Maximum reliable path under multiple failures

SHENGLI YUAN
Department of Computer and Mathematical Sciences
University of Houston – Downtown
One Main St, Houston, Texas 77002, USA
yuans@uhd.edu

Abstract: In this work, we study the problem of maximizing path reliability in mesh networks against simultaneous failures of multiple network links. The links belong to shared risk link groups (SRLGs) that have arbitrary failure probabilities. This problem is NP-hard and we propose a heuristic algorithm for networks with large numbers of SRLGs as well as optimal solutions for networks with smaller numbers of SRLGs. The solutions are evaluated through computer simulations.

Key-Words: path reliability, multiple failures, shared risk link group, routing

1. INTRODUCTION

Recent advances in networking technology have increased the data rate of path connections to 40Gbits/s or even 100Gbits/s [1][2]. With such high data rates, a failure of a communication path can potentially cause significant service disruption to the customers. Therefore it is among the top priorities of network operators to ensure path reliability.

A common approach to achieve high reliability is through path protection [3]. Such schemes provide 100% reliability against single-link failures. However, it is not uncommon that when a network failure occurs, multiple links that belong to the same shared risk link group (SRLG) fail simultaneously [4][5]. In this case, only a working and a protection path that are SRLG-disjoint can survive the failure [6]. However, the study in [7] proved it NP-hard to find SRLG-disjoint working and protection paths. Consequently, attempts were made to minimize the probability of simultaneous failures of the pair. Yuan et al. proved this problem also NP-hard and proposed heuristic algorithms for the special cases in which all SRLGs have equal failure probabilities [8][9]. Various heuristic solutions were also proposed for the general case in which the SRLGs have different failure probabilities [10][11].

In addition to the difficulties in finding the maximum reliable working and protection paths, another limitation of the protection schemes is that they require the reservation of a significant amount of network resources. Shared path protection may be used to improve resource utilization at the expense of signaling and network management [11][12]. Hence, for certain types of traffic and customers, it may be more cost-effective to use a single-path connection without protection, if the reliability of the path can be maximized.

As later shown in Section 2, for networks with single link failures, finding the maximum reliable path can be easily solved. However, the problem becomes NP-hard with simultaneous multiple link failures [8][9]. In this paper, we propose both heuristic and optimal algorithms to solve the problem.

The remainder of the paper is organized as follows. In Section 2, we describe the maximum reliable path problem. In Section 3, we propose an optimal solution for networks with small number of SRLGs. We also propose a heuristic solution for networks with large number of SRLGs. The algorithms are evaluated through computer simulations in Section 4. We conclude the paper in Section 5.

2. PROBLEM DESCRIPTION

The problem is defined as follows. Given network $G = (N, L, S, P_s)$ where N is the set of nodes, L is the set of links (assume they are bidirectional), and S is the set of SRLGs in the network, $P_s = \{p_1, p_2, p_3, \ldots, p_i, \ldots\}$ is the set of non-failure probabilities of each SRLG $s_i \in S$ and $0 < p_i < 1$, also given $S^l\ \forall\ l \in L$ and $S^l \subseteq S$ is the set of SRLGs to which link l belongs, find one path P from source node s to destination node d such that P has the maximum reliability.

The reliability of a path is defined as follows. Let S^P be the set of all SRLGs to which the links of a path P

belong. Then the reliability of P is

$$r^P = \prod_{s_i^P \in S^P} p_i \ , \quad p_i \in P_s \ \text{ and } p_i \text{ is the non-failure probability of SRLG } s_i^P \qquad \text{(Eq. 1)}$$

In this study, we are only considering link failures and not node failures because nodal devices in modern transport networks often have built-in redundancies and are located in well-controlled facilities, which make them least likely to fail.

A single SRLG may contain multiple links while a link may belong to multiple SRLGs. However, we can transform a link belonging to multiple SRLGs into multiple links each belonging to a single SRLG without altering the reliabilities of the paths in the network. This is explained as follows.

Let the set of SRLGs to which link l belongs be $S^l = \{ s_1^l, s_2^l, \ldots, s_i^l, \ldots \}$ and $|S^l| > 1$. Now replace l with $|S^l|$ concatenating links $l_1, l_2, \ldots, l_i, \ldots$, each belongs to a unique SRLG $s_1^l, s_2^l, \ldots, s_i^l, \ldots$ from S^l. The reliability of l_i is the non-failure probability of s_i^l, i.e., p_i . Thus the reliability of the combined links l_1-l_2- …-l_i- … is $p_1 \times p_2 \times \ldots \times p_i \times \ldots$, which is the same as the reliability of l. Also because there is no other links branching off from these links, a path going through one of the links must also go through all of them. Therefore their effects on the path reliabilities are the same as that of l. Consequently, without the loss of generality, we assume for the remainder of the paper, all links each belongs to a single SRLG, i.e., $|S^l| = 1 \ \forall \ l \in L$.

3. ANALYSIS AND SOLUTIONS

In the special case in which every link belongs to a unique SRLG, the maximum reliable path can be readily obtained using the following algorithm which is similar to Dijkstra's Algorithm:

Algorithm OA-1

```
Notations
    s_{i-j}:  the SRLG to which link <i,j> belongs;
    p_{i-j}:  the non-failure probability of s_{i-j};

Step 1:  initialize set ND to contain all nodes except s;
Step 2:  initialize array R[] so that R[u] = p_{su} if u is
         adjacent to s; and 0 otherwise;
Step 3:  initialize entries of array PR[] so that PR[u]
         is assigned s if u is adjacent to s; NULL
         otherwise;
Step 4:  initialize array of SRLG sets S[] so that S[u]
         = ∅ ∀ u ∈ N;
Step 5:  while (ND ≠ ∅) {
             choose a node u from ND such that R[u]
             is maximum;
             if (R[u] is 0)
                 output "no path exists" and quit;
             delete u from ND;
             S[u]=S[PR[u]] ∪ {s_{PR[u]-u}};
             if (u is d) {
                 generate the preferred path from
                 PR[] as:
                 s - … - PR[PR[u]] - PR[u] - u;
                 quit;
             }
             for (each node v that is adjacent to u) {
                 if (v ∈ ND) {
                     if (s_{u-v} ∈ S[u])
                         p = R[u];
                     else
                         p = R[u] × p_{u-v};
                     if (p > R[v]) {
                         PR[v] = u;
                         R[v] = p;
                     }
                 }
             }
         }
```

The running time is the same as that of the Dijkstra's Algorithm, i.e., $O(|N|\log|N|)$ [13].

If each SRLG contains multiple links that may fail simultaneously, we propose an optimal solution (OA-2) that has running time polynomial to the number of network nodes and exponential only to the number of SRLGs.

OA-2 utilizes two subfunctions: GenB() and GenR(). Gen$B(S_x)$ takes a subset S_x of S as the input. It generates an integer number whose binary representation has $|S|$ bits. Each of the bits represents a SRLG in S. If SRLG $s_i \in S_x$, then bit b_i is set to 1; otherwise to 0.

GenR(B) uses Eq. 1 to compute the reliability of a path whose links belong to the SRLGs represented by the bits in the binary representation of integer B. The details of OA-2 are described as follows:

Algorithm OA-2

```
Notations
    s_{i-j}:  the SRLG to which link <i,j> belongs;

Step 1:  create a 2-dimensional array T[2^{|S|}][|N|] of
         structure {int p_node;int p_srlg;bool status;};
```

```
Step 2:   for (every element of T) {
              p_node = UNKNOWN;
              p_srlg = 0;
              status = FALSE;
          }
Step 3:   for (every node, u, that is adjacent to s) {
              B_{s-u} = GenB({s_{s-u}}) ;
              T[B_{s-u}][u].p_node = s;
          }
Step 3:   continue = TURE;
          while (continue) {
              continue = FALSE;
              for (every index i from 1 to 2^{|S|}-1) {
                  for (every index j from 0 to |N|-1) {
                      if ( T[i][j].p_node ≠ UNKNOWN and
                      T[i][j].status = FALSE) {
                          continue = TRUE;
                          T[i][j].status = TRUE;
                          for (every node u that is adjacent
                          to  node j) {
                              B_{j-u} = GenB({s_{j-u}});
                              k = i OR B_{j-u}; //Bitwise OR
                              T[k][u].p_srlg = i;
                              T[k][u].p_node = j;
                          }
                      }
                  }
              }
          }
Step 4:   r_m = 0;
          for (every index i from 1 to 2^{|S|}-1) {
              if ( T[i][d].node ≠ UNKNOWN and
                      T[i][d].status = true) {
                      if (GenR(i) > r_m) {
                          r_m = GenR(i);
                          s_m = i;
                      }
              }
          }
          if (r_m == 0)
              print "path does not exist" and quit;
Step 5:   print d;
          u = s_m;
          v = d;
          while (u ≠ s ){
              print T[u][v].p_node;
              next_u = T[u][v].p_srlg;
              next_v = T[u][v].p_node;
              u = next_u;
              v = next_v;
          }
```

The running time of this algorithm is $O((N^2 2^{2|S|})$ which is exponential only to $|S|$, i.e., the number of SRLGs. Hence OA-2 is suited for networks that have small numbers of SRLGs (e.g., 20 or less). For networks with arbitrarily large numbers of SRLGs, we propose a heuristic solution. The heuristic algorithm (HA-1) executes OA-1 repeatedly while trying to generate more reliable paths. The details are as follows:

Algorithm HA-1

```
Step 1:   run OA-1 to find a path P between s and d;
          if (failed)  quit;
          compute P's reliability r^P using (Eq. 1);
          r' = r^P;
          P' = P;
          S' = S;
Step 2:   s' = null;
          for (every SRLG s_i ∈ S') {
              set the non-failure probability of s_i to 1;
              run OA-1 to find a new path P_i ;
              restore the non-failure probability of s_i ;
              if (succeeds in generating P_i) {
                  compute reliability r_i^P using (Eq. 1);
                  if (r_i^P > r') {
                      r' = r_i^P ;
                      P' = P_i ;
                      s' = s_i ;
                  }
              }
          }
          set the non-failure probability of s' to 1;
Step 3:   while (s' ≠ null) {
              S' = S - s' ;
              repeat Step 2;
          }
          return P' ;
```

Step 1 repeats OA-1. It generates a path which we use as the lower bound to generate more reliable paths in the next two steps. In Step 2, we select one SRLG at a time from all SRLGs in the network, set its non-failure probability to 1, and generate a new path between s and d. The intent is to determine whether we can increase the path reliability if we make the links belonging to one SRLG more attractive than others. After going through all the SRLGs, we identify a SRLG which leads to the highest reliability of a path. We set its non-failure probability to 1 before going to Step 3. The running time of this step is $O(|S||N|log|N|)$. Step 3 repeats the previous step until no new path can be found with greater reliability. The maximum number of iterations in this step is $|S|$, which results in a total running time of $O(|S|^2 |N|log|N|)$ for the entire algorithm.

The performances of the algorithms are evaluated by computer simulations in the next section.

4. SIMULATIONS

We use LEDA programs to randomly generate network graphs for the simulations [14]. The network size ranges from 10 to 40 nodes. The nodal degree ranges from 2.6 to 3.0. The number of SRLGs in each network ranges from 2 to 10. The non-failure probabilities of the SRLGs are uniformly distributed in the range from 0.9000 to 0.9999. Each link is assumed to support an unlimited number of paths and is assumed to belong to any SRLGs with equal probability.

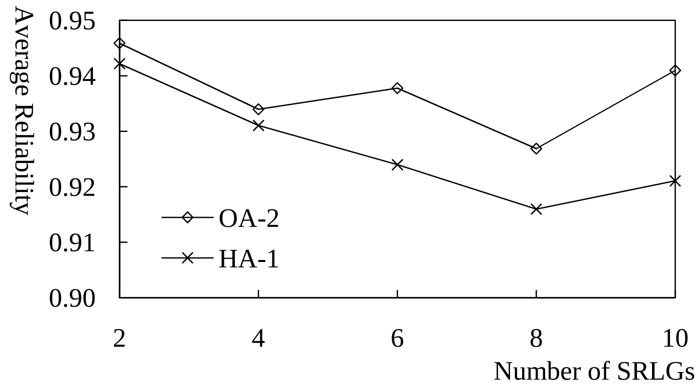
The optimal solution OA-2 and the heuristic solution HA-1 are evaluated. For all pairs of end nodes, we ran the algorithms to obtain the maximum reliable paths. We then compared the average reliabilities of the paths. Two sets of the results are depicted in Fig. 1 and Fig. 2. Results from other network topologies are similar.

The simulations demonstrate that the performance of the heuristic solution is very close to that of the optimal solution. In the worst case, the average reliability of the path generated by HA-1 is only 1.8% lower than that of the optimal paths generated by OA-2.

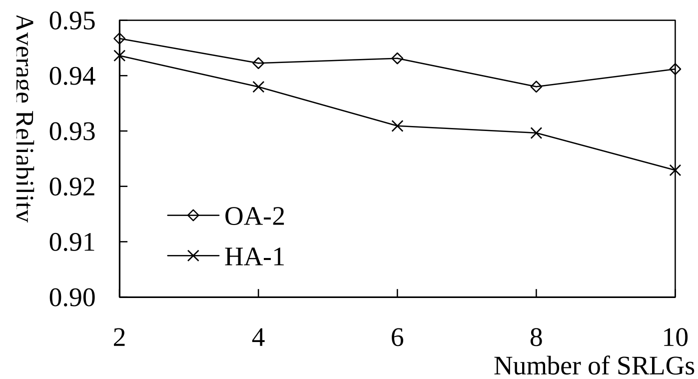
We note that the number of SRLGs and their non-failure probabilities have a significant impact on the average reliabilities of the paths. The reliabilities improve along with the increase of the non-failure probabilities of the SRLGs. In addition, when the number of SRLGs is small, a path is more likely to consist of links that belong to fewer SRLGs. Consequently, from (Eq. 1), the path has higher reliability if the non-failure probabilities of the SRLGs do not vary significantly. As the number of SRLGs increases, the links of a path are more likely to belong to a bigger variety of SRLGs, which reduces the reliabilities of the paths.

We also note that, as the nodal degree increases, the average reliabilities of the paths increase. The reason for this behavior is that an increase in nodal degree results in higher average number of links belonging to each SRLG, thereby making a path more likely to consist of links that belong to fewer SRLGs.

The network topology and the size of the network also affect the reliabilities of the path. Larger networks with more nodes result in a higher average hop count for paths; hence, for the same nodal degree and the number of SRLGs, the links of the paths in a network with more nodes will belongs to a greater number of SRLGs than paths in a network with fewer nodes, which reduces the reliabilities of the paths in a larger network.

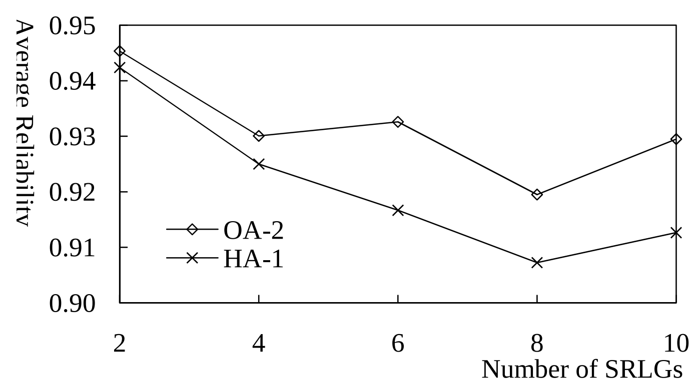


a. Nodal degree = 2.6

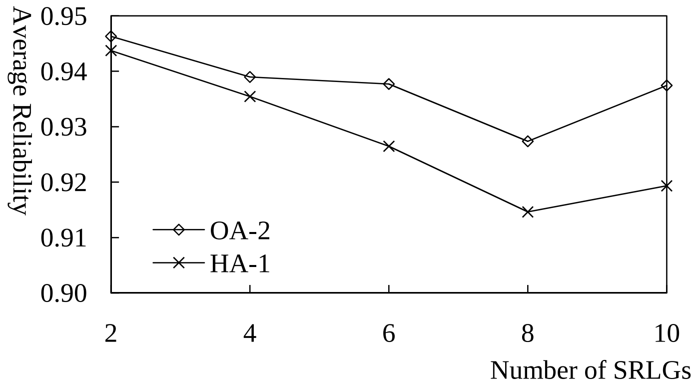


b. Nodal degree = 3.0

Fig. 1. Average Path Reliability vs. Number of SRLGs. Number of nodes = 20



a. Nodal degree = 2.6



b. Nodal degree = 3.0

Fig. 2. Average Path Reliability vs. Number of SRLGs. Number of nodes = 40.

5. Conclusions

In this paper we discussed the problem of finding the maximum reliable path between two nodes under multiple failures. For arbitrarily large networks that contain a small number of SRLGs, we proposed an optimal solution with running times that are polynomial to the network size and exponential to the number of SRLGs. For networks that contain a large number of SRLGs, we proposed a heuristic solution with running times that are polynomial to both the network size and the number of SRLGs. Despite the simplicity of the heuristic algorithm, computer simulations demonstrated that it yields solutions that are close to optimal.

References:

[1] T. Wuth, M.W. Chbat and V.F. Kamalov, “Multi-rate (100G/40G/10G) Transport over Deployed Networks,” *Proceedings, Fiber communication/National Fiber Optic Engineers Conference 2008*, February, 2008, pp. 1-9.

[2] D.C. Lee, “100G and DWDM: Application Climate, Network and Service Architecture,” *Proceedings, Fiber communication/National Fiber Optic Engineers Conference, 2008,* February, 2008, *pp. 1-3*.

[3] T. Wu, *Fiber Network Survivability*, 1st ed, Artech House, 1992.

[4] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks,” *Journal of Lightwave Technology*, Vol. 21, 2003, pp. 870–883.

[5] W. D. Grover, *Mesh-based Survivable Transport Networks: Options and Strategies for , MPLS, SONET and ATM Networking*, Prentice Hall PTR, 2003.

[6] P. Sebos, J. Yates, G. Hjalmtysson, and A. Greenberg, “Auto-discovery of shared risk link groups,” in *Fiber Communication Conference*, 2001 OSA Technical Digest Series (Society of America, 2001), paper WDD3.

[7] S. Yuan and J. P. Jue, “Dynamic path protection in WDM mesh networks under risk disjoint constraint,” *Proceedings of IEEE Globecom 2004,* Dallas, TX, November 2004, pp. 1770–1774.

[8] S. Yuan, S. Varma and J. P. Jue, “Path Routing for Maximum Reliability in Mesh Networks,” *Journal of Networking (OSA)*, Vol. 7, No. 5, May 2008, pp. 449-466.

[9] S. Yuan, S. Varma and J. P. Jue, “Minimum Color Problem for Reliability in Mesh Networks,” *Proceedings, IEEE INFOCOM 2005*, Vol. 4, Miami, FL, March 2005, pp. 2658- 2669.

[10] C. Huang, M. Li and A. Srinivasan, “A Scalable Path Protection Mechanism for Guaranteed Network Reliability under Multiple Failures”, *IEEE Transactions on Reliability,* Vol.56, Issue 2, June 2007, pp. 254-267.

[11] L. Guo, “Heuristic Survivable Routing Algorithm for Multiple Failures in WDM Networks,” *2nd IEEE/IFIP International Workshop on Broadband Convergence Networks, BcN '07*, Munich, Germany, May 2007.

[12] S. Yuan and J. P. Jue, “Shared protection routing algorithm for network,” *Networks Magazine,* Vol. 3(3), 2002, pp. 32–39.

[13] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed., McGraw Hill, 2001.

[14] Algorithmic Solutions Software GmbH, http://www.algorithmic-solutions.com/leda/index.htm