Diverse Path Computation Algorithms for Dynamic Service Provisioning with Shared Protection in SDH Networks

O. Manikumar Reddy1, R. Madangopal2, N. Usha Rani2, Timothy A. Gonsalves1

1Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India
{madan,usha}@lantana.tenet.res.in
2NMSWorks Software Pvt. Ltd., Chennai, India
{kumar,tag}@lantana.tenet.res.in

Abstract—SDH is the dominant transport technology for delivery of voice and private-line bandwidth services in access and backbone networks. Dynamic service provisioning requires the use of on-line algorithms which automatically compute the path to be taken to satisfy the given service request. Bandwidth services require two paths, one working path and one protection path, so that it can withstand certain network failures. Shared Risk Link Group (SRLG) has been widely recognized as an important concept in survivable optical networks. For an SRLG survivable service request, dedicated protection or shared protection can be used to establish the service request. In this paper, SRLG-disjoint path computation algorithms are proposed for shared protection that take into account the multiplexing structure defined by SDH. The performance of the algorithms is evaluated. The improved protection does not cause appreciable performance degradation.

I. INTRODUCTION

Service providers are employing diverse technologies like PDH, SDH/SONET, WDM/DWDM, ATM, DSL to fulfill the increasing bandwidth service requirements. With the increasing complexity of network management activities service providers are looking to migrate from manual, static provisioning systems to the more dynamic service-oriented automated provisioning systems to meet customer demands for rapid service turn-up and obtain more customers and maximize revenue opportunities.

Dynamic service provisioning systems (SPS) should be able to communicate with all the network elements in the service path, either directly or through their respective element manager systems. To activate a new service, a network operator selects a pre-defined service profile, inserts a few customer-specific parameters, and presses GO. The provisioning system should do the rest. It should be able to check the network for available resources, determines the optimal path through the infrastructure, configures each device along the path, and updates the network inventory. One important task of SPS is to find two diversely routed paths to satisfy the requested service taking into account the capacity that is already in use, free capacity available such that more future requests can be accommodated and the network resources are utilized optimally.

In transport networks, services require two paths, one working path and one protection path, so that it can recover the traffic from failures. Services are protected using dedicated or shared protection schemes. In dedicated protection each working path is protected by its protection path. In Shared protection one protection path is shared among many working paths such that any one working path failure is overcome by this protection path at the maximum. The requirement for working path and protection path is that they have to be diversely routed so that at least one path can survive a single failure in the network, where a single failure may represent a fiber cut or any failure. Because of the large amount of traffic a fiber carries, a single failure will cause a severe service loss. So, survivability is an important issue.

Shared Risk Link Group (SRLG) [1] has been widely recognized as an important concept in survivable optical networks. A SRLG is a set of network links that will break down simultaneously if a given failure occurs. For example, two fibers in the same conduit would be in the same SRLG. A link may belong to multiple SRLGs. Finding a pair of SRLG disjoint paths is more complicate than finding link or node-disjoint path. The issue is to find a pair of SRLG-disjoint paths for a given request such that the total bandwidth consumption is minimized.

SDH as a transport technology is used by many service providers to carry their main backbone traffic and also to provide bandwidth services to customers in the access side. SDH defines a multiplexing hierarchy [2] by which the low rate G.703 [3] signals such as E1, E4, DS1, DS3 etc. are multiplexed into high rate signals for transmission. We propose heuristic algorithms to find a pair of least cost SRLG-disjoint paths between a source and a destination in SDH optical networks under shared protection.

The rest of the paper is organized as follows. Section II gives details regarding related work. Section III describes the path computation algorithms for SDH networks under shared protection. The performance of the algorithms are analyzed in Section IV. We conclude in Section V.
II. RELATED WORK

The problem of finding diversely routed paths with the SRLG constraint is proved to be NP-complete in [4]. An iterative heuristic for computing SRLG-disjoint paths for dedicated protection is proposed in [5]. Heuristics for computing SRLG disjoint paths under shared protection are proposed in [6], [7]. A failure-dependent SRLG protection scheme where the working path is partitioned into overlapping segments and each individual segment protected using a backup is proposed in [8]. The static provisioning problem incorporating realistic constraints for dedicated, shared and unprotected SRLG-disjoint path protection is considered in [9].

All these works consider a graph of the network topology containing nodes representing the network elements present and links inter-connecting them with integer units of capacity. But when it comes to provisioning in transport networks particularly SONET/SDH networks, links cannot be simply considered to have certain integer units of capacity as the multiplexing structure defined by those technologies has to be followed while provisioning the requested services. Because of this multiplexing structure of SDH, each link cannot be assumed to have simply some integer units of capacity and the free capacity cannot be obtained simply by subtracting the allotted capacity from the maximum capacity and allocations cannot be made until the free capacity becomes zero. In SDH, higher order containers like VC-4 have to be used to create trails (logical connections) between the source and destination nodes before provisioning any bandwidth between two points. This means that even to have an E1 bandwidth which maps to VC-12 in SDH between two points, a single VC-4 trail or a sequence of VC-4 trails have to be available already or have to be established newly. All service requests must fit into one of the standard rates supported by SDH.

Path Computation algorithms taking into account the above constraints have been proposed in [10]. The network is treated as a graph containing physical links and logical trails and dynamic weights are assigned to them before computing a path with the least cost. Weights are assigned such that the trails are given higher preference to physical links so that existing paths are SRLG disjoint. This is commonly refereed as 1+N protection. When computing paths across the SDH network, computation should be aware of links and trails that appear to be disjoint but actually share same risk/failure condition. This is usually achieved by defining a SRLGs and having each link and trail maintaining its membership of the same group.

For computing a SRLG disjoint path pair for service requests in SDH networks under shared protection, the constraints mentioned in [10] are taken into account. So the graph of the network will contain physical links and trails. Since in SDH bandwidth allocations are in terms of only the standard rates supported by SDH and not in integer units of bandwidth, the sharing information is maintained with respect to those rates and not as a whole for the trail.

Two ways of maintaining the sharing information for SDH networks proposed in [11]. They are Complete Information Scenario (CIS) and Minimum Information Scenario (MIS). In CIS, for each VC-4, VC-3, VC-2, VC-12 or VC-11 that is allocated for a protection path, the list of links in the working paths that can be protected by this allocation is maintained. In MIS for each trail, for every standard rate supported by SDH, only the working paths protected by the last allocation of that rate for protection is maintained.

In both the scenarios, the weight of a trail for sharing will be controlled by a factor $\gamma$ which is defined in [11]. A trail which is not SRLG disjoint with the working path should be given less preference while computing the protection path. The weight of a trail $t$ for rate $r$ under shared protection for a working path $p$ is defined as follows:

$$w_{sp}(t, r, p) = \begin{cases} M & \text{if trail } t \text{ is not SRLG disjoint with working path } p \\ \gamma \ast w_t(t) & \text{if for some } a \text{ of rate } r, P_a(t) \cap P_p = \emptyset \\ w_t(t) & \text{otherwise} \end{cases} \tag{1}$$

where $M$ is sum of the weights of all the links and trails in network, $P_a(t)$ indicates the set of links protected by some allocation $a$ of rate $r$ in trail $t$, $P_p$ indicates the links traversed by the working path $p$, $w_t(t)$ indicates the initial weight of the trail $t$. The weight of the trails which are not SRLG disjoint with the working path $p$ is increased by $M$. This is to give less priority to these trails while computing SRLG disjoint path in the graph.

A. Proposed algorithm

The heuristic for computing least cost SRLG disjoint paths taking into account the constraints mentioned above is outlined in Algorithm 1. The input to the heuristic is a graph $G$, a node pair $(s,d)$, rate $r$ and a SRLG set $R$. The heuristic executes $K$ times iteratively. In the $i^{th}$ iteration, the heuristic computes $i^{th}$ shortest path $s_i$ from $s$ to $d$ using Yen’s K shortest path algorithm [12]. A new implementation of the Yen’s algorithm proposed in [13] is used during the implementation which in turn uses one of the algorithms proposed in [10]. The heuristic computes the SRLG disjoint path pair $(p_i,q_i)$ using the heuristic SRLGShortestPath (Algorithm 2) with $s_i$ as seed path. If a protection path $q_i$ cannot be found for the $i^{th}$ shortest path, the algorithm continues with the next iteration. Else if the SRLG-disjoint path pair found in this iteration is of lesser cost compared to the best path pair computed so far, then this is updated as the new best path pair. If the weight of the $i^{th}$ shortest path is greater than the cost of the best path pair
computed so far, then the cost of all further path pairs will be greater than the best one computed so far. This means that the path pair found so far is actually the optimal one and so the algorithm breaks from the loop. Finally, the algorithm returns the best path pair that was computed. If the algorithm had executed less than K times, then it is the optimal path pair else it is not an optimal one.

B. SRLGShortestPath

This algorithm is modified form of algorithm in [5]. The outline of the SRLGShortestPath heuristic is given in Algorithm 2. The input to the heuristic is a graph G, a node pair (s,d), rate r, SRLG set R and seed path p. Given the seed path p between source s and destination d the heuristic returns a pair of SRLG disjoint paths if such an SRLG disjoint path can be computed, else it returns NULL. It first constructs the graph G’ of the graph G based on p. It then removes the links and trails in the forward direction and sets the weight to zero in the reverse direction. It then finds the protection path q using the one of the three algorithms proposed in [10] to which the requested rate and the working path are passed. A modified form of those algorithms which use the weight computation specified in (1) is the one that is used. Depending on whether CIS or MIS is used, the amount of sharing information that will be referred varies. It then removes the overlapping links and trails from the paths p,q and the rest is grouped in to two paths p’ and q’. Finally, it will check the two paths p’ and q’ for SRLG disjointness and paths are returned.

Algorithm 2 SRLGShortestPath(G,s,d,R,rate,p)

1: Copy the graph G to a modified graph G’
2: remove all the links and trails in p
3: remove all the links and trails from the graph G’ that traverse the links traversed by belongs to path p
4: weight(i,j) ← 0, ∀(i,j) ∈ p
5: if A shortest path q in the modified G’ from s to d exists then
6: if capacity not available in the path q then return (∞)
7: end if
8: Take the union of p and q, remove from union the links and trails that are part of both p and q
9: and then group the remaining links and trails into p’ and q’
10: if p’,q’ are SRLG disjoint then
11: Compute the cost of path pair (p’,q’), C(p’,q’)
12: return (C(p’,q’))
13: else
14: return (∞)
15: end if
16: else
17: return (∞)
18: end if

IV. PERFORMANCE EVALUATION AND ANALYSIS

TABLE I

<table>
<thead>
<tr>
<th>Network no.</th>
<th>No. of Nodes</th>
<th>No. of Links</th>
<th>No. of SRLGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>33</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>70</td>
<td>103</td>
<td>20</td>
</tr>
</tbody>
</table>

The performance of the described algorithms are evaluated on three SRLG networks, whose parameters are listed in Table I. Network 1 (Fig. 1) and 2 are taken from [5] and Network 3 is taken from [10]. In Fig. 1 R1, R2, ... , R9 represent SRLGs defined in the network . The service requests generated randomly and the bandwidth requirement is generated in the following proportion VC-4-4%, VC-3-10%, VC-2-6%, VC-12-80%. Since the algorithm uses the algorithms proposed in [10], they are evaluated in terms of the parameters, factors used to evaluate those algorithms.

The requests that are accepted are weighted according to the relative bandwidth to get better indication of performance, This is achieved by keeping VC-12 as the base and multiplying each VC-4 request by 63, VC-3 request by 21 and VC-2 request by 3.

In each experiment the following parameters are considered: 1. Weighted number of service requests accepted. 2. Number of service requests rejected. 3. Number of trails created.
4. Percentage of the total bandwidth consumed to satisfy the accepted requests.

The factor $\alpha$ is some value between 0 and 1 and it refers to the relative weight of a trail. This is used so that the weight of a trail is less than the sum of the weights of the links it traverses so that existing trails are given higher preference to creating new trails. The weight of a link can be thought of as the cost of creating a new trail on that link. $\beta_1, \beta_2$ are the dynamic weight adjustment factors for VC-12 and VC-11 rates. $\beta_1$ is used when both VC-3 and VC-2 need not be broken to accommodate a VC-12 or VC-11 request. $\beta_2$ is used when a VC-3 need not be broken but a VC-2 has to be broken to accommodate a VC-12 or VC-11 request. $\beta_3$ is the dynamic weight adjustment factor for VC-2 requests if a VC-3 has to be broken to accommodate a VC-2 request. All these factors take a value between 0 and 1. The value of $\alpha$ varied from 0.1 to 1.0 with increments of 0.1. The values for $\beta_1, \beta_2$ and $\beta_3$ are varied from 0.3 to 0.7, 0.4 to 0.8 and 0.5 to 0.9 respectively.

Two sharing information scenarios are used (CIS or MIS) and the relative weight for sharing is denoted by $\gamma$. The value of $\gamma$ is varied from 0 to 0.5 with increments of 0.1. The experiments are run for the three networks 10 times each with different sets of service requests. The value for K that was used for Network 1 and Network 2 were 50 and for Network 3 was 100. The results obtained in one of the iterations for the four parameters mentioned above are shown in Figs. 2-5 for Network 1. Similar results were obtained in the other iterations as well. The results obtained with dynamic weight adjustment for values of 0.7, 0.8, 0.9 for $\beta_1, \beta_2$ and $\beta_3$ and for values of 0.6 to 1.0 for $\alpha$ are shown since this combination is comparatively better than the other combinations. The experiments are run on a 2 GHz Pentium 4 machine with 512 MB RAM.

From the results obtained, it is found that CIS is only marginally better (less than 10%) than MIS. This is observed for the networks that were evaluated and for the four parameters mentioned above. This means that maintaining the complete sharing information for all the allocations that have been made until now does not offer a significant advantage compared to maintaining the sharing information only for the last allocation that was made. Since the memory that is required to maintain the complete information will be very large for large high capacity networks, it may not be practically suitable to use. With small sacrifice in performance, MIS can be practically used in those cases. The results are consistent when we vary position of SRLGs in the network. These results are consistent with those obtained in [10]. Similar results were obtained for other two networks as well.

V. CONCLUSIONS

In this paper, path computation algorithms are proposed for service provisioning with shared protection in SDH networks under SRLG constraints. These algorithms take into account the multiplexing hierarchy defined by SDH which imposes restrictions on the allocation of bandwidth.

The algorithms are evaluated for different sets of relative weights for different factors considered. The performance of the minimum information scenario is found to be only slightly lower than that of the complete information scenario. Since the memory requirement for maintaining the complete sharing information is huge for large networks, the minimum information scenario can be used in practical systems. Similar algorithms can be used for dedicated protection.

REFERENCES

Fig. 4. Number of trails created in Network 1

Fig. 5. Bandwidth utilized in Network 1


