Differentiated Contention Resolution for QoS in Photonic Packet-Switched Networks

Tao Zhang, *Student Member, IEEE*, Kejie Lu, *Member, IEEE*, and Jason P. Jue, *Senior Member, IEEE*

***Abstract*—Packet contention is a major challenge in photonic packet-switched networks due to the lack of random access buffers in the optical domain. Existing contention resolution approaches such as wavelength conversion and fiber-delay-line buffering may significantly increase the overall system cost and may be difficult to implement. To avoid such issues, this paper proposes a framework for providing label-based differentiated contention resolution by exploiting recirculation buffering and deflection routing. To accommodate more options for differentiation and to avoid the potential problem of forwarding packets in a network indefinitely, two classes of loopless deflection algorithms are provided. An analytical model is also developed to evaluate the packet loss probability and the end-to-end delay for different buffering and deflection routing schemes. The paper also investigates the effectiveness of the control schemes in providing differentiated loss and delay through simulation and analysis. The accuracy of the analytical model is confirmed by simulation.**

***Index Terms*—Analytical model, contention resolution, deflection routing, loss probability, photonic packet switching, recirculation buffer.**

I. INTRODUCTION

As THE number of Internet users and broad-band applications continue to increase, next-generation communication networks are expected to provide huge bandwidth as well as to support diverse service demands. The tremendous bandwidth requirement can be accommodated by using *dense-wavelength-division-multiplexing* (DWDM) transmission systems [1], which can support more than 1 Tb/s in a single optical fiber. However, traditional electronic packet switches are not suitable for handling such high data rates due to limitations in electronic processing speeds and the expense of optical–electronic–optical (O/E/O) conversion. To provide high-speed switching without O/E/O conversion, it may be desirable to utilize all-optical switching techniques [2], [3].

From the network perspective, there are generally three approaches for optical switching. In *wavelength-routed networks*, data can be transmitted only after an all-optical circuit-switched connection is established. In *optical burst-switched networks*, upper-layer packets are first assembled into bursts at ingress edge nodes; bursts are then forwarded toward their destinations and disassembled at egress edge nodes. Compared with these two approaches, a *photonic packet-switched network* can provide finer granularity and possibly lower end-to-end delay [3], [4]. In spite of significant challenges in the processing and buffering of bits at high speeds, solid progress has been made toward realistic photonic packet switches. For example, an 8×8 optical switch with 5-ns switching time is now commercially available in [5]; a 1×2 all-optical packet switch element in which header processing can be carried out all-optically is presented in [6]; larger systems are also developed in [7] and [8], where experimental photonic packet switches can provide 200 Gchip/s all-optical processing, 40 Gb/s/port packet switching, and optical buffering.

Photonic packet-switched networks can be divided into two categories: time-slotted synchronous networks with fixed-length packets and unslotted asynchronous networks with fixed-size or variable-size packets [2]. Although the slotted schemes may lead to higher throughput, they are more difficult to deploy due to the synchronization requirements of all incoming packets [2], [9]. This paper focuses on the asynchronous switching scheme with fixed-size packets.

In most packet-switched networks, packet loss and end-to-end delay are two major *quality of service* (QoS) factors besides the bandwidth requirement. These two effects result mainly from packet contention, which occurs when two or more incoming packets need to be forwarded to the same output at the same time. Approaches for resolving contention in photonic packet-switched networks include wavelength conversion, buffering, and deflection routing.

In wavelength conversion, a contending packet can be converted from one wavelength to another in order to avoid conflict. Although this method has been studied in previous literature, it may significantly increase the system cost, since all-optical wavelength conversion technology is immature and wavelength converters are expected to remain expensive in the foreseeable future [10].

In buffering, contending packets are temporarily stored and are forwarded at a later time. Buffering in electronic packet-switched networks is implemented by storing packets in random-access-memory (RAM) buffers; however, RAM-like buffering is not yet available in the optical domain. In optical networks, optical-fiber-delay lines [11]–[15] can be utilized to delay packets for a fixed amount of time. Optical buffers are either single-stage or multistage and can be further categorized into feedforward architectures and feedback architectures [16]. A limitation of optical buffering is that the fiber delay lines may be bulky, since the length of the fiber is directly proportional to the propagation time of light in the fiber. In addition, buffering can lead to an increase in the end-to-end delay.

Manuscript received December 15, 2003; revised May 6, 2004. This work was supported in part by the National Science Foundation (NSF) under Grant ANI-01-33899.

T. Zhang and J. P. Jue are with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083 USA (e-mail: txz021000@utdallas.edu; jjue@utdallas.edu).

K. Lu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: lukejie@ieee.org).

Digital Object Identifier 10.1109/JLT.2004.836750

In deflection routing, a packet is forwarded to an alternative output port if its primary output port is occupied by another packet. Thus, the links in the network are acting as dynamic buffers without increasing the cost of the photonic switch [17]–[21]. Consequently, packet loss probability in the network can be decreased. In [22], deflection is studied in an unslotted packet network with a Manhattan Street Network topology. In [23] and [24], deflection routing is studied in irregular mesh networks. The most important advantage of the deflection routing method is that it does not require a huge effort to be implemented, either in terms of hardware components or in terms of control algorithms. The effectiveness of this technique critically depends on the network topology; meshed topologies with a high number of nodal degrees greatly benefit from deflection routing, whereas minor advantages arise from more simple topologies.

Nevertheless, deflection routing may lead to increased end-to-end packet delay and may raise several other issues. One of the issues raised by deflection routing is the *packet ordering* problem, in which packets that belong to the same session may arrive out of order. This problem can generally be resolved by network-layer protocols [14]. Another potential problem is the *packet-loop* problem, in which a packet may be forwarded within the network indefinitely. Looping of packets may increase delays, degrade signal quality for the looping packets, and increase the traffic load for the entire network. Several approaches to solve the packet-loop problem in deflection routing have been studied in previous literature. In [23], a deflection policy is applied at certain hub nodes in order to prevent loops in a specific network topology. However, the policy for determining the hub nodes and the corresponding deflection tables in an arbitrary network topology are not specified. In [25], a method for determining deflection alternatives in a manner that avoids routing loops is presented. Traditional time-to-live (TTL) technologies [26] are also widely used to limit the number of hops that a packet will traverse in the network, but these kinds of approaches may not suitable for high-speed photonic packet switching networks, since they require the packet header to be modified at each node.

While most research on contention resolution [13], [14] focuses on how to improve overall packet-loss performance, a comprehensive study on how to provide differentiated loss and delay for QoS in photonic packet-switched networks is not yet available. Service differentiation is a very important issue in the Internet, as a consequence of the variety of applications with different QoS requirements. An intensive research and standardization effort has taken place in order to define a paradigm for service differentiation support [27]. These architectures and concepts should be applied to optical routers where, in spite of the huge bandwidth available on the optical links, contention may still occur in the switches [28].

This paper first proposes a framework for differentiated contention resolution that exploits recirculation buffering and deflection routing schemes. The proposed differentiated contention resolution scheme includes different buffering and different deflection schemes for different classes of traffic. Specifically, for buffering schemes, we can limit the buffer usage and the recirculation times of packets; for example, traffic with relaxed loss requirements may be restricted to use only a subset of the total available buffers. In this case, some buffers will be reserved only for higher priority traffic. The deflection policy can also be varied depending on the class of traffic. We observe that, while deflection routing methods reduce packet loss, they can lead to increased delay. However, by restricting deflection routing to be loop-free, an upper bound may be placed on the maximum end-to-end delay of a packet. Two classes of loopless deflection-routing schemes are provided in order to support delay-limited QoS requirements. With the proposed loopless deflection algorithms, the average delay in the network can be reduced, and a strict delay bound can be provided, so that delay-restricted requirements can be supported. By varying the deflection method for different classes of traffic, differentiated loss and delay can be achieved. For example, traffic that is loss sensitive could be deflected, while traffic that is delay sensitive would not be deflected. Within the framework, the buffering and deflection parameters that can be varied to provide differentiation are defined, and then the buffering and deflection policies applied to different classes of traffic are specified.

An analytical model that analyzes recirculation buffering and deflection routing for different classes of traffic is then developed to evaluate the performance of the proposed schemes, and the accuracy of the analysis is verified through simulation.

The remainder of this paper is organized as follows. In Section II, we elaborate on the framework for differentiated buffering and differentiated deflection routing in photonic packet-switched networks. Section III proposes a destination-based looplesss deflection-routing (D-LLD) routing algorithm and a source–destination-based loopless deflection-routing (SD-LLD) algorithm. An analytical model is developed in Section IV to evaluate the performance of different buffering and deflection-routing schemes. Section V provides numerical results and discussions. Finally, Section VI concludes the paper.

II. Differentiated Contention Resolution in Photonic Packet-Switching Networks

In this section, we provide a switch architecture and a label-based control scheme implementation for supporting differentiated buffering and differentiated deflection routing in photonic packet-switched networks. We then discuss the various parameters for providing differentiated service in photonic packet-switched networks.

A. Switch Architecture

A fundamental principle in network design is the minimization of implementation costs. Thus, we assume that there are no wavelength converters in the network, and we also assume that only a small number of optical buffers, i.e., fiber delay lines, are available in a switch. Since there is no wavelength conversion in the switch, packets with different wavelengths will be switched separately. Therefore, in this paper, only a single wavelength plane is considered. Based on the previously stated assumptions, we consider the generic asynchronous switch architecture shown in Fig. 1.

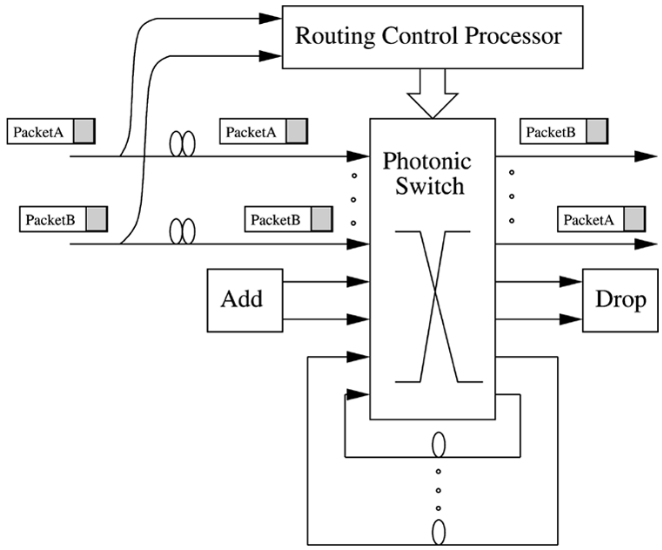


Fig. 1. Generic architecture for asynchronous photonic packet switch.

In this switch, there is a central routing control processor which has the capability to perform control functions, including processing the packet header, determining the route of the incoming packet, prioritizing packets based on class of service, and maintaining topology information. Since the controller needs some time to process the header information and to configure the switching element, each input is connected to the switching element via a fixed-length fiber delay line, such that the propagation delay in the fiber provides sufficient time for the controller to process the header and configure the switch. Some ports of the switching element are connected to add/drop units, which can be used to add and drop local traffic.

The recirculation buffers, which connect the switch outputs to the switch inputs via fiber delay lines, are used to resolve packet contention. In this paper, since we assume fixed-size packets, we let the length of each delay line correspond to the size of a packet. The advantage of the recirculation buffers is that packets from different input ports can share a fiber delay line at different times. Therefore, this scheme is more effective than output buffering at resolving contentions [1]. The tradeoff is that the switch size may increase depending on the number of recirculation buffers [16]. Another problem is that feedback may attenuate the signal-to-noise ratio and possibly lead to packets looping indefinitely within a switch. To avoid such situations, we can limit the number of times that a packet can traverse a fiber delay line. The amount of time that a packet stays in the switch can be maintained in the controller, and a packet will then be dropped if it cannot be forwarded to any output within a given amount of time. We will show that this method can also provide additional differentiation options.

B. Label-Based Forwarding Scheme

A control scheme must be defined in order to support buffering and deflection routing and to accommodate differentiated service in the network. This control scheme must also maintain the simplicity of packet forwarding in the optical domain.

TABLE I
LABEL-BASED FORWARDING TABLE AT NODE 2

	Output			
Label	1	2	3	4
A	4	-	-	-
B	4	3	1	0
C	4	3	-	-

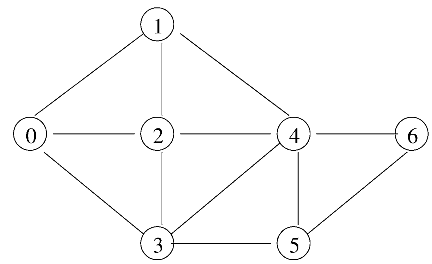


Fig. 2. Example network topology.

In Internet protocol (IP)-based networks, routing protocols are used to exchange topology and resource utilization information among nodes. Based on this information, each node can calculate its routing table. In general, only the destination information is required in order to forward a packet. However, to support QoS, additional information may be required. For example, when source routing is used, the path information must be saved in the header. Such schemes may not be suitable for photonic packet-switching networks since they introduce extra complexity on header reading and processing. Moreover, these schemes are not suitable for deflection routing since they do not provide coordination among nodes, which is required by loopless routing.

To support differentiated services and deflection routing in photonic packet-switched networks, we can apply a label-based control scheme for forwarding packets. This method is similar to the method used in multiprotocol label switching (MPLS) or generalized MPLS (GMPLS) [29]. In this scheme, before a packet can be sent into the network, a label path that satisfies certain QoS requirements must be determined and distributed to all possible nodes in the network. Each packet must carry its label information in the packet header, and each node maintains a forwarding table that specifies, for each incoming label, the corresponding output port. Thus, when a packet arrives to a node, the node will read the header and forward the packet to the correct output port according to the packet's label. In standard label-based forwarding approaches, a unique path is specified for each label. In this paper, we extend the concept of label switching to include multiple paths for each label. The multiple paths are defined by the deflection alternatives at each node. Thus, for each incoming label, the forwarding table will specify an ordered set of possible output ports. The first output port in a set will specify the primary output port, while the remaining output ports in the set will specify output ports to which a packet may be deflected if the primary output port is busy.

Table I gives an example of a label-based forwarding table at node 2 for the network topology shown in Fig. 2. In this example, we assume that all three label paths have the same destination—node 6. In this case, a packet with label "C" arriving at node 2 will be forwarded to output 4 if output 4 is available. If output 4 is not available, the packet will be deflected to output 3. If both output 4 and output 3 are not available, then the packet will be dropped.

The label-based forwarding and deflection mechanism can be used to support differentiated service by appropriately defining the set of alternate output ports for each label. In the previous example, the different labels can represent different classes of services that utilize different routing schemes. For example, label "A" may represent best-effort traffic using shortest path routing (no deflection); label "B" may represent traffic that requires low packet loss and can be deflected to any output port; and label "C" may represent traffic that has delay constraints and thus can only be deflected to a limited number of output ports.

Label paths can be set up statically for different classes of traffic and can be updated periodically for traffic engineering purposes. In this manner, the control overhead for establishing label paths is minimum. On the other hand, label paths can also be set up dynamically based on individual label-path requests. The dynamic approach may be more efficient and allow for greater flexibility; however, it has the additional cost of control complexity and overhead. In this paper, as the first step of our study, we assume that label paths are established offline in a static manner.

C. Framework for Differentiated Contention Resolution

In this section, we outline the various parameters for supporting differentiated service within the label-based contention resolution framework.

Based on different loss and delay requirements, differentiated service can be implemented by different buffering and different deflection-routing policies within the label-based control scheme. We define a control vector $V_l = [K^l, M^l, D^l, A^l]$ corresponding to a label path l. K^l is the number of buffers that packets of a given label l can use at a node, M^l is the maximum number of times that a packet is allowed to recirculate in the buffers at a node, D^l represents the deflection-routing policy, and A^l is the number of deflection alternatives at a node for a packet of label l.

In this paper, we consider a buffering-first policy. That is, a packet will be forwarded to buffers if the primary output port is occupied by another packet, then the packet will be forwarded to one of its predefined deflection output ports if the buffers are not available. The details of the control scheme follows.

Buffering options

- K^l: *Limit of buffer usage*: We can limit the number of buffers that a flow or class can use at a node. Specifically, from a packet-loss perspective, all of the buffers at a node may be used by high-priority traffic, while only a subset of the buffers may be used for low-priority traffic. In this manner, some buffers are dedicated to higher priority applications in order to guarantee lower packet loss.
- M^l: *Limit of recirculation time*: In general, we must limit the overall time that a packet can stay in the switch; otherwise, one packet may recirculate indefinitely in the system. The maximum number of times that a packet may traverse a recirculation buffer may also be utilized for service differentiation purpose. Intuitively, a larger limit may lead to lower packet loss while increasing the buffering delay.

Deflection options

- D^l: *Deflection routing algorithm*: Deflection schemes can be classified based on whether or not loops are allowed. Loops occur when a packet is forwarded to a node that has already been visited on the path to the destination. An example of deflection routing that allows loops is *shortest path deflection* (SPD), in which all possible outputs are sorted by distance to the destination. If the primary output of an incoming packet is occupied, the switch will try to forward it to the output that is available and has the shortest distance to the destination. By allowing loops in the network, SPD provides lower packet loss under lower traffic load but incurs higher end-to-end delay. Examples of loopless deflection are *destination-based loopless deflection* (D-LLD) [25] and *source-destination based loopless deflection* (SD-LLD), which will be presented in the next section. In these schemes, forwarding tables are set in a manner that avoids loops. Since loops are not allowed in the network, the maximum end-to-end propagation delay is limited.
- A^l: *Number of deflection alternatives*: The number of deflection alternatives is the maximum number of alternative output ports on which a packet can be forwarded. In nondeflection schemes, the number of deflection alternatives is equal to one. Therefore, each node has to maintain only the primary output port in a forwarding table for a given label path. In this case, if the primary output is occupied, then the packet will be dropped. For example, when the *shortest path* (SP) scheme is used, the primary output to a given destination can be determined by running Dijkstra's shortest path algorithm to find the output. In this case, the control vector for the label path can be expressed as $V_l = [K^l, M^l, \mathrm{SP}, 1]$. If the number of deflection alternatives is greater than one, then each node must maintain multiple alternative output ports for a given label path. In this case, the packet-loss ratio may be smaller than nondeflection schemes. However, deflection may also increase the end-to-end propagation delay.

To support different delay requirements, a routing scheme can be delay-limited or delay-unlimited. If the applications do not have delay requirements, such as e-mail and file transfer protocol (FTP), then a delay-unlimited routing algorithm can be used. For example, SPD routing can be used to achieve lower packet loss, while possibly incurring packet loops and higher end-to-end delay.

If the application has delay requirements, such as a video stream, then a delay-limited routing algorithm must be used. To support the delay constraint when using deflection routing, two classes of loopless algorithms, D-LLD and SD-LLD, will be proposed in Section III, where additional factors, such as the

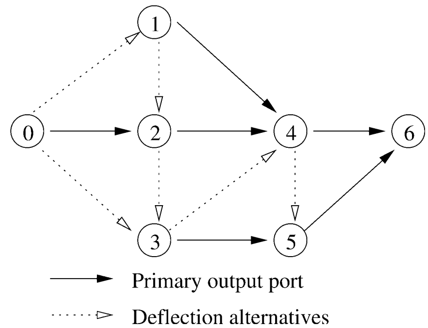


Fig. 3. Sample of D-LLD scheme. (Destination is node 6.)

number of deflection alternatives, can be used to obtain further tradeoffs between delay and loss.

We now use an example to illustrate the differentiation scheme. Consider a network where each node has three recirculation buffers and suppose there are three classes of traffic: loss-sensitive, delay-sensitive, and best-effort traffic. To support loss-sensitive traffic, we can define control vector $V_0 = [3, 3, \mathrm{SPD}, 2]$, in which three buffers are used, recirculating times are limited to three, SPD is used, and the number of deflection alternatives is equal to two. To support delay-sensitive traffic, a control vector $V_1 = [3, 1, \text{D-LLD}, 2]$ can be defined, in which three buffers are used, recirculating times are limited to one, D-LLD is used, and the number of deflection alternatives is equal to two. To support best-effort traffic, a control vector $V_2 = [0, 0, \mathrm{SP}, 1]$ can be defined, in which no buffering is allowed, and no deflection is allowed. We will show the performance of this scheme in Section V.

III. Loopless Deflection-Routing Algorithms

In this section, we first present a D-LLD routing algorithm, and we then present an SD-LLD routing algorithm.

A. Destination-Based Loopless Deflection

In this section, we assume that labels are defined on a per-destination basis. By utilizing destination-based labeling, the number of label entries at each node can be kept to a minimum, since packets from different sources can share the same entry in the forwarding table as described in Section II-A.

Fig. 3 illustrates the primary paths and deflection alternatives on all nodes for a given destination node 6. Note that in destination-based labeling, the label of a packet will remain the same throughout the network. Furthermore, the primary paths specified for a given destination will define a spanning tree on the network, with the destination node at the root of the tree. By examining the spanning tree for a given destination, we note that deflection-alternative links may be added to the tree in a manner that avoids routing loops.

The process of defining the label entries at each node can be divided into two subproblems. The first problem is to determine the primary outgoing link for each destination at each node. In this paper, we assume that the link that is on the shortest path to the destination is chosen as the primary outgoing link at a node for that destination. These links may be found by running Dijkstra's shortest path algorithm for each source–destination pair in the network and choosing the first-hop link on each of the shortest path routes. The link weights in the shortest path algorithm are determined by the physical distance of each link. An alternative approach for determining the primary links is to choose the links in a manner that balances the load in the network. Such approaches are beyond the scope of this paper.

The second problem is to find the set of deflection alternatives for each destination at each node, given the set of primary links defined in the previous problem. Since the shortest spanning tree for a given destination node does not have any cycles by definition, the deflection alternatives at each node must be defined in a way that eliminates the possibility of routing loops. The main idea of the D-LLD algorithm is as follows: 1) select nodes one at a time, and for each node select its deflection alternatives and 2) once the deflection alternatives have been selected, remove the node from the graph. By selecting leaf nodes that are furthest from the root one by one and by deleting the node after its deflection output port has been selected, the algorithm ensures that no deflections are made to nodes that are further from the destination than the selected node and that packets, upon departing from the selected node, can never return to that node. A detailed discussion on how to choose deflection alternatives can be found in [25].

Since the proposed algorithm does not allow loops, it is possible that a node will not have any deflection alternatives for a given destination. In particular, those nodes that are closer to the destination are less likely to have deflection alternatives than nodes that are further from the destination. For example, node 5 in Fig. 3 has only one path to destination. By restricting deflection at these nodes, the packet losses may increase.

B. SD-Based Loopless Deflection

In this section, we assume that labels are defined on the basis of per-source–destination pairs. Although the number of label entries at each node will be increased, this scheme can nevertheless improve the routing flexibility and provide more traffic engineering options.

Similar to that of the D-LLD, the objective of the SD-LLD routing algorithm is to find the primary path and deflection alternatives at each node such that there are no routing loops.

For each source–destination pair, the algorithm selects nodes one at a time, starting with the source node, and determines the primary and alternate output ports for the selected node. To ensure the loopless aspect, nodes that have already been selected cannot be chosen as deflection alternatives. The order of selected nodes is such that any deflected packet will eventually reach the destination.

An example of the algorithm is illustrated in Fig. 4. Suppose node 5 is the source, and node 6 is the destination. We first remove the source node and run a shortest path algorithm in the remaining graph [see Fig. 4(a)]. Then, based on the distances from each node to the destination, we determine the forwarding table for a packet from node 5 to node 6. At node 5, the output list in the forwarding table will be set to (6, 4, 3). In this case, the primary output is given as node 6, and the second and third outputs are node 4 and node 3, respectively.

We then choose a node that is adjacent to the source node and that has the largest distance to the destination [node 3 in

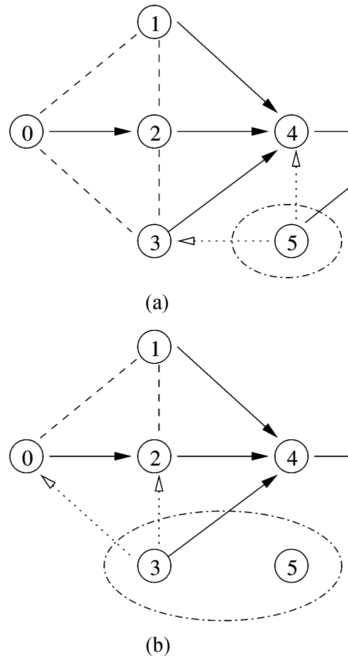


Fig. 4. Sample of SD-LLD. (Source is node 5, and destination is node 6.)

Fig. 4(b)]. By using the same technique, we can quickly determine the forwarding table entry at node 3 as (4, 2, 0), in which node 4 is given as the primary output.

Similarly, we choose another node in the remaining graph and then set its forwarding table. The node we choose next is a node that is adjacent to node 3 or 5 (denoted as the source group hereafter) and has the largest distance to the destination in the remaining graph. This algorithm will stop when all of the forwarding tables have been set.

We observe from Fig. 4 that we have several options for forwarding a packet from node 5 to node 6. In comparison, there is only one choice for a packet from node 5 to 6 in Fig. 3. We predict that our algorithm will utilize network resources as much as possible; thus, the probability of successfully forwarding packets to their destination may increase. In the remainder of this section, we present a formal description of the algorithm and prove its correctness.

1) Notation:

- N: total number of nodes in the network.
- V: set of all nodes in the network.
- E: set of all edges in the network.
- $G(V,E)$: a graph containing node set V and edge set E.
- $e(i,j)$: edge between nodes i and j.
- s: source node $s \in \{1,2,\cdots,N\}$.
- d: destination node $d \in \{1,2,\cdots,N\}$.
- W: a set of nodes $W \subset V$.
- $A(W)$: set of nodes where $A(W) \cap W = \emptyset$ and $\forall i \in A(W), \exists j \in W$ such that $e(i,j)$ exists.
- $E(W)$: set of edges where $\forall e(i,j) \in E(W), i \in W$ and $j \in W$.
- $G'(W)$: a subgraph of $G(V,E)$ which is equivalent to $G(W, E(W))$.
- V^s: source group, a set of nodes including source node s.
- V^d: destination group; a set of nodes, including destination node d.
- A^s: adjacent group, $A^s = A(V^s)$.
- $D(G,i,j)$: shortest distance between node i and j in graph G.
- P^{sd}: set of nodes in the primary path for source–destination pair sd.
- v_k^{sd}: the kth ($k \in \{1,2,\cdots,N\}$) node chosen while setting forwarding table for source–destination pair sd.
- A_k^{sd}: a group of nodes that can be put in the forwarding table on node v_k^{sd} for source–destination pair sd.

2) Algorithm Description: For each source–destination pair sd, we initialize the forwarding table on each node as follows:

Step 1) Let $k=1, V_1^s = \{s\}, v_1^{sd} = s, P_1^{sd} = P^{sd} - \{s\}$.

Step 2) Let $V_k^d = V - V_k^s, A_k^s = A(V_k^s)$.

Step 3) Calculate shortest path spanning tree T_k^{sd} for $G'(V_k^d)$.

Step 4) Let $A = A(\{v_k^{sd}\})$, let $A_k^{sd} = A \cap V_k^d$. Set the forwarding table on v_k^{sd} as

$$(w_1, w_2, \cdots, w_n)$$

where w_1 is in the primary path for s and d; $w_i \in A_k^{sd}$; and

$$D(G'(V_k^d), w_i, d) \leq D(G'(V_k^d), w_j, d), \forall i,j (1 < i < j).$$

Step 5) Select a node $v^* \in A_k^s$ such that

$$D(G'(V_k^d), v^*, d) = \max_{v \in A_k^s} D(G'(V_k^d), v, d).$$

If $v^* \notin P^{sd}$, then let $v_{k+1}^d = v^*$ and $P_{k+1}^{sd} = P_k^{sd}$; otherwise, choose v_{k+1}^d from P_k^{sd} such that $v_{k+1}^d \in A_k^s$ and $P_{k+1}^{sd} = P_k^{sd} - \{v_{k+1}^d\}$ is still connected.

Step 6) $k = k+1$, if $k = N$, stop.

Step 7) $V_k^s = V_{k-1}^s + \{v_k^{sd}\}$; go to Step 2).

3) Algorithm Correctness: In this section, we first prove that the algorithm can ensure the loopless characteristic. We then show that a packet can be successfully forwarded to its destination node regardless of the packet's current node.

Suppose a packet from source s to destination d arrives to node v, where $v = v_k^{sd}$. We can see that $A_k^{sd} \cap V_k^s = \emptyset$ in Step 4) of the algorithm. Therefore, the packet cannot be forwarded to any node in V_k^s. In addition, $\forall v_l^{sd} \in V_k^d, l \geq k$ and $A_l^{sd} \cap V_k^s = \emptyset$, since $V_l^s \supset V_k^s, \forall l > k$. We conclude that the algorithm itself ensures the loopless forwarding of packets in the network.

To prove the correctness of packet forwarding in our algorithm, we first make two assumptions:

1) $G(V,E)$ is a connected graph.
2) $\forall v \in V, G'(V - \{v\})$ is a connected graph.[1]

We first derive that for any source–destination pair sd and $k, 1 \leq k \leq N-1, G'(V_k^d)$ is connected.

Suppose $\exists w \in V$ such that the path between w and d exists in $G'(V_{k-1}^d)$ but does not exist in $G'(V_k^d)$. We let W be a set of nodes where 1) $w \in W$, 2) $W \subset V_k^d$, and 3) $G'(W)$ is a connected graph.

[1]If this assumption does not hold, then Step 4) must be slightly modified such that $\forall w_i$ in the forwarding table $D(G'(V_k^d), w_i, d) < \infty$.

Obviously, $P_k^{sd} \cap W = \emptyset$, since there is no path between w and d in $G'(V_k^d)$. We further conclude that for all $v \in W$, the path between v and d in $G'(V_{k-1}^d)$ must contain node v_k^{sd}. This is because $V_k^d = V_{k-1}^d - \{v_k^{sd}\}$. Therefore

$$D(G'(V_{k-1}^d), v, d) > D(G'(V_{k-1}^d), v_k^{sd}, d), \quad \forall v \in W$$

since, in our algorithm, we choose the next node from the adjacent group that has the largest distance to the destination if it is not in P^{sd}. We can claim that $W \cap A_{k-1}^s = \emptyset$. Consequently, $G'(W)$ is not connected to $G'(V - W - \{v_k^{sd}\})$. This means that the graph is not connected after we remove node v_k^{sd} which contradicts Assumption 2). Therefore, w does not exist in our algorithm, and $G'(V_k^d)$ is always connected for all source–destination pairs and any k.

IV. Analytical Model

In this section, we present an analytical model to evaluate the packet loss and delay performance of different buffering and deflection-routing schemes. The analysis may be applied to any arbitrary mesh topology. The main idea in this analytical model is to calculate loss probability for each connection by using a reduced-load approximation algorithm.

We first separate traffic in a network into different flows, where each flow consists of packets that have the same source and destination and that require the same QoS. For any flow f, we assume that packets arrive according to a Poisson process with rate λ^f. To simplify the analysis, we consider only the case that $K^l = K$ for each class, where K is the total number of fiber delay lines in the switch.

We also assume that there is no wavelength conversion; thus, we only consider a single wavelength plane. Therefore, we can model each link as an M/D/1/1 queuing system. Let λ_{ij} be the packet arrival rate on link ij and P_{ij} be the packet blocking probability on link ij. The packet-blocking probability on link ij can be given as

$$P_{ij} = \frac{\lambda_{ij} L}{1 + \lambda_{ij} L}. \tag{1}$$

For a given flow f, we denote λ_{ij}^f as the packet arrival rate on link ij. Then, we have

$$\lambda_{ij} = \sum_f \lambda_{ij}^f. \tag{2}$$

λ_{ij}^f depends on whether link ij is on the primary path or on the deflection path. To analyze λ_{ij}^f, we first define λ_i^f as the total arrival rate of f from all incoming links of node i and define λ_i^{fb} as the arrival rate of packets that pass through buffers and eventually depart from the primary output of f on node i. Let S_f and D_f be the source and destination node of flow f. If link ij is on the primary path to D_f, then the applied load will be both the load offered by all previous-hop links toward node i and the load from local buffers. If link ij is the lth $(l > 1)$ deflection link at node i, and link ik is the $(l-1)$th deflection link to destination D_f, then the applied load on link ij will be the load that is blocked on link ik and cannot enter the buffer. To simplify the notation, we let $w_i^f(l)$ be the next node corresponding to the lth alternate deflection port at node i. For example, $w_i^f(1)$ is the primary output port at node i. The parameter λ_{ij}^f can be calculated as

$$\lambda_{ij}^f = \begin{cases} \lambda_i^f + \lambda_i^{fb}, & j = w_i^f(1) \\ \lambda_i^f \cdot P_{iw_i^f(1)} - \lambda_i^{fb}, & j = w_i^f(2) \\ \lambda_{iw_i^f(l-1)}^f \cdot P_{iw_i^f(l-1)}, & j = w_i^f(l), (l > 2). \end{cases} \tag{3}$$

The parameter λ_i^f can be calculated as

$$\lambda_i^f = \begin{cases} \lambda^f, & i = S_f \\ \sum_{k, i = w_k^f(1)} \left[\lambda_k^f \cdot (1 - P_{ki}) + \lambda_k^{fb}\right] & \\ \quad + \sum_{k, i \neq w_k^f(1)} \left[\lambda_{ki}^f \cdot (1 - P_{ki})\right], & i \neq S_f. \end{cases} \tag{4}$$

From (3) and (4), we can observe that, for a given flow f, the attempted arrival rate to buffers at node i is

$$\lambda_i^f \cdot P_{iw_i^f(1)}$$

where $P_{iw_i^f(1)}$ is the probability that a packet of flow f is blocked on its primary output at node i.

We let P_i^{fb} be the conditional probability that packets cannot depart from the primary output $w_i^f(1)$ given that these packets have entered buffers at node i. Therefore, λ_i^{fb} can be calculated as

$$\lambda_i^{fb} = \left(\lambda_i^f \cdot P_{iw_i^f(1)}\right) \cdot (1 - P_i^{fb}). \tag{5}$$

To calculate the conditional probability P_i^{fb}, we make two assumptions: 1) that each packet can enter the fiber delay line at a given node up to M^f times and 2) that the probability that no fiber delay line is available at node i is P_i^b.

Since the length of any fiber delay line is L units, one packet that leaves a buffer and enters the switching element can use the same buffer if there is no limitation to enter again. Therefore, we have

$$P_i^{fb} = 1 - (1 - P_i^b) \times \left(1 - \left(r_i^f\right)^{M^f}\right) \tag{6}$$

where we define r_i^f as the probability that the primary output $j (j = w_i^f(1))$ is not available when the packet leaves the buffer. r_i^f can be estimated by using the approximation

$$r_i^f = 1 - e^{(-\lambda_{ij} \times L/2)}. \tag{7}$$

We now discuss how to calculate P_i^b in (6). Let d_i^f be the average duration that a packet in flow f stays in a buffer. Since a packet in flow f can stay in a buffer for up to M^f times, we have

$$d_i^f = \sum_{m=1}^{M^f - 1} \left[m \cdot L \cdot \left(r_i^f\right)^{m-1} \cdot \left(1 - r_i^f\right)\right] + M^f \cdot L \cdot \left(r_i^f\right)^{M^f - 1}. \tag{8}$$

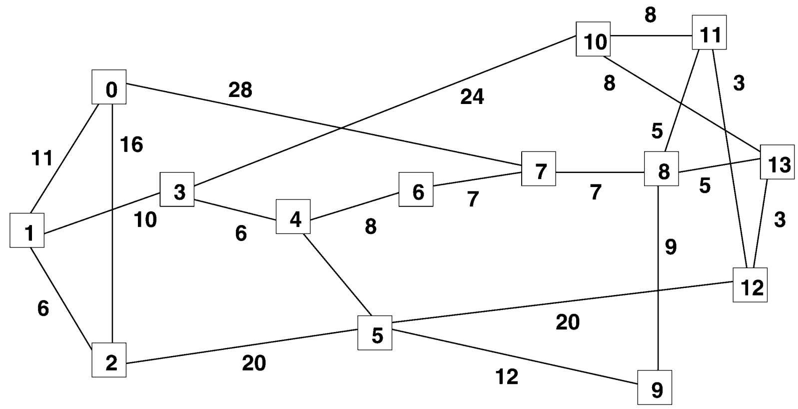


Fig. 5. The 14-node network topology.

Let d_i be the average duration that a packet of any flow stays in a buffer at node i. Notice that the attempted arrival rate of flow f to buffers at node i is $\lambda_i^f \cdot P_{iw_i^f(1)}$. Thus, we have

$$d_i = \frac{\sum_f d_i^f \cdot (\lambda_i^f P_{iw_i^f(1)})}{\sum_f \lambda_i^f P_{iw_i^f(1)}}. \tag{9}$$

We can then use the well-known Erlang-B formula to calculate P_i^b, since the Erlang-B formula is insensitive to the distribution of the service time, as follows:

$$P_i^b = \frac{\rho_i^K/K!}{\sum_{k=0}^K \rho_i^k/k!} \tag{10}$$

where

$$\rho_i = d_i \times \sum_f \lambda_i^f P_{iw_i^f(1)} = \sum_f d_i^f \cdot \left(\lambda_i^f P_{iw_i^f(1)}\right). \tag{11}$$

The packet-loss probability of flow f can be calculated as

$$P^f = 1 - \frac{\lambda_{D_f}^f}{\lambda^f} \tag{12}$$

and then the average packet-loss probability P in the network can be calculated as

$$P = \frac{\sum_f P^f \cdot \lambda^f}{\sum_f \lambda^f}. \tag{13}$$

To calculate packet-loss probability recursively, we use the following reduced-load approximation algorithm.

1) Initialize all $P_{ij} = 0$ for any link ij.
2) Calculate all $\lambda_{ij}, \lambda_i^f$ and λ_{ij}^f by using (2)–(4).
3) Calculate all P_{ij} by using (1).
4) Calculate all λ_i^{fb} by using (5)–(11).
5) Calculate P by using (12) and (13).
6) Stop if P converges; otherwise, go to Step 2).

For a given flow f, the average end-to-end delay for packets that successfully reach their destination D_f can be calculated as

$$D^f = \frac{\sum_{pq} d_{pq}\hat{\lambda}_{pq}^f}{\lambda^f(1-P^f)} + \frac{\sum_p d_p\hat{\lambda}_p^f}{\lambda^f(1-P^f)} \tag{14}$$

where d_{pq} denotes the propagation delay on link pq, $\hat{\lambda}_{pq}^f$ denotes the f traffic that passes through link pq and successfully reaches the destination D_f, and $\hat{\lambda}_p^f$ denotes the f traffic that passes through node p and successfully reaches the destination D_f.

To calculate $\hat{\lambda}_{pq}^f$, we first define $\lambda_i^f(pq)$ and $\lambda_{ij}^f(pq)$ as the packet arrival rate at node i and link ij, provided that all packets have traversed link pq. By using the method we developed for calculating λ_i^f and λ_{ij}^f, we can calculate $\lambda_i^f(pq)$ and $\lambda_{ij}^f(pq)$. Therefore, we have

$$\hat{\lambda}_{pq}^f = \lambda_{D_f}^f(pq). \tag{15}$$

We can calculate $\hat{\lambda}_p^f$ in a similar manner.

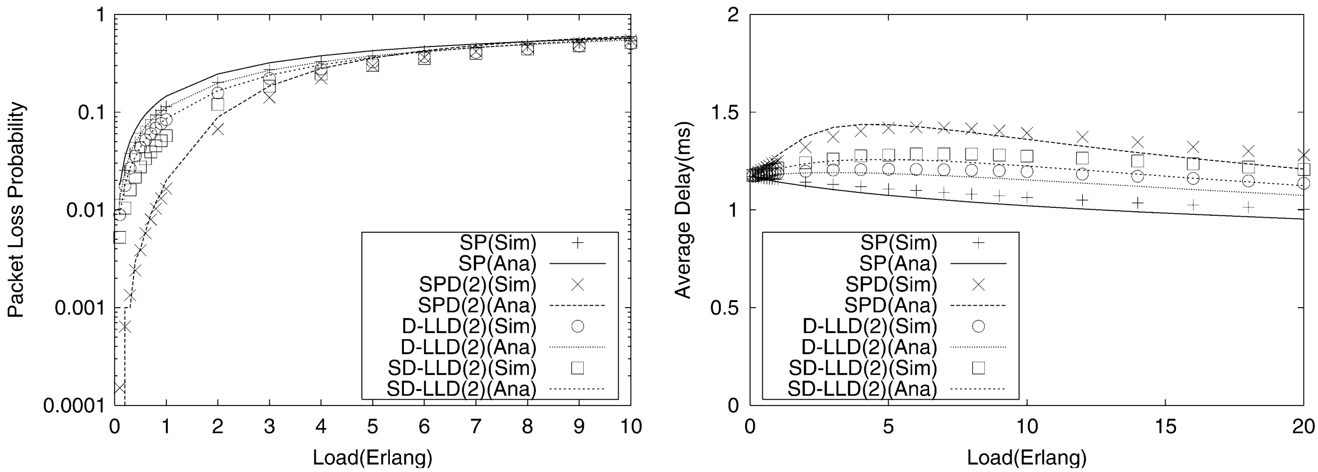
Finally, the average end-to-end delay can then be calculated by using

$$D = \frac{\sum_f D^f \cdot \lambda^f(1-P^f)}{\sum_f \lambda^f(1-P^f)}. \tag{16}$$

By using the reduced-load approximation, the above analysis can be applied to any loopless routing scheme. In order to analyze schemes that result in routing loops, the analysis must be modified slightly in order to avoid infinite path lengths. When calculating P_{ij} and λ_{ij}^f, the analysis will stop evaluating a path if the additional load on the next link in the path is less than some small value ϵ. When evaluating the packet-loss probabilities, the analysis will stop evaluating a path once it reaches a certain number of hops.

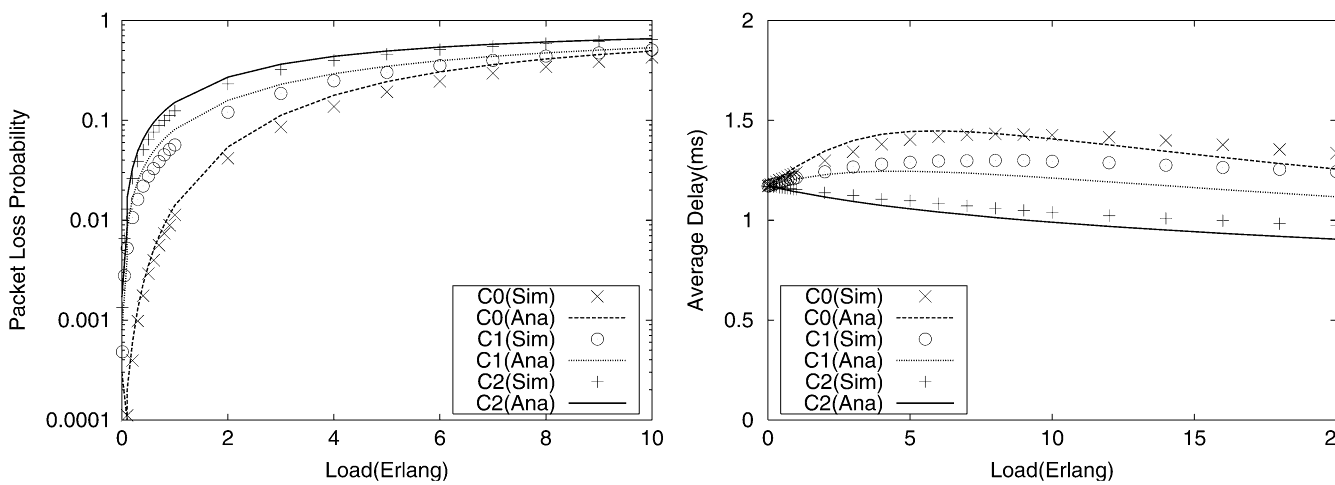
V. Numerical Results

In this section, we evaluate the performance of different contention resolution schemes through simulation and analysis. We use two topologies to conduct the experiments: a 14-node network topology, illustrated in Fig. 5, in which the numbers on each link denote the physical length in units of 10 km; and a 16-node torus–mesh network in which the length of each link is 10 km. In both topologies, each link represents a bidirectional channel with 10-Gb/s transmission rate. We assume that the arrival of packets follows a Poisson process and that the length of each packet is 10 Kb. We further assume that the traffic is uniformly distributed over all source–destination pairs. For buffering, we assume there are three recirculation buffers at each



(a) (b)

Fig. 6. Performance of different deflection-routing schemes in the 14-node network with no buffer for a single class ($A^l = 2$).



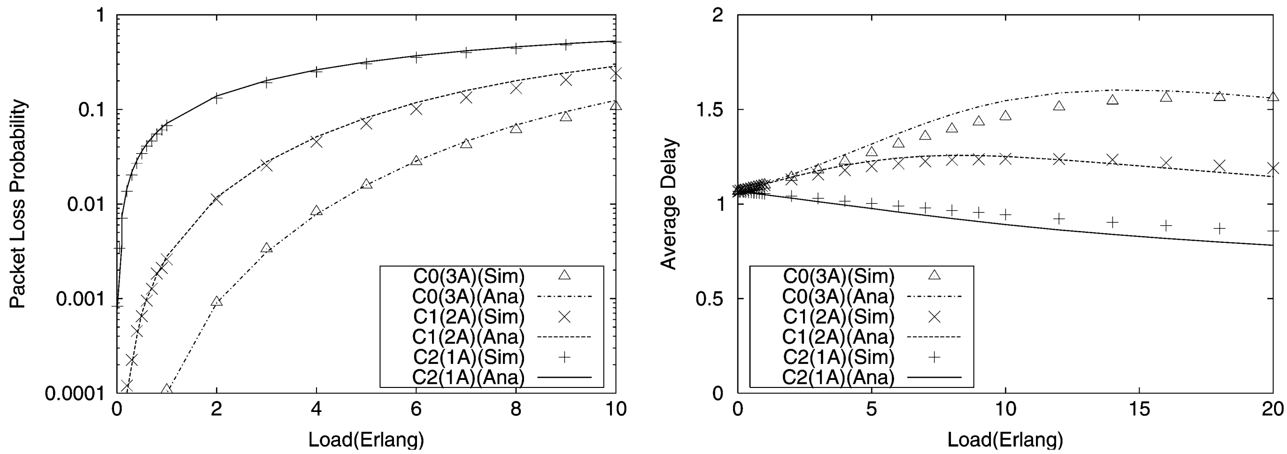
(a) (b)

Fig. 7. Performance of different deflection-routing schemes in the 14-node network with no buffer for three classes, in which SPD with $A^l = 2$ is used for C_0; SD-LLD with $A^l = 2$ is used for C_1; and SP is used for C_2.

node, all with one unit of length. If there are multiple classes of traffic in the network, we assume that the total traffic load is distributed evenly among all classes.

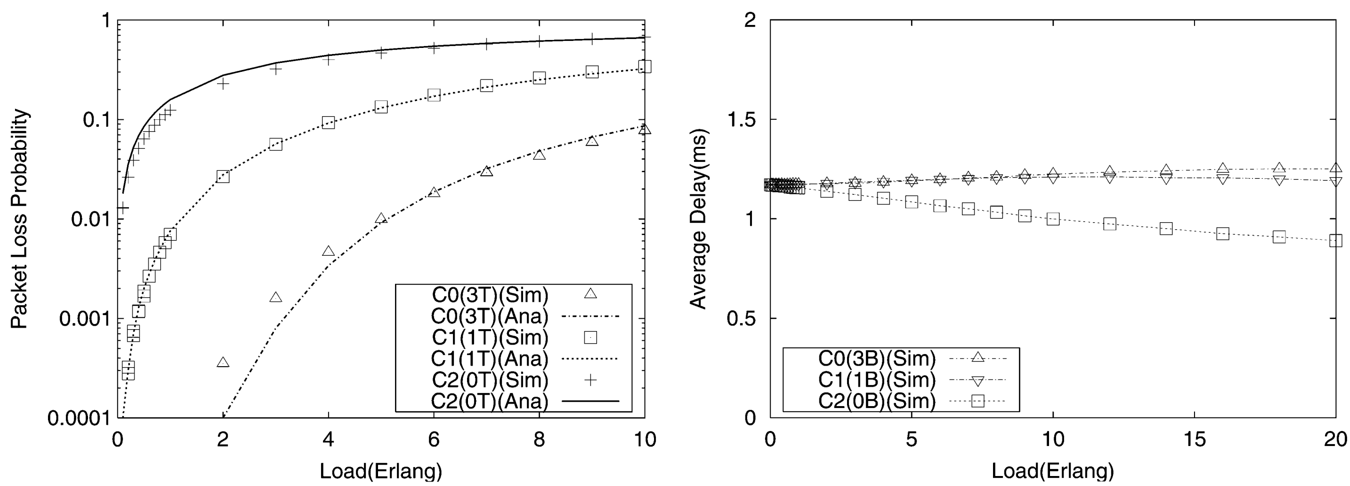
Fig. 6 shows the packet loss and average delay performance of different deflection-routing schemes in the 14-node network with no buffer. We assume that there is only one class of traffic in each experiment, and we evaluate four routing schemes (SP, D-LLD, SD-LLD, and SPD), in which we set the number of deflection alternatives A^l to two for all deflection cases. We observe that, under low and modest traffic loads, SPD outperforms all other schemes in terms of packet loss; however, it has the highest average delay. SPD results in low packet loss because it allows loops in the networks; therefore, it utilizes the network resources more efficiently when the traffic is low; however, since packets traverse longer paths before they arrive at their destinations, the end-to-end delay will be increased. We observe that, under high traffic loads, SPD will incur a higher packet-loss ratio. This effect is due to the deflected packets consuming additional network resources and affecting the delivery of other packets. In Fig. 6(a), we note that the D-LLD and SD-LLD schemes provide a slight improvement in loss performance over the case with no deflection (SP), while having larger packet loss over the deflection scheme without any looping restrictions (SPD). The reason for this behavior is that, with deflection routing, a packet has a greater chance to reach its destination when the total packet-loss probability is low. The restrictions placed on deflections in the loopless deflection case limit the number of nodes at which deflections can take place. These limitations lead to higher packet losses compared with the SPD scheme. From Fig. 6(b), it can be seen that both D-LLD and SD-LLD achieve significantly lower delay than SPD and slightly higher delay than SP. Deflection increases the average delay; however, the loopless deflection schemes limit loops so that the total distance that a packet travels is bounded. For D-LLD and SD-LLD, we can see that SD-LLD has better loss performance but slightly larger average delay. The reason for this performance is that SD-LLD has more deflection alternatives and more routing diversity than D-LLD, since packets with different sources but the same destination will have different routing and deflection alternatives.

Fig. 7 compares different deflection-routing schemes in the 14-node network without buffers, where we assume that there



(a) (b)

Fig. 8. Performance of SPD with different alternatives in the 16-node torus–mesh network with no buffer for three classes, in which SPD with $A^l = 3$ is used for C_0; SPD with $A^l = 2$ is used for C_1; and SPD with $A^l = 1$ is used for C_2.



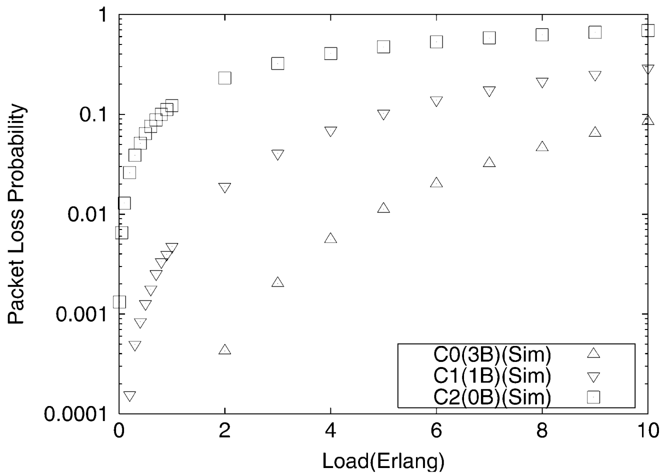
(a) (b)

Fig. 9. Performance of buffering schemes with different recirculation times in the 14-node network for three classes, in which SP with $K^l = 3$ and $M^l = 3$ is used for C_0; SP with $K^l = 3$ and $M^l = 1$ is used for C_1; and SP with no buffer is used for C_2.

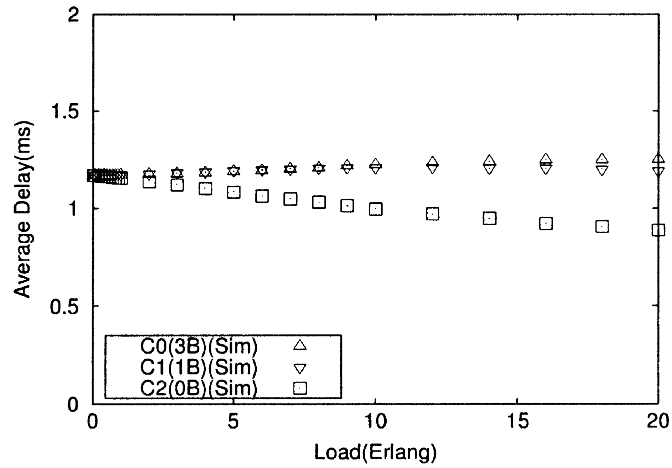
are three classes of traffic in the network. In this paper, we assume that there are three classes of traffic in the network and each class of traffic has different QoS requirements. Loss-sensitive traffic strictly requires low packet loss and does not care about delay; delay-sensitive traffic is delay limited; and best-effort traffic has no strict loss or delay requirements. The control vectors for the Class 0 (C_0), Class 1 (C_1), and Class 2 (C_2) label paths are $V_0 = [0, 0, \mathrm{SPD}, 2], V_1 = [0, 0, \mathrm{SD\text{-}LLD}, 2]$, and $V_2 = [0, 0, \mathrm{SP}, 1]$, respectively. In other words, different classes of traffic each use a different deflection policy. It can be observed that C_0 traffic has the lowest packet-loss rate; thus, C_0 can be used to support loss-sensitive traffic that does not have strict delay requirements. We also note that the average delay of C_1 traffic remains stable over a large range of traffic loads; thus, C_1 can be used to support traffic with guaranteed delay requirements. Finally, C_2 can be used to support best-effort traffic.

In Fig. 8, we demonstrate the performance of SPD with different number of deflection alternatives in the 16-node torus–mesh network. We assume that C_0 label paths have up to three output alternatives; C_1 label paths have two output options; and C_2 traffic can be forwarded to only one output. From Fig. 8, we can see that a larger number of alternatives will lead to smaller packet loss but will also increase the end-to-end delay. For loopless deflection algorithms, our experiments (not included) show that the benefit of having more than two deflection outputs is not significant in both topologies due to the small nodal degree and the limitations of the loopless constraint.

Fig. 9 shows the performance of buffering schemes with different number of recirculation times in the 14-node network. We select SP as the routing scheme for all three classes of traffic and the control vectors for three classes of label paths are $V_0 = [3, 3, \mathrm{SP}, 1], V_1 = [3, 1, \mathrm{SP}, 1]$, and $V_2 = [3, 0, \mathrm{SP}, 1]$, respectively. Thus, C_0 packets may traverse the recirculation buffers three times; C_1 packets may traverse the buffers once; and C_2 packets may not traverse the buffers. Fig. 9(a) illustrates packet loss versus load for different schemes. We observe that, by limiting the number of times that a packet can enter a buffer, we can provide differentiated packet loss, while also avoiding the sig-



(a)



(b)

Fig. 10. Performance of buffering schemes with different buffer usage in the 14-node network for three classes, in which SD-LLD with $K^l = 3, M^l = 3$, and $A^l = 2$ is used for C_0; SD-LLD with $K^l = 1 and M^l = 3, A^l = 2$ is used for C_1; and SD-LLD with no buffer is used for C_2.

nificant attenuation introduced by recirculation. As expected, a larger limit can lead to a smaller loss ratio but can also lead to higher end-to-end delay.

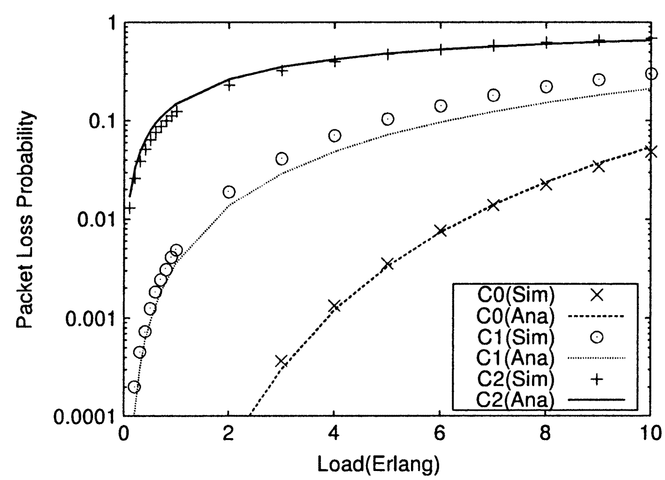
Fig. 10 illustrates the performance of buffering schemes with different buffer usages in the 14-node network. We use SD-LLD as the routing scheme for all three classes of traffic and the control vectors for three classes are $V_0 = [3, 3, \text{SD-LLD}, 2], V_1 = [1, 3, \text{SD-LLD}, 2]$, and $V_2 = [0, 0, \text{SD-LLD}, 2]$, respectively. Thus, C_0 traffic may use up to three buffers if all three buffers are available; C_1 may traverse only the lowest index buffer, even if other higher index buffers are available; and C_2 traffic will not traverse the buffers at all. We note that classes with larger buffer usage will have better performance in terms of packet loss. From Figs. 9 and 10, we can observe that our buffering schemes have successfully provided differentiated service.

Fig. 11 shows the simulation and analysis results for combined buffering and deflection-routing schemes. The corresponding control vectors for three classes of label paths are $V_0 = [3, 3, \text{SPD}, 2]$, $V_1 = [3, 1, \text{D-LLD}, 2]$, and $V_2 = [0, 0, \text{SP}, 1]$, respectively. Thus, SPD with $K^l = 3$, $M^l = 3$, and $A^l = 2$ is used to support C_0 traffic; D-LLD with $K^l = 3, M^l = 1$, and $A^l = 2$ is used to support C_1 traffic; and SP with no buffer is used to support C_2 traffic. Comparing Fig. 11(a) with Fig. 9(a), we can see that the loss for C_0 and C_1 traffic has been decreased by exploiting deflection routing.

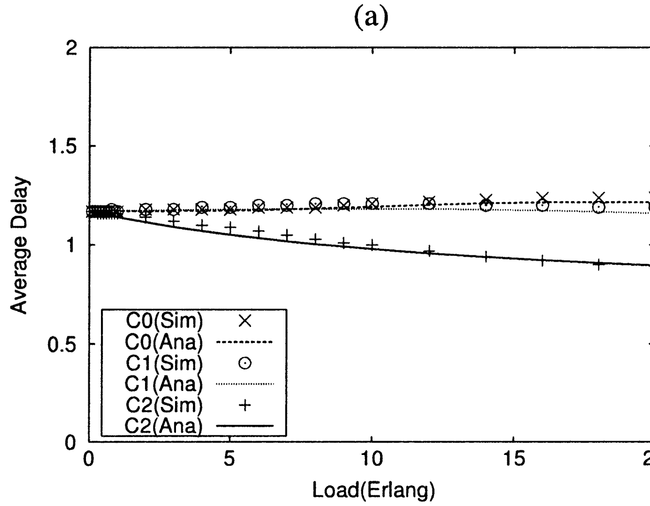
Fig. 12 shows the simulation results for deflection-routing schemes in the 14-node network for three classes under Pareto traffic. For Pareto traffic, we use

$$\frac{\alpha t_{\min}^{\alpha}}{t^{\alpha+1}}, \quad t > t_{\min}$$

where $\alpha = 1.5, t_{\min} = (\alpha - 1/\alpha)\bar{t}$, and $\bar{t}$ is the average interarrival time. We adopt the same control vectors for the three classes of label paths as in Fig. 7. We observe that the proposed schemes can still achieve sufficient differentiation under Pareto arrivals. Moreover, the performance is similar to that of the exponential arrivals in terms of both loss and delay.



(a)



(b)

Fig. 11. Performance of combination of buffering and deflection-routing schemes in the 14-node network for three classes in which SPD with $K^l = 3, M^l = 3$, and $A^l = 2$ is used for C_0; D-LLD with $K^l = 3, M^l = 1$, and $A^l = 2$ is used for C_1; and SP with no buffer is used for C_2.

The above results also show that the packet loss and delay analysis are highly accurate for all buffering and routing schemes under different traffic loads.

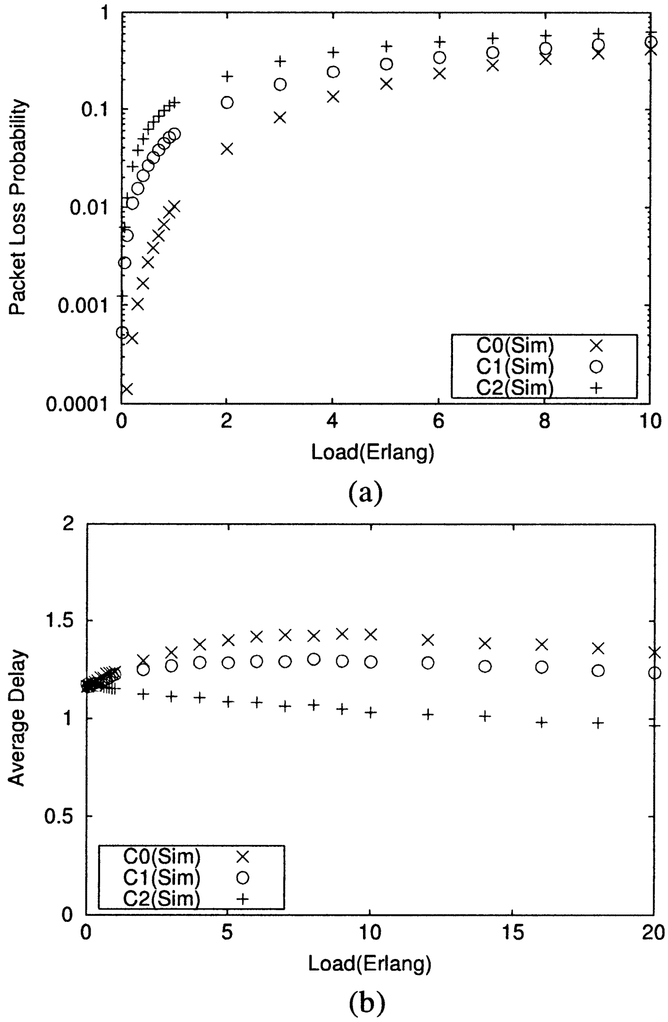


Fig. 12. Performance of deflection-routing schemes in the 14-node network with no buffer for three classes under Pareto traffic in which SPD with $A^l = 2$ is used for C_0; SD-LLD with $A^l = 2$ is used for C_1; and SP is used for C_2.

VI. CONCLUSION

This paper proposed a framework for differentiated contention resolution in photonic packet-switched networks. In this framework, different recirculation buffering and deflection routing schemes can be applied. Two classes of loopless deflection algorithms are provided in order to support different loss and delay requirements. An analytical model is also developed to evaluate the packet-loss probability and the end-to-end delay for different schemes. Simulation and analysis results show that our scheme can provide sufficient options for supporting differentiated service requirements. The results also show that our analytical model is highly accurate.

One area of further work would be to incorporate traffic engineering and load balancing into the deflection routing schemes. In addition, the proposed framework can be extended to support dynamic label assignment in which label paths requests arrive to the network dynamically.

REFERENCES

[1] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 2001.

[2] S. Yao, B. Mukherjee, and S. Dixit, "Advances in photonic packet switching: An overview," *IEEE Commun. Mag.*, vol. 38, pp. 84–94, Feb. 2000.

[3] G. I. Papadimitriou, C. Papazoglou, and A. S. Pomportsis, "Optical switching: Switch fabrics, techniques, and architectures," *J. Lightwave Technol.*, vol. 21, pp. 384–405, Feb. 2003.

[4] D. K. Hunter and I. Andronovic, "Approaches to optical internet packet switching," *IEEE Commun. Mag.*, vol. 38, pp. 116–122, Sept. 2000.

[5] Lynx Photonic Networks [Online]. Available: http://www.lynx-networks.com/prod/

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

[7] N. Wada, H. Harai, and F. Kubota, "40 Gbit/s interface, optical code based photonic packet switch prototype," in *Proc. Optical Fiber Communication Conf. (OFC 2003)*, Altanta, GA, Mar. 2003, pp. 216–217.

[8] N. Wada, H. Harai, W. Chujo, and F. Kubota, "Multi-hop, 40 Gbit/s, variable length photonic packet routing based on multi-wavelength label switching, waveband routing, and label swapping," in *Proc. Optical Fiber Communication Conf. (OFC 2002)*, Anaheim, CA, Mar. 2002, pp. 216–217.

[9] C. Guillemot *et al.*, "Transparent optical packet switching: The European ACTS KEOPS project approach," *J. Lightwave Technol.*, vol. 16, pp. 2117–2134, Dec. 1998.

[10] J. M. H. Elmirghani and H. T. Mouftah, "All-optical wavelength conversion: Technologies and applications in DWDM networks," *IEEE Commun. Mag.*, vol. 38, pp. 86–92, Mar. 2000.

[11] I. Chlamtac *et al.*, "CORD: Contention resolution by delay lines," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1014–1029, June 1996.

[12] I. Chlamtac, A. Fumagalli, and C.-J. Suh, "Multibuffer delay line architectures for efficient contention resolution in optical switching nodes," *IEEE Trans. Commun.*, vol. 48, pp. 2089–2098, Dec. 2000.

[13] F. K. H. Harai, N. Wada, and W. Chujo, "Contention resolution using multi-stage fiber delay line buffer in a photonic packet switch," in *Proc. IEEE Int. Conf. Communications (ICC 2002)*, New York, May 2002, pp. 2843–2847.

[14] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, "A unified study of contention-resolution schemes in optical packet-switched networks," *J. Lightwave Technol.*, vol. 21, pp. 672–683, Mar. 2003.

[15] A. G. Fayoumi and A. P. Jayasumana, "Effects of optical buffering on the performance of manhattan street networks," *Photonic Network Communications*, vol. 3, no. 1–2, pp. 161–171, Jan. 2001.

[16] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *J. Lightwave Technol.*, vol. 16, pp. 2081–2094, Dec. 1998.

[17] S. H. G. Chan and H. Kobayashi, "Packet scheduling algorithms and performance of a buffered shufflenet with deflection routing," *J. Lightwave Technol.*, vol. 18, pp. 490–501, Dec. 2000.

[18] A. Bononi, G. A. Castanon, and O. K. Tonguz, "Analysis of hot-potato optical networks with wavelength conversion," *J. Lightwave Technol.*, vol. 17, pp. 525–534, Apr. 1999.

[19] F. Forghieri, A. Bononi, and P. R. Prucnal, "Analysis and comparison of hot-potato and single-buffer deflection routing in very high bit rate optical mesh networks," *IEEE Trans. Commun.*, vol. 43, pp. 88–98, Jan. 1995.

[20] A. S. Acampora and I. A. Shah, "Multihop lightwave networks: A comparison of store-and-forward and hot-potato routing," *IEEE Trans. Commun.*, vol. 40, pp. 1082–1090, June 1992.

[21] A. G. Greenberg and B. Hajek, "Deflection routing in hypercube networks," *IEEE Trans. Commun.*, vol. 40, pp. 1070–1081, June 1992.

[22] T. Chich, J. Cohen, and P. Fraigniaud, "Unslotted deflection routing: A practical and efficient protocol for multihop optical networks," *IEEE/ACM Trans. Networking*, vol. 9, pp. 47–59, Feb. 2001.

[23] S. Yao, B. Mukherjee, S. J. Yoo, and S. Dixit, "All-optical packet-switched networks: A study of contention-resolution schemes in an irregular mesh network with variable-sized packets," in *Proc. SPIE Opticomm 2000*, Dallas, TX, Oct. 2000, pp. 235–246.

[24] G. Castanon, L. Tancevski, and L. Tamil, "Routing in all-optical packet switched irregular mesh networks," in *Proc. IEEE GLOBECOM 1999*, vol. 1B, Rio de Janeiro, Brazil, Dec. 1999, pp. 1017–1022.

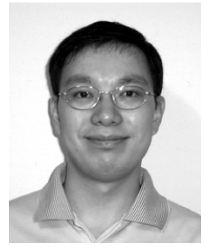
[25] J. P. Jue, "An algorithm for loopless deflection in photonic packet-switched networks," in *Proc. IEEE Int. Conf. Communications (ICC 2002)*, vol. 5, New York, May 2002, pp. 2776–2780.

[26] T.-S. Wang and S. Dixit, "Deflection routing protocol for burst switching WDM mesh networks," in *Proc. SPIE, Terabit Optical Networking: Architecture, Control, and Management Issues*, vol. 4213, Boston, MA, Nov. 2000, pp. 242–252.

[27] B. E. Carpenter and K. Nichols, "Differentiated services in the Internet," in *Proc. IEEE 2002*, vol. 90, Sept. 2002, pp. 1479–1494.
[28] F. Callegati, G. Corazza, and C. Raffaelli, "Exploitation of DWDM for optical packet switching with quality of service guarantees," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 190–201, Jan. 2002.
[29] L. Berger. (2003, Jan.) Generalized multi-protocol label switching (GMPLS) signaling functional description. IETF RFC 3471 [Online]. Available: http://www.ietf.org/rfc/rfc3471.txt

Tao Zhang (S'04) received the B.S. and M.S. degrees in electrical engineering from Wuhan University of Science and Technology, Wuhan, Hubei, China, in 1992 and 1998, respectively. She is currently working toward the Ph.D. degree in computer science at the University of Texas at Dallas.

Her research interests include wavelength-division-multiplexing optical networks and wireless networks, focusing on the design and analysis of network architectures and protocols.



Kejie Lu (S'01–M'04) received the B.S. and M.S. degrees in telecommunications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical engineering from the University of Texas at Dallas in 2003.

He is currently a Postdoctoral Research Associate at the Department of Electrical and Computer Engineering, University of Florida, Gainesville. His research interests include architecture and protocols design in wavelength-division-multiplexing (WDM) optical networks and wireless networks, performance analysis, and wireless communications systems.

Jason P. Jue (M'99–SM'04) received the B.S. degree in electrical engineering 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 computer engineering from the University of California, Davis, in 1999.

He is an Associate Professor in the Department of Computer Science at the University of Texas at Dallas. His research interests include optical networks, network control and management, and network survivability.