To appear in IEEE Journal on Selected Areas in Communications

Distributed Route Computation Algorithms and Dynamic Provisioning in Intelligent Optical Mesh Networks

Hang Liu, Eric Bouillet, Dimitrios Pendarakis, Nooshin Komaee,Jean-Francois Labourdette, Sid Chaudhuri
Tellium, Inc., 2 Crescent Place, Oceanport, NJ 07757

Abstract

Optical mesh network infrastructures have emerged as the technology of choice for next generation transport networks. At the same time, distributed, IP-based, control architectures have been proposed for intelligent optical networks, as a means to automate operations, enhance interoperability and facilitate the deployment of new applications. While distributed control in general enhances scalability and flexibility, it has also been observed that the requisite topology and link state information summarization may result in sub-optimal path computation, especially for shared mesh restoration paths. This paper presents a distributed control plane for optical mesh networks, focusing on the distributed path computation and provisioning mechanisms. It discusses the tradeoff between path computation efficiency for shared mesh restoration paths and the type of network link state and topology information that is disseminated via distributed routing protocols. We show that with appropriately aggregated link resource availability and sharing information the proposed distributed path computation algorithms are able to determine the shareability of restoration links with remarkable accuracy. A local channel assignment scheme, which is used in conjunction with the distributed path computation to assign shared channels when provisioning the restoration path, is also proposed. The information required to carry in the provisioning signaling messages in order to assign the shared channels at each node along the restoration path is discussed. In addition, we specify the extensions to the OSPF routing protocol that are necessary for disseminating the topology and link state information needed by the distributed path computation. We analyze the performance of path computation algorithms as well as the OSPF extensions. In particular, we study the tradeoffs of network capacity usage, computation complexity of restoration paths, control bandwidth usage for disseminating extended OSPF link state information and the amount of memory required to store the link state database.

1 Introduction

Optical mesh network infrastructures, in which multiple optical cross-connects (OXCs) are interconnected via Dense Wavelength Division Multiplex (DWDM) links in a general mesh topology, have emerged as the technology of choice to implement next generation transport networks [1]. In these architectures, an OXC is capable of switching an optical channel from an input port to an output port. The optical connections (lightpaths) are routed within the OXC network between an ingress port and an egress port to provide services to the clients connected to the optical transport network (OTN).

Optical transport networks have been traditionally controlled from Element Management or Network Management Systems (EMS/NMS). In this mode of operation, most of the functions related to topology discovery and connection provisioning are performed in a central manner, often requiring operator intervention and manual configuration. In this case, connection provisioning includes path computation and establishment of suitable cross-connects in each OXC along the path, so that the end-to-end connectivity is realized. This centralization is in some degree a consequence of the limited control plane intelligence of network equipment (NE) and is related to the mostly static operation of the optical transport network.

Recently, distributed control architectures have been proposed for optical transport networks as a means to automate operations, enhance interoperability and scalability as well as facilitate the deployment of new applications, such as unified traffic engineering [3]. In particular, the ability to provision connections/services dynamically and automatically through a distributed control plane has attracted a lot of interest. Efforts to standardize such a distributed control plane have reached various stages in several bodies such as the Internet Engineering Task Force (IETF) [4], International Telecommunications Union (ITU) [5] and Optical Internetworking Forum (OIF) [6]. The IETF is defining “Generalized MPLS” (GMPLS) which describes the generalization of MPLS protocols to control not only IP router networks but also various circuit switching networks including OXCs, photonic switches, ATM switches, and SONET/SDH systems. GMPLS extends MPLS signaling and Internet routing protocols to facilitate automatic service provisioning and dynamic neighbor and topology discovery across multi-vendor intelligent transport networks, as well as their clients. The OIF has assumed an overlay model of integration between optical transport networks and their clients [3] and defined an implementation agreement for the optical User-to-Network Interface (UNI) based on the GMPLS protocols. It is in the process of defining Network-to-Network Interface (NNI). The ITU has also assumed the overlay model and is defining a distributed control plane for Automatic Switched Optical Network (ASON) [5].

With the application of a distributed control plane, each network node participates in routing protocols that disseminate topology and link state information and is thus able to perform path computation for connection provisioning. In general, connections are signaled using explicit routes, i.e., the connection path is available at the ingress node and is carried in the connection establishment message. Connection requests may be initiated either by the management plane (Console, EMS or NMS) or from client directly connected to the optical network (via a UNI). In the former case, it is possible that the path is computed by the management plane and provided to the ingress node of the connection. However, in our work we focus on a fully distributed path computation model where the ingress node is responsible for computing the explicit route.

Path computation must take into account various requirements and constraints, including bandwidth and delay constraints, recovery and survivability, optimal utilization of network resources. This requires dissemination of information about the network topology and various link attributes to every node in the network using routing protocols. GMPLS has extended traditional IP routing protocols such as OSPF to support explicit path computation and traffic engineering in transport networks [9]. In order to reduce overhead and improve routing scalability, however typically only aggregated link information is disseminated via these routing protocols, leading to loss of information. For example, in the case of core optical switches with hundreds of ports there may be multiple data links between a pair of nodes. Data links between the same pair of nodes, with similar characteristics, can be bundled together and advertised as a single link bundle or a traffic engineering (TE) link [11] into the routing protocol. Therefore, unlike the (centralized) management plane, each network element does not have complete information about the network topology and link state. Clearly, path computation with complete information is expected to achieve higher efficiency compared to the distributed case. The challenge of distributed path computation is, therefore, to disseminate appropriately aggregated information so that computed explicit paths meet all the requirements and constraints and incur minimal penalty in network utilization. As will be shown in the remaining of the paper, the proposed algorithms incur a relatively small penalty, while offering the benefits of distributed path computation, namely faster provisioning and higher flexibility.

2 Protection and Restoration in Optical Mesh Networks

End-to-end path protection and restoration techniques are commonly used in transport networks in order to support high service availability and quick recovery from network failure. Several significant steps are necessary during the path provisioning [12-17, 40]. (1) Computing the path. Note that the recovery path may be computed with the working path during the path provisioning process or may be dynamically computed at the restoration. (2) Provisioning the path with signaling. For a recovery path, it may be fully signaled all the way and its resources (the individual channel or slot) may be fully selected (i.e. allocated) and cross-connected between the ingress and egress nodes. In this case, no signaling takes place to establish the protection path when a failure occurs. Alternatively, the restoration resources may be signaled and reserved a priori, but not cross-connected for the recovery path. The complete establishment of the restoration path occurs only after a failure of the working path, and requires some additional signaling.

Therefore, based on the method of recovery path provisioning and utilization of network resources, the path protection and restoration schemes may be classified to dedicated protection, dynamic restoration and shared mesh restoration. The concept of Shared Risk Link Group (SRLG) has been introduced to select the paths so that they will not be affected by a single failure [18][19]. An SRLG expresses the relationship that associates optical lines (or channels) with a single failure. An SRLG may consist of all the optical lines in a single fiber, or the optical lines through all the fibers wrapped in the same cable, or all the optical lines traversing the same conduit. Since a fiber can traverse several conduits, an optical line may belong to several SRLGs.

In 1+1 dedicated protection, the path computation algorithm computes the working path and its protection path simultaneously. The working path and protection path are selected to guarantee SRLG disjoint, i.e. they don't pass through the same SRLG, so that a single SRLG does not interrupt both of them. Note that it suffices that a working path and its protection path are SRLG disjoint to ensure that at least one path survives any single failure affecting all the optical lines in an SRLG. The protection path is pre-provisioned through signaling protocols such as GMPLS extended RSVP-TE or CR-LDP [20, 21, 22] and is dedicated to protect a working path. During normal operation mode, it is pre-established and carries active traffic, where the egress selects the best copy of the two. This type of schemes offers the shortest recovery time in the event of network failures, but uses more network resources.

In dynamic restoration, no recovery path is pre-computed and pre-provisioned. The network tries to establish the restoration path only after the working path fails. Although it does not require reserving any network capacity for protection and restoration, this type of schemes generally needs long and unpredictable restoration time and does not guarantee successful failure recovery.

In shared mesh restoration [18, 19, 23-25, 39, 40], the path computation algorithm computes the restoration path. The restoration path is pre-provisioned through signaling protocols and its resource is reserved along the path. However no cross-connections are performed along the restoration path. The complete establishment of the restoration path occurs only after the working path fails, and requires some additional signaling. The common restoration resource reserved on a link may be shared by multiple restoration paths to restore multiple working paths. In order to avoid contention for the reserved restoration resource (bandwidth or channel) during a single SRLG failure, two restoration paths may share the common reserved restoration resource only if their respective working paths are mutually SRLG disjoint. One failure then does not disrupt both working paths simultaneously. In shared mesh restoration, the bandwidth reserved for restoration on a link can be smaller than the total bandwidth required by all the working paths recovered by it because the bandwidth is only soft reserved. For example, two OC-48 restoration paths can share one OC-48 channel on one of their common links to recover two OC-48 working paths. Shared mesh restoration achieves more efficient utilization of network resources than dedicated protection by sharing the restoration resource. It can achieve reasonably fast restoration time and guarantees successful failure recovery from a single failure. The resource reserved for restoration can even be used by the other paths to carry the extra traffic during normal operation mode (i.e. while there are no failure on the working paths). However, the restoration path needs to be activated when the working path fails, which involves more processing to signal and establish the cross-connections along the restoration path. It may result in a restoration time longer than the dedicated 1+1 protection. Furthermore, since multiple restoration paths may share the common reserved restoration resource. The contention may occur on the reserved restoration resource when more than one of the working paths fails simultaneously due to multiple failures. Of course, this can be resolved by limiting the maximum number of working paths restored by each reserved channel. As a special case in 1:1 restoration, a reserved channel can restore a maximum of one working path, that is, the same reserved channel can not be shared by multiple restoration paths to recover multiple working paths (however it can be used to carry pre-emptible traffic during normal operation mode).

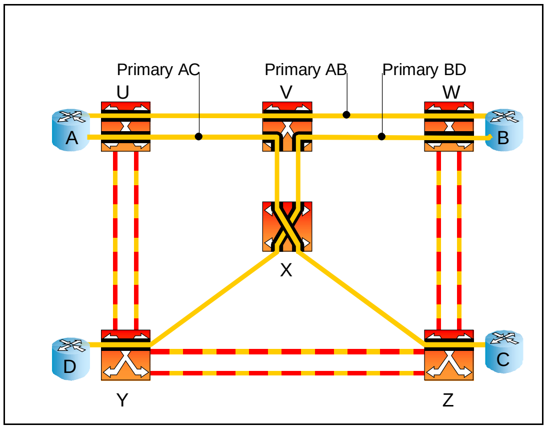


Figure 1. Failure dependent shared mesh restoration

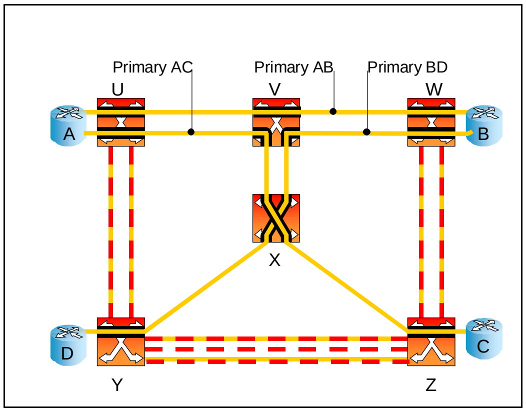


Figure 2. Failure independent shared mesh restoration

For shared mesh restoration, there are two different policies to allot channels to restoration paths [25, 26, 39, 40]: (1) A failure independent strategy assigns the restoration channel at the time of provisioning before failures occur. (2) A failure dependent strategy assigns the restoration channels along a precomputed route after failure occurrence and relies on the protection mechanism to select the channels on each link along the route from a pool of reserved channels [27]. A proper spare channel-provisioning scheme reserves enough channels on each link during provisioning time so that all lightpaths can be restored for every type of single failure. Figure 1 and 2 illustrate the examples of a failure dependent strategy and a failure independent strategy for shared mesh restoration, respectively. The example consists of three demands, routed across a 6-nodes optical network in such a way that every combination of primary lightpath pairs, but not all three primary lightpaths at once, can fail simultaneously. If a failure dependent strategy is used, only two channels need to be reserved on the restoration path, since at most two lightpaths will fail simultaneously. Now, if a failure independent strategy is used, we must reserve three restoration channels on link Y-Z in order to accommodate all failure scenarios along links (U,V), (V,W) or (V,X), even though at most two of the three channels will be used at any time.

Although more capacity-efficient, the second approach requires time-consuming inter-node communication to agree on the channel assignment during restoration, especially when the restoration initiated from both ends of the path. Therefore, we prefer the failure independent approach, which can achieve sub-200ms restoration times in large networks [2]. However the gain in restoration time requires filling up the restoration-to-channel lookup tables at each node during provisioning (when speed is less of an issue.) We assume that the failure independent strategy is used in the remainder of the paper, where the channels on restoration paths are pre-assigned.

In summary, various protection and restoration techniques offer different tradeoffs in term of network resource utilization, restoration time, restoration guarantee under single and/or multiple network failures. The network operators can choose the protection and restoration mechanism based upon the performance requirements of the end user applications, the cost related to the usage of restoration, and Service Level Agreements with the customers, etc. Therefore the path computation algorithms should support various protection and restoration mechanisms. As discussed before, unlike that in centralized NMS systems, the path computation algorithm in the distributed control plane does not have complete network information. In particular, complete restoration resource sharing information of all the links in the network required to compute the recovery path in shared mesh restoration is not available. This imposes additional constraints on the path computation. Two related research issues should be addressed in distributed path computation. (1) New path computation algorithms are able to compute various paths (unprotected, dedicated protection, shared mesh, etc.) to meet the constraints and optimize the utilization of network capability, but only use as little as possible network topology and link state information about resource availability and sharing. (2) Efficient methods to aggregate and disseminate the routing information including optical link resource availability and sharing so that the amount advertised by the routing protocols is minimized but information necessary for efficient path computation is not lost. In general, the more detail information is available, the better results the path computation algorithms can achieve. On the other hand, in order to reduce the amount of information handled by routing protocols and improve the routing scalability, it may be desirable to aggregate the information on a TE link (bundle).

This paper discusses the distributed control plane for intelligent optical mesh networks, especially the algorithms and mechanisms required for computing and provisioning shared mesh restored lightpaths. Section 3 describes the details of the distributed algorithms to compute mesh restored paths and the required input information of link resource availability and sharing. Since the path computation does not provide the specific channels along the restoration path, we discuss in section 4 what information needs to be carried in the signaling message in order to assign the local channel when provisioning the restoration path. A local channel assignment scheme is also proposed to assign the shared channels for optimizing capacity usage in individual links. In section 5, we specify how to aggregate and disseminate the link resource information in support of shared mesh restoration by extending the OSPF routing protocol. In section 6, the performance of path computation algorithms and the extended OSPF is analyzed, including network capacity usage, computation complexity of restoration paths, control bandwidth usage for disseminating extended OSPF link state information, the amount of memory required to store the link state database. Section 7 concludes this paper.

3 Distributed Path Computation Algorithms

The path computation algorithm finds shortest-cost path(s) between the ingress and egress nodes of the connection under certain constraints (e.g. bandwidth, diversity, and delay). For non-protection and 1+1 protection, path computation does not require link resource sharing information of the restoration paths, which significantly reduce the amount of information to be disseminated by routing protocols. Routing protocols such as OSPF can be extended to carry sufficient information so that the same path computation algorithms can be used for both distributed and centralized cases [7, 9]. In addition, the path computation algorithms for unprotected and 1+1 protected paths are very well understood. The standard algorithms can be used to compute the unprotected and 1+1 protected paths, such as Bellman-Ford algorithm, Suurballe algorithm and K-shortest path algorithm [24, 25, 28], which are based on the primitive graph. In the following, we'll focus on the computation algorithms for shared mesh restored paths.

The path computation algorithm for shared mesh restored paths finds a shortest-combined-cost primary path and a diversely routed restoration path between the ingress and egress nodes, so that sharing of channels is maximized. Before describing the distributed path computation algorithms, we consider how the mesh-restored paths are computed with a centralized Path Computation Module (PCM) because the centralized PCM can use the deterministic approach with complete and detailed link resource availability and sharing information to obtain the optimal paths in term of the total network capacity usage [24]. For distributed PCM, the heuristic approach is used to compute the shared mesh restored paths because it is difficult to disseminate complete link resource availability and sharing information to all the nodes in the network as described later. The local database of each node may contain a summarized information that is necessary to compute the routes using the heuristic approach. Since this information is relatively small, it can easily be disseminated by link-state routing protocols, such as OSPF. Using this information each demand's ingress node can compute a path. It is thus expected that the distributed PCM will result in certain performance tradeoff on the network capacity usage because of incomplete link resource sharing information. However it will be shown later that with carefully aggregated link resource information the proposed distributed path computation algorithms are able to achieve the network capacity usage close to that of centralized PCM and the required computation time of the distributed algorithms is much less.

With complete link resource information in centralized PCM, computation for shared mesh restored path is proved to be a NP-complete problem if minimization of the total capacity usage (working and restoration) is sought [24, 26]. A possible approach is then to enumerate a list of K minimum cost working paths and for every one of them compute the corresponding minimum cost restoration path [24]. The PCM returns the pair of paths with the lowest combined cost. The cost of a pair is the cost of the channels along both paths, excluding the cost of (preexisting) shareable reserved channels along the restoration path. Given a working path, we compute the minimum cost restoration path by: (i) setting the cost of the links with SRLGs traversed by the working path to infinite, (ii) setting the cost of links with shareable channels to a constant $\varepsilon \ll 1$, (iii) run a shortest path algorithm (Bellman-Ford algorithm) using the modified link cost metric. Steps (i) and (ii) respectively ensure that working and restoration paths are SRLG-diverse, and that the minimum cost restoration path is found using shareable reserved channels whenever possible.

With the distributed path computation, it does not scale well for large networks in term of required control network bandwidth and processing resource if the complete link resource availability and sharing information is distributed to every node. It is desirable that the path computation algorithms use the input information as little as possible to compute the paths, with no penalty or small penalty in terms of capacity efficiency. Note that in Step (ii) of the above restoration path computation algorithm the shareable reserved channels are identified deterministically by examining the protected SRLGs of all sharable channels. If we apply a heuristic approach to execute this operation with a certain probability of accuracy, i.e. assigning the cost based upon the probability that there are sharable channels on a TE link between two particular nodes, it will reduce the amount of input information necessary to compute the paths. In addition, it has been shown the time-complexity of this operation, if deterministic, is proportional to the total number of reserved channels, and thus does not scale well when the number of lightpaths established in the network becomes large [28]. On the other hand, the heuristic approach is independent of the number of reserved channels, and the processing power required for path computation is less. In the following, we'll describe the distributed path computation algorithms using the heuristic approach.

Given: a topology represented as a graph G(V,E) where vertices represent optical cross-connects (OXC) and links represent TE links between OXCs. A network state database, which indicates the aggregated information about the resource availability and sharing on each TE link.

Input: a pair of nodes A and Z, data rate

Output: a pair of paths from A to Z, working and restoration with minimum cost

Method:

1. Compute k-shortest paths (potential candidates for the working paths). Sort the paths by cost and denominate them w_1 to w_k.
2. Set the candidate path set S=0
3. For a shortest path w_i, assign the cost to each TE link, that is a function of default link cost, shareability, and resource availability on the TE link. Note that the link cost function results in different algorithms with different requirement for the knowledge of aggregated network state information, as discussed below.
4. Compute the shortest path s_i (potential candidates for the restoration paths) using the cost metric defined in 3 and set S =S+{w_i, s_i}
5. Repeat step 3 and 4 for each of k paths
6. Select the minimum cost path pair {w_k,s_k}. If no path can be found, return NO_PATH.

Let's consider step 3 in the above algorithm, different link cost function in restoration path computation can be used and they require different input information of network states. We describe several algorithms as follows. Let UC (unassigned channel) denote the channels that are available and have not been assigned to any path, RC (Reserved Channel) denote the channels that have been reserved for at least one shared restoration paths but are still idle (no restoration occurs); D_{cost} denotes the default link cost for this TE link. The following three algorithms are considered for determining the restoration routing graph given a working path for a mesh-restored path in step 3:

Algorithm 1 :

(a) To each TE link that shares an SRLG with the working path, assign infinite cost, i.e. *cost = infinite*
(b) For each remaining TE link, set the cost based on #UC, D_{cost}, i.e. ***Cost = f(#UC, D_{cost})***. A simple function can be expressed as ***f(#UC, Dcost) =***
  - D_{cost} if #UC>0
  - infinite if #UC=0

Algorithm 1 uses only the number of UC channels during the path computation and it can guarantee a channel for the restoration path. As a matter of fact, this is the same algorithm as used for computing 1+1 protection path (it is called shortest disjoint path algorithm because computing a pair of shortest disjoint paths based on the topology information without trying to share existing reserved channels) [24]. Actual resource sharing on the TE links along the restoration path can be achieved during provisioning by local channel assignment mechanism. It is because unlike the centralized deterministic approach the distributed path computation approach does not provide the channels along the path and this assignment must be done separately as provisioning of the path takes place on a link-by-link basis. We'll discuss channel assignment in next section.

Algorithm 2:

(a) To each TE link that shares an SRLG with the working path, assign infinite cost, i.e. *cost = infinite*

(b) For each remaining TE link, set the cost based on #UC, #RC and Dcost, i.e. ***Cost = f(#UC, #RC, Dcost)***. A simple function can be expressed as ***f(#UC, #RC, Dcost) =***

- Dcost if #UC > 0 and #RC=0 (a TE link with available channel but no reserved channel)
- infinite if #UC=0 and #RC=0 (a TE link with neither available channel nor reserved channel)
- *MAX_COST* if #UC=0 and #RC>0 (a link with reserved channel but no available channel). MAX_COST is larger than the default cost of any TE link. A special case is MAX_COST = infinity, which guarantees that the resource for the shared restoration path is available)
- Dcost x weight (weight is a constant and less than 1) if #UC > 0 and #RC>0

Note that if MAX_COST is equal to infinite and weight is equal to 1, the algorithm 2 is the same as algorithm 1. In another word, algorithm 1 is a special case of algorithm 2 without information of #RC in each TE link.

Algorithm 3:

(a) To each TE link that shares an SRLG with the working path, assign infinite cost, i.e. *cost = infinite*

(b) For each remaining TE link, set the cost ***f*** equal to

- infinite if #UC=0 and #RC=0
- Dcost if #UC > 0 and #RC=0
- Dcost x weight and weight = ε + (1-ε)P if #RC>0, where ε is a small constant and P is the probability that no reserved channel is sharable

This algorithm is based on the probabilistic approach [28]. Evidently the same SRLG cannot be protected multiple times by the same reserved channel otherwise contention would exist through their respective working paths if the SRLG fail. Thus, the problem of computing the probability that there is at least one shareable reserved channel (complement to the probability that no reserved channel is sharable) is equivalent to the probability that at least a reserved channel exists, which does not contain any of the SRLGs traversed by the new working path. Given the number of RCs (#RCs), the number of times (#times) by which an SRLG is protected in the TE link, the probability of no sharable channel can be approximately estimated as we illustrated in [28],

$$P = \left[1 - \prod_{i \in \{1,2,\ldots N\}} \left(1 - \frac{n_i}{M}\right)\right]^M$$

where M denotes the number of the reserved channels in a given TE link, N the number of SRLGs traversed by the working path for which a reserved channel is sought, n_i the number of times that the ith SRLG of this working path has been protected on the TE link.

Several variations for Algorithm 3 can be obtained. One is to estimate the un-sharable probability P based upon #RCs and the total number of SRLGs protected by the TE link. The other is to estimate P based upon #RCs and whether a SRLG has been protected by the TE link.

Note that the difference in the above algorithms and the deterministic algorithm is only in step 3. In the deterministic algorithm the cost of a TE link is set to the default link cost times ε<<1 if it contains a shareable reserved channel and default link cost if it does not. In the above heuristic algorithms this cost is replaced by a link cost function that determines the cost based on the default link cost, resource availability, and shareability. The deterministic approach requires additional information to compute the routes. In particular it needs to know whether a SRLG is protected or not for every reserved channel. Whereas in the heuristic approach, only certain aggregated network information is needed based on the link cost function in the algorithm. We'll see that there are tradeoffs between the network capacity efficiency and the required amount of input information for distributed path computation algorithms.

Finally, although only single SRLG failure is considered here in the path computation algorithms, the description of algorithms can easily be transposed to restore against both SRLG and node failure as well [35].

4 Provisioning through Signaling and Local Channel Assignment Mechanisms

In distributed control architecture, the ingress node initializes the provisioning along the path after computing it. Standard signaling protocols such as GMPLS extended RSVP-TE [21] and CR-LDP [22] can be used for this purpose. The path information computed by the ingress node is carried in the signaling messages. Note that the distributed path computation algorithms only give the nodes and TE links used by the paths because they don't have the detailed link utilization information. Channels are assigned locally node-by-node along the computed path during provisioning. Let's assume GMPLS extended RSVP-TE that is a dominant signaling protocol. In the case of unprotected path or 1+1 protected paths, the signaling messages contain a list of all the nodes and corresponding TE links used on each node for the working path or 1+1 protection path (e.g. in the Explicit Route Object (ERO) of the RSVP PATH message). It traverses node-by-node along the path. The upstream node assigns an available channel within the specified TE link between it and the next hop and informs the selection to the downstream node in the outgoing signaling message [38].

In shared mesh restoration, the objective of the path computation is to compute the paths so that sharing is maximized during channel assignment of restoration path. The channel assignment procedure during path establishment will guarantee that there is no sharing violation. In order to guarantee this, the scheme used for channel assignment at each node along the restoration path requires the SRLG information used by the corresponding working path. This information will be carried through signaling. RSVP-TE has been extended to carry the SRLG information of the corresponding working path in the signaling message (PATH) during restoration path establishment [15, 38]. It allows that each node build a local database regarding to a list of SRLGs protected by each reserved channel terminating into it. A list of SRLGs protected by a given reserved channel consists of all distinct SRLGs traversed by all the working paths whose respective restoration paths are assigned to this reserved channel. Thus a reserved channel can be reused to protect a new working path only if no SRLG traversed by the working path appears in the list of SRLGs already protected by this channel.

The nodes used by the restoration path and the SRLGs traversed by its corresponding working path are carried in the signaling message traveling node-by-node along the restoration path. Various schemes can be implemented to assign the reserved channel to the new restoration path. One scheme is that selection of specific channel in the TE links between two adjacent nodes is made by the upstream node [38]. The upstream node searches all the channels already reserved for shared mesh restoration between it and the next hop. The scheme employs the channel sharing information in the local database and the SRLG information of currently restored working path in the signaling to find a sharable channel. Note that only the nodes (no TE links) traversed by the restoration path are required to carry in the signaling message. If there are multiple TE links between two adjacent nodes, the search starts from the least cost TE link and then the higher cost TE links. Of course, only the TE links that are SRLG disjoint to the working path are considered, which will suffice that the restoration path and the working path are SRLG disjoint each other. It maximizes sharability by not limiting the TE link used between two adjacent nodes. If no present reserved channel is able to share with the restoration path, a new unassigned channel is reserved for this restoration path. The state of this unassigned channel becomes reserved channel. The unassigned channel with the least cost will be reserved if there are multiple TE links with different cost between this node and the next hop. The database will be updated after channel assignment. Even though the TE links between a pair of nodes may be erroneously tagged as having a shareable channel during path computation, the local channel assignment can guarantee there are no sharing violation.

Note that it is possible that there is neither sharable channel nor unassigned channel between two adjacent nodes along the restoration path. It is because some path computation algorithms (algorithms 2 and 3 in the previous section) used at ingress node do not have complete network resource sharing information and do not guarantee an unassigned channel available along the restoration path. For this case, a message with certain error indication can be sent back to the ingress node. The ingress node will decide whether to compute another path by excluding the failed links or to simply return error to the user.

Given a group of restoration paths traversing the common links between a pair of nodes, the issue for the optimal local channel assignment is to assign the minimum number of restoration channels to the paths in the links in accordance to the sharing rules. However the online channel assignment scheme described above assigns restoration channels on a first-come first serve basis and reserve new channels when sharing is not possible with present reserved channels. In this approach the number of reserved channels depends on the order of arrival of the restoration paths, which can not achieve the optimization. However it is possible to invoke an optimization algorithm at regular intervals to reassign the channels. It can be shown that finding the optimum channel assignment is equivalent to solving a vertex-coloring problem [29].

The allotment of reserved channels is tantamount to a vertex-coloring problem. Given the set of all restoration paths that intersect on the links between a pair of nodes, represent every path as a vertex, and connect with an edge every pair of vertices whose corresponding paths are conflicting. Two restoration paths are compatible and may share a reserved channel if their respective working paths are SRLG disjoint. Otherwise they are said to be conflicting. If we assign a distinctive color to each reserved channel, the allocation of a reserved channel to each path is to color the vertices. Clearly two vertices cannot be allotted the same color if they are connected by an edge, since the corresponding restoration paths are conflicting and cannot share the same channel.

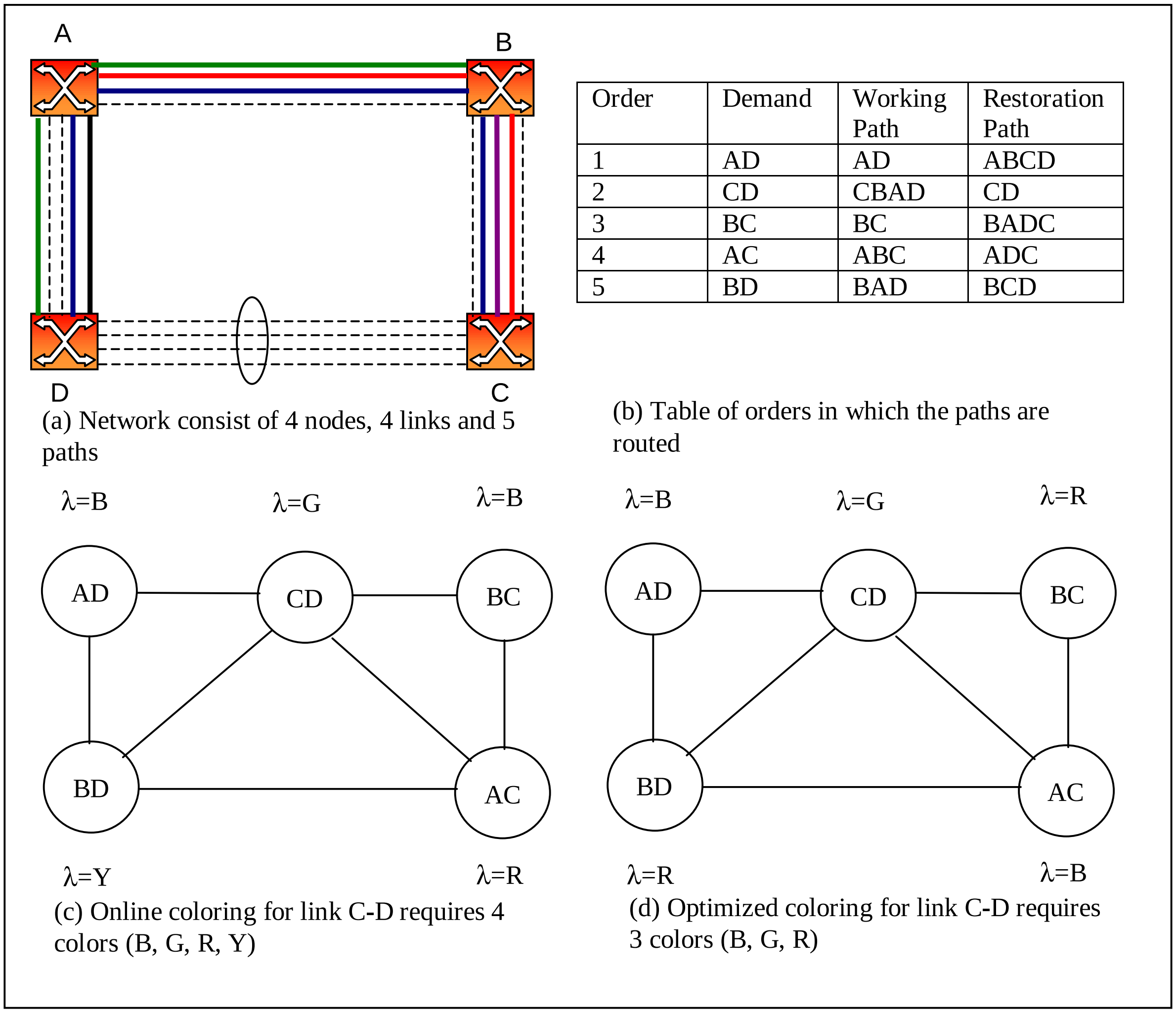
The objective is to minimize the number of reserved channels (respectively number of colors) required to accommodate all restoration paths (respectively color all vertices), while avoiding conflicts. This problem is known to be NP-hard, however there are many heuristics that can be used to compute sub-optimal solutions. A vertex-coloring algorithm that offers a good tradeoff between quality and runtime complexity is DSATUR [32].

Let's consider an example as shown in Figure 3 below. The figure illustrates five paths {AD, CD, BC, AC, BD} and their restoration paths, routed in a four-node ring network. All the restoration paths traverse link C-D (here we assume that the link C-D is a very high-cost link so the working path for demand CD is CBAD). The demands are provisioned following the sequence indicated in Table 3b. If we use the online shared mesh restoration channel assignment scheme above, and apply the graph representation presented earlier to C-D, we obtain the "coloring" shown in Figure 3c. Even though a single failure in this example affects at most three working paths, this coloring consumes four colors, indicating that four restoration channels are required. An optimized coloring yields the solution shown in Figure 3d, which consumes only three colors. Comparing figures 3c and 3d, we observe that a new channel (R) should have been allotted to the restoration path of demand (BC) instead of sharing channel (B) with the restoration of demand (AD). This solution however is not considered because not optimal when the third demand (demand {BC}) is being provisioned since at that time it would consume three channels (B, G, R) instead of two (B, G).

With the distributed routing algorithms, the restoration channels are not determined by the path computation algorithms. The online local channel assignment scheme can be used to allot the reserved channel for the restoration path node-by-node on the fly each time a lightpath is being provisioned and signaled. Furthermore, the optimized channel reassignment mechanism described above can be invoked at regular intervals or upon certain events. The optimized channel reassignment mechanism can be a low priority program thread running in background. The information necessary to accomplish this task is available locally in every OXC and independent of non-adjacent OXCs. Thus each OXC can run a copy of the algorithm in a distributed manner, locally and independently of other OXCs. A change in the allocation of a restoration channel needs only to be signaled to the node at the other end of the link. Since reserved channels are "booked" and actually not cross-connected until a restoration occurs, the task amounts to no more than modifying and exchanging sharing databases between pairs of adjacent nodes. For every OXC-pair connected by at least one optical link, the OXC with highest IP address can be delegated to perform the task. Note that the algorithm only optimizes the reserved channels assigned to the restoration paths locally. However, it does not optimize the routes of the restoration paths, and the resulting solution is thus not as efficient as a reoptimization algorithm that re-routes the restoration paths to maximize sharing [37].

Finally the channel re-optimization procedure closes an advantage gap of the failure dependent strategy over the failure independent strategy in term of channel utilization efficiency. Furthermore in the case of multiple failure scenarios there is a higher probability in the failure independent strategy that two services affected by two distinct failures contend for the same restoration channel, even if there are parallel reserved channels available. This is because the restoration channels are pre-assigned. Although re-provisioning mechanisms that compute restoration path on the fly when the planned restoration fails would mitigate this problem [37], they are not covered here.

Instead, a background channel re-optimization process can be used to detect the prospect for such contentions after the first failure, and re-assign the channels to eliminate them.



(a) Network consist of 4 nodes, 4 links and 5 paths

Order	Demand	Working Path	Restoration Path
1	AD	AD	ABCD
2	CD	CBAD	CD
3	BC	BC	BADC
4	AC	ABC	ADC
5	BD	BAD	BCD

(b) Table of orders in which the paths are routed

(c) Online coloring for link C-D requires 4 colors (B, G, R, Y)

(d) Optimized coloring for link C-D requires 3 colors (B, G, R)

Figure 3. Example of sub-optimal (first-fit) and optimal local restoration channel assignment.

5 Required Extensions to Routing Protocols

As discussed before, the path computation algorithm requires the network topology and optical link resource information. It is the responsibility of the routing protocols to disseminate this information. Topology discovery and routing in traditional IP networks is performed with the IP routing protocol, such as Open Shortest Path First (OSPF) [10]. OSPF allows hierarchical routing, where a large network may be divided into a collection of smaller areas with limited routing information exchange among areas. For smaller networks, it is sufficient with a single backbone area.

In traditional IP network, each router uses the information provided by the OSPF and a routing algorithm to compute a routing table. The routing table allows forwarding the packets to next hop based on their destination addresses. In the optical networks, however, the entire path is explicitly computed at the ingress node when the connection is requested. Then the path computation algorithm at the ingress node needs to have sufficient information to compute the path.

Standard OSPF does not distribute enough link state and resource information that is required to compute the explicit path in optical networks. OSPF-TE and GMPLS have extended the standard OSPF to support un-numbered link and to carry new link attributes, including link type, traffic engineering metric, available resource (maximum bandwidth, maximum reservable bandwidth, unreserved bandwidth), resource class/color, local and remote link identifier (or interface IP addresses), link protection type, interface switching capability descriptor, link SRLGs and etc.[7, 8, 9]. This information is necessary to support traffic engineering and routing in optical networks.

In addition, the link bundling is introduced to improve the scalability [11]. Each physical link in the standard OSPF would result in a routing adjacency so that routing messages would be exchanged over each such link and every link would be advertised to other nodes in the network. There are a large number of physical links between a pair of OXCs in the optical networks. This may then result in increased message flooding and processing overhead. Link bundling allows that multiple physical optical links between two adjacent nodes with the same characteristics are aggregated into a TE link, which is treated as a single virtual link (TE link) when advertised by OSPF. It also de-couples the physical data links and the control channel. Even if there are multiple data links between a pair of OXCs, there may be only one logical routing adjacency between them and the routing messages are sent over one single control channel.

OSPF with traffic engineering and GMPLS extensions is able to disseminating sufficient information for computing unprotected and 1+1 protected paths. However, in shared mesh restoration, multiple restoration paths can share the same reserved restoration resource on their common TE links only if the sets of SRLGs traversed by their respective working paths are disjoint in order to guarantee recovery from a single SRLG failure. This imposes additional constraints on the path computation. To compute the restoration path for the shared mesh restoration, the path computation module needs to have the restoration resource sharing information of the links in the network, as discussed in Section 3. To support path computation for shared mesh restoration, all or some information below should be disseminated by routing protocol, depending on the path computation algorithms.

(1) Summarized information about the restoration resource sharing on a TE link for shared mesh restoration, such as the total number of restoration paths sharing the restoration resource reserved on the TE link for shared mesh restoration, the total number of SRLGs protected by the reserved restoration resource on the TE link, the total sharable restoration bandwidth at each priority level.
(2) The list of SRLGs protected by the reserved restoration resource on the TE link and their respective sharable restoration bandwidth since the SRLG-disjointness is required to guarantee recovery in the event of a single SRLG failure.

The list of SRLGs protected by the TE link is defined as the union of SRLGs traversed by all the working paths whose respective restoration paths share the reserved restoration resource on this TE link. The sharable restoration bandwidth for a SRLG indicates the available restoration bandwidth on the TE link that can be reserved for recovering this SRLG failure. If a working path only traverses one SRLG, the available restoration bandwidth that its restoration path can share on this TE link is the sharable restoration bandwidth for this SRLG. When a working path traverses multiple SRLGs, the sharable restoration bandwidth available for its restoration path may become smaller on this TE link. The total sharable restoration bandwidth is the bandwidth reserved on the TE link for restoration, which is the union of the sharable restoration bandwidth for all SRLGs and nodes.

OSPF traffic engineering extensions [7] and GMPLS extensions [8, 9] make use of the Opaque LSA (Link State Advertisement) [24]. An Opaque LSA, called Traffic Engineering LSA is defined to carry the additional attributes related to traffic engineering and GMPLS links, and standard link-state database flooding mechanisms are used for distributing TE LSAs. The LSA payload consists of one or more nested TLV (Type/Length/Value) triplets for extensibility. There are two types of TE LSAs [9]. One contains a Router Address TLV (Router Address TE LSA) that specifies a stable IP address of the advertising node. The other contains a link TLV (Link TE LSA), which describes a TE link. The Link TLV is constructed of a set of sub-TLVs that specify the link attributes. We only concern the Link TE LSAs here. In the rest of this paper, TE LSA refers to Link TE LSA unless otherwise stated. Each TE LSA carries the summarized resource information regarding to a TE link. For each TE link, the attributes, including link type, traffic engineering metric, available resource, administrative group, local and remote link identifier (or interface IP addresses), link protection type, interface switching capability descriptor, are specified in the form of sub-TLVs in the Link TLV of the TE LSA. The information in the TE LSAs are used to build an extended TE link state database for the explicit path computation just as router LSAs are used to build a regular link state database for packet forwarding. The extensions in support of carrying link state information for the path computation of shared mesh restoration can be based upon the OSPF-TE and its GMPLS extensions. Specifically, the sub-TLVs carrying the above sharing information of the restoration resource on a TE link can be added to the link TLV of the TE LSA so that the information can be used by the path computation algorithm to compute the restoration path. Two sub-TLVs can be defined [35]. Restoration information summary sub-TLV specifies the sharing information of the restoration resource reserved for the shared mesh restoration on the TE link, including #SRLGs protected and total sharable restoration bandwidth (i.e. #RC). SRLG sharable restoration bandwidth sub-TLV identifies the sharable restoration bandwidth for a protected SRLG on the TE link.

In OSPF, a node originates its TE LSAs like other types of LSAs when the contents of LSAs change or refreshes them as required by the protocol. This does not mean that every change must be flooded immediately. There are various mechanisms to limit the flooding rates. One approach is to increase the minimum LSA flooding interval, which the origination of TE LSAs should be limited to at most one every minimum LSA flooding interval. The other approach is to set the thresholds that trigger immediate flooding. For example, #UC thresholds (i.e. unreserved bandwidth thresholds in the word of OSPF-TE) is set to trigger the LSA update flooding when the number of unassigned channels on a TE link increases or decreases across a set of fixed values. If we set the fixed threshold to be 1, a LSA update is flooded when #UC changes from a value greater than 0 to zero or vice versa. In another way, the threshold can be based on the relative change between the current and the previously advertised link state. When the change (increase or decrease) in the number of unassigned channels on a TE link exceeds the #UC change threshold or the change is over a percentage value of the total #UC on the link, a LSA update flooding is originated. Furthermore, multiple thresholds can be set. The flooding is generated when one of them is triggered (e.g. the change in #UC exceeds the #UC change threshold or the change in #RC exceeds the #RC change threshold). Note that in any case the origination of TE LSAs should be rate limited to at most one every minimum link state flooding interval as required by the protocol[10].

6 Performance Results

In this section, we investigate the performance of various path computation algorithms and extended OSPF. More specifically, we compare the computation complexity and network capacity efficiency of the algorithms. We also analyze the control bandwidth usage for disseminating link state information required by these algorithms through extended OSPF.

6.1 Complexity of Path Computation Algorithms

We first evaluate the complexity of the distributed path computation algorithms using heuristic approach and centralized algorithm using deterministic approach in term of processing time when determining a restoration path. We assume that a failure independent strategy is used, in which the protection channels are specifically assigned to the restoration paths at the time of provisioning before failures occur. Note that we measure here the complexity of computing the restoration path of a new service. This time should not be confounded with the restoration latency, which is the delay required to recover all the services on the pre-computed restoration paths when failures occur.

In shared mesh restoration, a list of SRLGs protected by a given reserved channel consists of all distinct SRLGs traversed by all the working paths whose respective restoration paths are assigned to this reserved channel. Thus a reserved channel can be reused to protect a new working path if no SRLG traversed by the working path appears in the list of SRLGs already protected by the channel.

We denote by h the average working path length expressed in number of (TE) links, m the number of (TE) links in the network, and x the total number of restoration channels reserved throughout the network. We also assume where the total number of SRLGs in the network is on the order of $O(m)$ and the average number of SRLGs on the working path is on the order $O(h)$. For the centralized path computation, the reserved channel is assigned by the path computation algorithm at the network management system using deterministic approach. Shareable reserved channels in the network are identified by verifying that for each reserved channel in each link the list of SRLGs protected by the channel does not intersect with the list of SRLGs traversed by the working path. Therefore, the complexity of identifying all the links with shareable reserved channels in the network is $O(hx)$ [28]. Note that it is assumed here that each reserved channel maintains a fixed length array in which each entry indicates whether a

SRLG is used or not. The complexity becomes $O(xhlog(m))$ if instead each reserved channel maintains a variable length list of protected SRLGs (search is required to find whether the SRLG is used or not). The number of restoration channels is a function of g, the number of paths in the network, and can be approximated by $x=O(gh')$, where h' is the average length of a restoration path (usually $h' \geq h$.) Substituting x, the complexity of this operation is $O(ghh')$. In term of required network information to compute the restoration path, the centralized algorithm needs to know the list of SRLGs protected by each channel. The size of this information is thus on the order of $O(gh'm)$. Note that both the computation complexity and the required information size for the centralized algorithm depend on the number of paths established in the network.

As described in Section 3, for the distributed path computation, the computation algorithm at the ingress node determines the restoration path in a series of node that the path traverses through. The specific channel reserved for the restoration path is assigned hop-by-hop locally during provisioning by the upstream node.

When computing the restoration path at the ingress node, the complexity of identifying all the links with sharable reserved channels in the network (i.e. determining the cost for each link) is then $O(m)$ for distributed path computation algorithm 1 and $2O(m)$ for algorithm 2 in Section 3. The algorithm 1 requires #UC be known and algorithm 2 #UC and #RC for each of link in the network. Therefore the size of required information is $O(m)$ and $2O(m)$, respectively.

In distributed path computation algorithm 3, the link cost depends on the probability that there is at least one shareable reserved channel. It is the probability that a reserved channel does not contain any of the N SRLGs traversed by the corresponding working path. The complexity of computing the probability involves computing N products and an Mth power, which is realizable in $O(N+logM) \approx O(N)$. Typically N is the average path length h. Therefore the time complexity of identifying all the links with sharable reserved channels in the network is $O(hm)$. Algorithm 3 requires #UC, #RC, and # of times each SRLG is protected in the link. Therefore the size of required information is $O(m^2)$.

As a summary, table 1 lists the complexity and required amount of input information to compute the restoration path for the above distributed path computation algorithms and centralized algorithm.

Table 1. The complexity and required information to compute the restoration path for different algorithms

	Distributed Algorithm 1	Distributed Algorithm 2	Distributed Algorithm 3	Centralized Algorithm
Complexity to identify all the links with sharable reserved channels	$O(m)$	$2O(m)$	$O(hm)$	$O(ghh')$
Required input information	#UC	#UC, #RC	#UC, #RC, # of times a SRLG protected by each TE link	the list of SRLGs protected by each channel
Quantity of input information	$O(m)$	$2O(m)$	$O(m^2)$	$O(gmh')$

For the centralized algorithm using deterministic approach to determine the sharable channel, both the complexity and input information amount depend on the number of lightpaths established in the network. However, for the distributed algorithms using heuristic approach, the complexity and input information amount remains constant with respect to the number of paths. Of course, nothing prevents the centralized algorithm from using the heuristic approach. However we'll see later that the heuristic approach results in performance loss in term of network capacity utilization.

In addition, the distributed path computation needs to assign the reserved channel hop-by-hop during the provisioning along the restoration path. Considering the above online provisioning algorithm that assigns restoration channel on a first-come first-serve basis, the complexity is $O(he)$. e is the number of reserved channels per link, which is $e=x/m= O(gh'/m)$. Then the complexity can be expressed as $O(ghh'/m)$. Once again, it is assumed here that each reserved channel has a fixed length array in which each entry indicates whether a SRLG is used or not. The complexity becomes *O(ghh'log(m)/m)* if instead each reserved channel maintains a variable length list of protected SRLGs (search is required to find whether the SRLG is used or not).

6.2 Performance Comparison of Path Computation Algorithms

In this section, we compare the performance of the centralized algorithm using deterministic approach and the distributed algorithms using heuristic approach described in Section 3. As for the distributed algorithms, the assignment of the channels is done locally during path-setup signaling and the channel sharing for the restoration paths is also performed with the local channel assignment. We use the online channel assignment algorithm described in Section 4 for this study in order to isolate and limit our measurement to the performance of path computation algorithms. The benefits of a local channel assignment optimization are measured separately in next sub-section. We assume here that every TE link costs one unit of currency and corresponds to one SRLG (i.e. one SRLG per TE link and one TE link per SRLG). For this comparison, we experimented the algorithms on various network topologies inspired from real carrier networks with realistic demands for shared mesh restored paths. We route different demands on various network topologies using each of path computation algorithms. Given a demand and a network, we measure the total number of channels required by the working paths and restoration paths (used for working paths and reserved for restoration paths) for a path computation algorithm. The relative performance of a distributed algorithm in term of total channel usage can then be obtained by comparing with that of the centralized algorithm. We use the average relative channel usage obtained from a distributed path computation algorithm under various networks as a measurement of the algorithm performance. The results are shown in Figure 4. They indicate that distributed algorithm 3 is comparable to the deterministic approach. In comparison, distributed algorithm 1, which ignores the possibility of sharing existing reserved channels in the path computation, performs relatively poorly, and requires about 9% more channels than the centralized algorithm.

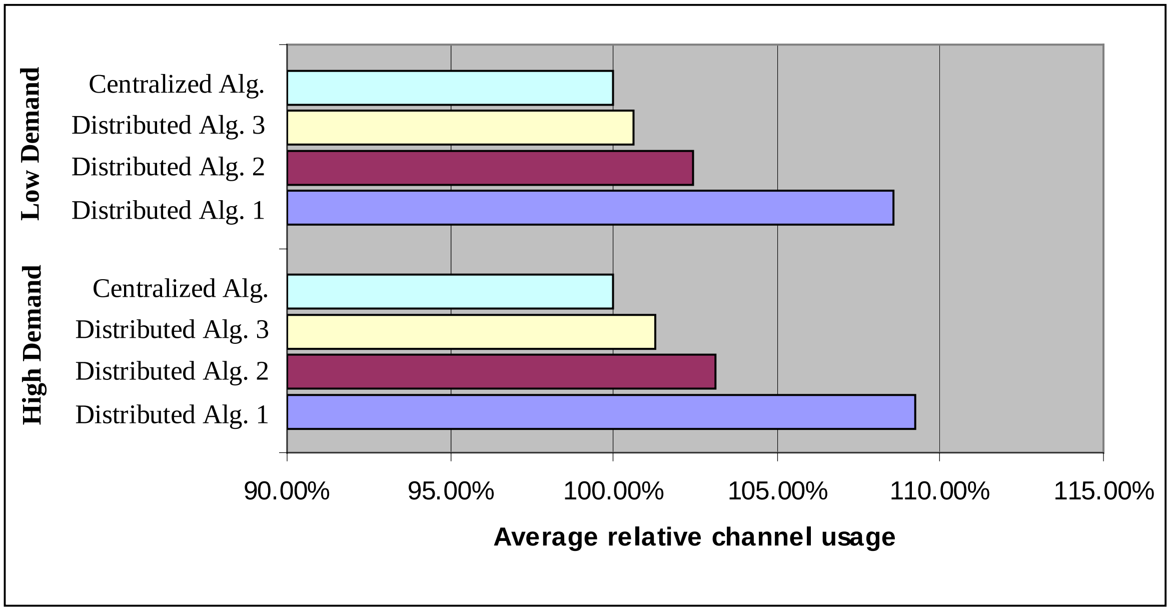


Figure 4. Comparison of performance in term of average channel usage for the heuristic path computation algorithms relative to the centralized algorithm using deterministic approach

6.3 Performance of Local Channel Assignment Optimization

In this sub-section we investigate the performance of local restoration channel optimization mechanism in term of number reduction in required reserved channels comparing to the online first-fit algorithm. We considered two realistic core mesh networks. Network A consists of shared-mesh capable optical switches in 46 cities interconnected by 75 links and loaded with 570 lightpaths. Network B consists of 61 switches, 88 links, and 419 lightpaths. For each network, we provision all the demands in sequence using various values of demand churns (expressed in percent of the total demand). We use an online first-fit channel assignment during provisioning, and then apply a local channel optimization after all the demands are routed. We measure the amount of reserved channel required before and after local channel assignment optimization and report the saving in percentage of total restoration capacity in Figure 5. Our measurements indicate that as the demand churn increases, the number of reserved channels that can be saved becomes substantial.

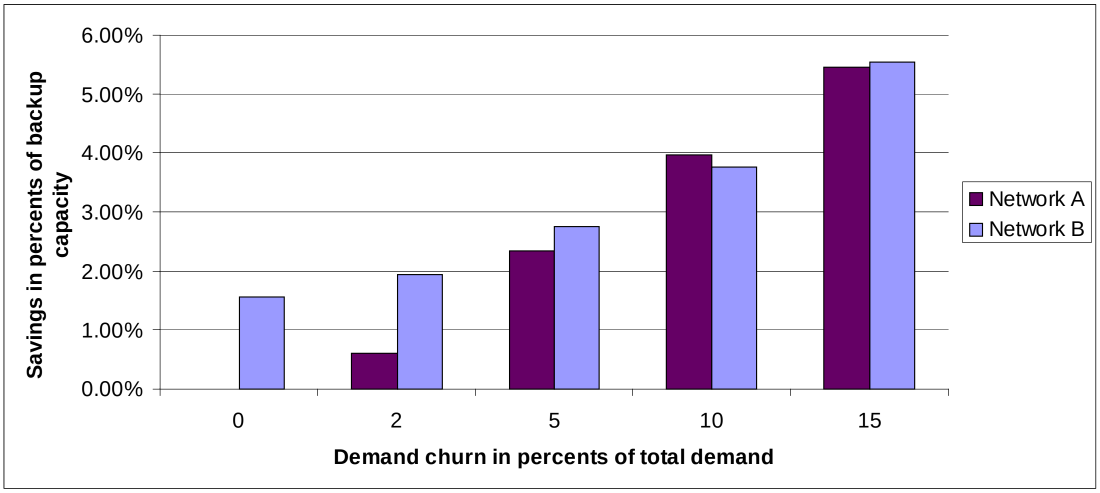


Figure 5. Percentage saving of reserved restoration capacity by applying local channel assignment optimization compared to a first-fit online channel assignment.

6.4 Bandwidth and Memory Requirements of OSPF-TE with GMPLS and Shared Mesh Restoration Extensions

Next we estimate the bandwidth usage due to TE LSA and router LSA updates and the memory requirement for storing the LSAs. For OSPF, a node originates its LSAs when the attributes of the links, i.e. the contents of LSAs change. This does not mean that every change must be flooded immediately. The flooding rate depends on the link attributes advertised in the LSAs and the thresholds triggering immediate flooding. In addition, the OSPF requires a router to refresh self-originated LSAs periodically (the default refresh time is 30 minutes). A new instance of the LSA is originated even though the contents of the LSA (apart from the LSA header) will be the same every 30 minutes. We consider the TE LSA flooding rate for three scenarios.

(1) The link attributes specified in standard GMPLS OSPF-TE (without shared mesh restoration extensions) are advertised in TE LSA. For this case, only the number of unassigned channels (#RC) is advertised by the OSPF, i.e. the link state information required by path computation Algorithm 1 in Section 3 is advertised.

    To estimate the average interval of TE LSA flooding for this case, we consider that the new TE LSAs shall be originated whenever the change in the number of unassigned channels exceeds a threshold Ψ_1. Let λ and μ denote the average rate of creation and deletion of shared mesh restored paths in the network, respectively; and *h* and *h'* denote the average number of hops for the working and restoration paths, respectively. Let *s* be the average number of restoration paths sharing a reserved channel and *m* the total number of TE links in the network. It is assumed that Ψ_1 is small so that path creation and deletion do not occur on the same TE link during a TE LSA flooding period. That is, the worst case in term of TE LSA flooding rate is considered (because if the threshold is greater than 1, the deletion and creation of paths may offset each other for the change in #RC on a TE link). Then the change rate in the number of unassigned channels per TE link per second is $(\lambda+\mu)(h+h'/s)/m$. The TE LSA flooding rate due to the change in the TE link attributes is $(\lambda+\mu)(h+h'/s)/m\Psi_1$. The average flooding interval triggered by the change in the TE LSA contents is

$T=m\Psi_1/[(\lambda+\mu)(h+h'/s)]$. OSPF requires the LSAs refreshed every $E=1800$ sec even if the contents of the LSA is not changed. If $T<E$, we can assume that the refreshes don't occur. The effective flooding interval for a TE LSA is T and the effective rate of a TE LSA flooding is $R=1/T$. If $T>E$, T/E refreshes occurs before a change triggered flooding. Thus the effective rate of a TE LSA flooding is given by $R=([T/E]+1)/T$ [33]. Note that this expression $R=([T/E]+1)/T$ holds for both $T<E$ and $T>E$ cases since $[T/E]=0$ for $T<E$. The stead state path creation and deletion rate can be obtained by Little's formula, $\lambda=\mu=g/\tau$, where g is the total number of paths in the network and τ is the average hold time of the paths.

(2) The link attributes specified in standard GMPLS OSPF-TE and Restoration Information Summary sub-TLV are advertised in TE LSA. For this case, both the number of unassigned channels (#RC) and the number of reserved channels (#RC) on the TE link are advertised by the OSPF, i.e. the link state information required by path computation Algorithm 2 in Section 3 is advertised.

To estimate the average interval of TE LSA flooding for this case, we consider that the new TE LSAs shall be originated whenever the sum of absolute changes in the number of unassigned channels and the number of reserved channels (i.e. |change in #UC| + |change in #RC|) exceeds a threshold Ψ_2. Hence, the average flooding interval triggered by the change in the TE LSA contents can be obtained as $T=m\Psi_2/[(\lambda+\mu)(h+2h'/s)]$ for $\Psi_2>1$ and $T=m/[(\lambda+\mu)(h+h'/s)]$ for $\Psi_2=1$. The effective rate of a TE LSA flooding is also given by $R=([T/E]+1)/T$.

(3) The link attributes specified in standard GMPLS OSPF-TE, Restoration Information Summary sub-TLV, and SRLG Sharable Restoration Bandwidth Sub-TLV are advertised in TE LSA. For this case, the number of unassigned channels (#RC), the number of reserved channels (#RC) and the number of times (#times) by which an SRLG is protected in the TE link are advertised by the OSPF, i.e. the link state information required by path computation Algorithm 3 in Section 3 is advertised.

To estimate the average interval of TE LSA flooding for this case, we consider that the new TE LSAs shall be originated whenever the link attributes specified in the TE LSA, i.e. the TE LSA contents changes. It represents the maximum flooding rate for a TE LSA. Hence, The average flooding interval triggered by the change in the TE LSA contents can be obtained as $T=m/[(\lambda+\mu)(h+h')]$ and the effective rate of a TE LSA flooding is also given by $R=([T/E]+1)/T$.

Table 2. OSPF parameters

Type	Size (Bytes)
IP header	20
OSPF header	24
LSA header	20
TE link LSA contents without mesh restoration extensions	148 + #SRLGperLink x 4
Restoration Information Summary sub-TLV	44
SRLG Sharable Restoration Bandwidth Sub-TLV	#SRLGprotected x 20
Router LSA content	4 + #LinkperNode x 12

OSPF protocol runs on the top of IP. Table 2 lists the parameters used in the estimation. For the simplicity, we assume that there is only one TE link between two neighboring nodes and every TE link corresponds to one SRLG. The size of TE LSA for a TE link can be obtained from the summation of the sizes of all sub-TLVs it carries. Each OSPF link state update packet may carry multiple LSAs, which depends on the number of LSAs flooded at the same time and the interface MTU(the size of the largest IP datagram that can be sent out the associated interface without fragmentation). In OSPF, a newly received LSA must be acknowledged. It can be done by sending Link State Acknowledgment packets. Many acknowledgments may be grouped together into a single Link State Acknowledgment packet. The packet can be sent in one of two ways: delayed and sent on an interval timer, or sent directly to a particular neighbor. Acknowledgments can also be accomplished implicitly by sending Link State Update packets if a duplicate LSA is treated as an implied acknowledgement [10]. Let P denote the effective size of each LSA, including the required bytes for IP header, OSPF header and acknowledgement, $P=(L+48/h_1+20+44/h_2)$, where L is the real size of the LSA, h_1 and h_2 are the number of LSAs per link state update packet and the number of LSAs per link state acknowledge packet, respectively. Because each TE link is advertised by both of the adjacent nodes, the bandwidth usage due to TE LSAs flooding is $2mPR = 2mP([T/E]+1)/T$ where T is determined by the above for the different cases and $E=1800$ sec is refresh time interval. We assume that the LSAs are flooded independently and each IP packet contains one LSA below.

For a network with n nodes and an average node degree d, the total number of links in the network is $m=nd/2$. When the node degree is small, it can be assumed that the average path length of the working path grows linearly with network size n, i.e. $h = an+b$, where constants a and b (depending on d) could be determined by experiments [33]. The length of the restoration path can be determined by $h'=\alpha h$. We assume that each node has K ports. The total number of ports in the network is nK. If β is the ratio of drop port number to total port number, the total number of drop ports in the network is then $nK\beta$. The maximum number of paths in the network is $\eta=nK\beta/2$ because each path uses two drop ports.

For router LSAs, the flooding due to the LSA content changes occurs rarely (only when the status of OSPF control channels changes, for examples, a node fails or a control channel fails). Therefore the steady flooding rate of a router LSA is the standard refresh rate $R=1/E = 1/1800$. The effective router LSA size, including the required bytes for IP header, OSPF header and acknowledgement, is $P=(L+48/h_1+20+44/h_2)$, where L is the real size of the router LSA ($L=24+12d$), h_1 and h_2 are the number of LSAs per link state update packet and the number of LSAs per link state acknowledge packet, respectively. We assume that the router LSAs are flooded independently and each IP packet contains one LSA. Then the bandwidth usage for the router LSA flooding is $nPR=n(136+12d)/1800$ bytes/sec.

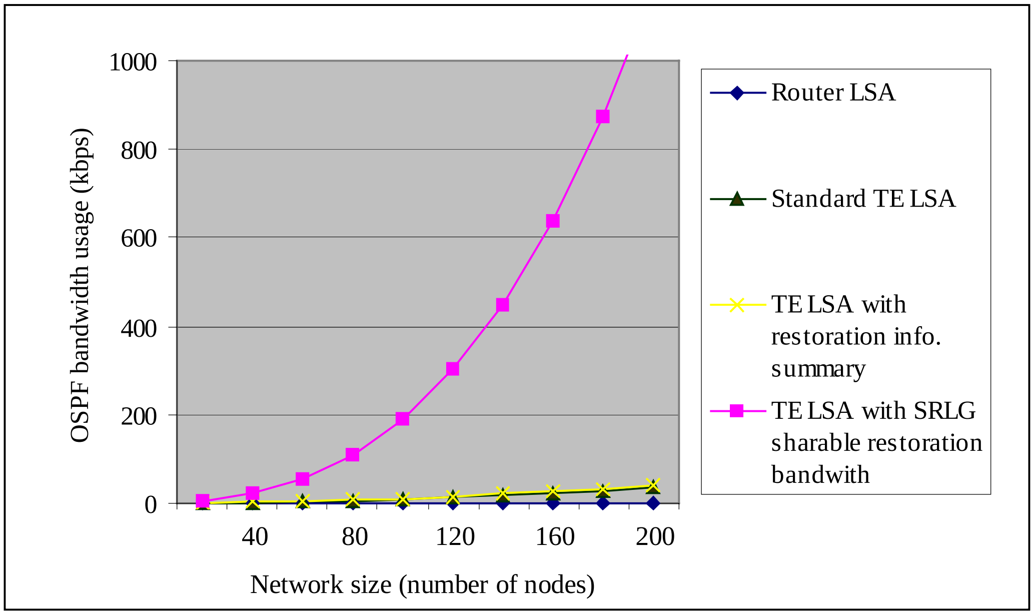


Figure 6. Bandwidth usage for router LSAs and TE LSAs with different abstraction of restoration resource information in various sizes of networks

Figure 6 shows the bandwidth usage for TE LSAs flooding under (1) standard GMPLS OSPF-TE without shared mesh restoration extensions, (2) GMPLS OSPF-TE with Restoration Information Summary sub-TLV, (3) GMPLS OSPF-TE, Restoration Information Summary sub-TLV, and SRLG Sharable Restoration Bandwidth Sub-TLV, and router LSA flooding with different network sizes. The trigger thresholds are set at $\Psi_1=1$ and $\Psi_2=1$ for scenario 1 and scenario 2, respectively. The node degree is equal to a typical value of 4 and the path hold time 12 hours. The flooding for TE LSAs with Restoration Information Summary sub-TLV and SRLG Sharable Restoration Bandwidth Sub-TLV (Scenario 3) requires more bandwidth because it advertises more information about the reserved resource sharing on the TE link. The difference in OSPF bandwidth usage between Scenario 3 and others becomes much larger as the network size increases. It is because the number of SRLGs protected on each link increases with the network size. The OSPF bandwidth usage for Scenario 3 (not for Scenarios 1 and 2, and router LSAs) depends on the number of SRLGs protected on each link since it needs to advertise the sharable restoration bandwidth for each SRLG. As shown in Section 6.2, however, path computation algorithms 3 that uses more resource sharing information on the TE link achieves better performance in term of network capacity usage. This verifies the tradeoffs between the path computation efficiency and the bandwidth requirement to disseminate the abstracted link resource information.

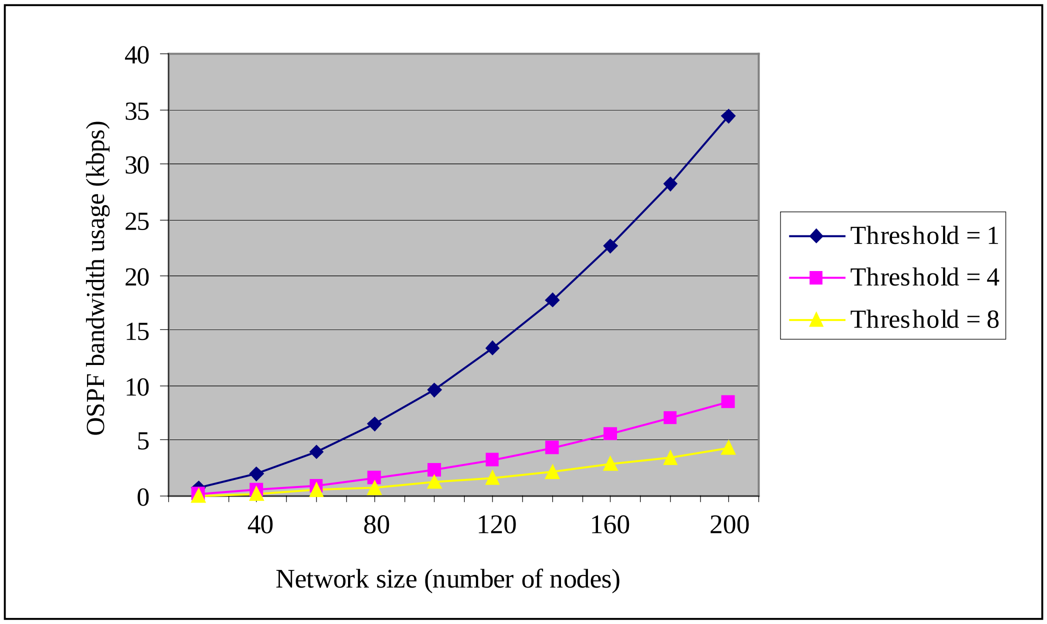


Figure 7. Bandwidth usage for TE LSAs with different values of flooding triggering thresholds in various sizes of networks when the standard GMPLS OSPF-TE is considered and the change in the number of unassigned channels on the TE link is set as threshold.

One approach to limit the flooding rate is to set larger thresholds that trigger immediate flooding. Figure 7 shows the TE LSA flooding bandwidth requirements for different values of #UC threshold under the above scenario 1. The bandwidth usage reduces with a large threshold. However the changes in link attributes cannot be advertised appropriately if the thresholds are set too large. It may result in path computation errors.

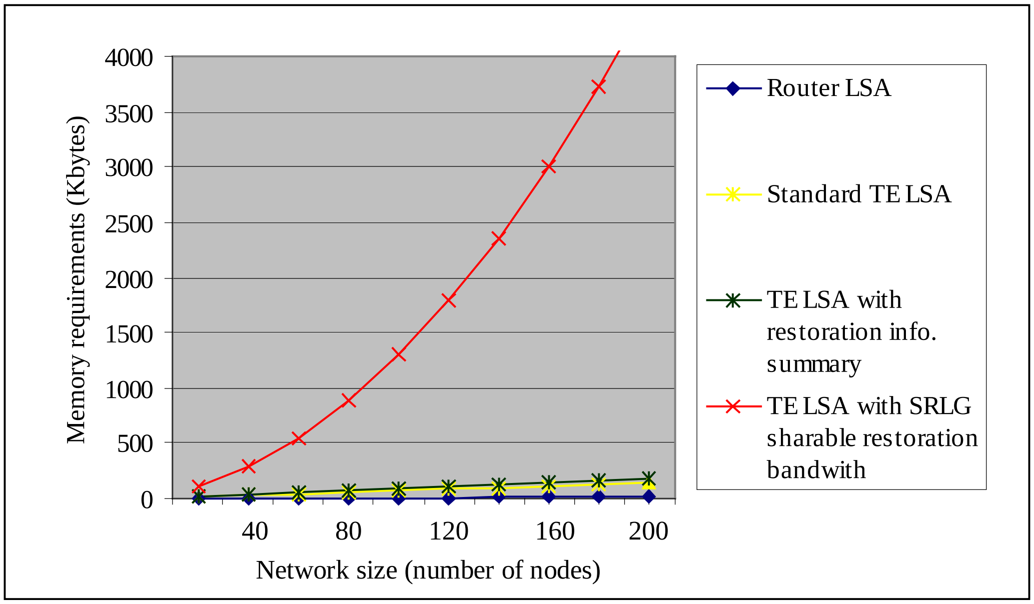


Figure 8. Node memory requirements for storing router LSAs and TE LSAs with different abstraction of reserved restoration resource information in various sizes of networks

The OSPF link state database is synchronized among all the nodes in the network. Figure 8 shows the node memory requirements to store the TE LSAs under the above three scenarios and the router LSAs for different sizes of networks. Note that there are *2m* TE LSAs and *n* router LSAs in the link state database.

7 Conclusions

We described several distributed path computation algorithms that use heuristic approach to identify shareable channels in a network when computing shared mesh restoration paths. These algorithms require different summarized resource information on the TE links. The tradeoff between the path computation efficiency and the degree of summarization and abstraction of link resource information are studied. The performance of these distributed algorithms is compared to that of the centralized path computation algorithm using deterministic approach, which requires the complete and detailed resource availability and sharing information. It is shown that with carefully aggregated resource information on TE link the proposed distributed path computation algorithms are able to achieve the network capacity usage close to that of centralized PCM and the required computation time of the distributed algorithms is much less. For the distributed algorithms, the summarized information consisting of one fixed length array for every TE link and independent of the amount of traffic demand is sufficient to compute the paths efficiently while maximizing sharing opportunities. In contrast, the deterministic approach needs one such array for every reserved restoration channel and the detailed information proportional to the number of active lightpaths, so it does not scale well when the traffic demand and network grow.

In general, the more detail information of resource availability and sharing on the TE links is available, the better results the path computation algorithms can achieve. On the other hand, in order to reduce the amount of information disseminated by routing protocols and improve the routing scalability, it is desirable to aggregate the information on a TE link. If only the number of unassigned channels (unreserved bandwidth) on a TE link is given to compute the restoration path for shared mesh restoration, the path computation (algorithm 1) results in a penalty of 9% more network capability usage than that of the deterministic algorithm in the centralized path computation. On the other hand, given the number of unassigned channels, the number of reserved channels, the number of times by which an SRLG is protected in the TE link, it can be estimated very accurately whether there exists a sharable reserved channel for the restoration path on the TE link. Based on the determined probability of available shared channel on the TE link, the path computation (algorithm 3) could achieve a network capability usage very close to that of the deterministic algorithm (the penalty is less than 2% in our simulation results). However this probabilistic approach uses several orders of magnitudes less information than what is necessary for the deterministic approach and completes the path computation significantly faster than deterministic approach.

We also discussed dynamic provisioning of shared mesh restored paths. The GMPLS RSVP-TE signaling protocol is extended to carry necessary SRLG information of the working path when the restoration path is provisioned. The information is used for the upstream node to assign the reserved channel to the restoration path locally based on the SRLG disjoint rule for guaranteeing single failure recovery. A local channel optimization method is described, which rearranges the allocation of shared channels reserved for restoration, with objective to minimize the number of allotted channels. This algorithm can be implemented as an independent background process to supplement either centralized or distributed provisioning algorithms. It is effective to correct sub-optimality inherent to a first fit based online channel assignment, or seize on improvement opportunities that are brought forth by demand churn.

Efficient methods are needed to aggregate and disseminate the routing information including optical link resource availability and sharing so that the amount advertised by the routing protocols is minimized but information necessary for path computation is not lost. We proposed to extend the GMPLS OSPF-TE routing protocol to carry necessary sharing information of reserved resource on the TE link in support of computing shared mesh restored paths. These new extensions provide the link resource information for the distributed path computation algorithms to compute the shared restoration path efficiently. The performance of the extended OSPF is analyzed. It needs much more control bandwidth and node memory to advertise and store the list of SRLGs protected by the reserved restoration resource on the TE link and their respective sharable restoration bandwidth than the link attributes only in GMPLS OSPF-TE. However it is shown that with a reasonable size of network, the total OSPF control channel bandwidth usage and node memory requirement is reasonable compared to available bandwidth and memory in the current control networks and nodes.

8 References

[1] T.E. Stern and K. Bala, “Multiwavelength Optical Networks: A Layered Approach”, Reading MA: Adison Wesley 1999.

[2] B. Doshi, et al “Optical Network Design and Restoration” Bell-labs Technical Journal, Jan-Mar 1999.

[3] B. Rajagopalan, D. Pendarakis, D. Saha, R. Ramamurthy and K. Bala, “IP over Optical Networks: Architectural Aspects”, IEEE Communications Magazine, September 2000.

[4] E. Mannie, editor, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture,” Internet Draft, Work in Progress, draft-ietf-ccamp-gmpls-architecture-03.txt, March 2002.

[5] ITU-T Recommendation G.8080, “Architecture for the ASON”.

[6] The Optical Internetworking Forum (OIF), “User Network Interface (UNI) v1.0 Signaling Specification”, December 2001.

[7] D. Katz, et al., “Traffic Engineering Extensions to OSPF Version 2,” Internet Draft, Work in Progress, October 2002.

[8] K. Kompella, et al., “Routing Extensions in Support of Generalized MPLS,” Internet Draft, Work in Progress, December 2002.

[9] K. Kompella, et al., “OSPF Extensions in Support of Generalized MPLS,” Internet Draft, Work in Progress, December 2002.

[10] J. Moy, "OSPF Version 2", RFC 2328, April 1998.

[11] K. Kompella, et al., “Link Bundling in MPLS Traffic Engineering”, Internet Draft, Work in Progress, January 2003.

[12] D. Papadimitriou, et al., "Analysis Grid for GMPLS-based Recovery Mechanisms," Internet Draft, work in progress, April 2002.

[13] E. Mannie, et al., "Recovery (Protection and Restoration) Terminology for GMPLS," Internet Draft, work in progress, February 2002.

[14] J. P. Lang, B. Rajagopalan, et al., "Generalized MPLS Recovery Functional Specification," Internet Draft, work in progress, August, 2002.

[15] G. Li, et al., "RSVP-TE Extensions for Shared-Mesh Restoration in Transport Networks," Internet Draft, work in progress, November 2001.

[16] G. Li, et al, “Efficient Distributed Path Selection for Shared Restoration Connections,” IEEE Infocom 2002, New York, NY, June 2002.

[17] R.R. Iraschko, M.H MacGregor, and W.D. Grover, “Optimal Capacity Placement for Path Restoration in STM or ATM Mesh-Survivable Networks”, IEEE/ACM Transactions on Networking, vol 6, Jun. 1998.

[18] S. Chaudhuri, G. Hjalmtysson, and J. Yates, “Control of lightpaths in an optical network,” Optical Internetworking Forum, Jan 2000.

[19] R. Ramamurthy et al, “Capacity Performance of Dynamic Provisioning in Optical Networks”. IEEE JLT, vol 19, Jan 2001.

[20] L. Berger (Editor), et al., "Generalized MPLS Signaling Functional Description," RFC 3471, January 2003.

[21] L. Berger (Editor), et al., "Generalized MPLS Signaling - RSVP-TE Extensions," RFC 3473, January 2003.

[22] P. Ashwood-Smith and L. Berger (Editors), et al., "Generalized MPLS Signaling - CR-LDP Extensions," RFC 3472, January 2003.

[23] R.D Doverspike and J. Yates “Challenges for MPLS in Optical Network Restoration”, IEEE Communication Magazine, Aug. 1999.

[24] R. Ramamurthy, J-F. Labourdette, S. Chaudhuri, and et al, “Comparison of Centralized and Distributed Provisioning of Lightpaths in Optical Networks,” OFC 2001, Anaheim CA.

[25] J-F. Labourdette, et al, “Routing Strategies for Capacity-Efficient and Fast-Restorable Mesh Optical Networks”, Photonic Network Communications, vol. 4, no. 3/4, pp. 219-235, July/Dec. 2002.

[26] G. Ellinas, et al, “Routing and Restoration Architectures in Mesh Optical Networks”, Optical Networks Magazine, Issue 4:1, Jan/Feb 2003.

[27] S. Datta, et al, “Efficient Channel Reservation for Backup Paths in Optical Mesh Networks”, IEEE GLOBECOM 2001, San Antonio, TX, Nov. 2001.”

[28] E. Bouillet, J-F. Labourdette, G. Ellinas, R. Ramamurthy, and S. Chaudhuri, "Stochastic Approaches to Route Shared Mesh Restored Lightpaths in Optical Mesh Networks", Proc. of IEEE Infocom 2002, New York, NY, June 2002.

[29] E. Bouillet, J.-F. Labourdette, S. Chaudhuri, “Local Optimization of Shared Backup Channels in Optical Mesh Networks,” OFC 2003, Atlanta GA.

[30] E. Bouillet, J. Labourdette, R. Ramamurthy, S. Chaudhuri, “Enhanced Algorithm Cost Model to Control tradeoffs in Provisioning Shared Mesh Restored Lightpaths”, OFC 2002, Anaheim, CA, March 2002.

[31] J. Doucette, W. Grover, T. Bach, “Bi-Criteria Studies of Mesh Network Restoration: Path Length vs. Capacity Tradeoffs”, OFC 2001, Anaheim, CA, March 2001.

[32] D. Brélaz, “New Methods to Color the Vertices of a Graph”. Communications of the ACM, Vol 22, Num 4. April 1979.

[33] S. Segupta, D. Saha, and S. Chaudhuri, “Analysis of Enhanced OSPF for Routing Lightpaths in Optical Mesh Networks,” InfoComm2002, New York.

[34] R. Coltun, “The OSPF Opaque LSA Option,” RFC 2370, July 1998.

[35] H. Liu, et al, “OSPF-TE Extensions in Support of Shared Mesh Restoration”, Internet Draft, work in progress, Oct. 2002.

[36] A. Akyamac, et al, “Ring Speed Restoration and Optical Core Mesh Networks”, NOC’02, Darmstadt Germany, Jun. 2002.

[37] E. Bouillet, et al, “Lightpath Re-optimization in Mesh Optical Networks”, NOC’02, Darsmstadt Germany, Jun. 2002.

[38] H. Liu, et al, “GMPLS-Based Control Plane for optical Networks: Early Implementation Experience,” SPIE ITCom 2002, Boston MA.

[39] C. Qiao, et al “Distributed Partial Information Management (DPIM) Schemes for Survivable Networks - Part I,” IEEE Infocom 2002, New York, NY, June 2002.

[40] Z. Dziong, et al, “Efficient Capacity Sharing in Path Restoration Schemes or Meshed Optical Networks ,” NFOEC, 2002.