An Algorithm for Loopless Deflection in Photonic Packet-Switched Networks

Jason P. Jue
Center for Advanced Telecommunications Systems and Services
The University of Texas at Dallas
Richardson, TX 75083-0688
jjue@utdallas.edu

***Abstract*– In this paper, we present a label-based approach for implementing deflection in a photonic packet network, and we introduce an algorithm for determining deflection options in a manner which eliminates looping. A general analytical model is developed to evaluate packet loss probabilities in networks with deflection. The analysis may be applied to a wide class of deflection schemes and may be applied to networks with any arbitrary topology. The analysis is verified through simulation.**

I. Introduction

As telecommunication networks gradually evolve towards data-centric architectures, packet-based networks will be required to provide increasingly stringent quality of service requirements in order to support a growing number of high-speed multimedia applications. While electronic packet-switching technology is capable of providing differentiated services based on various priority and buffering schemes, the increasing transmission rates and high wavelength densities over each fiber will eventually make it prohibitively expensive to process each packet electronically at each node. Emerging all-optical switching technologies will enable packets to traverse nodes transparently without conversion to electronics. In particular, photonic packet switches have the potential to offer the flexibility and statistical multiplexing benefits of existing packet-switched networks, while also eliminating the cost of electronic conversion and processing at each node.

A significant issue in photonic packet switching is contention resolution. Contention occurs when two or more packets contend for the same output port at the same time. Typically, contention in traditional electronic packet-switched networks is handled through buffering; however, in the optical domain, it is more difficult to implement buffers, since there is no optical equivalent of random-access memory. Instead, optical buffering is achieved through the use of fiber delay lines [1]. Note that, in any optical buffer architecture, the size of the buffers is severely limited, not only by signal quality concerns, but also by physical space limitations. To delay a single packet for 5 μs requires over a kilometer of fiber. Because of this size limitation of optical buffers, a node may be unable to effectively handle high load or bursty traffic conditions.

Another approach to resolving contention is to route the contending packets to an output port other than the intended output port. This approach is referred to as deflection routing or hot-potato routing [2], [3], [4]. While deflection routing is generally not favored in electronic packet-switched networks due to potential looping and out-of-sequence delivery of packets, it may be necessary to implement deflection in photonic packet-switched networks, where buffer capacity is limited, in order to maintain a low level of packet losses. For deflection to be practical in photonic packet networks, methods for overcoming the limitations of deflection must be developed and investigated.

Deflection has been studied in a number of previous works. In [2], hot-potato routing is compared to store-and-forward routing in a ShuffleNet topology. [3] and [4] compare hot-potato and deflection routing in ShuffleNet and Manhattan street network topologies. In [5], deflection is studied in an unslotted packet network with a Manhattan Street Network topology, and a heuristic for scheduling packets to minimize contention is presented. It is shown that the heuristic improves the performance of unslotted networks almost to the level of slotted networks. Since both the ShuffleNet and Manhattan Street Network are two-connected (each node has an outgoing degree of two), the choice of the deflection output port is obvious. When the nodal degree is greater than two, a method must be developed to select the alternate outgoing link when a deflection occurs. In [6] and [7], deflection routing is studied in irregular mesh networks. Rather than choosing the deflection output port arbitrarily, priorities are assigned to each output port, and the ports are chosen in the prioritized order.

When deflection is implemented, a potential problem that may arise is the introduction of routing loops. If no action is taken to prevent loops, then a packet may return to nodes which it has already visited and may remain in the network for an indefinite amount of time. The looping of packets contributes to increased delays and degraded signal quality for the looping packets, as well as increased load for the entire network. Standard approaches for eliminating looping, such as maintaining a hop counter for each packet, can lead to increased complexity when processing packet headers. An alternative approach to resolving routing loops is to define the deflection alternatives at each node in a manner which eliminates all possibility of routing loops. In [7], deflection is studied together with optical buffering in irregular mesh networks with variable-length packets. The nodes at which deflection can occur, as well as the options for the deflection port, are limited in such a way as to prevent routing loops in the given network; however, a general methodology for selecting deflection options to avoiding looping in any arbitrary network is not given.

While analytical models have been developed to evaluate deflection in regular-topology networks [2], [3], [4], no previous work has been done to develop analytical models for evaluating deflection in general mesh-topology networks.

In this work, we investigate approaches for implementing deflection in a manner which eliminates looping, and we present an analytical model for evaluating deflection schemes in arbitrary mesh networks. Section II describes the basic network architecture and the deflection algorithms. Section III presents the analytical model for evaluating packet losses. Section IV provides numerical results for specific network topologies, and Section V concludes the paper.

II. Deflection Routing Algorithms

In this work, we will assume that deflection is implemented within a label-switched environment. Each node maintains a label database with a number of label entries. Each label entry indicates, for a given input port and a given label, the corresponding output port and outgoing label. Deflection options are defined by adding additional output-port/label pairs to each entry in the label database. When a packet arrives to the node, the corresponding entry in the label database is referenced, the packet's label is updated to the new outgoing label, and the packet is sent to the appropriate output port. If the primary output port is occupied, then the packet will be deflected to the alternate output port after updating the packet's label to the alternate outgoing label.

In general, labels may be defined on a per-destination basis, on a source/destination pair basis, or on a per-flow basis. For simplicity, we will 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; however, destination-based labeling also reduces the routing flexibility and traffic engineering options in the network.

Figure 1 illustrates the label-switched paths for a given destination node 8, and Fig. 2 shows the corresponding label entry at node 2. Note that, in destination-based labeling, the label of a packet will remain the same throughout the network. Furthermore, the primary label-switched 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 which avoids routing loops.

The process of defining the label entries at each node can be divided into two sub-problems. The first problem is to determine the primary outgoing link for each destination at each node. In this paper, we assume that the link which 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 which balances the load in the network. Such approaches are beyond the scope of this paper.

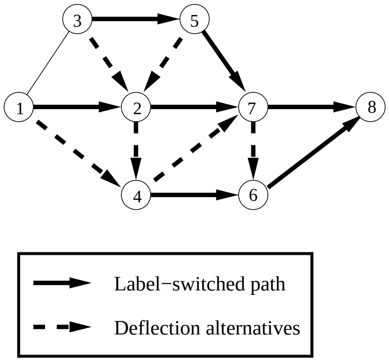

Fig. 1. Label switched paths and deflection alternatives for destination node 8.

input port	label	output port	label	deflect port	label
any	8	7	8	4	8
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•

Node 2

Fig. 2. Label entry at node 2 for destination node 8.

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. The deflection alternatives at each node must be defined in a way which eliminates the possibility of routing loops. The deflection-finding problem can be formulated in graph theoretic terms. Given a graph $G(V, E)$, and a directed spanning tree, T_v, rooted at vertex v, and with edges, E_{T_v}, directed towards the root, the problem is to find a set of edges $E_{D_v} \subset E$ such that the directed graph $R_v = (V, E_{T_v} \cup E_{D_v})$ is acyclic. The following *loopless-deflection* algorithm is proposed to find a feasible set of deflection edges. We define $\delta(v)$ to be the nodal degree of a vertex v, and $dist(u, v)$ to be the hop distance from node u to node v.

```
LOOPLESS-DEFLECTION:
Given :
    A graph G=(V,E).
    $|V|$ spanning trees, $T_v = (V_{T_v}, E_{T_v})$, each with a unique
    root vertex, $v \in V$.
Find :
    $E_{D_v}$, the set of deflection edges.
    Directed routing graphs, $R_v = (V, E_{R_v}), \forall v \in V$.
Step 0 :
    Set $V^* = V$.
    Set $E_{D_v} = \{\emptyset\}, \forall v \in V$.
Step 1 :
    Select a vertex $v \in V^*$.
    Set $E^* = \{E - E_{T_v}\}$.
    Let $d_v$ be the depth of tree $T_v$.
    Let $S_i, i \in \{1, 2, \ldots, d_v\}$ be the set of vertices which are distance $i$ from the root node $v$.
    Set $k = d_v$.
Step 2 :
    Select vertex $u \in S_k$ such that $\delta(u) = min_{u' \in S_k} \delta(u')$.
Step 3 :
    Select a directed edge, $e(u, w)$, such that $dist(w, v) = min_{w' : e(u,w') \in E^*} dist(w', v)$, if such an edge exists.
    Set $E_{D_v} = E_{D_v} \cup e(u, w)$.
Step 4 :
    Remove all edges $e(i, u), \forall i$ and $e(u, j), \forall j$ from $E^*$.
    Remove node $u$ from $S_k$. If $S_k = \{\not{\phi}\}$, then $k = k - 1$.
    If $k \neq 0$, then go to Step 2, otherwise, go to Step 5.
Step 5 :
    Set $R_v = (V, E_{T_v} \cup E_{D_v})$.
    Remove node $v$ from $V^*$.
    If $V^* \neq \{\not{\phi}\}$, then go to Step 1, otherwise, stop.
```

The loopless-deflection algorithm selects nodes one at a time, and attempts to find a deflection output port at the selected node. By selecting leaf nodes which are furthest from the root, and by deleting the node after its deflection output port has been selected, the algorithm ensures that no deflections are made to nodes which are further from the destination than the selected node, and that packets, upon departing from the selected node, can never return to that node. In Step 3, the algorithm attempts to choose the deflection edge which results in the shortest path to the destination. The algorithm can be further customized in Step 3 by choosing the deflection edges based on estimated link loads, or by allowing multiple deflection options for each destination. If multiple deflection options are allowed, the options may be prioritized based on distance or load considerations.

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 which are closer to the destination are less likely to have deflection alternatives than nodes which are further from the destination. By restricting deflection at these nodes, the packet losses may increase.

For comparison, we also consider a deflection-finding algorithm which allows looping. The algorithm starts with the graph $G(V, E)$ and the directed tree, T_v, rooted at destination v. Each node is selected one at a time in any order. For each node, the algorithm selects the deflection edge which results in the shortest physical-distance path to destination v. Edges in the tree T_v are used for primary-path routing, therefore these edges are not considered when selecting the deflection edge. Unlike the loopless-deflection algorithm, nodes and edges are not deleted once a deflection edge has been chosen; thus, routing loops are possible. We will refer to this algorithm as the *shortest-path deflection* algorithm.

III. Analytical Model

In this section, we develop an analytical model for evaluating the packet loss probabilities of the proposed deflection scheme. The model is general, and can be applied to any irregular mesh topology. The analysis can also be used to evaluate any deflection scheme in which the deflection alternatives at each node are ordered and pre-defined.

The model assumes that the network is asynchronous, and that packet have a fixed length of L seconds. Packets arrive to the network according to a Poisson process with rate λ^{sd} packets per second for source-destination pair sd.

Each link in the network is modeled as an M/D/1/1 queue with no buffers. The arrival rate, λ_{ij} of packets to a link l_{ij} is determined by aggregating the packet arrivals from all source-destination pairs which route packets over the link. The arrival rates will also depend on the probability of contention, P_{ij}, on each link in the network, and the deflection policy.

To find the contention probabilities, we examine the time between packet departure instants and calculate the fraction of time that a link is busy. The expected cycle time, T, between two consecutive packet departures is found by adding the expected time until the next packet arrival to the expected packet transmission time:

$$E[T] = \frac{1}{\lambda_{ij}} + L. \tag{1}$$

The probability that a packet arriving to link ij encounters contention is equal to the probability that the link is busy:

$$P_{ij} = \frac{L}{E[T]}. \tag{2}$$

Based on the link blocking probabilities, we can find the offered load, λ_{ij}, on each link l_{ij}. We define r'_{sd} as the primary route from source s to destination d without deflection.

The load on a link is the sum of the loads contributed by each source-destination pair which routes packets over the link:

$$\lambda_{ij} = \sum_{s,d} \lambda_{ij}^{sd}, \tag{3}$$

where λ_{ij}^{sd} is the rate at which packets sourced at s and destined for d arrive to link l_{ij}.

The load placed on a link l_{ij} by traffic going from source s to destination d depends on whether link l_{ij} is on a primary path to destination d, or whether the link is a deflection link to destination d. If link l_{ij} is on a primary path to d, and node i is the source node s, the load applied to link l_{ij} by sd traffic is simply λ^{sd}. If link l_{ij} is on a primary path to d, but node i is not the source node, then the applied load will be the load offered by previous-hop links, l_{hi}, which are sending packets to d through link l_{ij}. The load is reduced by the amount of contention on the links l_{hi}. If link l_{ij} is the deflection link for a primary link l_{sk} at the source node s, then packets will arrive to link l_{ij} if the packets experienced contention on link l_{sk}; thus, the load on l_{ij} will be $\lambda^{sd} P_{sk}$. If link l_{ij} is a deflection link at an intermediate node, and link l_{ik} is the corresponding primary link to destination d from node i, then the applied load on link l_{ij} will be the incoming load from all previous-hop links which are sending traffic to d through i, and which encounter contention on link l_{ik}. Thus, we have:

$$\begin{array}{lll}
\lambda_{ij}^{sd} & & \\
= & \lambda^{sd} & \text{if } l_{ij} \in r'_{id}, i = s \quad (4) \\
= & \sum_h \lambda_{hi}^{sd}(1 - P_{hi}) & \text{if } l_{ij} \in r'_{sd}, i \neq s \quad (5) \\
= & \sum_h \lambda_{hi}^{sd}(1 - P_{hi}) & \text{if } l_{ij} \notin r'_{sd}, i = s, l_{ij} \in r'_{id} \quad (6) \\
= & \lambda^{sd} P_{sk} & \text{if } l_{ij} \notin r'_{sd}, i = s, l_{sk} \in r'_{id} \quad (7) \\
= & \sum_h \lambda_{hi}^{sd}(1 - P_{hi}) P_{ik} & \\
 & & \text{if } l_{ij} \notin r'_{sd}, i \neq s, l_{ik} \in r'_{id}. \quad (8)
\end{array}$$

Once the above equations are solved to find P_{ij}, the packet loss probability can be found for packets traveling from source s to destination d. A packet will be lost if, on a given hop, both the primary and deflection links are blocked. Let node i be the primary next-hop node, and let j be the deflection next-hop node. The packet loss probability for packets travelling from source node s to destination node d is given by:

$$P_{loss}^{sd} = P_{si}P_{sj} + (1 - P_{si})P_{loss}^{id} + P_{si}(1 - P_{sj})P_{loss}^{jd}. \quad (9)$$

The packet loss probability for the entire network is found by calculating the weighted average of the packet losses for each source-destination pair:

$$P_{loss} = \sum_{s,d} P_{loss}^{sd} \cdot \left(\frac{\lambda^{sd}}{\sum_{s',d'} \lambda^{s'd'}} \right). \quad (10)$$

Although the analysis assumes that there is at most one deflection alternative for each destination at each node, the analysis can be extended in a straightforward manner to accommodate the case in which there is an arbitrary number of ordered deflection alternatives for each destination at each node.

The analysis may also be applied to deflection schemes in which looping is present; however, the analysis must be modified slightly in order to avoid infinite path lengths. When calculating P_{ij} and λ_{ij}^{sd}, 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.

IV. Numerical Results

In this section, we evaluate the deflection algorithms in the 15-node network topology illustrated in Fig. 3, and in a bidirectional Manhattan street network topology in which the nodal degree at each node is equal to 4. Packets are fixed in length, consisting of 10,000 bits each, and the transmission rate is assumed to be 10 Gb/s. Packets arrive to the network according to a Poisson process, and traffic is uniformly distributed over all source-destination pairs.

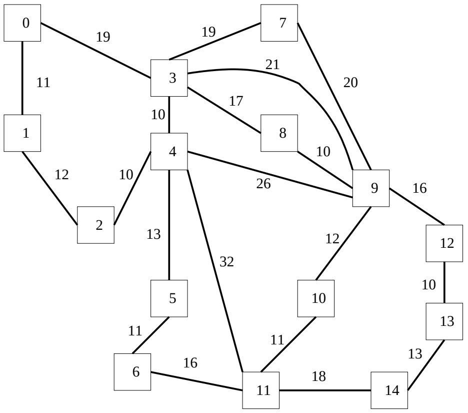

Fig. 3. 15-node network topology.

Figure 4 shows the packet loss probability as a function of load for the 15-node network. We observe that the loopless deflection scheme provides a slight improvement in performance over the case with no deflection, while the deflection scheme without any looping restrictions offers significantly lower packet loss probabilities than either of the other two cases. 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 to the shortest-path deflection scheme.

The packet loss probability for the bidirectional Manhattan street network is shown in Fig. 5. In this network, the higher nodal degree enables a larger number of nodes to have deflection options in the loopless deflection case. Consequently, in a network with a higher nodal degree, the loopless deflection case will provide greater benefits over the case in which no deflections take place.

In Fig. 6, we plot the packet loss probability for higher network loads in the 15-node network. We observe that, for high loads, the shortest-path deflection scheme has the highest packet loss rate, while the scheme with no deflection has

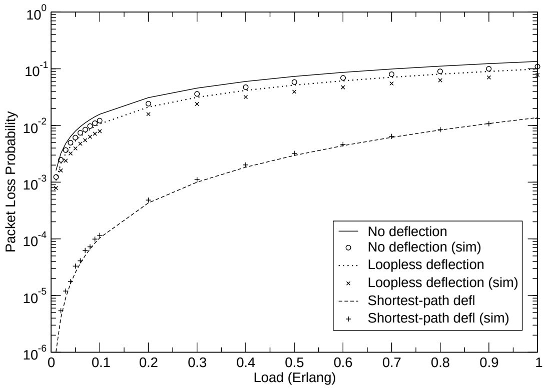

Fig. 4. Packet loss for 15-node network topology.

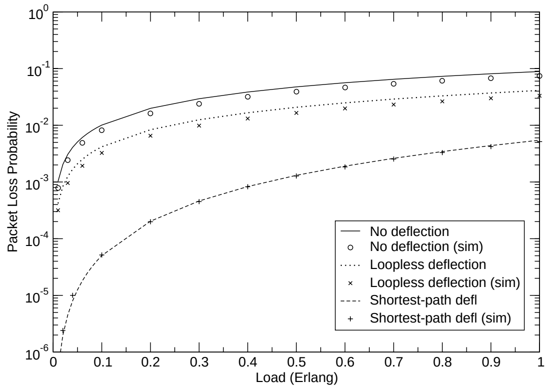

Fig. 5. Packet loss for Manhattan street network topology.

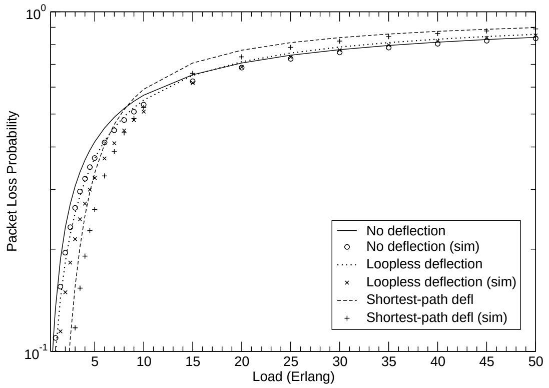

Fig. 6. Packet loss for 15-node network topology under high loads.

the lowest packet loss rate. Under higher loads, the number of contentions will increase, resulting in a greater number of deflections in networks. The increased deflections will increase the effective network load and cause higher blocking probabilities. The same general results are found for the Manhattan street network, with the cross-over points occuring at higher loads than in the 15-node network. Results for the Manhattan street network are not shown due to space constraints.

Figure 7 shows the average hop distance traversed by each packet as measured by simulation in the 15-node network. For the shortest-path deflection scheme, as the load increases, more packets are being deflected, leading to higher average hop distances; however, packets are still reaching their destinations, indicating that deflection is successfully resolving contention. As the load increases further, the average hop distance begins to drop. The drop indicates that, while packets are still being deflected, the deflected packets are not as likely to reach their destination. At this point, deflection starts to become detrimental rather than beneficial. For both the loopless deflection scheme and the scheme with no deflection, the average hop distance decreases as load increases. This effect indicates that there is some degree of unfairness in the network, since a packet which must travel a greater number of hops to its destination has a higher chance of being dropped.

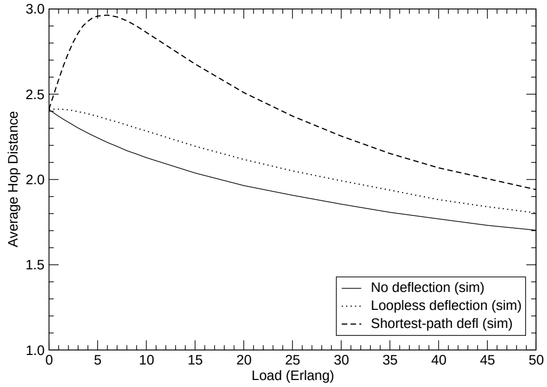

Fig. 7. Average hop count for 15-node network topology.

V. Conclusion

In this paper, we investigated the use of deflection routing to resolve contentions in a label-switched photonic packet network, and we introduced an algorithm for determining the deflection alternatives in a manner which eliminates the possiblity of routing loops. A general analytical model, which can be applied to a wide range of deflection schemes and on arbitrary network topologies, was developed to evaluate the loopless deflection scheme.

It was shown that, under low loads, the loopless deflection scheme provides modest gains over the scheme with no deflection, and that the gains increase for networks which have a higher average nodal degree. An unrestricted deflection scheme, in which loops are allowed, provides significantly lower packet losses at low loads, but results in higher average packet hop distances. At high loads, the deflection schemes result in higher packet losses. Overall, the loopless deflection scheme offers a reasonable trade-off between packet losses and average hop distance.

References

[1] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, et al., "CORD: Contention Resolution by Delay Lines," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 1014-1029, June 1996.

[2] A. S. Acampora and I. A. Shah, "Multihop Lightwave Networks: A Comparison of Store-and-Forward and Hot-Potato Routing," *IEEE Transactions on Communications*, vol. 40, no. 6, pp. 1082-1090, June 1992.

[3] 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 Transactions on Communications*, vol. 43, no. 1, pp. 88-98, Jan. 1995.

[4] A. Bononi, G. A. Castanon, and O. K. Tonguz, "Analysis of Hot-Potato Optical Networks with Wavelength Conversion," *IEEE Journal of Lightwave Technology*, vol. 17, no. 4, pp. 525-534, April 1999.

[5] T. Chich, J. Cohen, and P. Fraigniaud, "Unslotted Deflection Routing: A Practical and Efficient Protocol for Multihop Optical Networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 47-59, Feb. 2001.

[6] G. Castanon, L. Tancevski, and L. Tamil, "Routing in All-Optical Packet Switched Irregular Mesh Networks," *Proceedings, IEEE Globecom '99*, Rio de Janeiro, Brazil, pp. 1017-1022, Dec. 1999.

[7] S. Yao, B. Mukherjee, S.J.B. Yoo, and S. Dixit, "All-Optical Packet-Switched Networks: A Study of Contention Resolution Schemes in an Irregular Mesh Network with Variable-Sized Packets," *Proceedings, SPIE OptiComm 2000*, Dallas, TX, pp.235-246, Oct. 2000.