Fault Management with Fast Restoration 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

This paper studies the important fault management issue with focus on the fast restoration mechanisms for Optical Burst Switched (OBS) networks. In order to reduce the burst losses during the restoration process, effective fast restoration schemes are necessary. This is illustrated via two basic fast restoration schemes, the distributed deflection scheme and the local deflection scheme, compared with the slow global routing update mechanism. A novel priority-based QoS restoration scheme is also proposed to provide differentiated restoration services. Through detailed descriptive analysis and a comprehensive simulation study, these fast restoration schemes demonstrate fast restoration process, low fault management overheads, and excellent burst loss performance. As far as we know, this is the first comprehensive study on the restoration mechanisms for OBS networks.

1 Introduction

As a promising future all-optical data-centric network technique, optical burst switching (OBS) has been attracting more and more research interest from both industry and academia. OBS is usually viewed as a technique bridging the optical circuit switching and the optical packet switching. [2] and [3] provide good general discussions on OBS techniques.

While the majority of previous studies have been focused on the resource reservation and scheduling [12] [8], burstification [13], signalling protocol definition and implementation [1], and performance analysis [5] [11], few studies have been done on the subject of fault management for OBS networks. However, with the terabits transmission capacity of a single fiber in a DWDM system, the possible vast data loss as a result of a link failure has made the fault management a critical topic for OBS networks.

The objective of network fault management is to recover the interrupted service as quickly as possible upon network element failure(s). The fault recovery techniques can be broadly classified into two categories: pre-planned protection and dynamic restoration. The protection technique relies on dedicated resources which are reserved in advance during connection setup. The restoration technique, on the other hand, is invoked right after the detection of a failure to discover spare capacity and reserve the network resource for the interrupted services. Compared with protection switching, restoration is more efficient in terms of resource utilization, but usually requires longer restoration time [14].

Protection is very suitable for the existing circuit switched networks or connection-oriented data networks (e.g. SONET and GMPLS). Since the service connections in these networks are explicitly set up and exist for a relatively long period, it is easy to set up and maintain a *backup path* for every *primary path*. The fundamental fault recovery mechanism for IP/MPLS networks is *global routing table update* triggered by link failure messages advertised by the router adjacent to the failure. The major drawback of this approach is the rather long convergence times. A fast topology-driven constraint-based rerouting restoration scheme is proposed for the MPLS network in [4] in which the router adjacent to the link failure will recalculate alternative routes based on the network topology excluding the failed link.

For the 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 impossible for a node to maintain the path (or path segment in the link-based case) information and pre-reserve spare resources for a connection requirement. Therefore, fast restoration is a better choice for the OBS network fault management. [10] presented a brief operation and maintenance framework for OBS networks. [9] studied a simple deflection routing scheme to achieve fault tolerance upon link failures for OBS networks. In [6], 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

---

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

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.

In this paper, we focus on the fast restoration mechanisms for the dominant short duration bursts in the OBS networks. Typically, routes of these bursts are dynamically decided on a hop-by-hop basis. Thus pre-planned protection techniques are not suitable in this case and an efficient dynamic restoration mechanism seems to be the best choice for the fault management of OBS networks. The restoration mechanism in this paper is proposed with the following objectives: (1) The restoration process should be very fast such that interrupted services can be restored in short time; (2) The overall burst losses during the failure and restoration are 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; (4) Differentiated restoration service is supported with the limited network spare capacity. As far as we know, this is the first comprehensive study on the restoration mechanisms for OBS networks.

The paper is organized as follows. A general discussion on the fault management for OBS networks is presented in Section 2. Two basic fast restoration mechanisms are discussed in Section 3. A novel priority-based restoration mechanism to provide differentiated restoration services is presented in Section 4. Results from a comprehensive simulation study are presented in Section 5. Section 6 concludes the paper.

2 Routing mechanism and fault management for OBS networks

In this paper, we assume that the OBS network operates under the JIT signaling protocol [1]. The basic routing mechanism is much like that of the IP network, 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 coming burst control packets and set up the connections based on 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 the OBS network relies on a link state protocol similar to OSPF. Each node collects link state information for all its adjacent links periodically and reports the information to the routing entity in the network. The routing entity then computes a new set of routes between every pair of OBS nodes and updates the forwarding table in the OBS nodes. In the centralized case, there is a given *Routing Decision Node* (RDN) in the network who fulfills the routing function for the entire network. In the case of distributed routing, there is no such RDN. All nodes flood to and receive from every other nodes 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 comprises the routing entity.

However, in this paper, whether the routing is centralized or distributed is irrelevant to our proposed fast restoration mechanisms. 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 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 shortest routes 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 per 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 and the number of hops of the 2 routes 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.

One thing that makes OBS networks different from circuit switched optical networks is that a wavelength channel not carrying a burst will not have the presence of light. Therefore, loss of light (LOL) does not necessarily mean a link failure. A ”Keep-Alive” control message is defined in [1] that is sent across adjacent nodes and can be used to detect link failures. There may also be other fault detection mechanisms available. However, we will not discuss the fault detection and localization techniques in this paper and assume that any link failure can be detected in a given time period.

Upon detection of a link failure, a fault notification message is sent to the RDN for routing re-computation, and the forwarding tables at all nodes will be updated. This is actually the 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, which will result in intolerable large amount of burst losses because all bursts, 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.

In the following sections, we will present two deflection-based fast restoration schemes and a priority-based QoS restoration mechanism. We note that all the discussions will be valid for a distributed routing architecture.

3 Fast restoration techniques for OBS networks

In this section, we illustrate the basic restoration mechanisms discussed in this paper through an example OBS network. The focus is on the burst loss under each restoration mechanism and only those bursts whose primary routes pass the faulty link are concerned.

3.1 Basic restoration schemes

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 $I \to 1 \to 2 \to 3 \to E$. Assuming the link between its *headend* node 2 and *tailend* node 3 is broken, every group of other 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. To work around the faulty link, we present following basic restoration schemes.

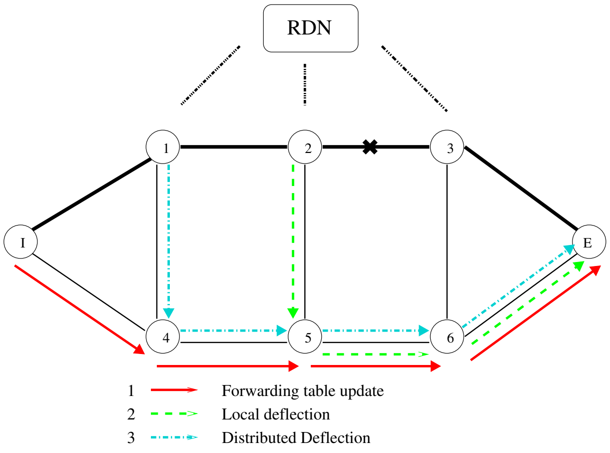

Figure 1. OBS fast restoration mechanisms

- 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 follow the new routes. For example, new bursts will follow the route $I \to 4 \to 5 \to 6 \to E$. 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 is the faulty link. In the example, new bursts from I to E will follow the alternative route $I \to 1 \to 2 \to 5 \to 6 \to E$. 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.

- Scheme 2. Distributed deflection: This is a novel fast restoration scheme we propose 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 about 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 $I \to 1 \to 4 \to 5 \to 6 \to E$. Comparing 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 may occur. However, this scheme requires extra fault notification time. And the capacity of the links between the headend (node 2) of the faulty link and its adjacent nodes (node 1) will not be utilized by the deflected bursts to their destination (node E).

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).

3.2 Burst losses during the restoration

Before the faulty link is worked around, bursts will be lost in the faulty link. After the successful restoration, burst losses would still be higher than that under normal condition due to the reduced network capacity. Therefore efficient fast restoration is usually desired. While the fault detection and processing time may normally be a constant, the fault notification time makes the major difference among different restoration schemes. The relationship between the restoration time and burst losses can be depicted in Figure 2.

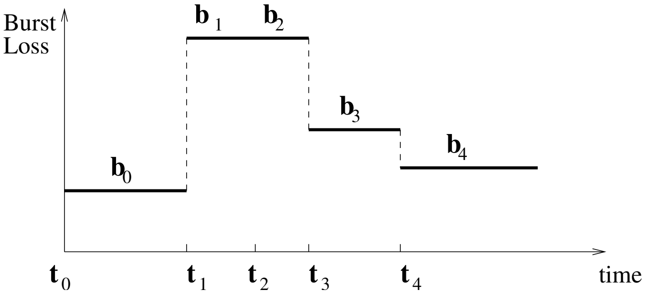

Figure 2. Burst losses vs. time

In this figure, the x-axis represents the time instant t_i in the life cycle of a fault management process. The network starts operation at t_0 and a link failure occurs at t_1. The fault detection and localization process is finished at t_2. The network element (the headend node in local deflection or adjacent nodes in distributed deflection) conducting the deflection task will receive the fault notification and restore the interrupted services at t_3. The routing tables get updated globally at t_4. The Y-axis represents the relative amount of burst losses b_i during the individual time periods between these time instant. While b_0, b_1, and b_4 are the same for all the restoration schemes since t_1 and t_2 are the same, the difference would be the b_2 and b_3. b_2 will increase with longer fault notification time between t_2 and t_3. b_3 is decided by the time period between t_1 and t_4 and will increase with longer routing table update time (t_4). Furthermore, during the period between t_3 and t_4, network congestion could increase when extra traffic is deflected to the alternative route. Therefore the total burst losses during the time period between t_2 and t_4 are jointly decided by the fault notification time and the network congestion condition.

In this paper, we use *Good-Phase*, *Detection-Phase*, *Notification-Phase*, *Deflection-Phase*, and *Heal-Phase* to represent the time periods $t_0 \sim t_1$, $t_1 \sim t_2$, $t_2 \sim t_3$, $t_3 \sim t_4$, and after t_4 respectively. We also use *Restoration-Phase* to represent the whole restoration period $t_1 \sim t_4$.

Upon a link failure, the alternative routes decided by the local deflection and the distributed deflection are different. First, it is likely that the longer route will result in higher burst loss probability since more resources are required. Furthermore, the longer route also results in longer burst delay which is not desirable as well. Either of the two alternative routes for a particular deflected burst could be the longer (shorter) one. Secondly, the distributed deflection requires longer fault notification time than the local deflection because of the transmission and processing delay of the control message from the headend/tailend nodes to their adjacent nodes. Therefore, the distributed deflection scheme has longer *Notification-Phase* period and results in more burst losses in this phase. However, the distributed deflection tends to make the deflected burst traffic more balanced such that the burst losses during the *Deflection-Phase* could be less than the local deflection case. In conclusion, the amount of total burst losses during the entire *Restoration-Phase* is decided by these factors jointly.

4 A novel priority-based QoS restoration scheme

As we have discussed in last section, fast restoration mechanisms are necessary to reduce the burst losses before the routing table update. However, the proposed two deflection schemes have different performance implications and the network congestion condition will deteriorate in different ways due to the deflected traffic. In this section, we present a novel priority-based restoration scheme, QoS restoration, to make the best use of the two fast restoration schemes and provide differentiated restoration services.

This scheme naturally leads to a new dimension of QoS definition for OBS networks: differentiated restoration time and burst losses during a network failure. This new dimension of QoS definition could be included into the Service Level Agreement (SLA). A similar concept can be found for MPLS QoS protection [7], though its underneath mechanism is totally different from ours.

Class	Restoration scheme
0	better one of local and distributed deflection
1	distributed deflection
2	global routing table update

Table 1. QoS restoration classes

In this QoS dimension, we define 3 restoration classes in Table 1 with different priority values. When a burst is generated at the edge node, it will be assigned a priority value in its control packet according to its restoration QoS requirement. The higher the priority value, the lower restoration quality the burst will receive. All the bursts in the lowest class (Class 2) whose primary routes pass the faulty link will not be deflected and will be directly dropped at the headend. Bursts in the highest class (Class 0) will pick up the best of the two fast deflection restoration schemes during different restoration periods. The local deflection will be chosen during the *Phase-Notification* period because of its shorter fault notification time. And the one of the local and distributed deflection schemes with shorter alternative route length (number of hops in this paper) during the *Phase-Deflection* period will be chosen because of its lower backup capacity requirement (and possible lower average burst loss probability). Bursts in the middle class (Class 1) will be restored via the distributed deflection. In this way, the middle class bursts would not compete for the bandwidth with the highest class bursts in the deflected alternative routes. We note that this scheme requires extra information about the route length available in the forwarding table, which is given by the RDN from routing computation.

This mechanism only functions before a global forwarding table update. The algorithm depicted as follows consists of two parts, one is performed at the headend node of the faulty link, and the other is performed at the adjacent nodes of the headend or the tailend of the faulty link. The bursts in the algorithm refer to those bursts whose primary routes pass the faulty link.

- Suppose x is the headend of the faulty link, y is one of the adjacent nodes of x. In the forwarding table of node i, an entry for the primary route is a triplex (d, n_i, l_i) and an entry for the alternative route is a triplex $(d, n_i^{'}, l_i^{'})$, where $i \in \{\mathtt{x}, \mathtt{y}\}$, d represents the destination, n_i and l_i represent the next hop node and the length of the primary route, $n_i^{'}$ and $l_i^{'}$ represent the next hop node and the length of the alternative route, respectively.

- **For the headend node** x: Upon detecting the link failure, x sends a fault notification message (d, δ) to all of its adjacent nodes (e.g., y), where $\delta = l_x^{'} - l_x$. An incoming burst will be deflected to its alternative route only if the burst belongs to Class 0, otherwise the burst will be dropped.

- **For the adjacent node** y: When it receives the fault notification message (d, δ), it forwards an incoming burst according to the algorithm described in Figure 3.

5 Simulation Study

In this section, we present the results from a comprehensive simulation study to illustrate and compare the performance of the four restoration schemes we discussed above. The simulation is conducted on an NSF network (14 nodes, 21 links) under the *JumpStart* JIT signalling protocol [1].

```
if the next-hop node is $x$, destination is $d$, and
fault notification $(d, \delta)$ is received then
    if the burst belongs to Class 0 then
        if $l_y + \delta > l_y^{'}$ then
            forward the burst to the alternative route $l_y^{'}$
        else
            forward the burst to $x$ ($x$ will forward it to
            its alternative path $l_x^{'}$)
    else if the burst belongs to Class 1 then
        forward the burst to the secondary path $l_y^{'}$
    else
        drop the burst if it belongs to Class 2
else
    forward the burst to the primary route $l_y$
    (not influenced by the faulty link)
```

Figure 3. Priority-based QoS restoration algorithm

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 bursts whose primary routes pass the faulty link. We will refer to the former as the overall burst loss (probability) and the later as the local burst loss (probability). We are especially interested in the burst loss (probability) during the *Restoration-Phase*, as the loss probability during the *Good-Phase* and *Heal-Phase* would be the same for all the three fast restoration schemes.

During the simulation, we let the most loaded link fail after the simulation runs enough time at steady state. We also assume the RDN is the center node of the network and is always working. The updated new burst routes will replace the deflection routes after the RDN update message arrives 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 $\rho = \frac{\lambda}{\mu * W}$, 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 different periods in the restoration cycle defined in Section 3.2. They are obtained within the 95% confidence interval by using the batch mean method. Using the terminology from Section 3.2, we set the fault detection time $t_2 - t_1 = 15\ ms$, the RDN update time is fixed to $t_4 - t_1 = 200\ ms$.

For the QoS restoration, we assume the total bursts are distributed into three classes as: Class 0: 30%, Class 1: 30%, Class 2: 40%. The class information is contained in the burst control packet. After the link failure, the OBS node looks at the restoration class of each burst and decides the restoration method used to restore the affected traffic according to the algorithm given in Section 4.

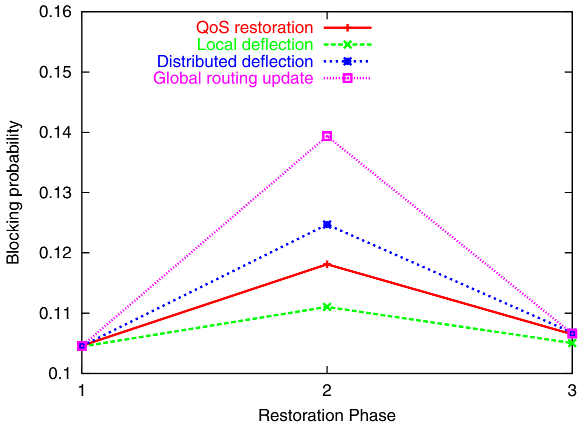

Figure 4. Overall burst loss probability vs. restoration phases ($\rho = 0.5$)

Fig. 4 and Fig. 5 depict the overall and local burst loss probability (y-axis) when $\rho = 0.5$ during the three major restoration phases. In the x-axis, 1 represents the *Good-Phase*, 2 the *Restoration-Phase*, and 3 the *Heal-Phase*. We observe that the burst loss probability is very low for both *Good-Phase* and *Heal-Phase*, though it is actually a little bit higher in *Heal-Phase* due to the lost link capacity. However, the loss probability increases significantly in *Restoration-Phase*. Purely relying on the forwarding table update would incur very high burst losses in this phase. However, the loss probability only increases moderately when the proposed three fast restoration schemes are used in this phase.

For the overall burst loss shown in Fig. 4, among the four restoration schemes, local deflection shows the best performance followed by priority-based QoS restoration and distributed deflection. Global routing update incurs the highest burst losses. Specifically, the improvements from using the three fast restoration schemes over the global forwarding table update are 20.3%, 15.3%, and 10.5%, respectively. The performance improvement in this phase is more obvious for the affected bursts whose primary routes pass the faulty link before new forwarding table update. As depicted in Fig. 5, these bursts will be directly dropped when no fast restoration schemes in place (the burst loss probability is 100% for the global routing update scheme), but only 27.8%, 49.1%, and 49.1% of the bursts are lost with local deflection, distributed deflection, and QoS restoration, respectively. We observe that local deflection achieves the least burst losses and distributed deflection and QoS restoration also reduce the burst loss dramatically over the global routing update. The obvious performance differentiation among the three basic restoration schemes forms the basis for the proposed differentiated QoS restoration scheme.

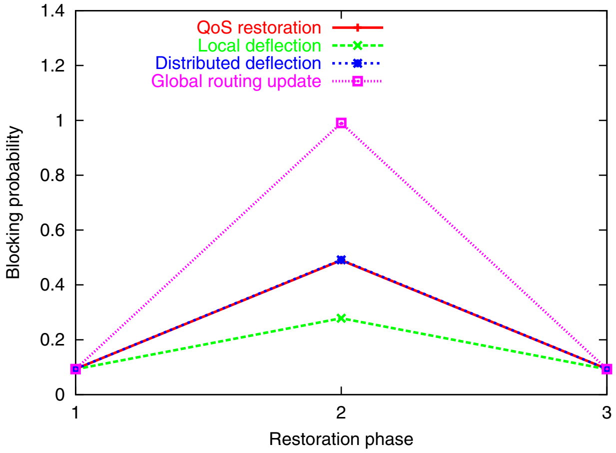

Figure 5. Local burst loss probability vs. restoration phases ($\rho = 0.5$)

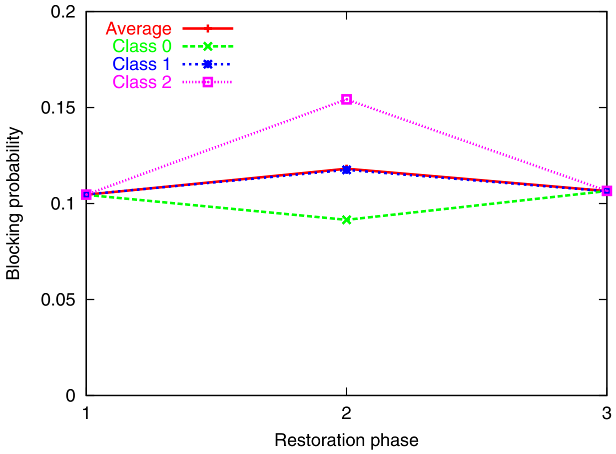

Figure 6. Overall burst blocking probability of priority classes vs. restoration phase ($\rho = 0.5$)

Fig. 6 and Fig. 7 show the burst loss probability of the three priority classes in the three phases when the QoS restoration scheme is used under the load of $\rho = 0.5$. We also compare them with the burst loss probability shown in Fig. 4 and Fig. 5 when individual local deflection or distributed deflection is used. The average loss probability over all the three classes is also depicted. The overall and local burst loss differentiation for bursts from different classes during the *Phase-Restoration* can be clearly observed. Bursts of Class 0 always achieve the lowest loss followed by bursts of Class 1. Bursts of Class 2 always encounter the highest burst loss.

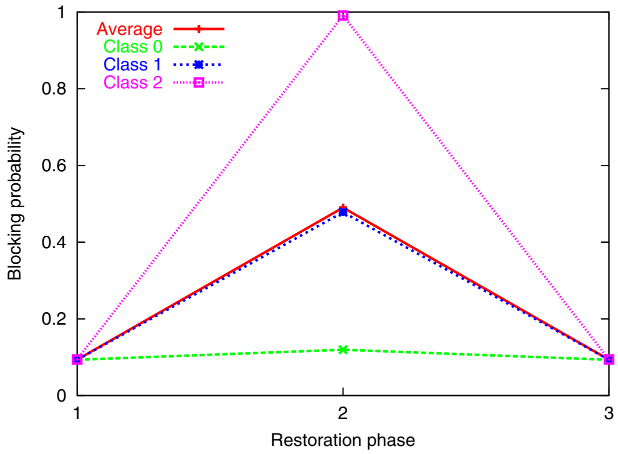

Figure 7. Local burst loss probability of priority classes vs. restoration phase ($\rho = 0.5$)

Comparing Fig. 6 with Fig. 4, we observe that the overall loss probability of Class 0 bursts is only 9.1%, an improvement of about 19% over that (11.1%) when pure local deflection is used. However the loss probability of Class 1 and 2 increases slightly compared with that when only distributed restoration and global routing update are used, respectively. For the local burst loss case, comparing Fig. 7 with Fig. 5 tells us that the local loss probability of Class 0 bursts is only 11.9%, an improvement of about 57% over that (27.8%) when pure local deflection is used. The local loss probability of Class 1 bursts is also lower (47.8%) than that (49.1%) when pure distributed deflection is used. The local burst loss probability of Class 2 will always be 1. We conclude that QoS restoration can reduce the burst loss of the highest class to the level that can never be achieved with any basic restoration mechanism. At the same time, its average performance is only next to the best basic restoration scheme.

Fig. 8 to Fig. 11 compare the performance of the four restoration schemes under different offered loads (different values of ρ) during the *Phase-Restoration*.

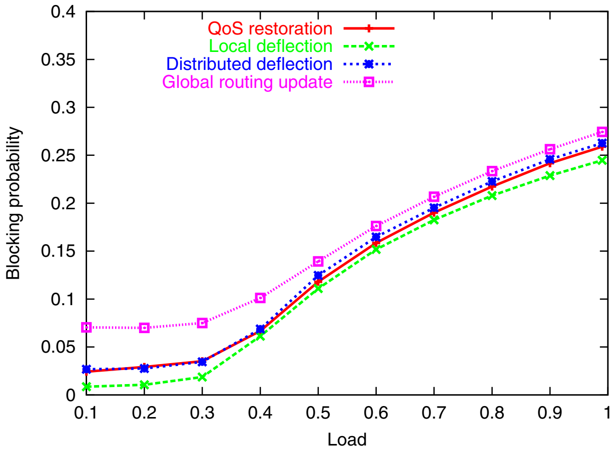

Figure 8. Overall burst loss probability with different schemes vs. offered load ρ

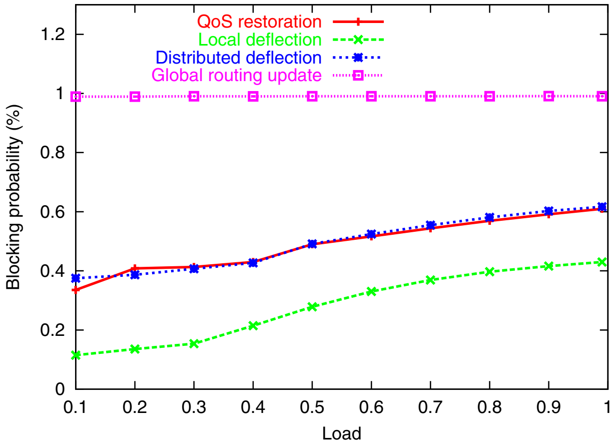

Figure 9. Local burst loss probability with different schemes vs. offered load ρ

Fig. 8 and Fig. 9 illustrate the overall and local burst loss probability under different restoration schemes. We observe that the local deflection has the best performance, waiting for the RDN update incurs the highest burst loss, and the performance of the distributed deflection and QoS restoration are in between. For example, at the load level of 0.1, the blocking probability of the global forwarding table update is about 9 times higher than that of local deflection, 3 times higher than the QoS restoration and the distributed deflection. However, the improvement with these fast restoration schemes tends to decrease as the load increases. For example, at the load of 1, the relative improvements are only 10.8%, 5.5%, and 4.4%. This is because the burst loss due to the network congestion begins to dominate as the load increases. This can be clearly seen in Fig. 4, where we observe that the burst loss probability during the first period and last period is already 10.4% while that during the second period (*Restoration-Phase*) is just 13.2%. This is also confirmed by Fig. 9 showing the local burst loss probability under different loads. We observe that the local burst loss probability increases almost linearly with the load increment when the three fast restoration schemes are used.

Fig. 10 and Fig. 11 show the overall and local burst loss probability of different classes when the QoS restoration is used. The performance differentiation among the three classes can be clearly observed, especially for the local bursts loss as depicted in Fig. 11.

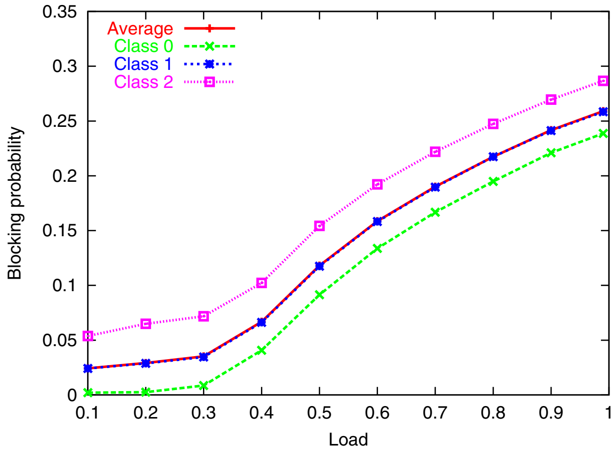

Figure 10. Overall burst loss probability of priority classes vs. offered load ρ

6 Conclusion

In this paper, we presented a fault management framework with the focus on fast restoration mechanisms for OBS

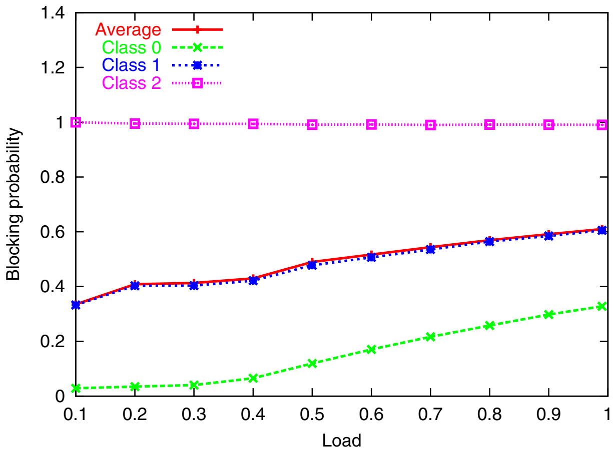

Figure 11. Local burst loss probability of priority classes vs. offered load ρ

networks. Two fast restoration schemes, distributed deflection and local deflection, along with the general global routing table update method, were discussed. We also proposed a novel priority-based restoration mechanism to provide differentiated restoration QoS. These schemes also feature low fault management overhead. The performances of the four schemes were analyzed via a comprehensive simulation study. The results showed that the presented fast restoration mechanisms can dramatically reduce the burst losses during the fault restoration process. When the priority-based scheme is used, the restoration quality differentiation can be clearly observed and the loss probability of bursts from the highest priority class can be reduced significantly.

References

[1] I. Baldine, G. Rouskas, H. Perros, and D. Stevenson. Jumpstart: A just-in-time signaling architecture for wdm burst-switched networks. *IEEE Communication Magazine*, 40(2), Feb. 2002.

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

[3] Y. Chen, C. Qiao, and X. Yu. Optical burst switching: A new area in optical networking research. *IEEE Network*, 18(3), May/June 2004.

[4] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, and P. Demeester. Data-centric optical networks and their survivability. *IEEE Journal on Selected Areas in Communications*, 20(1), Jan. 2002.

[5] K. Dolzer, C. Gauger, J. Spth, and S. Bodamer. Evaluation of reservation mechanisms for optical burst switching. *AEU International Journal of Electronics and Communications*, 55, Jan. 2001.

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

[7] J. L. Marzo, E. Calle, C. Scoglio, and T. Anjali. Adding qos protection in order to enhance mpls qos routing. In *Proceedings of ICC '03*. IEEE, 2003.

[8] C. Qiao and M. Yoo. Optical burst switching (obs)- a new paradigm for an optical internet. *IEEE Journal of Highspeed Network*, 8(1), 1999.

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

[10] J.-D. Shin, S. Charcranoon, H. C. Cankaya, and T. S. El-Bawab. Operation and maintainance issues in optical burst-switched networks. In *Proc. of Optical Transmission Systems and Equipment for WDM Networking*. SPIE, 2002.

[11] J. Teng and G. N. Rouskas. A comparison of the jit, jet, and horizon wavelength reservation schemes on a single obs node. In *Proc. of the First International Workshop on Optical Burst Switching*, Oct. 2003.

[12] J. Turner. Terabit burst switching. *IEEE Journal of Highspeed Network*, 8, 1999.

[13] V. M. Vokkarane, Q. Zhang, J. Jue, and B. Chen. Threshold-based burst assembly policies for qos support in optical burst-switched networks. In *Proc. of OptiComm'2002*. SPIE, 2002.

[14] Y. Xin and G. N. Rouskas. A study of path protection in large-scale optical networks. *Kluwer Photonic Network Communications*, 7(3), May 2004.