A Novel Fast Restoration Mechanism for Optical Burst Switched Networks [*]

Yufeng Xin[⋆] Jing Teng[‡] Gigi Karmous-Edwards[†] George N. Rouskas[‡] Daniel Stevenson[†]
⋆ Correspondending author, UMIACS, University of Maryland, College Park, MD, Email: `yxin@umiacs.umd.org`
† ANR, MCNC-RDI, RTP NC 27709
‡ Computer Science Department, NC State University, Raleigh NC 27609

Abstract

Survivability is a critical network design issue for optical networks since even a single failure for a short duration may result in huge data loss due to the large capacity of optical fibers. However, few studies have been done on this issue for optical burst switching (OBS) networks. In this paper, we extend our early work on the fast restoration technique for OBS networks and present a novel fast restoration mechanism based on distributed deflection routing. Compared to other survivability schemes, the proposed mechanism has the advantage of fast and low-overhead fault management process and demonstrates excellent burst blocking performance by balancing the deflected traffic load during the restoration process.

1 Introduction

Internet traffic is growing very fast and demanding more bandwidth and better bandwidth utilization. The bursty nature of Internet traffic has shifted recent research focus from optical circuit switching to optical packet switching (OPS) and optical burst switching (OBS) that feature fast service provisioning and efficient bandwidth usage [4]. OBS is usually viewed as a technique bridging optical circuit switching and optical packet switching. Since the implementation of the techniques for buffering and header processing in optical domains is still premature, OBS protocols were designed to avoid optical buffering and perform fast setup of the end-to-end data lightpath using out-of-band header processing. One representative protocol for OBS networks is the Just-In-Time (JIT) signaling protocol. Supporting variable payload size and designed for unslotted OBS switches, JIT requires no synchronization at the switches [6]. The bufferless mesh optical network under the control of the JIT signaling protocol is referred as the *JIT network* in this paper.

While the majority of the previous studies have been focused on resource reservation and scheduling, burstification, and performance analysis [5], few studies have been done on the survivability issue for OBS networks. Network survivability is a critical design issue for all types of optical networks since even a single failure such as a fiber cut or an interface card malfunction for a short duration may result in huge data loss due to the large capacity of optical channels. Unfortunately, the probability of such kind of failures is not low. FCC statistics show that metro networks annually experience 13 cuts for every 1000 miles of fiber, and long-haul networks experience 3 cuts for 1000 miles fiber. Even the lower rate for long-haul implies a cable cut every 4 days on average in a network with 30000 route-miles of fiber [1].

The techniques used for network survivability can be broadly classified into two categories: *preplanned protection* and *dynamic restoration* [2]. Compared to protection, restoration has more efficient resource utilization, but typically takes more time. Another advantage of dynamic restoration is the better scalability in term of the fault management overheads, as backup connections for the disrupted services need only to be discovered and maintained after the network failure(s). However, dynamic restoration requires long restoration time and does not guarantee the reestablishment of the disrupted services, since the backup resources may not be available at the time that a failure happens. Therefore, protection is often used for the premier long-term service connections in circuit-switched or connection-oriented data networks. Existing restoration mechanisms also focus on the recovery of disrupted existing connections in the circuit-switched or connection-oriented data networks. As to the connectionless networks, such as IP networks, the restoration is achieved by dynamic routing through global routing table update upon a network failure, where all new incoming connections whose initial routes traverse the failure will be dropped before this slow restoration process (up to minutes in the worst case) is completed.

For the JIT-based OBS networks we are looking at, service routes are discovered at every involved node by the hop-by-hop forwarding table lookup. It is almost impossi-

---

[*] The work was done when the author was with MCNC-RDI

BROADNETS 2004: Workshop on Optical Burst Switching
October 25-29, 2004. Copyright 2004 ICST.

ble for a node to maintain the path (or path segment) information and pre-reserve spare resources for a connection requirement. Therefore, fast restoration is a better choice for the fault management of OBS networks.

[9] presented a brief operation and maintenance framework for OBS networks. [8] studied a simple deflection routing scheme to achieve fault tolerance upon link failures for OBS networks. In [7], a 1+1 protection scheme was studied for OBS networks. However, this study was only concerned with the special long duration OBS sessions whose primary and backup paths are decided prior to actual burst transmission. Therefore only premium traffic, comprising a small fraction of the total load in an OBS network, would be afforded this type of protection.

The dominant traffic in a JIT network is short-term optical bursts that are set up by dynamic route discovery using the hop-by-hop forwarding table lookup at the intermediate nodes. The need for the minimization of restoration time in JIT networks is emphasized by the following facts: (1) restoration of the existing short-term connections may be meaningless because their duration may be shorter than the restoration time; (2) restoration through the global forwarding table update, although providing an optimal routing solution after the network failure, is too slow to prevent the heavy data loss under heavy traffic conditions. Fast restoration mechanisms must be in place so that the data loss can be minimized before the global forwarding table update is complete.

In this paper, we focus on the fast restoration mechanism for JIT-based OBS networks where short bursts are dominant. The proposed deflection-based restoration schemes are designed with the following objectives: (1) the restoration process is fast such that interrupted services can be restored in short time; (2) the overall burst loss during the failure and restoration is as low as possible with the limited network capacity; (3) the fault management overhead is low in terms of extra control message exchange and processing. We also show that optimization of the overall network blocking performance is achieved by balancing the traffic load during the restoration process in order to alleviate deteriorated network congestion due to the network failure. This work is an extension to our early work on this topic [10].

The paper is organized as follows. A fault management framework for JIT networks is presented in Section 2. A fast restoration mechanism based on deflection routing is discussed in Section 3. Results from a comprehensive simulation study are presented in Section 4. Section 5 concludes the paper.

2 Fault management framework for JIT networks

In JIT networks, the basic routing mechanism is much like that of IP networks, in which every OBS node maintains a local forwarding table. The entries in the forwarding table contain the next hop information for bursts per destination and per FEC (Forward Equivalent Class). OBS nodes forward the incoming burst control packets and set up the connections by looking up the next-hop information in their forwarding tables. We simply use *burst forwarding* or *burst routing* to represent this connection setup process.

Routing in JIT networks relies on a link state protocol similar to OSPF. Each node collects link state information for all its adjacent nodes periodically and reports the information to the routing entity in the network. The routing entity then calculates a new set of routes between every pair of OBS nodes and updates the forwarding tables in the OBS nodes. In the centralized case, there is a *Routing Decision Node* (RDN) in the network that performs the routing functionality for the entire network. In the case of distributed routing, there is no such RDN. All nodes flood to and receive from every other node with the link state information. Each node will compute the routes to other nodes and update the forwarding table locally. Therefore, all the OBS nodes in the network comprise the routing entity.

However, in this paper, whether the routing is centralized or distributed is irrelevant to our proposed fast restoration mechanism. For the purpose of better illustration, we assume centralized routing in this paper. Burst forwarding decisions are made at OBS nodes locally, based on the forwarding table given by the RDN. All OBS nodes report their link state information to the RDN periodically or by interrupt in abnormal cases. In addition, nodes may also exchange fault information with their adjacent nodes (neighbors) within the control plane.

The routing algorithm implemented at the RDN is based on the 2-shortest-path algorithm to support alternative routing. For every pair of OBS nodes in the network, two disjoint routes with the shortest overall length are computed, one as the primary route and another as the alternative route. Therefore, the resulting forwarding table at an OBS node contains 2 next-hop entries per destination or FEC. The route for a particular burst is discovered based on a hop-by-hop paradigm. For any end-to-end burst connection, a node along its route is only aware of the primary and alternative next hops for this burst given by the RDN.

We only consider the single link failure scenario in this paper. We also assume the control plane is independent of the burst data plane and is 100% reliable. The fault management for OBS networks contains three steps: fault detection and localization, fault notification, and service restoration.

Upon detection of a link failure, a fault notification message is sent to the RDN for routing re-computation and then the RDN advises all the nodes to update their forwarding tables accordingly. This is actually a default service restoration mechanism as the new routes will work around the faulty link. However, this global forwarding table update process could be very slow (from several seconds up to a few minutes) according to the experience from current IP networks. The slow restoration process may result in an intolerably large amount of burst loss because all bursts that are supposed to traverse the failed link will be discarded if no special action is taken during the update process. Therefore, we have to implement efficient fast restoration techniques complementary to the global forwarding table update to reduce the overall burst loss.

Since the JIT network is a loss network, a feasible restoration mechanism needs to make efficient use of spare network resources in order to minimize the burst blocking probability. Our study shows that two types of burst blocking contribute to the overall burst blocking performance: (1) *restoration blocking*, which is the burst blocking during the fault detection and notification periods; (2) increased *congestion blocking* arising from the diverted traffic and the reduced network capacity that results from the failure. However, there may exist complex trade-offs between these two types of blocking. A fast restoration scheme with shorter fault notification time (thus smaller restoration blocking) may incur larger congestion blocking, and *vice versa*. This phenomenon will be showed latter when we present the restoration schemes.

In the following section, we will present two basic deflection-based fast restoration schemes that can be integrated into the JumpStart JIT signaling protocol [3], with light management overhead and good blocking performance. Furthermore, by defining a *distribution ratio*, α, we generalize these two schemes to a general restoration mechanism, α *distributed deflection restoration*.

3 Fast restoration techniques for OBS networks

Figure 1 depicts an example OBS network with 8 OBS nodes and a RDN. We assume node *I* is the ingress node for a certain burst flow and node *E* is the respective egress node. The links represented by solid lines are the physical links connecting the OBS nodes (the links to *RDN* are represented by the dashed lines). The primary route from *I* to *E* is $I \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow E$. Assuming the link between its *headend* node 2 and *tailend* node 3 is broken, links in the same line style represents an alternative route decided by a restoration scheme. As mentioned before, there is at least one alternative next hop for every entry in the forwarding table. This information enables the JIT network to set up alternative routes to work around the faulty link.

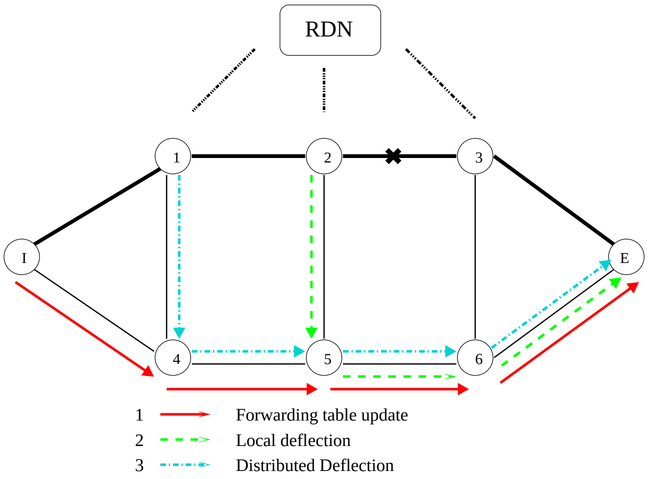

Figure 1. OBS fast restoration mechanisms

Before the faulty link is worked around, bursts will be lost in the faulty link. After the successful restoration, burst loss would still be higher than that under normal conditions due to the reduced network capacity. While the fault detection and processing time may normally be a constant, the fault notification time makes the major difference among different restoration schemes. Generally, we want it is preferable that the node that makes the re-routing decision be close to the faulty link so that the fault notification time is reduced. This also leads to light fault management overhead as only a small amount of fault notification message transmission is needed. In the list below, we present two fast restoration schemes based on deflection routing in which at most one-hop fault notification message transmission and processing are required. They are also illustrated in Fig 3. For the purpose of comparison, we first describe the default global routing update scheme.

- Scheme 0. **Global routing update**: When the headend 2 (or tailend 3) detects the link failure, it informs the RDN via the control plane. The RDN conducts the routing re-computation and updates the forwarding tables for all nodes, and new bursts will subsequently follow the new routes. For example, new bursts will follow the route $I \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow E$. This solution is optimal and the existing routing protocol can handle it well. However, global routing table updating is a slow process (in seconds or even minutes) due to the long round trip time for the signal transmission and processing between the OBS nodes and the routing entity. As a result, a large amount of bursts will be lost before the forwarding tables are updated.

- Scheme 1. **Local deflection**: This is similar to the traditional deflection routing usually seen in a congestion resolution scheme. When the headend 2 detects the link failure, it will automatically pick up the alternative next hop in the forwarding table for every new burst whose next hop on its primary route passes the faulty link. In the example, new bursts from I to E will follow the alternative route $I \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow E$. This would be the fastest restoration scheme since new bursts will be deflected to an alternative good link right after the link failure is detected locally. Therefore, it will incur the smallest restoration blocking. However, because all the affected bursts are deflected to one alternative path, this scheme would increase the congestion blocking.

- Scheme 2. **Distributed deflection**: This is a novel fast restoration scheme proposed in this paper. In this scheme, the headend 2 will also send a different fault notification message to all its adjacent nodes in addition to the one to the RDN. This fault notification message contains the destination information for all the primary routes passing the faulty link. After receiving this message, each of the adjacent nodes will pick up an alternative next hop for the affected bursts that are heading to the faulty link according to their primary route. In the example, bursts from I to E will take the new route $I \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow E$. Compared with the local deflection scheme, distributed deflection has the potential to make the re-routed traffic more distributed instead of being totally deflected to one alternative path. In this way, less congestion and therefore less burst loss may occur. However, this scheme requires extra one-hop fault notification. One possible problem is that, if the network traffic load is already very heavy, distributed deflection may have a negative impact as it may deteriorate the congestion condition all over the network.

  The actual algorithm is a combination of local deflection and adjacent deflection, i.e., the affected bursts are deflected locally until the adjacent nodes receive the fault notification. At that time the affected bursts will be deflected distributively.

Above analysis clearly shows that the last two restoration schemes will provide fast restoration as at most one-hop fault notification message transmission/processing are required and the alternative route is pre-computed before any failures. Furthermore, they only add a small amount of fault management overhead to the normal network operation (the alternative route information in both schemes and the one-hop fault notification message transmission in the distributed deflection).

One interesting observation from scheme 2 is that the capacity of the links between the headend (node 2) of the faulty link and its adjacent nodes (node 1) will not be utilized if all affected bursts are deflected at adjacent nodes. Therefore, we define a *distribution ratio*, α, to determine the portion of affected bursts that will be deflected at the adjacent nodes. That is, after the adjacent nodes receive the fault notification, α portion of affected bursts will be deflected distributively, and $(1-\alpha)$ portion of affected bursts will be forwarded to the headend node of the faulty link to be deflected locally. With a different value of $\alpha \in [0,1]$, we have a different variance of the distributed restoration scheme. When $\alpha = 0$, it is equivalent to scheme 1, local deflection based restoration. When $\alpha = 1$, it becomes scheme 2, the distributed deflection based restoration. We use α *distributed deflection* to denote the generalized distributed deflection mechanism. We also note that using α only introduces a tiny amount of management complexity in the adjacent nodes. We expect that there exists an optimal value of α that makes the affected bursts to be deflected in a most balanced way such that the minimum burst loss can be achieved.

4 Simulation Study

In this section, we present the results from a comprehensive simulation study to illustrate and compare the performance of the restoration schemes we discussed above. The simulation is conducted on an NSF network (14 nodes, 21 links) under the *JumpStart* JIT signaling protocol [3].

We consider two types of burst loss (probability) in this study. One is the burst loss (probability) for all the bursts in the network (since deflected bursts may have negative impacts on other bursts due to the increased congestion). Another is the loss (probability) of the affected bursts whose primary routes pass the faulty link. We will refer to the former as the overall burst loss (probability) and the latter as the local burst loss (probability). We are especially interested in the burst loss (probability) during the restoration process, the time period between the link failure and the global routing table update.

During the simulation, we let the most loaded link fail after the simulation runs enough time at the steady state. We also assume the RDN is the central node of the network and is always working. The updated new burst routes will replace the deflection routes after the RDN update messages arrive at every node.

We assume all the physical links in the network to be bi-directional and there are $W = 32$ wavelengths per link per direction. We also assume full wavelength conversion capacity at every node. Two shortest alternative routes are pre-computed for every pair of nodes with the shorter one as the primary route.

We assume a Poisson burst arrival process for every OBS node with an offered load of $\rho = \frac{\lambda}{\mu \times W}$, where λ and $\frac{1}{\mu}$ are the arrival rate and average burst duration time, respectively. The traffic is uniformly distributed to every other OBS node as the destination.

All the results are the mean burst loss probability (defined as the ratio of the number of lost bursts to the total offered number of bursts) during the link failure. They are obtained within the 95% confidence interval by using the batch mean method. The fault detection time is set to 15 ms and the RDN update time is fixed to 200 ms.

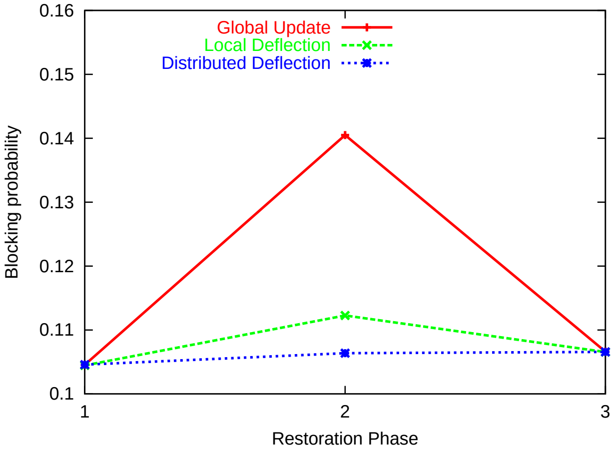

Figure 2. Overall burst loss probability vs. restoration phases ($\rho = 0.5$)

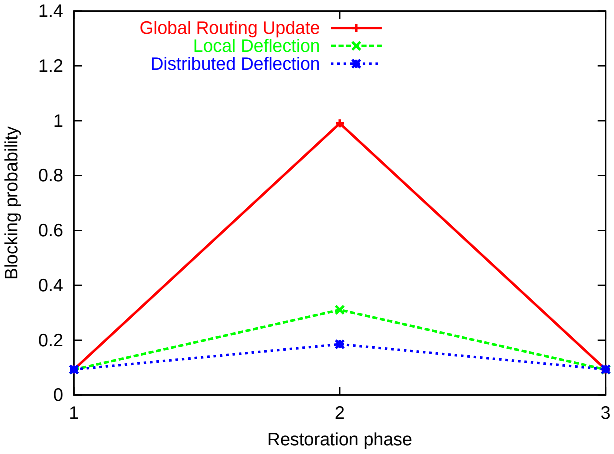

Figure 3. Local burst loss probability vs. restoration phases ($\rho = 0.5$)

Fig. 2 and Fig. 3 depict the overall and local burst loss probability (y-axis) when $\rho = 0.5$ during the three network operational phases around a link failure. In the x-axis, phase 1 represents the normal phase when the link failure has not occurred, phase 3 represents the phase after the global routing table update has been finished, and phase 2 represents the time period in between. We observe that the burst loss probability is very low for both phases 1 and 3, though it is actually a little bit higher in phase 3 due to the reduction of the network capacity. However, the loss probability could increase significantly in phase 2. Relying only on the forwarding table update would incur very high burst loss in this phase. However, the loss probability only increases moderately when the proposed fast restoration schemes are used in this phase. For the overall burst loss depicted in Fig. 2, among the three restoration schemes, the distributed deflection shows the best performance (almost no extra burst loss) followed by the local deflection. Global routing update incurs the highest burst loss. Specifically, the improvements from using the two fast restoration schemes over the global forwarding table update are 24.3% and 20.1%, respectively. The performance improvement in this phase is more obvious for the affected bursts whose primary routes pass the faulty link before the new forwarding table update. As depicted in Fig. 3, these bursts will be directly dropped with no fast restoration schemes in place (the burst loss probability is 100% for the global routing update scheme), but only 18.5% and 31% of the bursts are lost with distributed deflection and local deflection, respectively. We observe that the distributed deflection achieves the least burst loss and the local deflection also reduces the burst loss dramatically over the global routing update.

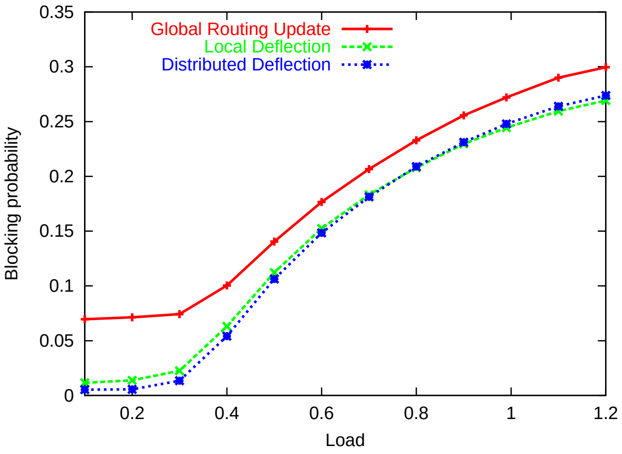

Figure 4. Overall burst loss probability with different schemes vs. offered load ρ

Fig. 4 and Fig. 5 illustrate the overall and local burst loss probability under different offered loads (different values of ρ) and different restoration schemes during the restoration period (phase 2). We observe that applying the proposed fast restoration schemes can dramatically reduce both overall and local burst loss. For example, at the load level of 0.1, the blocking probability of the global forwarding ta-

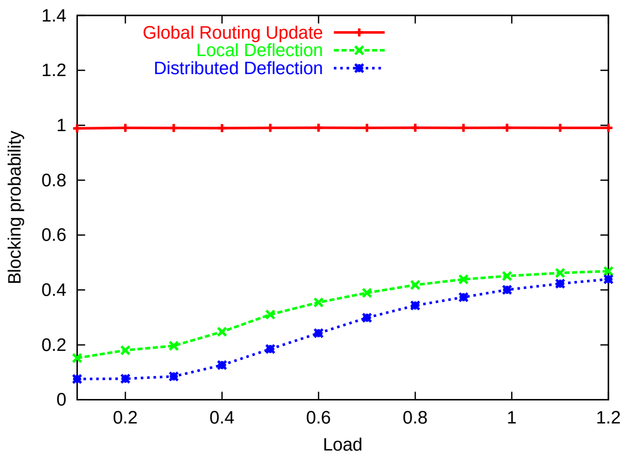

Figure 5. Local burst loss probability with different schemes vs. offered load ρ

ble update is about 13 times higher than that of the distributed deflection, and 6 times higher than the local deflection. However, the improvement with these fast restoration schemes tends to decrease as the load increases. For example, at the load of 1.2, the improvements of the burst blocking probability over the global routing update are only 8.6% and 10.1% for the distributed deflection and local deflection respectively. This is because the burst loss due to normal network congestion begins to dominate in the overall burst loss as the load increases. This can be clearly seen in Fig. 2, where we observe that the burst loss probability during the first period and last period is already 10.4% (due to normal network congestion) while the burst loss probability during the second period (phase 2) is just increased to 14.1% for the global routing update, and 10.6% and 11.2% for the distributed deflection and local deflection, respectively. This is also confirmed by Fig. 5 showing the local burst loss probability under different loads, in which we observe that the local burst loss probability increases slowly with the increment of load.

An interesting observation from Fig. 4 is that the distributed deflection achieves lower overall burst loss than the local deflection when the traffic load is relatively light ($\rho \leq 0.7$). But the local deflection starts to outperform when the traffic load is getting higher. However, Fig. 5 shows that the local deflection always incurs higher local burst loss, though the difference between these two schemes gets smaller as the traffic load increases. These observations actually suggest that distributed deflection may have a negative impact on the overall network blocking performance under heavy network traffic, although it always achieves smaller local burst loss.

To achieve minimal burst loss, we introduced a new con-

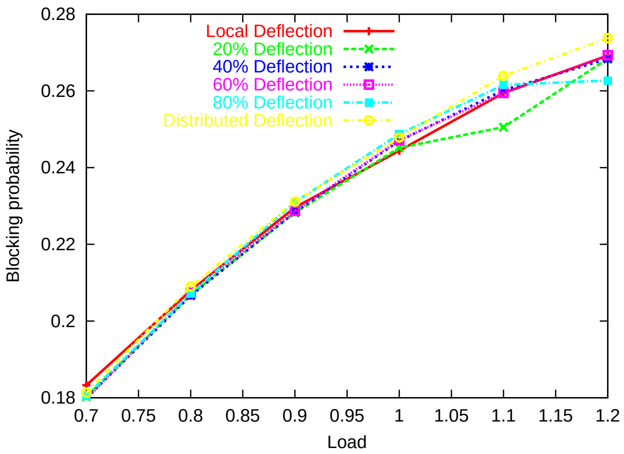

Figure 6. Overall burst loss probability with α Deflection vs. offered load ρ

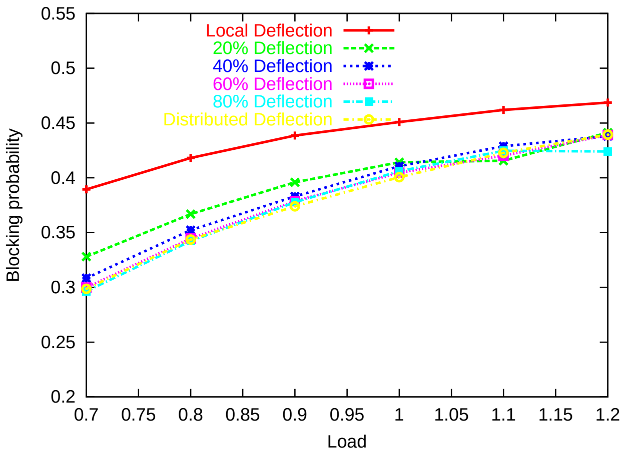

Figure 7. Local burst loss probability with α Deflection vs. offered load ρ

cept, α *distributed deflection* in last section. This mechanism helps the deflected traffic be more balanced during the restoration process. Fig. 6 and Fig. 7 illustrate the simulation results when α is given different values. We observe that the overall burst loss shown in Fig. 6 is almost the same for different values of α when the traffic is less than 1.0. However, when the traffic load continues to increase, there exists an optimal value of α, α_{opt} for different loads. For example, when $\rho = 1.0$, $\alpha_{opt} = 0$; when $\rho = 1.1$, $\alpha_{opt} = 0.2$; and when $\rho = 1.2.$, $\alpha_{opt} = 0.8$. Accordingly, similar observations can be made for the local burst loss from Fig. 7.

Generally, when the traffic load is small or moderate, the distributed deflection achieves the best burst blocking performance. When the traffic load becomes heavy, an optimal portion of bursts need to be locally deflected to achieve the best blocking performance.

5 Conclusion

In this paper, we present a novel fast restoration mechanism based on the distributed deflection routing for OBS networks. Through analyzing the burst loss performance during the restoration process, the resulting restoration schemes are designed with the advantages of (1) fast and low-overhead fault management process and (2) optimal overall burst blocking performance by balancing the deflected traffic during the restoration process. The comprehensive simulation study demonstrates the excellent performance of the presented restoration mechanism. It is also shown that there exists an optimum burst deflection distribution to achieve the minimum overall burst loss probability. How to decide this optimum deflection distribution in an adaptive way according to the network load would be an interesting topic that is worthy of further research.

References

[1] Wayne D. Grover. Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking. Prentice Hall, August 26, 2003.

[2] D. Sou and S. Subramaniam. Survivability in optical networks. *IEEE Network*, 14(6), Nov./Dec. 2000.

[3] I. Baldine, George N. Rouskas, Harry G. Perros and Dan Stevenson. JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks. *IEEE Communication Magazine*, 40(2), Feb. 2002.

[4] Y. Chen, C. Qiao and X. Yu. Optical burst switching: A New Area in Optical Networking Research. *IEEE Network Magazine*, 18(3), May/June 2004.

[5] T. Battistilli and H. Perros. An introduction to optical burst switching. *IEEE Communication Magazine*, 41(8), Aug. 2003.

[6] A. Halim Zaim, Ilia Baldine, Mark Cassada, George N. Rouskas, Harry G. Perros and Dan Stevenson. Fault Management with Fast Restoration for Optical Burst Switched Networks. *Optical Engineering*, 42(2), Feb. 2003.

[7] David Griffith and SuKyoung Lee. A 1+1 protection architecture for optical burst switched networks. *Journal of Selected Areas in Communications*, IEEE, 21(9), Nov. 2003.

[8] A. Sengupta, J. Qu and S. Bandyopadhyay. Fault management in burst switched optical WDM networks *Proc. of Terabit Optical Networking*, SPIE, 2000.

[9] J-D Shin, Saravut Charcranoon, Hakki C. Cankaya, and Tarek S. El-Bawab. Operation and maintenance issues in optical burst-switched networks. *Proc. of Optical Transmission Systems and Equipment for WDM Networking*, SPIE, 2002.

[10] Yufeng Xin, Jing Teng, Gigi Karmous-Edwards, George N. Rouskas, and Daniel Stevenson. Fault Management with Fast Restoration for Optical Burst Switched Networks. *Proc. of BroadNets*, IEEE, 2004.