

Spare Capacity Assignment in Telecom Networks using Path Restoration*

Jeyakesavan Veerasamy and S.Venkatesan†

Computer Science Program, University of Texas at Dallas
Richardson TX 75083
{veerasam,venky}@utdallas.edu

J.C.Shah

Alcatel Network Systems, 1225 N. Alma Road
Richardson, TX 75081

Abstract

Path restoration scheme can be used in augmenting existing telecommunication networks with adequate spare capacity to achieve a desired level of protection against link failures. Path restoration planning corresponds to the multi-commodity flow problem, which is computationally hard. In this paper, we present the details of an approximation scheme for the path based restoration planning problem and an implementation. We then compare the performance of link and path restoration schemes. These schemes have been incorporated into the network planning tool under development at Alcatel Network Systems.

1 Introduction

Catastrophic consequences of failures in today's high capacity transport networks have made the network survivability one of the prime concerns for the network planners. In this paper, we consider the problem of network planning to cope with single link failures. When a link fails, the traffic going through the faulty link is disrupted. To route the traffic of the faulty link, sufficient spare capacity must exist in the other links. There are several ways of adding spare capacity to the existing links in order to restore traffic in the event of a link failure. The following are the two major techniques used:

1. Link restoration: Link restoration allocates spare capacity to the links so that a faulty link's traffic

can be rerouted through one alternate path using the spare capacities of the links on that path. Thus, the link is viewed as a "black box."

2. Path restoration: In this scheme, each path passing through the faulty link is considered separately and is restored. A "failed path" is a path that goes through a faulty link. One alternate path is found for each failed path and spare capacities are assigned to the links of the alternate path.

A restoration network planning tool that uses link restoration scheme has been developed by Shah [9]. We have enhanced this tool to design spare capacity based on path restoration for single link failures in the given backbone network. The traffic and spare capacities are expressed in units of DS3s. (One DS3 corresponds to 672 circuits [2].) Each node represents high speed digital cross-connect switches (DCS). Digital cross-connects are used for connecting the traffic on one link (incident on that node) with the traffic on other links (incident on that node). Rerouting is achieved by reconfiguring the DCSs that belong to the nodes of the alternate paths. This type of topology (called mesh topology) promotes higher sharing of the spare capacity among various alternate paths and results in a more economical network than ring topology [9]. In this paper, we present the details of an implementation of a path restoration scheme and perform a comparative study between link restoration and path restoration.

1.1 Motivation for Path Restoration

Path restoration requires fewer spares compared to link restoration. We illustrate the potential advan-

*Supported in part by Alcatel Network Systems.

†Also supported by NSF under Grant No. CCR-9110177 and by the Texas Advanced Technology Program under Grant No. 9741-036

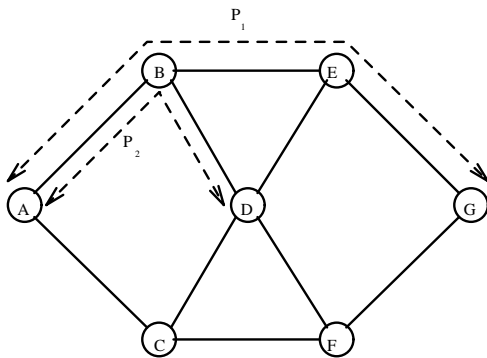


Figure 1: A network with 7 nodes and 10 links

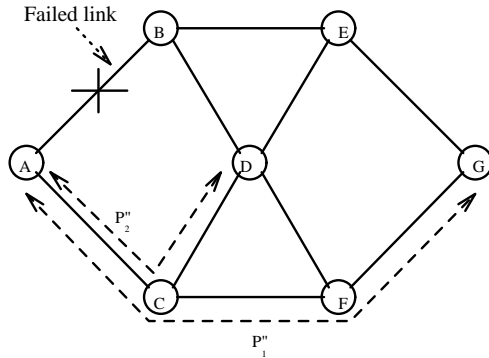


Figure 3: Path Restoration

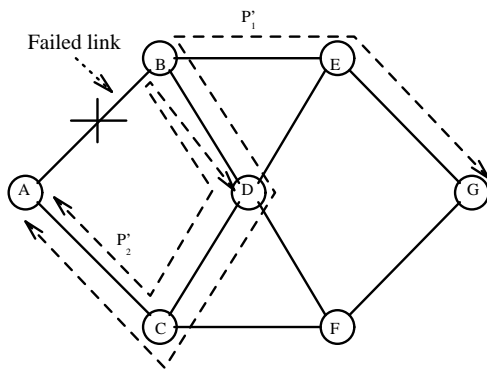


Figure 2: Link Restoration

tages of path restoration by an example. Figure 1 shows a example network and the routing of two paths P_1 and P_2 before any link failure. Assume that the link (A, B) has failed. The resulting network for link restoration is shown in Figure 2 and the same for the path restoration is shown in Figure 3.

In link restoration, a faulty link is replaced by one path. Assume that the faulty link (A, B) is replaced by the path $A-C-D-B$. Then all the traffic carried by the link (A, B) is now rerouted on the path $A-C-D-B$. Correspondingly, the paths P_1 and P_2 , which get disrupted by the failure of (A, B) , are rerouted as shown in the Figure 2. P_2 , which took the path $A-B-D$ before failure, takes the path $A-C-D-B-D$. The traffic of P_2 is routed in (B, D) in both directions (called back haul traffic [7]) and requires more capacity in that link. Link restoration forces all the traffic going through the faulty link to the other end of the faulty link, irrespective of the destination of the failed path. Thus, if a link

(X, Y) fails, then the link (X, Y) is replaced by, say, path $X-Z_1-Z_2 \dots -Z_m-Y$. If $P_i = S-S_1-S_2 \dots -S_j-X-Y-F_1-F_2 \dots -F_k-D$ is a path (going through (X, Y) before failure), then, after (X, Y) 's failure, the restored path for P_i is $S-S_1-S_2 \dots -S_j-X-Z_1-Z_2 \dots -Z_m-Y-F_1-F_2 \dots -F_k-D$. This is done even if one of the Z 's is equal to the destination D .

Link restoration scheme has one advantage over path restoration. The work performed at the time of failure is considerably simple with link restoration. In the case of path restoration, each path going through the faulty link must be considered separately and restored. This involves the execution of a more complex on-line restoration algorithm. However, path restoration is considerably more economical than link restoration. Path restoration takes advantage of the topology of the network and reroutes the paths more efficiently (Figure 3).

Another variation of path restoration is to assign sufficient spares to the links such that for every path, there exists a link disjoint alternate path. So whenever any link on a path fails, traffic of that path is immediately rerouted through the corresponding link disjoint alternate path. But this approach results in more spares and higher cost compared to our implementation. Also this approach fails in many low connected networks. For example, in the network shown in Figure 4, there is no link disjoint alternate path for the path $C-D-E-F-G$. But there exists an alternate path in the event of any single link failure in that path.

1.2 Problem Statement

The path restoration planning problem can be stated as follows: A DS3 backbone network with the traffic in every path and the total equipped capacity in

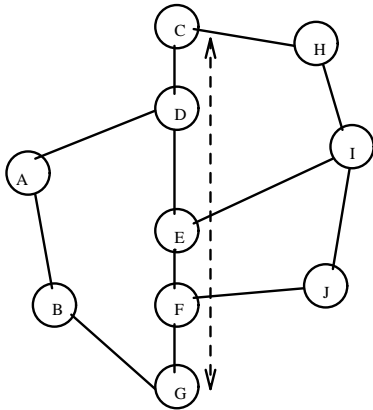


Figure 4: A low connected network

every link is given. Determine the quantity and placement of the spare capacity based on path restoration in a cost effective manner so that adequate spare capacity is available to restore the network in the event of a single link failure.

2 Algorithm

Strict optimal methods to realize the minimum redundancy network are found to be computationally infeasible for moderate size networks. This problem is equivalent to the multi-commodity flow problem, which is NP-complete [4]. So, we design heuristics for placement of spares.

Consider the failure of a link in the given backbone network. Let k be the number of different paths passing through that link. We have to restore these k failed paths by allocating spare capacities in the links used by alternate paths. (The alternate paths may use some of the links of the original paths.) The order in which we choose these k paths may influence the spare capacity network we design. We present six different approaches to determine the order. These approaches are similar to that of link restoration described by Shah [9].

- The paths are chosen in input order.
- The paths are chosen in non-increasing order of the traffic they carry.
- The paths are chosen in non-decreasing order of the traffic they carry.
- The paths are chosen randomly one by one.

{The user has two options: (i) protect¹ all paths, (ii) protect selected paths}

1. Find all the links that carry the paths to be protected.
2. For each such link, perform steps 3 to 7.
3. Mark that link as failed. Find the paths going through that link.
4. For each such path, follow its route information and free all working DS3s it is currently using. (Mark the freed DS3s as spare DS3s).
5. For each such path, find the alternate minimum cost path using Dijkstra's algorithm. For each link used by that alternate path, reduce the number of spare DS3s of that link by the amount of traffic of the path. Also add new spare DS3s if needed.
6. Reset the working DS3s to the original value for all links.
7. Mark the candidate link as usable.

Figure 5: Algorithm *Path_Restore*

- The paths are chosen in non-increasing order of path length.
- The paths are chosen in non-decreasing order of path length.

A restoring path for a failed path is computed such that the cost of allocating spare capacity in this path is the least among all the existing alternate paths. This is computed by using Dijkstra's shortest path algorithm [3].

To protect the network from any single link failure, we repeat the above procedure for all the links. The spare capacity allocated for one link failure can be used for other link failures also. Again the order in which the links are chosen may influence the spare network we design. A strategy similar to that of ordering the paths is adopted. An outline of the algorithm is shown in Figure 5.

Algorithm *Path_Restore* executes Dijkstra's algorithm once for every link of every path (assuming that all paths are protected). Let p be the number of paths, l be the average length (the number of hops) of the paths, e be the number of links and n be the number of nodes in the network. We use the adjacency list for network representation. So the time complexity of the algorithm is $O(pel \log n)$. Note that normally the average path length $l \ll n$.

¹By protecting a path P, we mean allocation spares in the links so that P can be restored using the spares should any link of P fail.

The link restoration algorithm described in [9] executes Dijkstra’s algorithm once for every link, irrespective of the number of paths. So its time complexity is $O(e^2 \log n)$.

3 Implementation

3.1 User Inputs

There are three optionally selectable network optimization criteria provided – cost, distance, and number of hops. The user may choose cost as the optimization criterion when the necessary cost data is available. Distance is used when the cost data is not available but the link distances are known. Number of hops is the chosen criterion if neither cost nor distance data is available.

The user inputs include (i) network optimization parameter options, (ii) link list for the existing backbone network, (iii) path list for the existing backbone network, and (iv) network node definition data.

If the user does not want to restore all the paths, a priority level for each path must be specified. In that case, only paths that have priority level greater than or equal to the user specified priority value are restored. The paths are numbered sequentially from 1 to p , where p is the total number of paths.

3.2 Tool Outputs

The main results that appear in the output are listed below. The tool outputs two sets of following information, one for link restoration, another one for path restoration: (i) total network cost (or DS3 miles) and average number of DS3 miles/link, (ii) total number of DS3s in the network, (iii) total number of DS3 cross-connect ports in the network, (iv) percentage of paths protected, (v) the required spare capacity, transmission cost and transmission equipment details for each link, (vi) the required number of DS3 cross-connect ports and cost for digital cross-connect systems for each node, and (vii) for each link to be protected, a list of alternate paths for each protected path that uses the link.

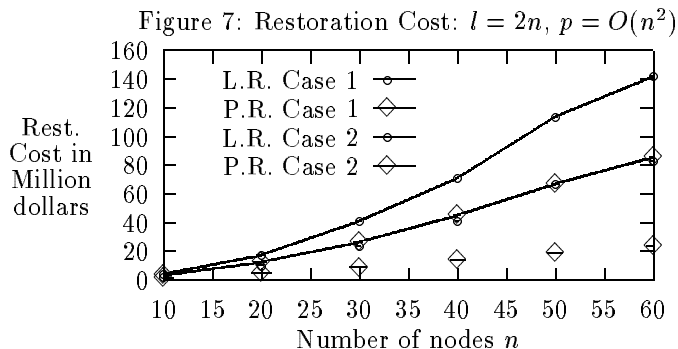
4 Comparative Study

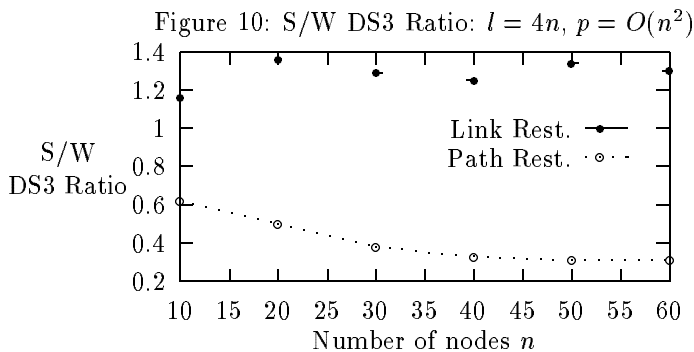
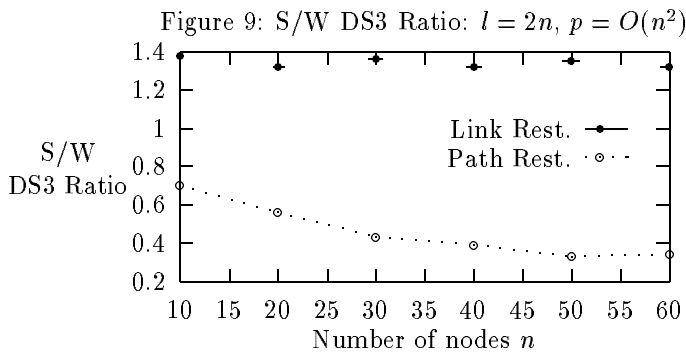
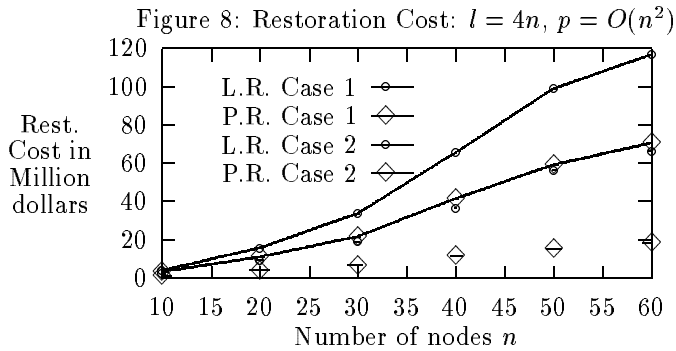
In this section, we analyze the performance of path restoration in comparison with link restoration. A random network generator takes the number of nodes

(N), number of links (L) and number of paths (P) as input from the user, generates a random network, and prints the link, path and node information into a file. This file is used as the input for the tool. Average performance of 10 randomly generated networks was used to determine every data point of the graphs.

The random network generator works in two steps. In the first step, it randomly decides the placement and length of L links in the network. It also ensures that the resulting network is connected, i.e., there exists a path between every pair of nodes. In the second step, the shortest path between every two nodes is found using an all-pairs shortest path algorithm. The probability for each such path (to be included in the path list) is chosen as a value inversely proportional to the number of hops in the path. Based on this probability, P paths are chosen and the traffic for the paths are randomly assigned. Then, required capacity (working DS3s) for each link is calculated by adding the traffic of all paths that are passing through it.

For each network, path and link restoration schemes are used to design spare capacity. **Case 1** represents the restoration cost (in million \$) based on reuse of the existing transmission equipment and new equipment for additional spare capacity. **Case 2** represents the restoration cost (in million \$) based on all new equipment for transmission capacity for the traffic and the spare DS3s without the reuse of existing equipment. From the results of all of these experiments, we found that path restoration is superior to link restoration in all experiments. Particularly, savings (in *Case 1* cost) are very attractive (50% to 80%) for networks with 20 or more nodes (Fig. 7 and Fig. 8). Also *Spare/Working DS3 ratio* for path restoration (Fig. 9 and Fig. 10) decreases as the network size increases, which implies that path restoration results in higher sharing of the spares as the network size goes up.





5 Conclusions

We have studied the advantages of path restoration over link restoration. We have also discussed the design and implementation of a path based restoration scheme. A comparative study was performed between our path restoration scheme and link restoration using 50 randomly created networks. We have found that our scheme results in higher sharing of spares and reduces the network augmentation cost significantly.

Spare capacity assignment problem is important since several real-time restoration algorithms reported in the literature [1, 2, 6, 8] assume that sufficient spares exist in the links for restoration to be possible.

Acknowledgements: We thank Alcatel Network Systems for providing sample data, equipment details, and financial support. Restoration Network Planning Tool described in this paper is a proprietary software of Alcatel Network Systems.

References

- [1] BAKER, J. A distributed link restoration algorithm with robust preplanning. In *Proceedings of IEEE Globecom91* (1991), IEEE, pp. 10.4.1–10.4.6.
- [2] COAN, B. A., VECCHI, M. P., AND T., W. L. A distributed protocol to improve the survivability of trunk networks. In *Proceedings of XIII International Switching Symposium* (1991), vol. IV, pp. 173–179.
- [3] DIJKSTRA, E. A note on two problems in connection with graphs. *Numerische Mathematik 1* (1959), 269–271.
- [4] EVEN, S., ITAI, A., AND SHAMIR, A. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.* 5 (1976), 691–703.
- [5] GROVER, W., BILLODEAU, T., AND VENABLES, B. Near optimal spare capacity planning in a mesh restorable network. In *Proceedings of IEEE Globecom91* (1991), IEEE, pp. 57.1.1–57.1.6.
- [6] KOMINE, H., CHUJO, T., OGURA, T., MIYAZAKI, K., AND SOEJIMA, T. A distributed restoration algorithm for multiple-link and node failures of transport networks. In *Proceedings of IEEE Globecom90* (1990), IEEE, pp. 403.4.1–403.4.5.
- [7] PEKARSKE, R. J. Restoration in a flash – using DS3 cross-connects. *Telephony* (September 1990), 35–40.
- [8] SAKAUCHI, H., NISHIMURA, Y., AND HASEGAWA, S. A self-healing network with an economical spare-channel assignment. In *Proceedings of IEEE Globecom90* (1990), IEEE, pp. 403.1.1–403.1.6.
- [9] SHAH, J. C. Restoration network planning tool. In *Proc. 8th Annual Fiber Optic Engineers Conference* (1992).