Fast Restoration Signaling in Optical Networks[*]

Radim Bartoš[†] and Swapnil Bhatia
Department of Computer Science
University of New Hampshire
Durham, NH 03824, USA
email: {rbartos, sbhatia}@cs.unh.edu

ABSTRACT
As WANs continue to form the framework for deploying mission-critical applications, the need for fault resilience and fast restoration assumes greater importance. This paper examines existing schemes for fast restoration signaling in a wavelength-routed optical core and introduces an improved signaling scheme based on Optical Burst Switching and Just-Enough-Time path establishment. Scaling experiments show that the proposed scheme outperforms existing schemes.

KEYWORDS
Optical Burst Switching, Restoration, Signaling.

1 Introduction

Over the years, large scale data networks have undergone significant change not only with respect to the volume and nature of the traffic they carry but also in regard to the criticality of their role, and the level of service demanded of them. WANs have evolved from simple, best-effort and specialized data networks used by a few, into complex, heterogenous, multi-layered, all-pervasive backbones guaranteeing an array of critical services to a global user population. Optical networks, with their terabit-wide links and ‘cut-through’, wavelength-routed architecture, seem well-suited to implement unified and homogenous, yet multipurpose networks to address the diverse data transport needs of the future. In such a scenario, since optical networks would provide all the services currently provided by separate networks, the issue of reliability and availability becomes all the more critical. Moreover, in most applications, the unified transport network may be the only resource driving a user’s business. Therefore, network downtime, no matter how small, may entail significant losses for the user. This paper proposes a new scheme for fast signaling for restoration, based on Optical Burst Switching (OBS), a unique, emerging paradigm for optical network design [1]. The paper discusses various issues involved in implementing fast restoration signaling in backbone transport networks. Existing schemes are described and a new scheme for restoration signaling is then proposed. Experiments to evaluate the proposed scheme are described followed by a discussion of the results.

2 Background

Efficient resource allocation has been at the center of much of the recent work in the area of restoration. As challenging as this problem may be, it is not the focus of our work. This paper addresses the problem of measuring and comparing the delay inherent to the signaling schemes currently used for lightpath setup. We assume that a path with sufficient resources has been found and the only task remaining is to reserve it. We proceed by making the following assumptions about the network. A cut-through, all-optical (i.e., no intermediate o-e-o conversion) wavelength-routed-network (WRN) is primarily[1] composed of fiber-links connecting nodes which contain optical switching elements such as Optical Add/Drop Multiplexers (OADMs) and Optical Cross Connects (OXCs) [2]. The all-optical nature of the signal path allows the network to be transparent and makes it ideal for use as a raw, high-bandwidth Optical Transport Network (OTN). A lower bandwidth IP network, known as the Data Communication Network (DCN) serves as a control plane to the OTN and is primarily used for exchange of signaling messages [3, 4]. A DCN node controls an OXC in the OTN and can instruct the OXC to execute a particular cross-connect operation. When a lightpath request arrives at the ingress, it computes a primary path to the egress node and using a signaling scheme, sets up the lightpath by exchanging messages with other controlling DCN nodes. Once the path has been set up completely, the ingress node forwards incoming traffic onto it. When a fault occurs, the flow of traffic to the egress stops and the traffic stream is subjected to heavy packet-loss. In this situation, traffic delivery to the egress node needs to be resumed as quickly as possible [5] in order to minimize packet-loss. We shall focus on single or multiple link or node failures[2] in the OTN. We assume the OTN to be 2-node connected and the DCN to be fault-free and reliable[3].

[*] This research was supported in part by NSF-ARI grant No. 9601602.

[†] Corresponding author.

[1] An all-optical WRN consists of a plethora of other devices as well, but none of these are pertinent from the perspective of the current discussion.

[2] A node failure can be modeled as a multiple link failure.

[3] The DCN can be protected using 1+1 protection switching as discussed in [4].

2.1 Current Schemes

We shall look at two different schemes that are currently used for activation of the alternate path. The first one is implemented by extensions to the RSVP protocol [6] while the second one can be implemented using plain IP or other standard signaling schemes such as GMPLS [4]. The following subsections discuss each scheme and derive average time estimates to complete each phase of the restoration process from detection to traffic-switching.

Term	Description
f	hop distance from ingress to node upstream to failed link, (nodes in OTN)
l_w	length of working path, (nodes in OTN)
l_a	length of alternate path, (nodes in OTN)
h	diameter of DCN, (nodes)
b_{len}	length of fiber link between adjacent OTN nodes, (miles)
c	speed of optical signal propagation through the OTN fiber links, (miles/second)
d	time by which the data burst must lag control header, (seconds)
t_{LOL}	time to decide Loss Of Light, (5 ms)
t_{prop}	propagation delay per link in DCN, (8 μs/mile)
t_{proc}	processing delay per node in DCN, (20 ms)
t_{detect}	time to decide link failure, (seconds)
$t_{restore}$	time to restore traffic, (seconds)
t_{FIS}	fault indication signaling delay, (seconds)
t_{setup}	time to setup alternate path, (seconds)
t_{switch}	time to switch traffic from working path to alternate path, (10 ms)
$t_{\times connect}$	time to perform a cross-connect operation (10 ms)

Table 1. Terms and values used in analysis and experiments as suggested in [4].

2.2 Preliminary Definitions

Please refer to Table 1 for a description of the terminology. Before we delve into estimating the delay for each scheme, we shall define the delay for a path. If $d_{DCN}(x)$ and $d_{OTN}(x)$ is the delay incurred due to a message travelling x hops through the DCN and the OTN respectively, then:

$$d_{DCN}(x) = x \cdot (t_{proc} + t_{prop}) + t_{proc} \tag{1}$$

$$d_{OTN}(x) = \frac{b_{len}}{c} \cdot x \tag{2}$$

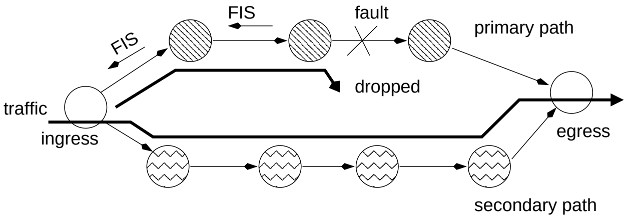

Figure 1. The RSVP/Backup Tunnel serial restoration scheme.

We shall follow the discussion in [7] and calculate the restoration delay as contributed by each phase of restoration. Hence:

$$t_{restore} = t_{detect} + t_{FIS} + t_{setup} + t_{switch} \tag{3}$$

2.3 RSVP/Backup Tunnels

Figure 1 illustrates the RSVP/Backup Tunnel restoration scheme. This restoration scheme is based on the MPLS protection scheme introduced in [6]. In this approach, the failed headend[4] informs the ingress node (which is f hops away from it) about the failure through a FIS message. The ingress node sets up the alternate path by sending a request to the first node on it. Each node forwards the request downstream without actually waiting to complete the cross connect. The egress node returns an acknowledgement which travels the same route upstream. Each node forwards it (upstream) as is, with a positive acknowledgement in it if the cross connect at its site was successfully completed. If not, it puts a negative acknowledgement. When the acknowledgement reaches the ingress, it knows if the path is setup by examining the type (positive or negative) of the acknowledgement. The time required to complete each phase of the restoration process is as follows:

$$t_{detect} = t_{LOL} \tag{4}$$

$$t_{FIS} = f \cdot d_{DCN}(h) \tag{5}$$

$$t_{setup} = 2 \cdot l_a \cdot d_{DCN}(h) \tag{6}$$

$$t_{restore} = t_{LOL} + f \cdot d_{DCN}(h) + 2 \cdot l_a \cdot d_{DCN}(h) + t_{switch} \tag{7}$$

2.4 Parallel Activation Architecture

Figure 2 illustrates the Parallel Activation architecture discussed in [4]. In this approach, as soon as a failure occurs, nodes downstream to the failed headend sense a loss of light. The egress node among these nodes, on sensing LOL, multicasts an alternate path setup message to each node on the alternate path. The message also contains the address of the node upstream to the node to which the message is sent. On receiving such a message a node on the alternate path extracts the address of its upstream neighbor from it and exchanges wavelength and port information

[4] The node immediately upstream to the failed link.

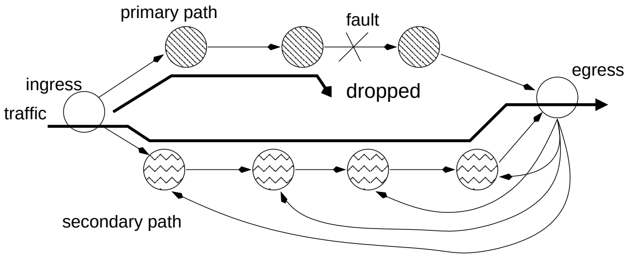

Figure 2. The parallel activation architecture.

with it. After this, each node sets up its cross connects and a complete alternate light-path is established. The time required to complete each phase of the restoration process is as follows:

$$t_{detect} = t_{LOL} \tag{8}$$

$$t_{FIS} = d_{OTN}(l_w - f) \tag{9}$$

$$t_{setup} = 2 \cdot d_{DCN}(h) \tag{10}$$

$$t_{restore} = t_{LOL} + d_{OTN}(l_w - f) + 2 \cdot d_{DCN}(h) + t_{switch} \tag{11}$$

3 Proposed OBS-based Scheme

Optical Burst Switching is a recent paradigm for optical network design. In OBS, a control packet is first sent to reserve bandwidth and configure switches along the path, followed by a burst of data without waiting for an acknowledgement of the connection establishment [1]. OBS thus belongs to the family of “tell-and-go”, one-way reservation protocols. OBS allows the data burst to be fairly large (order of megabytes) thus reducing overhead and can allow use of out-of-band signaling for transmission of control packet. OBS relies on the the Just-Enough-Time protocol for its operation. When a burst needs to be sent, the transmitting end calculates the time t required by the control packet to reach the farthest node on the path. It then transmits the control packet and after waiting for time t, sends the data burst. This allows just enough time for the switches along the path to configure themselves so that the following data burst is routed correctly to the destination. We now propose two alternate path activation schemes based on OBS.

3.1 Fast reroute with serial OBS activation

Figure 3 illustrates our proposed scheme. The scheme is based on the Fast Re-route scheme discussed in [7]. In this approach, the failed headend, upon sensing the failure, tries to setup an alternate path itself. In the path setup phase, the DCN headend constructs an OBS control packet containing information about timing and wavelength of the data burst to follow. When its upstream neighbor receives the control packet, it extracts information about the source of the following data burst and initiates a cross-connect operation in the OXC that it controls. It then updates the control packet with burst information for the next node and forwards it on. The ingress forwards the control packet further on downstream toward the egress through every node on the alternate path. Thus, the whole path is setup using the Just Enough Time (JET) burst switching technique. [8, 9, 10] suggest that such a signaling scheme which couples data bursts in a high speed OTN and control packets in a slower DCN is indeed practicable. Since the transmitting node (i.e., the node immediately upstream to the failed link) does not wait for an acknowledgement of path setup before forwarding traffic on the alternate path, the time to setup the path is significantly reduced. Moreover, the traffic stream can be resumed even before all the nodes on the path have received the control packet, as long as it can be guaranteed that the data burst always lags behind the control packet. The time required to complete each phase of the restoration process is as follows:

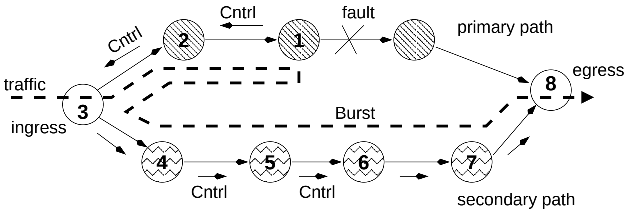

Figure 3. Serial OBS activation architecture.

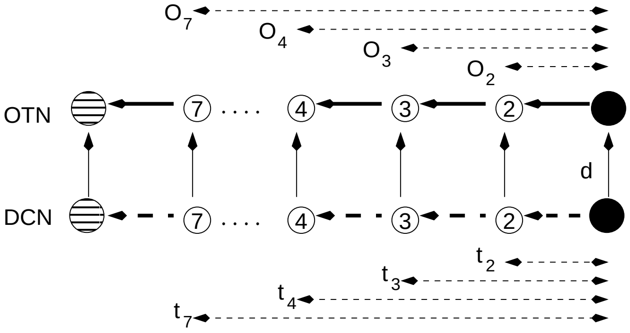

Figure 4. Calculating t_{setup} for serial OBS-based activation.

$$t_{detect} = t_{LOL} \tag{12}$$

$$t_{FIS} = 0 \tag{13}$$

As shown in Figure 3, the alternate path for this scheme begins at the failed-headend, goes through the ingress and ends at the egress. Thus, as per the figure, if the set of nodes in the alternate path is:

$$\mathcal{P} = \{node_2, node_3, \cdots, node_7\}$$

then, order to guarantee that the data burst always follows the control packet, the following constraint (illustrated by Figure 4) needs to be satisfied:

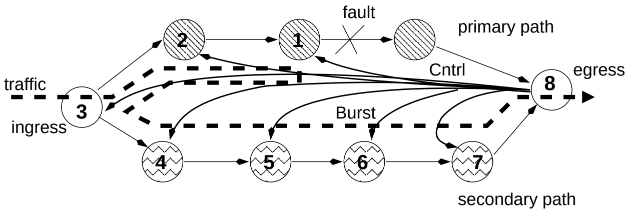

Figure 5. Parallel OBS activation architecture.

$$\forall\, n \in \mathcal{P} : d + O_n > t_n$$

where d is the time by which the data burst is required to lag behind the control packet, O_n is the time taken by the burst to travel to node n through the OTN, and t_n is the time taken by the control packet to travel to node n through the DCN. Hence:

$$t_{setup} = \max(\forall\, n \in \mathcal{P} : t_n - O_n) \qquad (14)$$

$$t_{restore} = t_{LOL} + \max(\forall\, n \in \mathcal{P} : t_n - O_n) + t_{switch} \quad (15)$$

3.2 Fast Reroute with Parallel OBS Activation

In the serial OBS activation architecture, route setup signaling messages flow serially starting at the node upstream to the failure and along the re-routed alternate path. However, this signaling scheme can be further improved by combining it with the parallel activation architecture introduced in [4]. Figure 5 illustrates our proposed scheme. In this hybrid signaling scheme, the egress node, multicasts separate OBS control packets to all the nodes on the alternate path over the DCN in parallel. Real internetworks are organized as transit-stub hierarchical structures [11]. As a result, the hop distance between a pair of nodes in an internetwork such as the DCN does not grow linearly with the geographical distance between them. Hence, the largest hop distance that an OBS control packet is required to travel before the data burst can be transmitted will always be smaller in case of the parallel OBS scheme than in the serial OBS activation architecture. This is reflected in the way t_{setup} is estimated for the serial and parallel architectures. For this scheme t_{detect} and t_{FIS} remain same as in Eqn. (12) and Eqn. (13). From Figure 5, following an argument similar to the one in the previous section, we arrive at the same[5] expression for $t_{restore}$ as Eqn. (15):

$$t_{restore} = t_{LOL} + \max(\forall\, n \in \mathcal{P} : t_n - O_n) + t_{switch} \quad (16)$$

We now describe the experiments carried out in order to compare the performance of the four schemes discussed above.

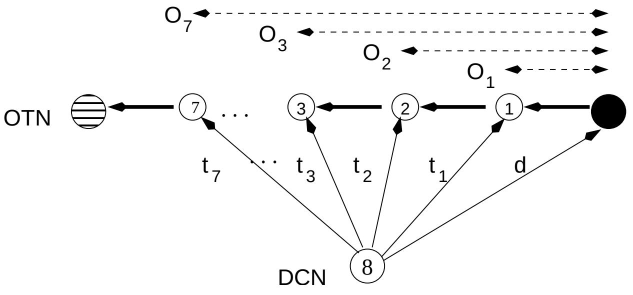

Figure 6. Calculating t_{setup} for parallel OBS-based activation.

4 Experimental Evaluation

Experimental measurements were performed in order to study the effect of each of the activation schemes on the restoration time. An experimental run consisted of measuring the average restoration time using the estimates calculated above for each of the four activation schemes for a given OTN and DCN network pair. Table 1 shows the typical values of various parameters used in the calculation of restoration time as suggested in [4]. The Georgia Tech Internet Topology Modeler (`gt-itm`) was used in order to generate pure-random [11], undirected OTN and DCN graphs. The OTN graphs were generated using the Flat Random Graph model provided by the `gt-itm` whereas the Transit-Stub Graph model was used to represent DCN structure [11, 12]. The `gt-itm` package offers ways to control the network topology model, the number of nodes, the edge probability, the node-link distribution and the geographical network size for the graphs generated. In order to be representative of real networks, the geographical sizes of the OTN and the DCN were always maintained equal. A mapping was developed from each node in the OTN to a unique node in the DCN to model its controller. Using the geographical position information provided by `gt-itm` for each node in a graph, the mapping function ensured that an OTN node was always mapped to the approximately[6] nearest DCN node. Furthermore, the lengths of edges between a pair of OTN nodes were then set to the distance between their controlling DCN nodes to ensure a sound model. The OTN generated was always much smaller than the DCN in node-count (except in Experiment 3). Each data point was obtained by running measurements on sets of at least ten different randomly generated OTN/DCN pairs. Only 2-node connected graphs were used as OTNs for any experiment. All $\frac{(p-1)\cdot p}{2}$ possible shortest paths in an p-node OTN were considered to be primary paths. All $\frac{(p-1)\cdot p}{2}$ possible node- and link-disjoint second-shortest paths in an p-node OTN were considered to be secondary or alternate paths. Each

[5] Here t_n still represents DCN delay, but unlike serial OBS activation, $\forall n \in \mathcal{P} : t_{n+1} > t_n$ does not necessarily hold for internetworks in this case.

[6] For each OTN node, the DCN node with the smallest Euclidian distance from it was chosen. However, this still does not guarantee the best geographical superimposition of the DCN on the OTN with all distances minimized.

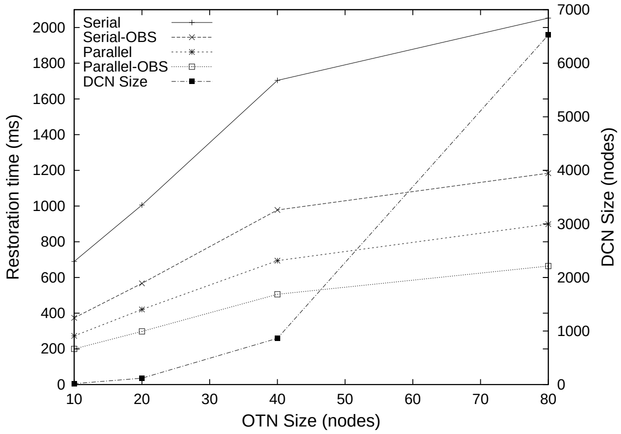

Figure 7. Restoration time v/s OTN and DCN size (Edge probability: DCN = 0.13, OTN = 0.23).

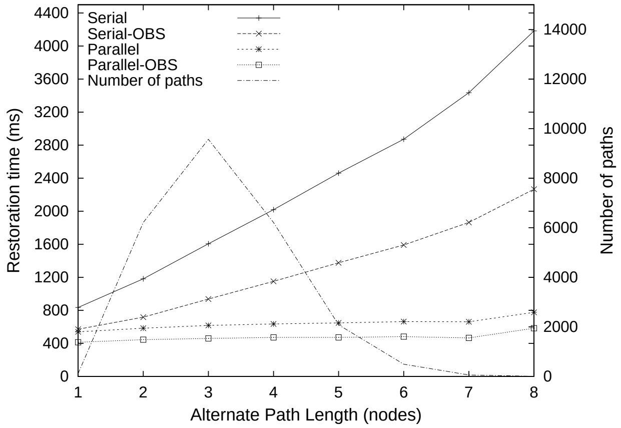

Figure 8. Restoration time v/s alternate path length (OTN size = 100 nodes, DCN size = 480 nodes; OTN edge probability = 0.13 and that of DCN = 0.19).

data point represents the mean of the time to restore traffic from a primary path (shortest path) to its node- and link-disjoint secondary (second-shortest) path. The Stanford GraphBase [13] library was used for writing routines to perform measurements. The Walrus Graph Visualization tool from CAIDA[7] was used for visualizing graphs.

4.1 OTN and DCN Size

In Experiment 1, DCN and OTN were expanded two-fold[8] at every run. The pure-random model was used for generating edges with edge probabilities 0.23[9] and 0.13 for the OTN and DCN respectively. The average restoration time from all primary paths to all secondary paths in each randomly generated OTN was measured for each of the four activation schemes. Figure 7 shows the performance of each of the four activation schemes for different sizes of the OTN and the size of the DCN (scaled down ten-fold) at which the measurement was performed. As seen in the figure, the proposed serial and parallel OBS-based schemes outperform their non-OBS counterparts.

4.2 Alternate Path Length

In Experiment 2, the OTN and DCN sizes were held constant. The restoration time for all alternate path-lengths was measured over a set of ten random DCN/OTN pairs. The pure-random model was used for generating edges with edge probabilities 0.13 and 0.19 and sizes fixed at 100 and 480 nodes for the OTN and the DCN respectively. Figure 8 shows the performance of each of the four schemes on varying path lengths, and the distribution of path lengths (scaled down ten-fold) in the OTN graphs on which the measurements were performed. As seen from the results, Parallel OBS activation scheme scales better than all the other schemes as path length increases. Similarly, parallel activation also scales well with path length but performs worse than its OBS cousin. Parallel schemes thus benefit from the structure of the DCN whereas serial schemes do not.

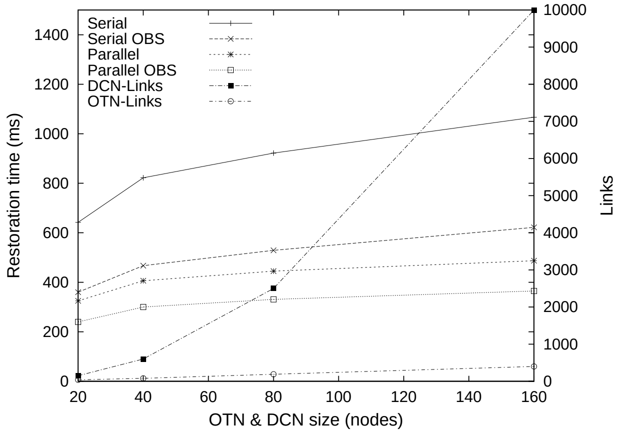

Figure 9. Restoration time v/s network size (edge probability of OTN = 0.4 and that of DCN = 0.18).

4.3 Network Size

In Experiment 3, the random DCN/OTN pairs were generated with both containing the same number of nodes. The network size (i.e., sizes of both the OTN and the DCN) were doubled for every run and the restoration time for all possible paths was measured for each of the four schemes. The pure-random model was used for generating edges with edge probabilities set to 0.4 and 0.18 for the OTN

[7] http://www.caida.org.

[8] Since the DCN is a transit-stub graph, the nodes in the DCN increased by a factor of $Z = 4 + 4 \cdot \frac{s}{s+1}$ where s is the number of nodes in a stub.

[9] A relatively higher edge-probability is required in order to guarantee 2-node connectedness using the `gt-itm` package.

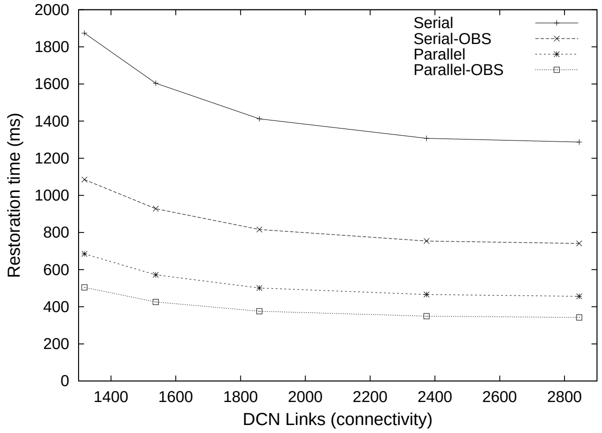

Figure 10. Restoration time v/s DCN connectivity (OTN size = 100 nodes, DCN size = 576 nodes; edge probability of OTN = 0.13).

and DCN respectively. Figure 9 shows the results of Experiment 3. It shows the performance of each of the four schemes for varying network size with information about the number of links in the OTN and DCN (scaled down ten-fold) in the OTN/DCN graphs on which measurements were performed. As seen in Figure 9, parallel OBS activation scheme exhibits the best performance as network size increases, followed by plain parallel activation.

4.4 DCN Connectivity

In Experiment 4, the connectivity of the DCN was gradually increased. The sizes of the OTN/DCN pairs generated were held constant. The pure-random model was used for generating edges with edge probabilities set to 0.13 for the OTN and varied for the DCN. At every run, the edge probabilities for the DCN were doubled. As seen from Figure 10, parallel activation schemes exhibit stable scaling performance as compared to serial schemes. Again, OBS-based schemes outperform their non-OBS counterparts.

5 Conclusion

This paper focused on the delay inherent to various lightpath activation schemes using the novel [4] OTN-DCN model with dissimilar data and control latencies. Two existing schemes were discussed and two variants of a new scheme were proposed. The four schemes were compared through experiments based on the time-to-restore metric. Initial results from the experiments confirm that the proposed schemes offer lower setup delays thus minimizing packet loss during restoration. The proposed schemes also exhibit resilient scaling behavior benefiting from the natural structure of DCN internetworks.

Acknowledgements

The authors would like to thank the reviewers for their helpful comments and suggestions.

References

[1] C. Qiao, “Optical Burst Switching (OBS) - A New Paradigm for an Optical Internet,” *Journal of High Speed Networks*, vol. 8(1), pp. 69–84, 1999.

[2] T. E. Stern and K. Bala, *Multiwavelength Optical Networks: A Layered Approach*. Addison Wesley Longman, 1999.

[3] G. P. Austin, B. T. Doshi, C. J. Hunt, R. Nagarajan, and M. A. Qureshi, “Fast, Scalable, and Distributed Restoration in General Mesh Optical Networks,” *Bell Labs Technical Journal*, vol. January-June, pp. 67–81, 2001.

[4] B. T. Doshi, S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang, “Optical Network Design and Restoration,” *Bell Labs Technical Journal*, vol. January-March, pp. 58–84, 1999.

[5] A. Gandhi and R. Bartoš, “Dynamic issues in MPLS service restoration.” Department of Computer Science, University of New Hampshire Technical Report TR 01-05, November 2001.

[6] K. Owens, V. Sharma, S. Makam, and C. Huang, “A path protection/restoration mechanism for MPLS networks.” Work in progress *[draft-chang-mpls-path-protection-03.txt]*, July 2001.

[7] D. Haskin and R. Krishnan, “A method for setting an alternative label switched paths to handle fast reroute.” Work in progress *[draft-haskin-mpls-fast-reroute-05.txt]*, November 2000.

[8] G. Li, J. Yates, D. Wang, and C. Kalmanek, “Control Plane Design for Reliable Optical Networks,” *IEEE Communications Magazine*, vol. 40 No. 2, pp. 90–96, February 2002.

[9] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, “JumpStart: A Just-In-Time Signaling Architecture for WDM Burst-Switched Networks,” *IEEE Communications Magazine*, vol. 40 No.2, pp. 82–89, February 2002.

[10] C. Qiao, “Labeled Optical Burst Switching for IP-Over-WDM Integration,” *IEEE Communications Magazine*, vol. 10, pp. 240–242, 2000.

[11] K. Calvert, M. Doar, and E. W. Zegura, “Modeling Internet Topology,” *IEEE Communications Magazine*, June 1997.

[12] E. W. Zegura, K. Calvert, and M. J. Donahoo, “A Quantitative Comparison of Graph-based Models for Internet Topology,” *IEEE/ACM Transactions on Networking*, vol. 5, No. 6, December 1997.

[13] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*. ACM Press, 1993.