Signaling for Fast Restoration in Heterogeneous Optical Mesh Networks

Bala Rajagopalan[a], Debanjan Saha[a], Greg Bernstien[b], Vishal Sharma[c]

[a]Tellium, Inc., USA
[b]Ciena Corp., USA
[c]Metanoia, Inc., USA
(Contact: braja@tellium.com)

ABSTRACT

With the advent of optical mesh networks, certain new protection schemes have been defined. This encompasses both local span and end-to-end path protection. But the implementations of these protection schemes have so far been based on proprietary mechanisms developed by each vendor. This has made it virtually impractical to construct a heterogeneous network with interoperable mesh protection schemes. Also, while the notion of a standard IP-centric control plane for optical networks based on Generalized Multi-Protocol Label Switching (GMPLS) has gained wide acceptance, the work in this area has so far focussed exclusively on connection provisioning rather than restoration. This paper defines standard, IP-based signaling protocols for restoration in optical mesh networks. These protocols focus on a new local span protection mode and end-to-end shared protection. The main requirements on these protocols are simplicity and speed. The signaling mechanisms described in this paper are complimentary to the GMPLS provisioning mechanisms.

1. INTRODUCTION

Restoration of switched connections under tight time constraints is a challenging problem in optical mesh networks. Such a network consists of optical cross-connects (OXCs) connected in a general topology [1]. Restoration typically involves the activation of an alternate path for a connection when a failure is encountered in the primary path. The ingress and egress port remain the same for primary and alternate paths, but the rest of the paths are typically resource-disjoint (e.g., node or link disjoint).

The manner in which restoration is accomplished depends on the type of protection afforded to the connection. In this regard, protection can be “local span” or “end-to-end”. Local span protection refers to the protection of the connection segment between two neighboring switches. End-to-end protection refers to the protection of the entire connection from the ingress to the egress port. A connection may be subject to both local span protection (for each of its segments) and end-to-end protection (when local protection does not succeed). Under local span and end-to-end protection schemes, it may be required that when a failure affects any one direction of the connection, both directions of the connection are switched to a new link or path, respectively. This paper describes signaling for the following two protection modes:

N:M local span protection with pooled protection links: Under this mode, a total of N links are assigned to protect a total of M other links between two adjacent OXCs A and B, where M+N is the total number of links between A and B. The value of M and N is pre-configured, and the specific links in the set of N protection links may also be pre-configured. Since N < M, it is possible that not all failed links in the set of M links may be protected from the same failure event. Note that this protection mode is different from (and more flexible than) typical protection groups where the protection links are pre-assigned as backup to specific working links. This sort of protection requires a higher layer signaling protocol, as defined in this paper.

End-to-end shared path protection: Suppose a connection's primary (bi-directional) path is from an ingress port in OXC A to an egress port in OXC B over a set of intermediate OXCs. A resource-disjoint alternate path is pre-assigned to protect the connection. But the links along the alternate path are shared among multiple connections being protected. In this case, the links are allocated in real-time for one of the protected connections whose primary path is affected by a failure. If more than one such connection is concurrently affected by a failure, only one of them will be allowed to use a shared link.

The main requirements on the signaling protocols described in this paper are simplicity and speed. The latency requirement on switching to protection paths is typically specified in tens to hundreds of milliseconds, the performance depending on the number of hops involved. It is clear that a protocol level specification itself cannot guarantee these performance numbers, since a lot depends on the system architecture of the OXCs that implement the protocols. Indeed, this is the reason why these protocols are presently being implemented in a proprietary manner. It is, however, a reasonable goal to aim for a lightweight protocol mechanism that has a good chance of achieving the target performance. This is precisely the objective of this paper. Given the fact that IP-centric (GMPLS) protocols have been widely accepted for provisioning of primary and back-up paths, it is natural to consider standard, IP-based restoration protocols to move away from proprietary restoration solutions.

In the next section, the basic signaling mechanism proposed for fast restoration is outlined. Sections 3 and 4 describe local span and end-to-end protection protocols in detail. Section 5 presents some discussion items and Section 6 presents the conclusions.

2. SIGNALING MECHANISM

The signaling protocols for local and end-to-end protection rely on a small set of short messages. A question that arises is whether existing RSVP-TE or CR-LDP protocols may be extended to support restoration signaling in the same way they are used for provisioning. In our view, restoration signaling must be given the highest priority and presently, there is no means to give relative priorities to different RSVP or CR-LDP messages (if used for both provisioning and restoration). Indeed, even if these protocols are extended for restoration signaling, they will in essence implement a logically distinct protocol framework for restoration as compared to provisioning. In this sense, it seems best to start with a new protocol dedicated for restoration signaling. This is indeed the approach of this paper.

The main aspects of the proposed signaling mechanism are as follows:

- The signaling mechanism consists of new protocol running directly on top of IP;
- The signaling mechanism supports a small set of messages, each of which is succinct, transported in a single short IP packet;
- The signaling mechanism supports reliable delivery via a simple retransmission mechanism;
- Signaling messages are sent over the IP control channel between neighboring OXCs. This control channel can be in-band or out-of-band.

The general format of signaling messages under this proposal is shown in Figure 1 below:

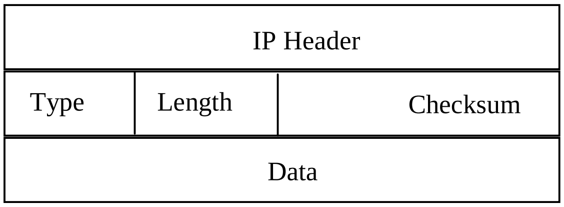

Figure 1: Signaling Message Format

The source and destination address in the IP header are set to those of the sending OXC and the (neighboring) receiving OXC. The protocol ID in the IP header is set to the (yet unassigned) new value corresponding to the restoration protocol.

The restoration message types and data are as defined in the following sections. The 16-bit checksum field covers the entire restoration message, starting from the type field (8 bits). The Length field (8 bits) indicates the number of 4-octet words in the message, starting with the Type field.

3. LOCAL SPAN PROTECTION

Local span protection is described with respect to two neighboring OXCs A and B. The following scenario is considered for local span protection:

- At any point in time, there are two sets of links between A and B, i.e., a *working* set of M (bi-directional) links carrying traffic subject to protection and a *protection* set of N (bi-directional) links. There is no apriori relationship between the two sets of links, but the value of M and N are pre-configured.
- When a link in the working set is affected by a failure, the traffic on it is diverted to a link in the protection set, if such a link is available. Note that such traffic might consist of more than one connection, for example, an OC-192 link carrying four STS-48 connections.
- More than one link in the working set may be affected by the same failure event. In this case, there may not be an adequate number of protection links to accommodate all of the affected traffic.
- Each OXC is assumed to have the mapping of its local link (or port) ID to the corresponding ID at the neighbor. This mapping could be configured, or obtained automatically using a neighbor discovery procedure (e.g., LMP [2]).
- When traffic must be diverted from a failed link in the working set to a protection link, the decision as to which protection link is chosen is always made by one of the OXCs, A or B. The OXC with the numerically larger IP address is considered the *master* and it is required to select specific protection links to divert working traffic. The other OXC is considered the *slave*.
- Failure events are assumed to be detected by lower layer mechanisms (e.g., SONET). Since bi-directional links are typically formed by a pair of unidirectional links, a failure in the direction from A to B is typically detected by B, and a failure in the opposite direction is detected by A. It is possible that a failure simultaneously affects both directions of the bi-directional link. In this case, A and B will concurrently detect failures, in the B-to-A direction and in the A-to-B direction, respectively.

The basic steps in local span protection are as follows:

1. If the master detects a failure of a working link, it autonomously invokes a process to allocate a protection link to the affected traffic.
2. If the slave detects a failure of a working link, it must inform the master of the failure. The master then invokes the same procedure as above to allocate a protection link.
3. Once the master has determined the identity of the protection link, it indicates this to the slave and requests the switchover of the traffic.
4. The slave sends an acknowledgement to the master. Prior to this, it starts sending the (failed) working link traffic on the selected protection link.
5. When the master receives the acknowledgement, it starts sending and receiving the (failed) working link traffic over the new link.

From the description above, it is clear that local span restoration requires three messages for each working link being switched: a failure indication message, a switchover request message and a switchover response message. The following identifier is also needed:

Link ID: A 32-bit identifier that uniquely identifies a bi-directional link at the sending and the receiving OXC. The link ID, for instance, could be the port ID at the receiving OXC.

The messages used are as follows.

1. Failure Indication Message

This message is sent from the slave to the master to indicate the failure of one or more working links. (This message may not be necessary when the underlying link technology itself provides for such a notification). The format of this message is as follows (Figure 2):

Type=1	Length	Checksum
ID of one or more failed links		

Figure 2: Failure Indication Message

The number of links included in the message would depend on the number of failures detected within a window of time by the sending OXC. An OXC may choose to send separate failure indication messages in the interest of completing the restoration for a given link within an implementation-dependent time constraint. Also, the ID of the failed link is the identification used at the slave OXC. The master must convert this to the corresponding ID at its side.

This message is transmitted periodically by the slave, as controlled by the configurable timer, *Retransmission Timer*. The default value of this timer is 30ms. The slave also starts a *Failure Indication Timer* when it sends the message the first time. The slave stops transmitting this message under the following conditions:

- A corresponding Switchover Request message is received;
- The Failure Indication timer expires (the default value for this timer is 500 ms); or
- The connection is de-provisioned.

2. Switchover Request Message

This message is sent from the master to the slave to indicate whether the traffic on the failed working link can be switched to a protection link, and if so, the ID of the protection link. The format of this message is as follows (Figure 3):

Type=2	Length	Checksum
Message ID		
ID of failed link		
ID of protection link		
⋮		⋮
ID of failed link		
ID of protection link		

Figure 3: Switchover Request Message

The link IDs are based on the identification used at the master. The slave must convert them to the corresponding local IDs. The message ID uniquely identifies the message at the master. If the message is generated in response to a Failure Indication message from the slave then the set of failed links in the message MUST be the same as the set received in the Failure Indication message. If the ID of the protection link is the same as the ID of the failed link then it is implicit that the traffic on the failed link cannot be switched to a protection link.

This message is transmitted periodically by the master, as controlled by the Retransmission Timer. Each retransmitted message has the same content, including the Message ID. The master also starts the *Switchover Timer* when it sends this message the first time. The master stops transmitting this message under the following circumstances:

- The corresponding Switchover Response is received;

- The Switchover Timer expires (the default value for this timer is 300 ms); or
- The connection is de-provisioned.

A failure event may result in the master sending more than one Switchover Request message to the same slave OXC. In this case, each of these messages will have different Message IDs and indicate different failed links. The retransmission of each is controlled by a different instance of the Retransmission Timer.

3. Switchover Response Message

This message is sent from the slave to the master to indicate the receipt of the Switchover Request message. The format of this message is as shown in Figure 4:

Type=3	Length	Checksum
Received Message ID		
ID of zero or more failed links		

Figure 4: Switchover Response Message

The Received Message ID field is taken from the corresponding Switchover Request message. ID of failed links, if present, indicates that the slave cannot switch over to the corresponding protection links for some reason. This identification is based on the IDs used at the slave. The action to be taken by the master when there are one or more failed link IDs in the Switchover Response message is undefined (for example, the master may abort the switchover of the traffic on the failed working link, and perhaps trigger end-to-end protection). The slave responds with the appropriate Switchover Response each time it receives a Switchover Request Message (including retransmitted Switchover Request messages).

4. END-TO-END PROTECTION

One of the significant differences between end-to-end protection and local span protection is that the latter is on a per-link basis while the former is on a per-connection basis. In other words, span protection switches over the entire traffic on a link which may consist of multiple connections. End-to-end protection, on the other hand, switches over individual connections. In this case, there is a *working* connection path and a *protection* path.

Another difference between end-to-end and local protection is that signaling messages may have to be transmitted multiple hops to effect restoration. The signaling messages are transmitted along the connection path, working or protection, where it is assumed that there is a control channel between each pair of intermediate OXCs.

The description below for end-to-end shared protection is in the context of a connection between a source OXC A and a destination OXC B. The source for such a connection is also considered to be the *owner* of the connection.

1. Shared Protection

Shared protection is illustrated in Figure 5. It requires prior soft-reservation of capacity along the protection path. Furthermore, after a failure event, the protection path must be explicitly activated for the failed connection. This requires actions at each intermediate OXC along the protection path. Shared protection will be successful only under single-failure scenarios, i.e., at most one of the connections sharing the protection path fails at one time. It is therefore possible that a protection path may not be successfully activated when multiple, concurrent failure events occur. In this case, shared protection capacity may be claimed for more than one failed connection and the protection path can be activated only for one of them (at most).

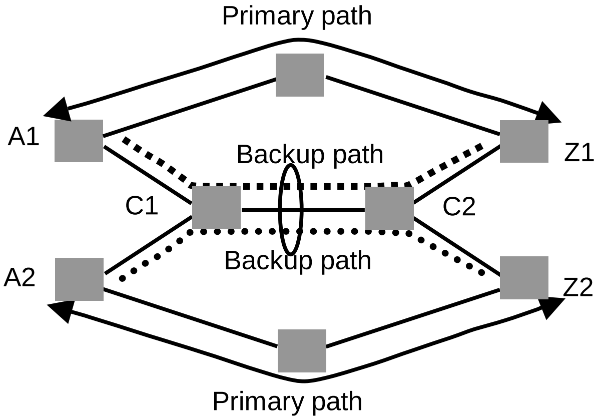

Figure 4: Shared Protection - Primary paths A1-Z1 and A2-Z2 are resource-disjoint and share a channel on link C1-C2 on their backup paths.

2. Identifiers

The following identifiers are used in the signaling protocol.

Source ID: The IPv4 address of the source OXC (connection owner). This is assumed to be unique for OXCs within the scope of the restoration domain.

Connection index: A 32-bit identifier that uniquely identifies a connection at the source.

Together, the Source ID and the Connection index identify a connection uniquely within the scope of the restoration domain. This combination is referred to as the *Connection ID* for convenience.

3. OXC Data Structure

Each OXC that is the working or protection path of a connection must maintain a *neighbor table* whose entries consist of the following information:

< *Connection ID* (*source ID, connection index*), *Previous OXC, Next OXC*>

The Previous and Next OXC fields are 32-bit IPv4 addresses of neighboring OXCs along the connection path. These are the same addresses used in the IP header of restoration messages sent to these neighbors. At the source and destination OXCs, the Previous and Next OXC values are both set to indicate the same neighbor towards the other endpoint of the connection.

The neighbor table may be built when the working and protection paths of the connections are provisioned using signaling, or may be configured in the case of management-system-based provisioning. The neighbor table entries must remain until the connection is explicitly de-provisioned.

In addition to the neighbor table, each OXC in the protection path must also keep information needed to establish the data plane during restoration. This information is maintained in a *connection table*, whose entries indicate the cross-connect that must be established to activate the protection path for each connection, as follows:

{ *Connection ID*, < *Incoming Port, Channel, etc.* >, < *Outgoing Port, Channel, etc.* > }

The precise nature of the Port, Channel, etc. information would depend on the type of OXC and connection (see the Generalized MPLS signaling paper, which describes different type of switches [3]).

4. Messages: End-to-End Failure Indication

This message is sent by an intermediate node towards the source of a connection along the working path of the connection. For instance, such a node might have attempted local span protection and failed (this message may not be necessary if the lower layer provides mechanisms for detection of connection failure by the endpoints). The format of this message is as shown in Figure 5:

<table>
<tr><td>Type=4</td><td>Length</td><td>Checksum</td></tr>
<tr><td colspan="3">Connection ID (Source ID)</td></tr>
<tr><td colspan="3">Connection ID (Connection Index)</td></tr>
</table>

Figure 5: End-to-End Failure Indication

Consider an OXC detecting a link failure. Suppose the failed span is to a neighbor whose identity is *N*. The OXC retrieves all neighbor table entries where *N* is the Previous or Next OXC. Each such table entry indicates the affected connection, and the above message is generated for each connection. The message is sent in an IP packet with the source address set to the address of the sending OXC and the destination address set to the Previous OXC.

This message is transmitted periodically by the OXC, as controlled by the configurable parameter, *End-to-End Retransmission Timer*. The default value of this parameter is 90ms. The OXC also starts the *End-to-End Failure Indication Timer* when it sends the message the first time. The OXC stops transmitting this message under the following conditions:

- A corresponding Failure Acknowledge message is received;
- The End-to-End Failure Indication Timer expires (the default value for this timer is 1 minute); or
- The connection is de-provisioned.

Each OXC in the working path receiving such a message does the following.

- If it is the source of the indicated connection, it stops propagating the message further and generates an acknowledgement message (see below).
- If it also has originated a failure indication message for the same connection, it stops re-transmitting such a message. (This action is needed when a bi-directional data link failure occurs between two OXCs, say, A and B. Suppose A is upstream from B (in relation to the connection source). Also suppose that A has generated a failure indication towards the source. If it later receives a failure indication generated by B, it must suppress the retransmission of failure indication messages it originated. Also, it must forward failure indication acknowledge messages received from the source to B (see below)).
- Otherwise, it looks up the neighbor table entry corresponding to the Connection ID:
  - ➢ If there is no table entry, the received message is silently discarded.
  - ➢ Otherwise, the Source Address in the IP header of the message is compared with the Next OXC field: If there is a match then the received message is sent in an IP packet to the Previous OXC, with the Source Address set to its own address and the Destination Address set to Previous OXC. If there is no match, the received message is silently discarded.

5. End-to-End Failure Acknowledge Message

This message is sent by the source OXC in response to an End-to-End failure indication message. This message is sent towards the originator of the failure indication message along the working path of the connection. The format of this message is shown in Figure 6:

<table>
<tr><td>Type=5</td><td>Length</td><td>Checksum</td></tr>
<tr><td colspan="3">Connection ID (Source ID)</td></tr>
<tr><td colspan="3">Connection ID (Connection Index)</td></tr>
</table>

Figure 6: End-to-End Failure Ack

This message is transmitted in response to each End-to-End Failure Indication message. The source OXC always sends this message to the Next OXC as found in the neighbor table entry corresponding to the Connection ID.

Each intermediate OXC receiving such a message does the following.

- If it is the originator of the corresponding Failure Indication message, it stops propagating the message further.
- Otherwise, it looks up the neighbor table entry corresponding to the Connection ID:
  - If there is no table entry, the received message is silently discarded.
  - Otherwise, the Source Address in the IP header of the message is compared with the Previous OXC field: If there is a match then the received message is sent in an IP packet to the Next OXC, with the Source Address set to its own address and the Destination Address set to Next OXC. If there is no match, the received message is silently discarded.

6. End-to-End Switchover Request

This message is generated by the source OXC receiving an indication of failure in a connection. It is sent to the Next OXC in the protection path. The format of this message is shown in Figure 7:

<table>
<tr><td>Type=6</td><td>Length</td><td>Checksum</td></tr>
<tr><td colspan="3">Connection ID (Source ID)</td></tr>
<tr><td colspan="3">Connection ID (Connection Index)</td></tr>
<tr><td colspan="3">Status</td></tr>
</table>

Figure 7: End-to-End Switchover Request

The Status field is set to 0x01 and the message is transmitted periodically by the source OXC, as controlled by the End-to-End Retransmission timer. The End-to-End Switchover Timer is also started the first time the message is sent. The OXC stops transmitting this message under the following conditions:

- A corresponding End-to-End Switchover Response message is received;
- The End-to-End Switchover Timer expires (the default value for this is 1 second); or
- The connection is explicitly de-provisioned.

Each intermediate OXC receiving such a message does the following. It first looks up the neighbor table entry corresponding to the Connection ID:

- If there is no table entry, the received message is silently discarded.
- Otherwise, the Source Address in the IP header of the message is compared with the Previous OXC field:
  - If there is no match then the received message is silently discarded.
  - Otherwise,
    - The value of the received Status field is checked. If it is 0x00 then the received message is sent in an IP packet to the Next OXC, with the Source Address set to its own address and the Destination Address set to Next OXC.

        - If the received Status field is 0x01 then it is checked if the protection path for the connection can be activated. If so, the message is passed on to the Next OXC as described above.
        - If the received Status field is 0x01, but the protection path for the connection cannot be activated then the message is passed on to the Next OXC as described above, with the Status field set to 0x00.

7. End-to-End Switchover Response

This message is sent by the destination OXC receiving an End-to-End Switchover Request message. It is sent to the previous OXC in the protection path. The format of this message is as follows (Figure 8):

Type=7	Length	Checksum
Connection ID (Source ID)		
Connection ID (Connection Index)		
Status		

Figure 8: End-to-End Switchover Response

This message is transmitted in response to each End-to-End Switchover Request message received. If the Status field was set to 0x00 in the Request message, it is also set to 0x00 in the Response message. Otherwise, the Status field is set to 0x01 or 0x00 depending on whether the destination OXC is able to switchover to the protection path or not, respectively. Each intermediate OXC receiving such a message does the following.

The OXC looks up the neighbor table entry corresponding to the Connection ID:

- If there is no table entry, the received message is silently discarded.
- Otherwise, the Source Address in the IP header of the message is compared with the Next OXC field:
    - If there is no match then the received message is silently discarded.
    - Otherwise,
        - The value of the received Status field is checked. If it is 0x00 then the received message is sent in an IP packet to the Previous OXC, with the Source Address set to its own address and the Destination Address set to Previous OXC. The protection path is not activated in this case.
        - If the received Status field is 0x01 then the protection path for the connection is activated by referring to the connection table entry corresponding to the Connection ID, and the message is passed on to the Previous OXC as described above.

5. DISCUSSION

1. Relationship between Local and End-to-End Protection Procedures

In general, local protection may be attempted before invoking end-to-end protection (see sub-section 6 below on the ramifications of this). The exception to this is when end-to-end 1+1 protection is used for a connection. In this case, it is better to directly invoke end-to-end protection since alternate path resources are already active for the connection. Thus, the general guideline that may be considered is to note the protection type of connections in intermediate OXCs during provisioning, and invoke local span protection only for working links carrying connections that are not 1+1 protected end-to-end. This implies that when a working link carries more than one connection, all the connections must have the same end-to-end protection type. The provisioning process must ensure this. If this is not possible then local span protection may be invoked for working links that have at least one connection that is not end-to-end 1+1 protected.

2. Connection Priorities During Protection

The local protection procedure described in this paper switches all the connections on a failed working link onto a protection link. The advantage of this approach is that the signaling between OXCs is at the level of links and not at the level of connections. This is beneficial if a link could potentially carry many connections. On the other hand, it limits flexibility, since a working link must carry connections of similar priority. Otherwise, it is not possible to ensure that higher priority connections are favored over lower priority connections when a failure event affects more than one working link and there are fewer protection links than the number of failed working links.

Also, under the above failure scenario, a decision must be made as to which working links (and therefore connections) are chosen to be protected and in what priority order. In general, an OXC might detect failures sequentially, i.e., all failed working links may not be detected simultaneously, but only sequentially. In this case, as per the proposed signaling procedures, connections on a working link may be switched over to a given protection link, but another failure (of a working link carrying higher priority connections) may be detected soon afterwards. In this case, the new connections may bump the ones previously switched over the protection link.

In the case of end-to-end shared protection, priorities may be implemented for allocating shared link resources under multiple failure scenarios. Note that shared protection works under the assumption that the primary path of connections whose backups share resources are resource-disjoint [1]. Under single-failure scenarios, this would ensure that exactly one connection will “claim” the allocated (shared) resources. But under multiple failure scenarios, more than one connection can claim shared resources. If such resources are allocated to a lower priority connection, they may have to be reclaimed and allocated to a higher priority connection. Furthermore, the lower priority connection must be de-provisioned along the protection path (this can be done using the signaling mechanisms developed for provisioning, rather than restoration signaling). The proposed signaling mechanisms can support connection-priority based allocation of shared resources during restoration signaling (specifically, during the Switchover Response step).

A way to simplify end-to-end shared protection is to allocate shared resources to connections of the same priority. This way, a connection will not be first allocated shared resources and then bumped from the protection path.

3. Multi-Domain Restoration

When an end-to-end connection follows a long path through multiple routing or administrative domains, it may be required to consider an intermediate form of restoration, called *intra-domain end-to-end restoration*. With this approach, a failure within a domain would result in end-to-end restoration between the connection ingress and egress points within the domain (perhaps after local span restoration is attempted). When this fails, or if a failure occurs in an inter-domain link, full end-to-end restoration will be attempted.

This type of a structured approach for restoration is particularly useful in the near term when an optical internetwork may be constructed by interconnecting multi-vendor optical sub-networks [1]. In this case, intra-domain restoration may be proprietary, with standard restoration signaling implemented between border OXCs. But this type of restoration also requires some hardware support at the border nodes.

4. Optical mesh restoration and MPLS-based recovery

Over the past year or so, there has been considerable work on MPLS-based recovery (see, for example, [2], [4], [5], [6], [7], [8], and [9]). The terminology used in this paper is also explained in the MPLS-recovery framework document [3], in the context of MPLS LSP-based recovery.

The failure indication message of previous sections, is quite similar to the failure indication signal (FIS) defined in [4], and elaborated on in [8] and [9]. A difference between the schemes and message formats discussed in this paper and those presented in [4], [8], and [9], is that these documents focus primarily on MPLS LSP restoration. As such, the messages defined therein contain explicit label information for packet LSPs, which is not required in optical networks.

Further, [4] does not specifically cover the case of the coordinated signaling required for local span protection and for M:N protection with pooled protection links, which are central to this proposal.

5. Implementation Considerations

As described in this paper, restoration signaling does not require any central actions (such as admission control or centralized resource allocation) within an OXC for end-to-end protection. Local span protection may require the consideration of all available protection link resources at the master. But end-to-end protection, which is more difficult from a latency perspective, can be controlled by distributing multiple, independent protocol instances in an OXC such that each instance covers a subset of connections passing through an OXC. Such optimizations would depend on the architecture of the systems implementing the proposed protocol.

6. Performance

The premise of this paper is that lightweight signaling aids towards achieving strict performance goals associated with restoration. The performance of the specific mechanisms proposed in this paper is yet to be evaluated. However, the performance of functionally similar protocols using proprietary messaging over SONET overhead bytes has been extensively evaluated and reported [11]. Some of the performance highlights are:

- The performance was evaluated on a 17-node US backbone network topology. There was a wide range of resource diversity in this network. Source and destination for connections were chosen randomly. The routing and restoration of hundreds of connections were studied.
- The primary and backup paths had an average of 2.55 and 4.25 hops, respectively. The average number of connections sharing a backup link was 2.54.
- For local restoration, the observed latency on the average was 50.8 ms. For end-to-end restoration, the average latency was 60 ms.
- The average latency when a connection is restored end-to-end after local restoration attempt failed was 113 ms. This indicates a problem with the two-level restoration. While local restoration speeds up restoration times when successful, its failure results in considerable deterioration of the latency.

The performace numbers above do not include any latency that may arise due to packet processing when an IP-based protocol is used. It is expected, however, that the additional latency will not be significant. Finally, the latency figures will increase with the number of connection hops. But this number, again, is not expected to be large in a backbone network. Thus, we can reasonably conclude that the proposed mechanisms are likely to be fast.

6. CONCLUSION

In this paper, a signaling mechanism for fast restoration in heterogeneous optical mesh networks was described. The proposed mechanism consisted of a protocol running directly over IP. The proposed messages were short and the interaction amongst nodes was rather simple. The proposal dealt with local span protection and end-to-end protection.

It is likely that the structure of the proposed mechanism would allow restoration to be completed quickly. Clearly, extensive quantitative evaluation is needed before the responsiveness of the proposed mechanism can be established. The evaluation of a functionally similar set of protocols using proprietary messaging has indicated good performance.

The protocols described in this paper have been presented to the IETF to aid in the development of standard restoration protocols [12]. The IETF document covers these proposals in more detail and may be referred to by interested readers.

7. REFERENCES

1. B. Rajagopalan, et al., “IP over Optical Networks: Architectural Aspects,” *IEEE Communication Mag*., September, 2000.
2. J. P. Lang, et al, “Link Management Protocol”, Work in Progress, Internet Draft, draft-ietf-ccamp-lmp-00.txt, July, 2001.
3. P. Ashwood-Smith, et al., “Generalized MPLS: Signaling Functional Specification,” Work in Progress, Internet Draft, draft-ietf-mpls-generalized-signaling-05.txt, July, 2001.
4. V. Sharma, et al, “Framework for MPLS-based Recovery,” Work in Progress, Internet Draft, draft-ietf-mpls-recovery-frmwrk-03.txt, February 2001.
5. K. Owens et al, “A Path Protection/Restoration Mechanism for MPLS Networks,” Work in Progress, Internet Draft, draft-chang-mpls-path-protection-03.txt, July, 2001.
6. Kini, S., et al, “Shared Backup Label Switched Path Restoration,” Work in Progress, Internet Draft, draft-kini-restoration-shared-backup-01.txt, May, 2001.
7. Hellstrand, F., and Andersson, L., “Extensions to CR-LDP and RSVP-TE for setup of pre-established recovery tunnels,” Work in Progress, Internet Draft, draft-hellstrand-recovery-merge-01.txt, November 2000.
8. D. Haskin, Krishnan, R., “A Method for Setting up an Alternative Label Switched Path to Handle Fast Reroute,” Work in Progress, Internet Draft, draft-haskin-mpls-fast-reroute-05.txt, November, 2000.
9. K. Owens et al, “Extensions to RSVP-TE for MPLS Path Protection,” Work in Progress, Internet Draft, draft-chang-mpls-rsvpte-path-protection-ext-02.txt, July, 2001.
10. K. Owens et al, “Extensions to CR-LDP for MPLS Path Protection,” Work in Progress, Internet Draft, draft-owens-mpls-crldp-path-protection-ext-01.txt, November 2000.
11. S. Biswas, S. Datta, and S. Sengupta, “Performance of Restoration Protocols in Optical Mesh Networks,” Proceedings of NFOEC, 2001, Baltimore, USA.
12. B. Rajagopalan, et al., “Signaling for Fast Restoration in Optical Mesh Networks,” Work in Progress, Internet Draft, draft-bala-restoration-signaling-01.txt, August, 2001.