# Fast Service Restoration in Optical Networks[1]

*Radim Bartoš*[2] and *Swapnil Bhatia*
Department of Computer Science, University of New Hampshire
Durham, NH 03824, USA, Phone: (603) 862-3792, Fax: (603) 862-3493
E-mail: {rbartos,sbhatia}@cs.unh.edu

## Abstract

**As WANs continue to form the framework for deploying mission-critical applications, the need for fault resilience and restoration assumes greater importance. This paper examines the issues involved in providing fast service restoration in a wavelength-routed optical core and proposes schemes that provide fast restoration with minimal packet-loss. The paper introduces an improved signaling scheme based on Optical Burst Switching and Just-Enough-Time path establishment which allow fast restoration without pre-reservation of resources. Simulation analysis for scalability and restoration time shows that the proposed scheme outperforms existing schemes.**

## 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. Heterogeneity, layering, bandwidth limitation, and complexity hamper the efficient and wide-spread application of infrastructure to user needs. Optical networks, with their terabit-wide links and 'cut-through', wavelength-routed architecture, appear to be 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. Many applications (trunk telephony) may also impose other constraints on the traffic, such as QoS guarantees etc., which may require restoration schemes to complete restoration within strict time bounds. 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.

On the other hand, guarantees regarding availability are not without additional costs themselves. The problem of building reliable and affordable backbone networks is essentially a problem of deciding optimal trade-offs between various conflicting variables. Broadly, the solution can be described in terms of three variables: reliability, flexibility, and efficiency. High reliability constraints may entail inefficient allocation of resources leading to higher costs. Low cost solutions may not be able to provide hard guarantees. In addition, any solution must allow for flexible control and administration of resources so that they can be allocated when and where they may be required.

The main contribution of this paper is a new scheme for fast service restoration based on Optical Burst Switching (OBS), an emerging and interesting paradigm for optical network design [1]. The paper starts with a discussion of various issues involved in implementing fast restoration in backbone transport networks from a practical standpoint. A general model of the process of restoration is derived and various existing schemes are described inline with the model. A new scheme for restoration is then proposed and a restoration-time estimate for each of the schemes is calculated. Simulation experiments are described in detail followed by a discussion of the results and directions which can be further explored.

[1]This research was supported in part by NSF-ARI grant No. 9601602.
[2]Corresponding author.

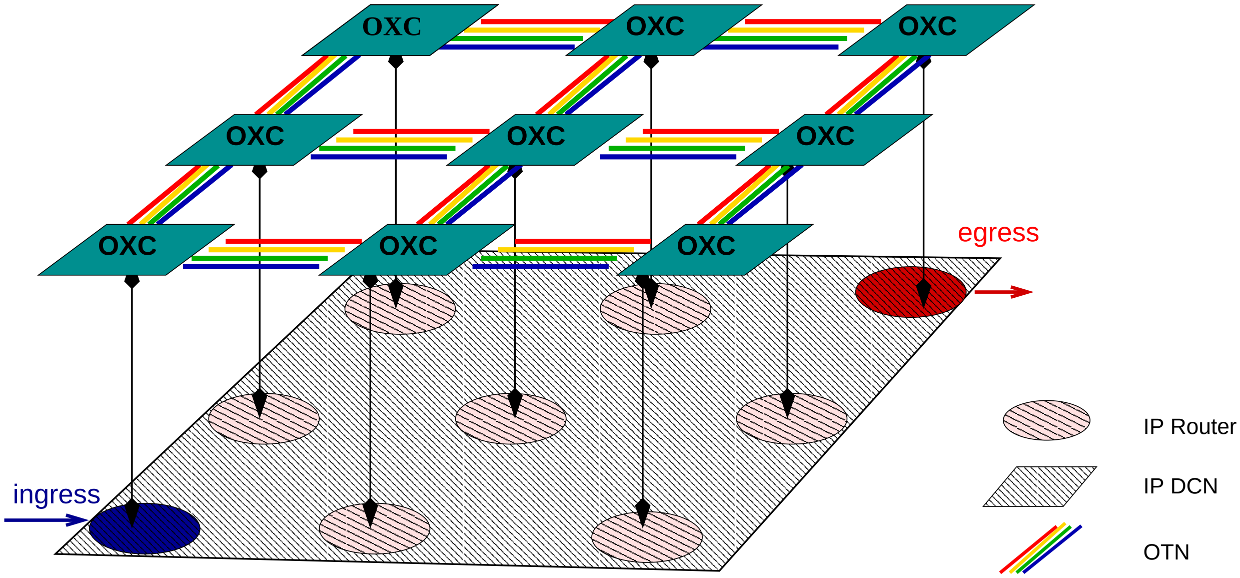


Figure 1: An Optical Transport Network (OTN) with a Data Communication Network (DCN).

## 2 Background & Problem Definition

In this paper we shall focus on the problem of fast restoration in an all-optical, wavelength-routed network (WRN). As shown in Figure 1, a WRN is primarily[3] composed of fiber-links connecting nodes which contain optical switching elements such as Optical Add/Drop Multiplexers (OADMs) and Optical Cross Connects (OXCs) [2]. Each fiber-link is capable of carrying multiple wavelengths (typically 20-100). OXCs can support any permutation of connections of optical signals received on an input wavelength on an input port to any output wavelength on any output port. Since the data signal travels on a different wavelength on each segment of its path, the network is called a wavelength-routed network. Each wavelength can provide data rates of the order of tens of gigabits per second, thus allowing a single fiber to be used as a terabit link. Moreover, the all-optical nature of the signal path allows the network to be transparent. This makes the all-optical network 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. For example, on receiving a $\texttt{connect}((\lambda_1, \texttt{port}_\texttt{a}), (\lambda_2, \texttt{port}_\texttt{b}))$ message, a DCN node initiates a cross-connect operation between the input wavelength $\lambda_1$ on $\texttt{port}_1$ and output wavelength $\lambda_2$ on $\texttt{port}_\texttt{b}$ in the OXC under its control. Standard protocols such as GMPLS or RSVP may be easily adapted to implement this signaling scheme. The DCN runs its own routing algorithm to route such control messages among the DCN nodes[4]. In addition, each node in the DCN also stores information about the physical and virtual topology of the OTN. The physical topology is used to compute a path through the OTN and the DCN nodes to be signaled in order to setup or tear down the path.

Given this background, the problem of restoration can now be summarized as follows. A demand for transportation of data between an ingress and an egress node arrives at an ingress node. The ingress node either possesses or gathers information about the topology of the OTN by exchanging messages with other DCN nodes over the DCN. Based on this information, the ingress node then computes a primary path route to the egress node. Using some signaling scheme, the ingress node then sets up a lightpath along the pre-computed route by exchanging messages with the controlling DCN nodes of those OTN nodes that lie on the path. Path setup actually takes place in the OTN and primarily consists of configuration of switches within

[3] 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.

[4] As for the OTN, traffic cuts through an all-optical network from ingress to egress over a lightpath purely in the optical domain and therefore does not require per-hop routing.

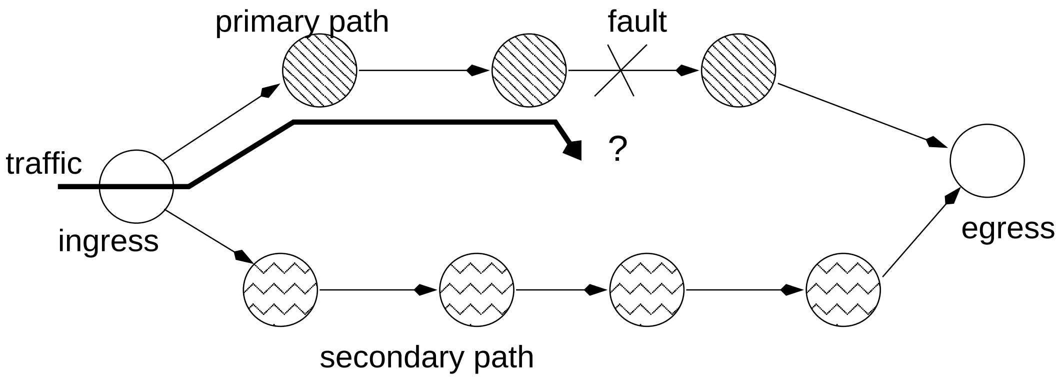


Figure 2: The problem of traffic restoration.

the OXCs along the path. Once the path has been set up completely, the ingress node forwards incoming traffic onto it. Figure 2 illustrates this process. Every lightpath is active for a fixed lifetime, after which it is disconnected using a process similar to the above. During the lifetime of a lightpath, a fault may occur on any of the links over which the lightpath is routed. Once 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 in order to minimize packet-loss.

This is important for a variety of reasons. Packet loss triggers retransmission and other adaptation schemes in the higher network layers which in turn may lead to packet re-ordering and lower throughput [5]. Furthermore, packet-loss is undesirable in itself if the OTN is being used to deploy mission-critical applications (telephony, video, IP storage, etc). A restoration scheme must be designed so as to meet these goals. We shall focus on single or multiple link or node failures in the OTN. We assume the OTN to be 2-node connected[5] and the DCN to be fault-free and reliable[6].

## 3 A General Model of Restoration

In general, the process of restoration can be cleanly divided into (at least) four different phases [6]. Once a failure occurs, the fault needs to be detected (detection phase). Once detected by a node, this information has to be propagated to other nodes for recovery procedures to be initiated (Fault Indication Signaling (FIS) phase). Once recovery has been initiated, a new path has to be setup (setup phase). Depending on the restoration scheme this may or may not involve computation of a new path to the egress. After an alternate path has been setup, traffic has to be switched to the secondary path (switching phase). Thus, the time taken to restore traffic will include the time required to complete each of the above phases. However, the time required to compute and setup a new route may depend on factors such as path length, network size, network load, etc. Route computation and route setup may thus be the largest contributors to restoration delays. The performance of a restoration scheme can be significantly affected depending on whether each of these tasks is executed before or after a fault occurs. [4] evaluates protection schemes on the basis of four performance criteria: restorability, scalability, restoration speed and capacity efficiency. Restorability is concerned with the kinds (node, link) and the number of failures a scheme can protect against. Scalability is a measure of the increase in cost of applying a scheme to a network as the size of the network and number of demands grow. Restoration speed is the time elapsed between the moment of failure and the point at which service is restored. Capacity efficiency measures the optimality with which a scheme uses the available bandwidth in order to protect a given number of demands given a fixed amount of total bandwidth.

Table 1 shows the impact of alternate route computation before or after failure on the above performance measures. Maximum restorability can be achieved by pre-computing routes for all failure scenarios (any number, any kind). This may be highly computationally intensive in which case it may be cheaper to compute routes after actual information regarding a fault is available. (The argument is similar for the

[5]A node failure can be modeled as a multiple link failure.
[6]The DCN can be protected using 1+1 protection switching as discussed in [4].

capacity efficiency of a restoration scheme). This also makes the restoration scheme scalable since a fixed, fewer (actual) number of failure scenarios have to be considered after a fault has occurred. On the other hand, computation after failure can delay restoration of service causing heavy traffic loss.

| Metric | Route Computation Before Failure | Route Computation After Failure |
|---|---|---|
| **Restorability** | Positive | Highly Positive |
| **Scalability** | Highly Negative | Positive |
| **Rest. Speed** | Highly Positive | Negative |
| **Capacity Efficiency** | Positive to Highly Positive | Highly Positive |
| **Computation** | Highly Negative | Highly Positive |

Table 1: Impact of route computation on performance metric.

Table 2 shows the impact of alternate route setup before or after failure on the same performance criteria. Restorability remains largely unaffected by route setup. Route setup involves signaling messages and these may not scale when a large number of connections have to be re-routed to alternate paths (however, this may be remedied by bundling requests together). Pre-reservation of alternate paths cuts down on signaling delay when an actual fault occurs, but results in inefficient utilization of bandwidth. Handling a large number of signaling requests by a few nodes along the alternate path may be computationally intensive.

| Metric | Route Setup Before Failure | Route Setup After Failure |
|---|---|---|
| **Restorability** | None | None |
| **Scalability** | Positive to none | Negative to none |
| **Rest. Speed** | Highly Positive | Highly Negative |
| **Capacity Efficiency** | Highly Negative | Highly Positive |
| **Computation** | Positive | Negative to none |

Table 2: Impact of route setup on performance metric.

Various existing schemes that perform pre-reservation and pre-computation also suggest interim use of pre-allocated alternate lightpaths to carry best effort (BE) traffic till the time a failure actually occurs. For these schemes, pre-allocation only guarantees the availability of an alternate lightpath after a failure. The path still needs to be established such that it discontinues delivery of BE traffic and dedicates itself to carrying the actual traffic being protected. This incurs at least a round-trip latency during which heavy traffic loss may occur. Furthermore, most schemes based on pre-allocation are practically inextensible to multiple faults. Making them applicable to multiple fault scenarios either involves a heavy price in computation and/or a large penalty in bandwidth. Real-time schemes on the other hand, which perform route computation and setup after the fault occurs are largely immune to bandwidth penalties. The cost in this case is usually in the form of added delay in computing routes and/or establishing alternate paths resulting in a significant amount of packet loss.

It is clear that a trade-off has to be chosen between pre- or post-computation of routes and pre- and post reservation of bandwidth. The remedies for these tradeoffs clearly lie in the algorithms and techniques used for route computation and setup. We shall separate the general restoration problem into two parts: the route computation problem and the route setup problem. This paper solely focuses on the route setup problem and considers schemes which perform route pre-computation without pre-reservation of bandwidth. We discuss two such schemes and propose a new, low delay scheme for route setup in the next section.

## 3.1 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 while the second one can be implemented using plain IP or other standard signaling schemes such as GMPLS. The following subsections discuss each scheme and derive average time estimates to complete each phase of the restoration process from detection to traffic-switching. (Table 3 lists all the terms involved in deriving time estimates).

| Term | Description |
|---|---|
| FIS | fault indication signal |
| DCN | Data Communication Network (control network) |
| OTN | Optical Transport Network (data network) |
| OBS | Optical Burst Switching |
| $m_x$ | length of message of type $x$, (bits) |
| $r_{DCN}$ | bandwidth of DCN, (bits/second) |
| $f$ | the $f^{th}$ node from the ingress in OTN is immediately upstream to failed link, (nodes in OTN) |
| $l_w$ | length of working path in the OTN, (nodes in OTN) |
| $l_a$ | length of alternate path in the OTN, (nodes in OTN) |
| $h$ | diameter of DCN, (nodes) |
| fiberLen | length of fiber link between adjacent nodes in the OTN, (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, (seconds) |
| $t_{trans}(x)$ | time to transmit message of length $x$ bits in the DCN, (seconds) |
| $t_{prop}$ | propagation delay per link in DCN, (seconds) |
| $t_{proc}$ | processing delay per node in DCN, (seconds) |
| $t_{detect}$ | 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, (seconds) |

Table 3: Terminology.

### 3.2 Definitions

Refer to Table 3 for a description of terminology.

$$t_{prop} = \frac{m}{r_{DCN}} + t_{trans}(m) \tag{1}$$

$$d_{DCN}(x) = x \cdot (t_{proc} + t_{prop}) + t_{proc} \tag{2}$$

$$d_{OTN}(x) = \frac{\texttt{fiberLen}}{\texttt{c}} \cdot x \tag{3}$$

$$t_{restore} = t_{detect} + t_{FIS} + t_{setup} + t_{switch} \tag{4}$$

### 3.3 RSVP/Backup Tunnels

Figure 3 illustrates the Backup Tunnel restoration scheme. This restoration scheme is based on the MPLS protection scheme introduced in [7]. In this approach, the failed headend[7] 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 (positive or negative) of the acknowledgement. The time required to complete each phase of the restoration process is as below:

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

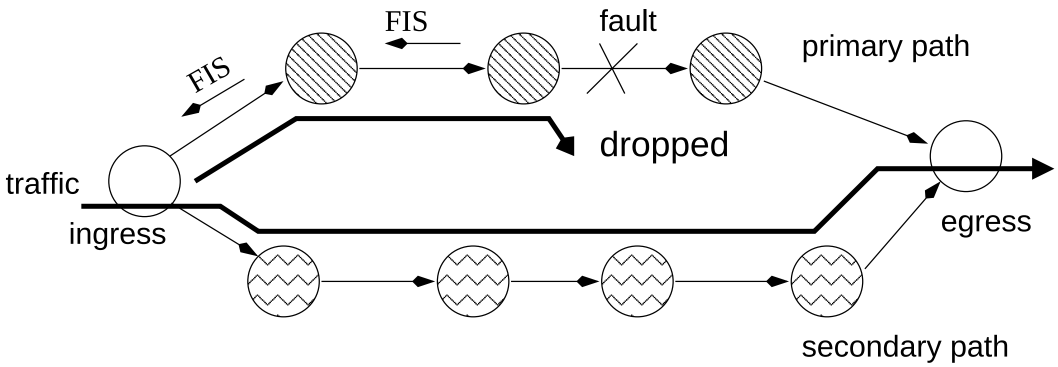


Figure 3: The RSVP/Backup Tunnel serial restoration scheme.

$$t_{detect} = t_{LOL} \tag{5}$$
$$t_{FIS} = f \cdot d_{DCN}(h) \tag{6}$$
$$t_{setup} = 2 \cdot l_a \cdot d_{DCN}(h) \tag{7}$$
$$t_{restore} = t_{LOL} + f \cdot d_{DCN}(h) + 2 \cdot l_a \cdot d_{DCN}(h) + t_{switch} \tag{8}$$

### 3.4 Parallel Activation Architecture

Figure 4 illustrates the Parallel Activation architecture. This restoration scheme is based on the MPLS protection scheme introduced 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 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 below:

$$t_{detect} = t_{LOL} \tag{9}$$
$$t_{FIS} = d_{OTN}(l_w - f) \tag{10}$$
$$t_{setup} = 2 \cdot d_{DCN}(h) \tag{11}$$
$$t_{restore} = t_{LOL} + d_{OTN}(l_w - f) + 2 \cdot d_{DCN}(h) + t_{switch} \tag{12}$$

## 4 Proposed OBS-based Activation 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 for 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.

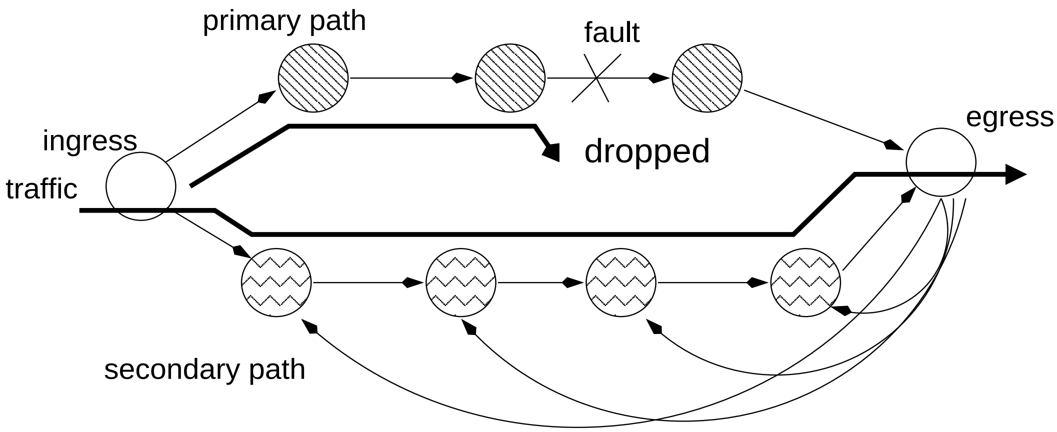


Figure 4: The parallel activation architecture.

### 4.1 Fast reroute with serial OBS activation

Figure 5 illustrates our proposed scheme. The scheme is based on the Fast Re-route scheme introduced in [8]. In this approach, the failed headend, upon sensing the failure, tries to setup an alternate path itself. (Hence, nothing needs to be done in the FIS phase, for this approach). In the path setup phase, the DCN node upstream to the failed link first constructs an OBS control packet containing information about timing and wavelength of the data burst to follow. It then transmits it upstream toward the ingress by forwarding it to its immediate upstream neighbor in the DCN. When the next DCN node receives the control packet, it extracts information about the source of the following data burst from the control packet and initiates a cross-connect operation in the OXC that it controls. The DCN node then updates the control packet with information for the next node on what port to expect the burst on and forwards it on. A section of the alternate path upto the ingress is the same as the working path, but in the opposite (upstream) direction. The remaining portion of the alternate path is node and link disjoint with the primary path and goes from the ingress to the egress node. The ingress forwards the control packet further on downstream toward the egress through every node on the alternate path till the control packet reaches the egress. Thus, whole path is setup using the Just Enough Time (JET) burst switching technique. [9, 10] suggest that such a signaling scheme which couples data bursts in a high speed OTN and control packets in a lower bandwidth 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 drastically 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 below:

$$t_{detect} = t_{LOL} \quad (13)$$

$$t_{FIS} = 0 \quad (14)$$

From Figure 5 it is clear that the alternate path for this scheme begins at the failed-headend, goes through the ingress and ends at the egress. Thus, if alternate path:

$$\mathcal{P} = \{\text{nodesInPath(failed headend} \rightarrow \text{ingress} \rightarrow \text{egress)}\}$$

then as per Figure 6, in order to guarantee that the data burst always follows the control packet, the following constraint needs to be satisfied:

$$\forall\ node \in \mathcal{P} : d + O_{node} > t_{node}$$

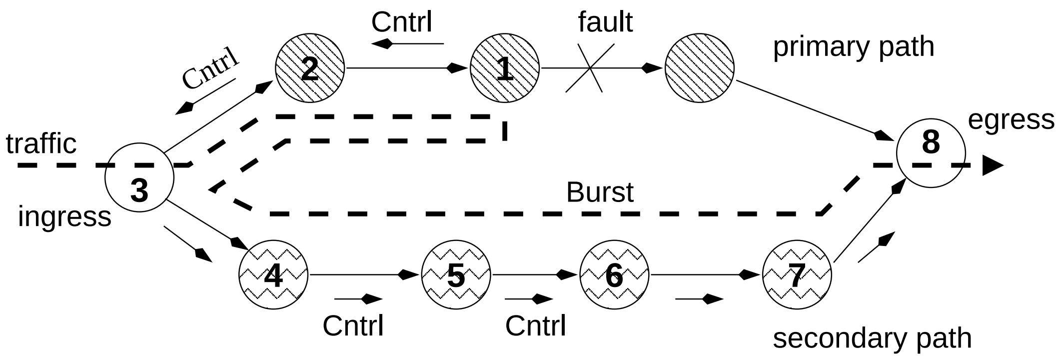


Figure 5: Serial OBS activation architecture.

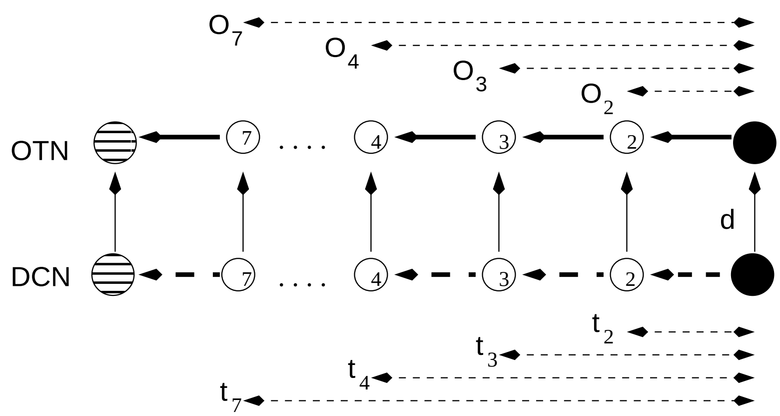


Figure 6: Calculating $t_{setup}$ for serial OBS-based activation.

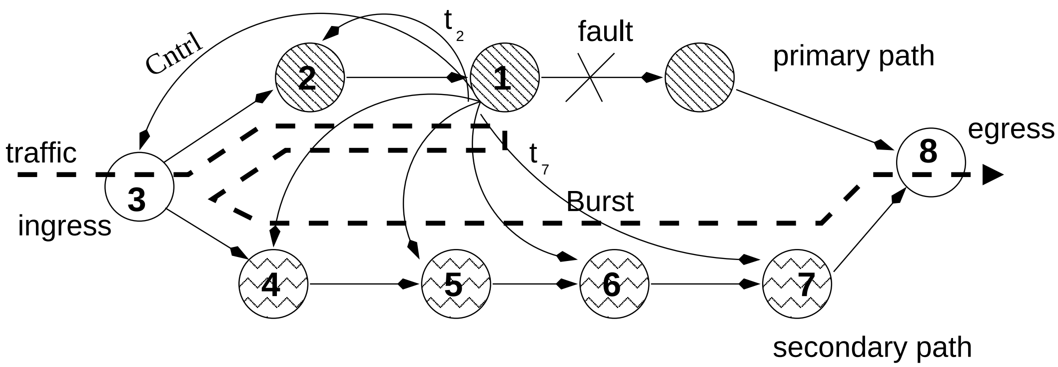


Figure 7: Parallel OBS activation architecture.

where $d$ is the time by which the data burst is required to lag behind the control packet, $O_{node}$ is the time taken by the burst to travel to node *node* through the OTN, and $t_{node}$ is the time taken by the control packet to travel to node *node* through the DCN.

Hence,

$$t_{setup} = MAX\{\forall\ node \in \mathcal{P} : t_{node} - O_{node}\} \tag{15}$$

$$t_{restore} = t_{LOL} + 0 + MAX\{\forall\ node \in \mathcal{P} : t_{node} - O_{node}\} + t_{switch} \tag{16}$$

## 4.2 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. Figure 7 illustrates our proposed scheme. In this hybrid signaling scheme, the node upstream to the failure, 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 than in case of the parallel OBS activation architecture that in the serial OBS activation architecture. This is reflected in the way $t_{setup}$ is estimated for the serial and parallel architectures. We now describe the simulations carried out in order to compare the performances of the four schemes discussed above.

$$t_{detect} = t_{LOL} \tag{17}$$

$$t_{FIS} = 0 \tag{18}$$

From Figure 7 it is clear that the alternate path for this fast-reroute scheme begins at the failed-headend, goes through the ingress and ends at the egress. Thus, if alternate path:

$$\mathcal{P} = \{\text{nodesInPath(failed headend} \rightarrow \text{ingress} \rightarrow \text{egress)}\}$$

then as per Figure 8, in order to guarantee that the data burst always follows the control packet, the following constraint needs to be satisfied:

$$\forall\ node \in \mathcal{P} : d + O_{node} > t_{node}$$

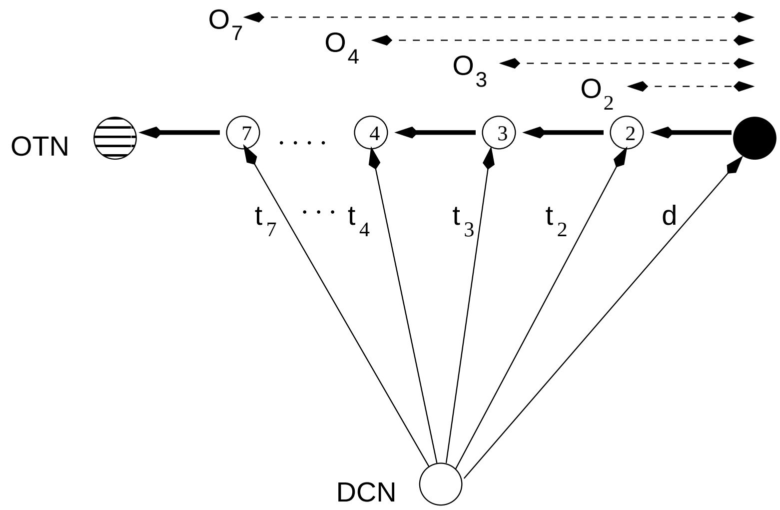


Figure 8: Calculating $t_{setup}$ for parallel OBS-based activation.

where $d$ is the time by which the data burst is required to lag behind the control packet, $O_{node}$ is the time taken by the burst to travel to node *node* through the OTN, and $t_{node}$ is the time taken by the control packet to travel to node *node* through the DCN.

Hence,

$$t_{setup} = MAX\{\forall\ node \in \mathcal{P} : t_{node} - O_{node}\} \tag{19}$$

$$t_{restore} = t_{LOL} + 0 + MAX\{\forall\ node \in \mathcal{P} : t_{node} - O_{node}\} + t_{switch} \tag{20}$$

# 5 Simulation Experiments

Simulations were carried out in order to study the effect of each of the activation schemes on the restoration time. A simulation run consisted of measuring the average restoration time for each of the four activation schemes for a given OTN and DCN network pair. The Georgia Tech Internet Topology Modeler (`gt-itm`) was used in order to generate pure-random, 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]. Figure 9 shows an example of an OTN/DCN pair generated using `gt-itm`[8].

The `gt-itm` package offers ways to control number of nodes, edge probability and geographical network size for the models 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 which would serve as 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[9] 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 stated otherwise).

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 $\frac{(n-1)\cdot n}{2}$ possible shortest paths in an $n$-node OTN were considered to be primary paths. All $\frac{(n-1)\cdot n}{2}$ possible node- & link-disjoint second-shortest paths in an $n$-node OTN were considered to be secondary or alternate paths.

[8]The Walrus Graph Visualization tool from CAIDA.org was used to for visualizing graphs.

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

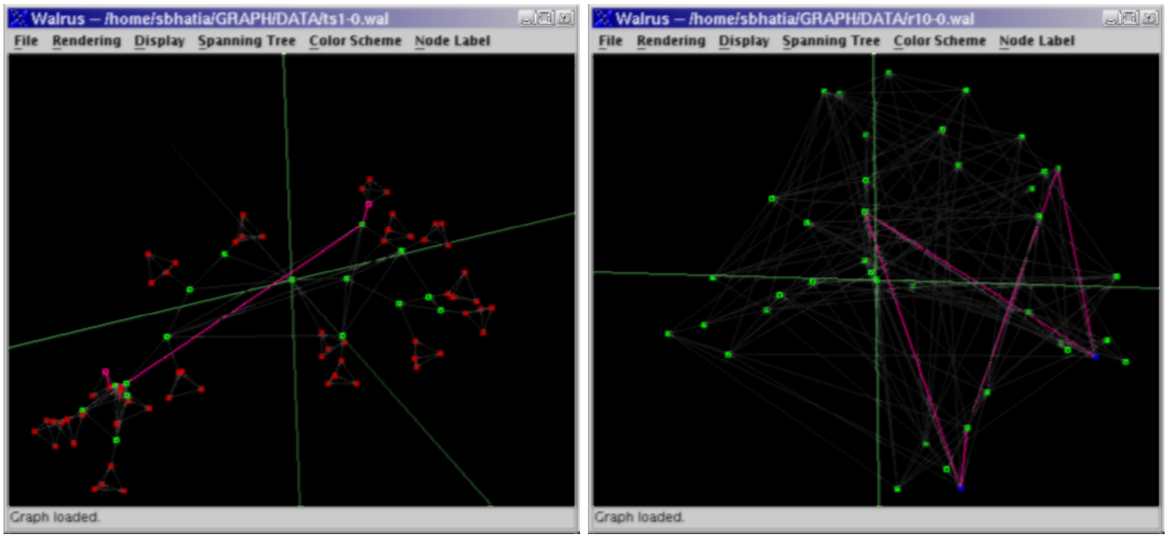


Figure 9: Random DCN (left) & OTN graphs.

| Parameter | Value |
|---|---|
| $t_{proc}$ | 20ms |
| $t_{LOL}$ | 5ms |
| $t_{switch}$ | 10ms |
| $t_{prop}$ | 8$\mu$s/mile |
| $t_{xconnect}$ | 10ms |

Table 4: Constant parameter values selected.

Each data point represents the mean of the time to restore traffic from a primary path (shortest path) to its node- & link-disjoint secondary (second-shortest) path. The Stanford GraphBase [13] library was used for writing routines to perform measurements. Table 4 shows the typical values of various parameters used in the calculation of restoration time as suggested in [4].

## 5.1 Mean Restoration time v/s OTN & DCN Size

In Experiment 1, DCN and OTN were expanded two-fold[10] at every run. The pure-random model was used for generating edges with edge probabilities 0.23[11] 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 10 shows the results of Experiment 1. The figure 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, serial and parallel OBS-based schemes outperform their non-OBS equivalents.

## 5.2 Mean Restoration Time v/s 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 11 shows the results of Experiment 2. It shows the performance of each of

[10] 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.

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

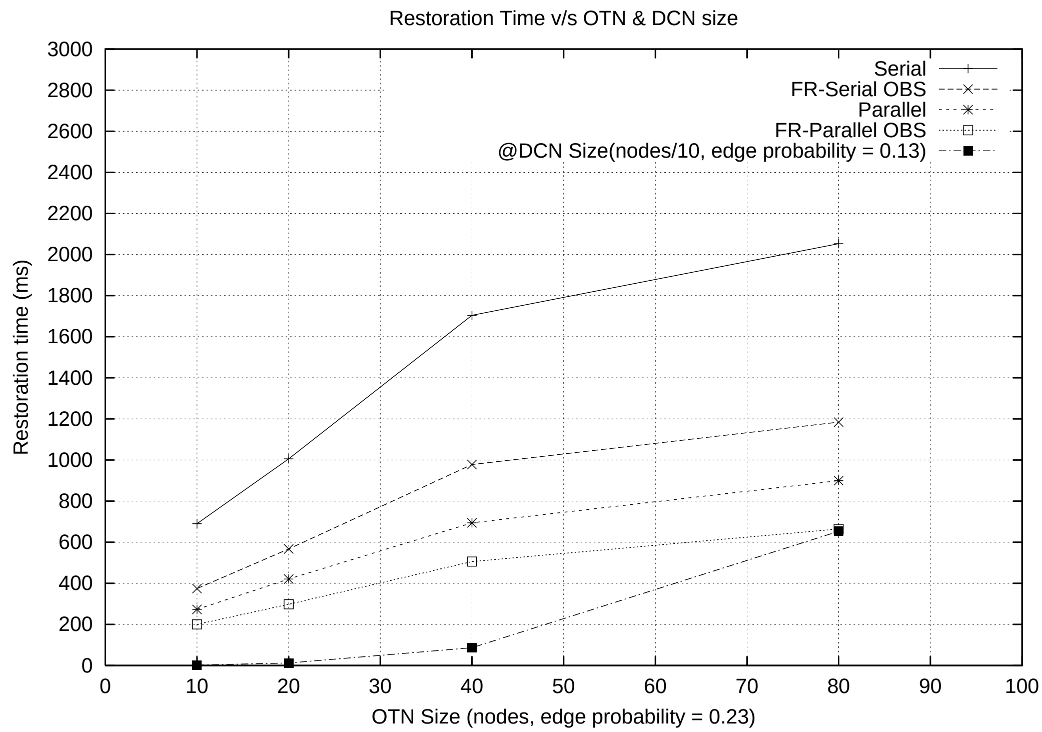


Figure 10: Restoration time v/s OTN & DCN size.

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.

## 5.3 Mean Restoration Time v/s 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 and DCN respectively. Figure 12 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 12, parallel OBS activation scheme exhibits the best performance as network size increases, followed by plain parallel activation.

## 5.4 Mean Restoration Time v/s 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 13, parallel activation schemes exhibit stable scaling performance as compared to serial schemes. Again, OBS-based schemes outperform their non-OBS counterparts.

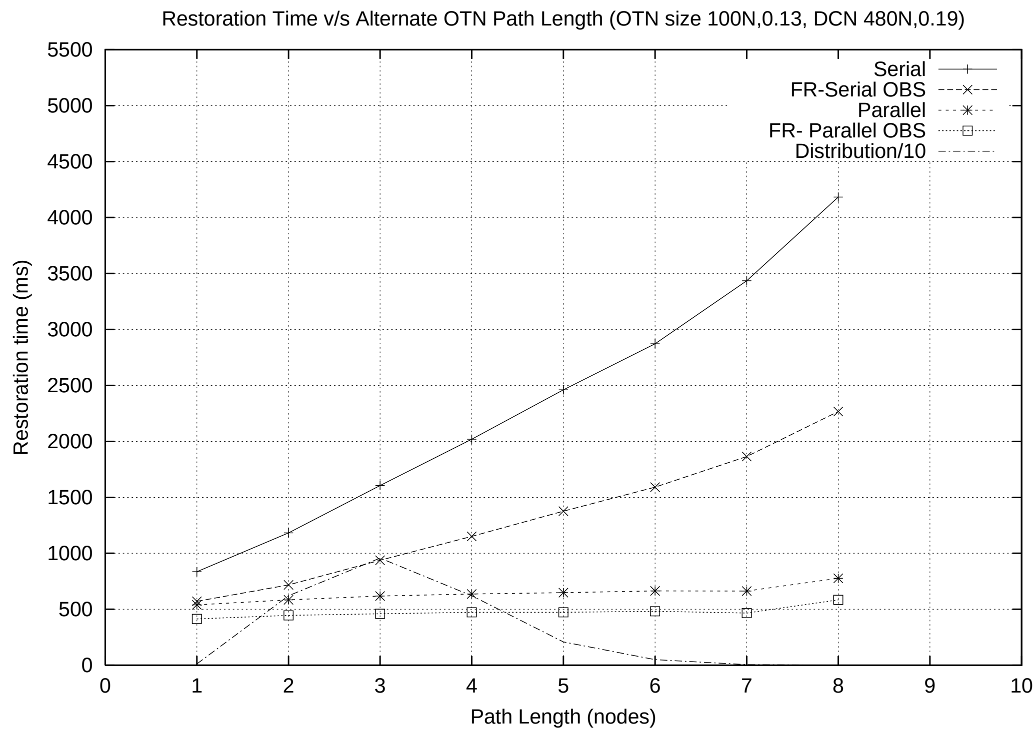


Figure 11: Restoration time v/s alternate path length.

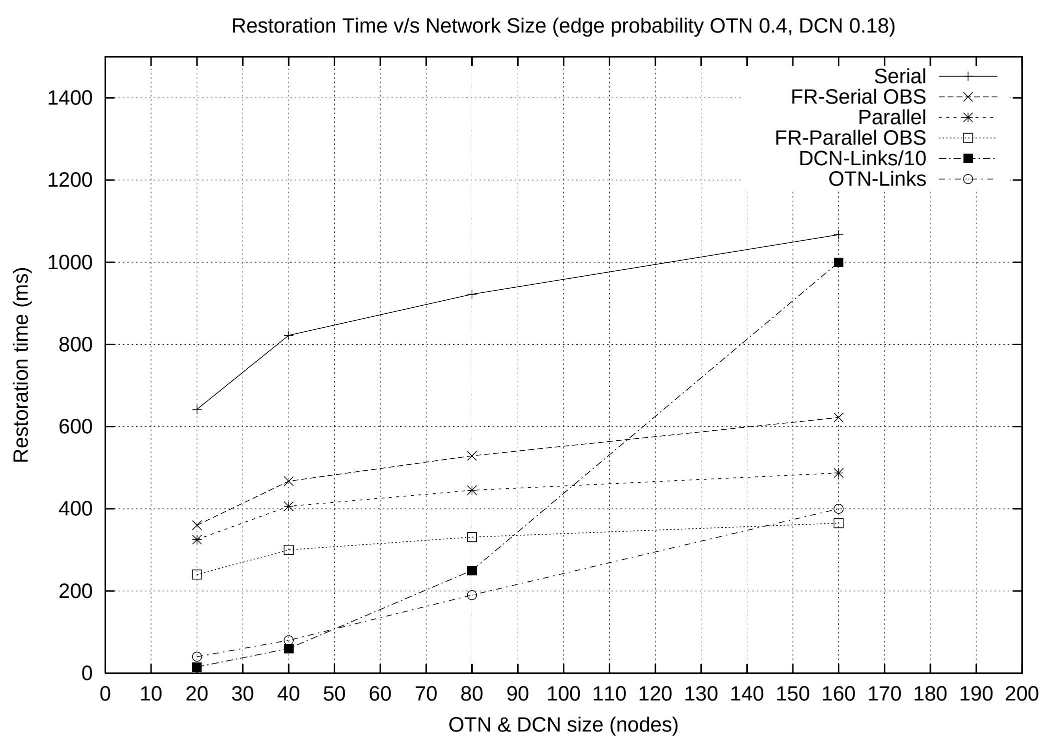


Figure 12: Restoration time v/s network size.

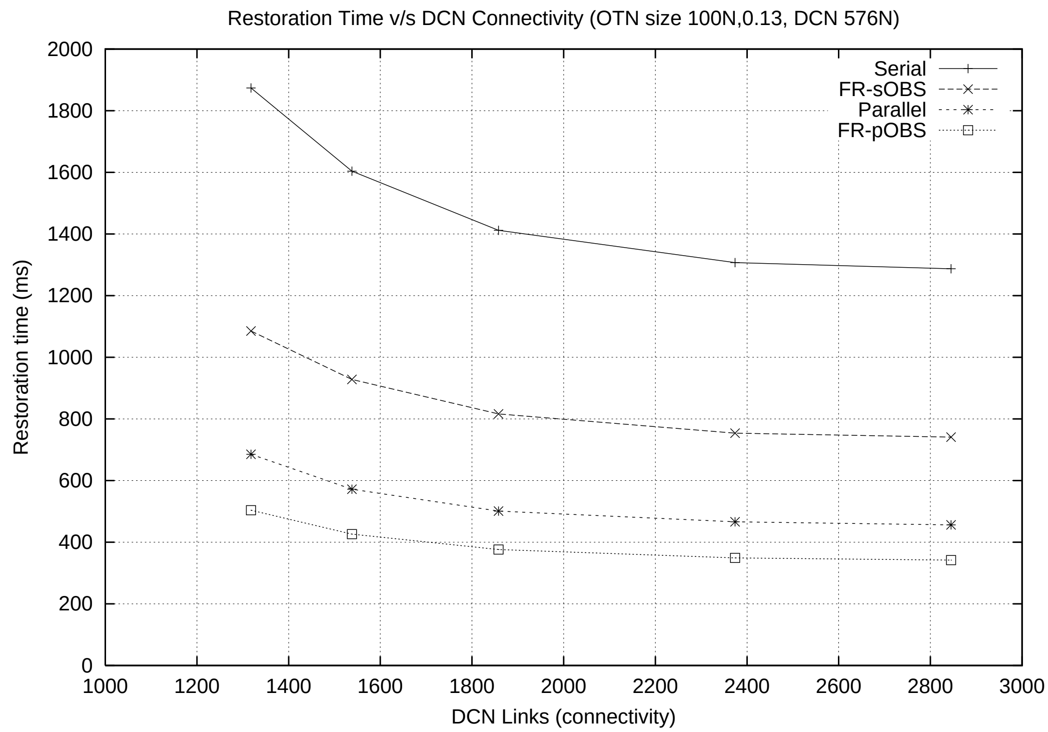


Figure 13: Restoration time v/s DCN connectivity.

# 6 Conclusion & Future Work

Initial results obtained from simulations certainly suggest that OBS based activation schemes can play an important role in minimizing packet loss during restoration. OBS based schemes also exhibit resilient scaling behavior benefiting from the natural structure of DCN internetworks. This property can be further leveraged to develop new signaling schemes with even better restoration bounds.

Many of the trade-offs mentioned in the earlier sections need further investigation in order to further minimize restoration time. Fast route computation may be possible by assigning special restoration sites within the OTN. Pre-wiring may further help reduce setup time as touched upon in [4]. This paper does not address the problem of route computation which is essential to the recovery cycle. New routing algorithms borrowed from wireless networks may allow computation of routes along with setup after a fault actually occurs, thus optimizing a maximal set of metrics. Thus, many avenues still remain open to be explored in the area of restoration schemes.

# 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] V. Sharma and F. Hellstrand, "Framework for MPLS-based Recovery," Tech. Rep. draft-ietf-mpls-recovery-frmwrk-05, IETF, May 2002.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[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.