On the Tradeoffs between Path Computation Efficiency and Information Abstraction in Optical Mesh Networks

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

Abstract—**Distributed, IP-based control architecture has been proposed for switched optical mesh 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 also offers challenges to path computation, especially for shared mesh restored paths, because of the summarization of link state information disseminated by the routing protocol. This paper discusses the tradeoffs between the path computation efficiency and the abstraction level of link state information. Several heuristic algorithms for computing shared mesh restored paths are described, which require different levels of abstraction and summarization of link resource sharing information. The performance of these algorithms is compared in term of the efficiency of network capacity utilization, the computation complexity, and the amount of required network information. We show that with appropriately aggregated link state information the proposed path computation algorithms are able to utilize the network resource very efficiently.**

Keywords-path computation algorithms; optical networks; shared mesh restoration

I. INTRODUCTION

Traditionally, optical transport networks have been controlled from Element Management or Network Management Systems (EMS/NMS). In this mode of operation, most of the functions related to topology discovery, path computation and connection provisioning are performed in a central manner, often requiring operator intervention and manual configuration. Recently, distributed control architecture has been proposed for optical mesh 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 [1]. Efforts to standardize such a distributed control plane have reached various stages in several bodies such as the Internet Engineering Task Force (IETF) [1], International Telecommunications Union (ITU) [2] and Optical Internetworking Forum (OIF) [3]. The IETF is defining Generalized Multi-Protocol Label Switching (GMPLS) that describes the generalization of MPLS protocols to control not only IP router networks but also various circuit switching networks including optical mesh networks.

With the application of distributed control plane, a connection request can be sent to the ingress node either from a client directly connected to the optical network or by the management plane. The ingress node computes the explicit path and then initiates the path provisioning through signaling protocols such as GMPLS extended RSVP-TE [14]. Path computation must take into account various requirements and constraints, including bandwidth constraints, recovery and survivability, optimal utilization of network resources, etc. 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 optical transport networks [4, 5, 12].

End-to-end path protection and restoration techniques are commonly used in optical transport networks in order to support high service availability and quick recovery from network failure. Dedicated (1+1) protection and shared mesh restoration are two appropriate recovery schemes in the context of mesh networks [6, 7, 8]. In dedicated (1+1) protection, the path computation algorithm computes and establishes the working path and its diverse protection path simultaneously. During normal operation mode, both paths carry the same live traffic. If the working path fails, the receiver simply recovers the signal on the protection path without extra signaling. Path diversity guarantees that the working and protection paths will not simultaneously succumb to the same failure. This scheme offers short recovery time in the event of network failures, but uses a large amount of network capacity.

In shared mesh restoration [7, 8, 9], the path computation algorithm computes the restoration path. The restoration path is pre-provisioned with signaling messages and its resource is reserved along the path. However no cross-connections are created along the restoration path. The complete establishment of the restoration path occurs only after the working path fails, and requires some additional signaling. Since it is only soft reserved, the same reserved network resource can be shared by multiple restoration paths as long as their corresponding working paths are mutually diverse. This scheme allows considerable saving in terms of network capacity required compared to the dedicated (1+1) protection. It can achieve reasonably fast restoration time and guarantee successful recovery from a single failure [10]. Furthermore, the restoration resources can be utilized for lower priority pre-emptible traffic in normal network operating mode. Of course, the restoration path needs to be activated via signaling when the working path fails. It may result in a recovery slower than that of the dedicated (1+1) protection. This paper focuses on the path computation for shared mesh restoration.

Although not necessary, deterministic path computation algorithms are often used in the centralized NMS system to achieve high efficiency of channel utilization [11]. The deterministic algorithms require a detailed level of link resource sharing information to compute a restoration path. The amount of this information is proportional to the number of provisioned paths in the network and could be very large. The deterministic approach is then not practicable for distributed route computation involving large networks. For distributed path computation, typically only aggregated link information is disseminated via the routing protocols in order to reduce overhead and improve routing scalability. Therefore, unlike the centralized management plane, the distributed path computation algorithm at each network element does not have complete information about the network topology and link states. In this paper we describe three algorithms for distributed path computation that use heuristic approach to compute shared mesh restored paths. These algorithms require different levels of abstracted link resource sharing information. We compare the performance of the proposed algorithms in term of the efficiency of network capacity utilization, the computation complexity and the amount of required network information. The tradeoffs between path computation efficiency and the abstraction level of link state information is studied. We also show that compared to the deterministic algorithm using complete network information, the proposed heuristic algorithms incur a relatively small penalty in term of network capacity utilization while they only need reasonable amount of summarized link state information and the computation complexity is greatly reduced.

The paper is organized as follows. Next section describes the details of the heuristic algorithms to compute shared mesh restored paths and the required input information of link resource availability and sharing. In section 3, the performance of path computation algorithms is evaluated. Section 4 concludes the paper.

II. Distributed Path Computation Algorithms

The path computation algorithm for shared mesh restored paths finds a shortest-combined-cost working path and a diversely routed restoration path between the ingress and egress nodes, so that sharing of channels is maximized. The term Shared Risk Link Group (SRLG) is used here to indicate a group of optical lines that share a common risk of failure [10]. Two restoration paths are "compatible" and may share a protection channel if their respective working paths are SRLG disjoint. Otherwise they are said to be "conflicting". For centralized path computation, the path computation algorithm has complete and detailed link resource availability and sharing information and can use the deterministic approach to compute shared mesh restored path. This is proved to be a NP-complete problem if minimization of the total capacity usage (working and restoration) is sought. A possible approach then is to enumerate a list of K minimum cost working paths, and for every one of them compute the corresponding minimum cost restoration path [11]. The path computation algorithm returns a pair of paths with the lowest combined cost. The cost of a pair is the cost of the links along both paths, excluding the cost of links with preexisting shareable reserved channels along the restoration path. Given a working path, we compute the minimum cost restoration path by: (i) set the cost of the links with SRLGs traversed by the working path to infinite. (ii) Set the cost of links with shareable channels to a constant $\varepsilon \ll 1$. (iii) Run a shortest path computation algorithm (e.g. 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.

It does not scale well for large networks in term of required control network bandwidth and memory if the complete link resource availability and sharing information is disseminated to every node. Then, the heuristic approach is used to compute the shared mesh restoration paths. The local database of each node may contain summarized link state information that is necessary to compute the paths using the heuristic approach. For optical cross-connects with hundreds of ports there may be multiple data links between a pair of nodes. In order to improve the routing scalability, 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 into the routing protocol [13].

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 shareable 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 shareable channels on a TE link between two particular nodes, it will reduce the amount of input information necessary to compute the paths.

The distributed path computation algorithms using the heuristic approach can be summarized as follows.

1. Compute k-shortest paths (potential candidates for the working path). Sort the paths by cost and denominate them w_1 to w_k.
2. Set the candidate path set $S=0$.
3. For a candidate working path w_i, assign the cost to each TE link, which 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 link state information, as discussed below.
4. Compute the shortest path s_i (potential candidate for the restoration path) 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 $\{w_i, s_i\}$. 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 link 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 path but are still idle (no restoration occurs); Dcost 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 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 and Dcost, i.e. *Cost = f(#UC, Dcost)*. A simple function can be expressed as *f(#UC, Dcost) =*
- Dcost 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 dedicated (1+1) protection path [11]. Note that unlike the centralized deterministic approach the distributed path computation algorithms only provide the links, not the specific channels, used by the restoration path and the channel sharing assignment can then be done separately as provisioning of the restoration path takes place on a link-by-link basis.

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 = infinite, 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, 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 = $\varepsilon + (1-\varepsilon)P$ if #RC>0, where ε is a small constant and P is the probability that no reserved channel is shareable

Algorithm 3 is based on the probabilistic approach [10]. Evidently the same SRLG cannot be protected multiple times by the same reserved channel, otherwise the contention would exist if this SRLG fails. 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 shareable) 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 shareable channel can be approximately estimated as [10],

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

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.

Note that the difference in the above algorithms and the deterministic algorithm is only in path computation step 3. In the deterministic algorithm the cost of a TE link is set to the default link cost times $\varepsilon << 1$ if it contains a shareable reserved channel and the 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 detailed information to compute the routes. In particular it needs to know whether a SRLG is protected or not for every reserved channel in the network. Whereas in the heuristic approach, only certain aggregated link state information is needed based on the link cost function of the algorithm.

III. PERFORMANCE RESULTS AND COMPARISONS

In this section, we investigate the performance of different path computation algorithms. We'll see that there are tradeoffs among the efficiency of network capacity utilization, the computation complexity, and the amount of required input information for different path computation algorithms.

A. Complexity of Path Computation Algorithms

We first evaluate the computation complexity and the amount of required input information of the distributed path computation algorithms using heuristic approach and the centralized algorithm using deterministic approach 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 path at the time of provisioning before a failure occurs. 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 to recover a service on the pre-computed restoration path when a failure occurs.

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 that 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 of $O(h)$. For the centralized deterministic 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 $O(hx)$ [10]. Note that it is assumed here that a fixed length array is maintained for each reserved channel in which each entry indicates whether a SRLG is used or not. The complexity becomes $O(xhlog(m))$ if instead a variable length list of protected SRLGs is maintained for each reserved channel (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 deterministic algorithm needs to know the list of SRLGs protected by each channel. The size of this information is thus on the order of $O(gh'm)$.

For distributed path computation, a heuristic algorithm proposed in Section II is used at the ingress node to determine the restoration path. The restoration path is specified in a series of node that the path should traverse. 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 shareable reserved channels in the network (i.e. determining the cost for each link) is then $O(m)$ for algorithm 1 and $2O(m)$ for algorithm 2. 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 heuristic 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 shareable reserved channels in the network is $O(hm)$. Algorithm 3 requires #UC, #RC, and number 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 to identify all the links with shareable reserved channels and the amount of required input information to compute the restoration path for the above distributed heuristic path computation algorithms and centralized deterministic algorithm. For the centralized deterministic algorithm, both the complexity and input information amount depend on the number of paths established in the network. However, for the distributed algorithms using heuristic approach, they remain 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 certain performance loss in term of network capacity utilization.

TABLE I. THE COMPLEXITY AND REQUIRED INFORMATION TO COMPUTE THE RESTORATION PATH FOR DIFFERENT ALGORITHMS

Algorithm	Complexity	Quantity of input information	Required input information
Heuristic Alg. 1	O(m)	O(m)	#UC
Heuristic Alg. 2	2O(m)	2O(m)	#UC, #RC
Heuristic Alg. 3	O(hm)	$O(m^2)$	#UC, #RC, # of times a SRLG protected by each TE link
Deterministic Alg.	O(ghh')	O(gmh')	the list of SRLGs protected by each channel

In addition, the distributed path computation needs to assign the reserved channel hop-by-hop during provisioning along the restoration path. To assign 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 for each reserved channel there is 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 a variable length list of protected SRLGs is maintained for each reserved channel (search is required to find whether the SRLG is used or not).

B. Network Utilization of Path Computation Algorithms

We then compare the network capacity efficiency of the centralized algorithm using deterministic approach and the distributed algorithms using heuristic approach. 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 by working paths and reserved for restoration paths) for an algorithm. The relative performance of a heuristic algorithm in term of total channel usage can be obtained by comparing with the deterministic algorithm. We use the average relative channel usage obtained from a heuristic path computation algorithm under various networks as a measurement of the algorithm performance. The results are shown in Fig. 1. They indicate that heuristic algorithm 3 is comparable to the deterministic algorithm. In comparison, heuristic 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 deterministic algorithm.

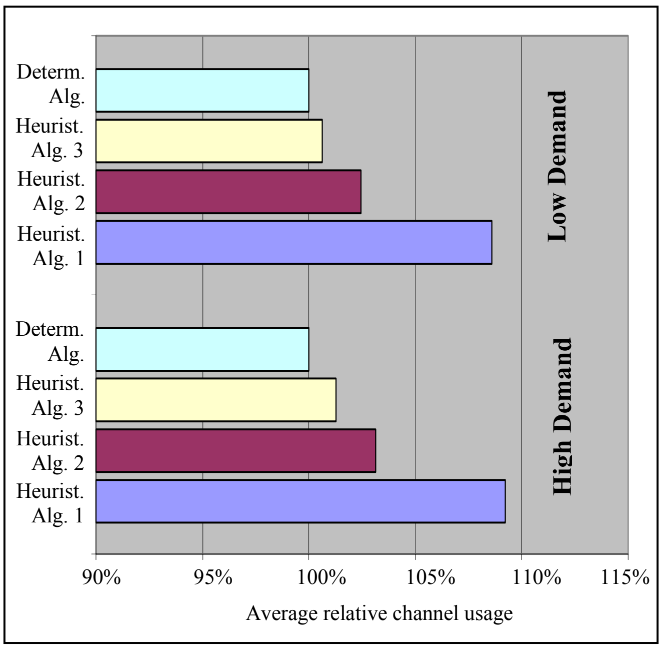

Figure 1. Performance comparison in term of average channel usage for the heuristic path computation algorithms relative to the deterministic algorithm.

IV. 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 levels of summarization of resource information on TE links. The tradeoffs between the path computation efficiency and the degree of summarization and abstraction of link state information are studied. The performance of these distributed algorithms is also compared to that of the centralized path computation algorithm using deterministic approach, which requires the complete and detailed link resource availability and sharing information. It is shown that with carefully aggregated link state information the proposed distributed path computation algorithms are able to utilize the network capacity very efficiently and require much less routing information.

In general, the more detailed information of link resource availability and sharing 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 (available 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, 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).

REFERENCES

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

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

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

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

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

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

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

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

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

[10] 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.

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

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

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

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