, "New solutions for optical packet delineation and synchronization in optical packet switched networks," in *Proc. 22nd ECOC'96* Oslo, Norway, Sept. 15–19, 1996, vol. 3, pp. 310–304.
- [14] D. Bertsekas and R. Gallager, *Data Networks*, second ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.



Hui Zang received the B.S. degree from Tsinghua University, China, in 1997 and the M.S. degree from the University of California, Davis, in 1998. She is currently working towards the Ph.D. degree at the Department of Computer Science at the University of California, Davis, as well as a principal R&D engineer at Sprint Advanced Technology Laboratories (ATL), Burlingame, CA.

From June 1999 to September 1999, she worked at IBM Almaden Research Center, San Jose, CA, as a summer intern. From October 1999 to September

2000, she worked at Sprint ATL as a part-time intern, where she became a principal R&D engineer in October 2000. Her research interests include photonic packet switching, WDM network control and management, and fault-tolerance in WDM networks.



Jason P. Jue received the B.S. degree in electrical engineering (with honors) from the University of California, Berkeley, in 1990, the M.S. degree in electrical engineering from the University of California, Los Angeles, in 1991, and the Ph.D. degree in electrical and computer engineering from the University of California, Davis, in 1999.

In 1999, he joined the faculty at the University of Texas, Dallas, where he is currently an Assistant Professor of Computer Science and a member of the Center for Advanced Telecommunications Systems and Services. His research interests are in the area of high-speed communication networks, with an emphasis on the design and performance evaluation of WDM optical networks.



Biswanath Mukherjee received the B.Tech. (Hons.) degree from Indian Institute of Technology, Kharagpur, India, in 1980 and the Ph.D. degree from the University of Washington, Seattle, in June 1987.

At the University of Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship. In July 1987, he joined the University of California, Davis, where he has been Professor of Computer Science since July 1995 and Chairman of Computer Science since September 1997. He is

author of the textbook *Optical Communication Networks* (New York: McGraw-Hill, 1997), a book which received the Association of American Publishers, Inc.'s 1997 Honorable Mention in Computer Science. His research interests include lightwave networks, network security, and wireless networks.

Dr. Mukherjee is co-winner of paper awards presented at the 1991 and the 1994 National Computer Security Conferences. He serves on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE NETWORK, ACM/Baltzer *Wireless Information Networks (WINET)*, *Journal of High-Speed Networks*, *Photonic Network Communications*, and *Optical Network Magazine*. He also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society. He served as the Technical Program Chair of the IEEE INFOCOM '96 Conference.