ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

An Approach for Implementing Dual Link Failure Resiliency through Backup Link Mutual Exclusion

Ashok kumar velpuri*
K L University, Greenfields,
Vaddeswaram,Guntur District,A.P,India
ashokkumarvelpuri@gmail.com

Sreenivas velagapudi
K L University, Greenfields,
Vaddeswaram,Guntur District,A.P,India
velagapudisreenivas@gmail.com

Abstract: In every network we see the link failures are common, for this purpose networks having the scheme to protect their links against the link failures. Link protection helps fast recovery from link failures .Existing schemes either pre-reserve two backup paths for each demand or compute new backup paths for unprotected demands after the first link failure occurs. Both approaches require a large amount of backup capacity. In this paper, we propose a capacity efficient hybrid protection/restoration scheme for handling two-link failures. The protection component reserves backup capacity intelligently to ensure the majority of the affected demands can be restored using the pre-planned backup paths upon a two-link failure. A remarkable feature of our approach is that it is possible to trade off capacity for restorability by choosing a subset of double-link failures and designing backup paths using our algorithm for only those failure scenarios. In this paper we discus implementation issues of dual link resiliency system along with simulations.

In this we use Backup link mutual exclusion(BLME), when the links fail simultaneously. The solution methodologies for BLME problem is 1).for mulating the backup path selection as an integer linear program;2)developing a polynomial time heuristic based on minimum cost path routing

Key Words: Optical networks, link protection, link failures, backup link mutual exclusion

I. INTRODUCTION

The growing transmission speed in the communication networks calls for efficient fault-tolerant network design. Current day's backbone networks use optical communication technology involving wavelength division multiplexing (WDM). One of the most gifted concepts for high capacity communication systems is wavelength division multiplexing (WDM). Each communication channel is allocated to a different frequency and multiplexed onto a single fibber. At the destination wavelengths are spatially separated to different receiver locations. In this configuration the high carrier bandwidth is utilized to a greater level to transmit multiple optical signals through a single optical fibre.

Optical networks at present operate in a circuit switched way as optical header processing and buffering technologies are still in the in the early hours stages of research for wide-scale commercial deployment. Protecting the circuits or connections established in such networks against single-link failures may be achieved in different ways:

a. ***Path Protection*:** Path protection is having the capability to protect one or more peer-to-peer paths via a predetermined or pre-established backup tunnel. This is for all time peer-to-peer protection and is similar to the shadow PVC model often used in the ATM networks. The backup tunnel is link and node diverged from the primary tunnel, such that if any element (link or node) along the primary path fails, the head end reroutes the traffic onto the backup path. Many schemes for backup can be used, such as 1 to N or 1 to 1. In the 1-to-N scheme, there is one backup tunnel for N primary tunnels between the same pair of routers. The 1-to-1 back up implies that for every primary tunnel a backup tunnel exists. The number of backup tunnels needed for path protection is twice the number of primary tunnels. The past is referred to as failure independent path protection (FIPP) while the latter is referred to as failure-dependent path protection (FDPP).

b. ***Link protection*:** As clear by the name itself, link protection involves protecting against link failures[1]. These days, links have become more reliable, but statistics still show that most unplanned failures in the network occur because of link. failures. So, protecting against link failures is necessary in any network. To protect against link failures it can use multiple circuits or SONET APS protected circuits. This can result in expensive circuits. Because providing circuits is usually a recurring cost especially if the fiber circuit is not owned by the carrier you might want to reduce the operating cost by eliminating the redundant circuits if fast reroute of traffic can be done by using other paths in the network. Link protection enables you to send traffic to the next hop on a backup tunnel should the primary link fail. Off-course link protection does not work if the only means of reaching the next hop is through the primary link (singly connected cases). Link protection reduces the communication requirement as compared to path protection, so providing fast recovery. On the other hand, the downside of link protection is that its capacity requirement is higher than that of path protection, explicitly when protection is employed at the connection granularity [2].

c. ***Node protection:*** In link protection, the backup tunnel is always set up to the next hop node and the failure detection is performed based on loss of carrier or SONET alarms. In node protection, the mechanism described is similar to the link protection except that the backup tunnel is always set up to the node beyond the next hop that is, next-next hop. Upon detection of failure via a hello timeout, the point of local repair (PLR) node reroutes traffic onto the backup tunnel to the next-next-hop (nnhop). However, when MPLS packets emerge at the tail of the nnhop backup tunnel, they might not have the right labels for the merge point to carry the traffic further. To avoid discarding traffic at the tail of the backup tunnel, the head of the backup tunnel (also known as the point of local repair) swaps the primary tunnel label to the label expected by the merge point and then imposes the backup tunnel label. This ensures that the MPLS packets coming out of the backup tunnel carry the correct labels and hence are switched to the correct destination.

Algorithms for protection against link failures have traditionally considered single-link failures [3]–[5]. However, dual-link failures are becoming more and more important due to two reasons. First, links in the networks share resources such as conduits or ducts and the failure of such shared resources result in the failure of multiple links. Second, the average repair time for a failed link is in the order of a few hours to few days [6], and this repair time is satisfactorily long for a second failure to occur. Although algorithms developed for single-link failure resiliency is shown to cover a good percentage of dual-link failures [7]–[10], these cases often include links that are far away from each other. Considering the fact that these algorithms are not developed for dual-link failures, they may provide as an alternative to recover from independent dual-link failures.

However, reliance on such approaches may not be preferable when the links close to one another in the network share resources, leading to correlated link failures.

Dual-link failures may be modeled as shared risk link group (SRLG) failures. A connection established in the network may be given a backup path under every possible SRLG failure. This approach assumes a precise knowledge of failure locations to re-configure the failed connections on their backup paths. An alternative is to protect the connections using link protection, where only the nodes adjacent to the failed link (and those involved in the backup path of the link) will perform the recovery. The focus of this paper is to protect end-to-end connections from dual-link failures using link protection.

II. DUAL-LINK FAILURE RESILIENCY WITH LINK PROTECTION

Assume that two links, *l* and *l*’, failed one after the other (even if they happen together, assume that one failed first followed by the other) in a network. The backup path of the first failed link is analogous to a connection (at the granularity of a fiber) established between two nonadjacent nodes in the network with link removed. The connection is required to be protected against a single-link failure. Therefore, strategies developed for protecting connections against single link failures may be directly applied for dual-link ailures that employ link protection to recover from the first failure. Dual-link failure resiliency strategies are classified based on the nature in whichthe connections are recovered from first and second failures. The recovery from the first link failure is assumed to employ link protection strategy. Fig. 1 shows an example network where link 1-2 is protected by the backup path 1-3-4-2. The second protection strategy will refer to the manner in which the backup path of the first failed link is recovered.

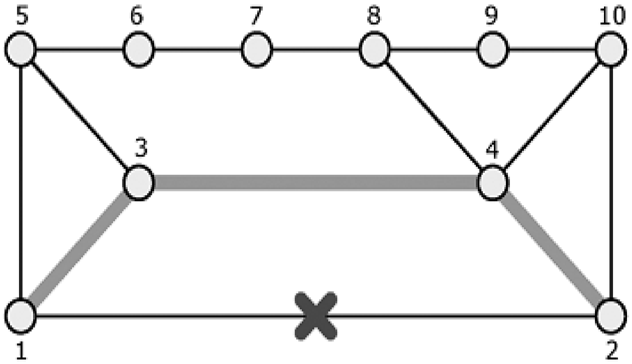

Figure.1: Link 1-2 Protected by Backup Path 1-3-4-2 when Failed

A. Link Protection—Failure Independent Protection (LPFIP):

One approach to dual-link failure resiliency using link protection is to compute two link-disjoint backup paths for every link. Given a three-edge-connected network, there exists three link-disjoint paths between any two nodes [11]. Thus, for any two adjacent nodes, there exists two link-disjoint backup paths for the link connecting the two and *B'l* denote the two link-disjoint backups for link *Bl*. If any link in the backup path *Bl* fails, the backup path of will be reconfigured to *B'l*. Hence, the nodes connected to link *l* must have the knowledge of the failure in its backup paths (not necessarily the location).

III. BACKGROUND AND PRIOR WORK

A network must be three-connected for it to be resilient to any two arbitrary link failures, irrespective of the protection strategy employed. In [12] and [13], a heuristic solution to the BLME problem for arbitrary dual-link failures is developed. In [14], a polynomial time algorithm is developed for solution to the BLME problem considering only adjacent link failures. To the best of our knowledge, there is no prior work that establishes the existence of a solution to the BLME problem.

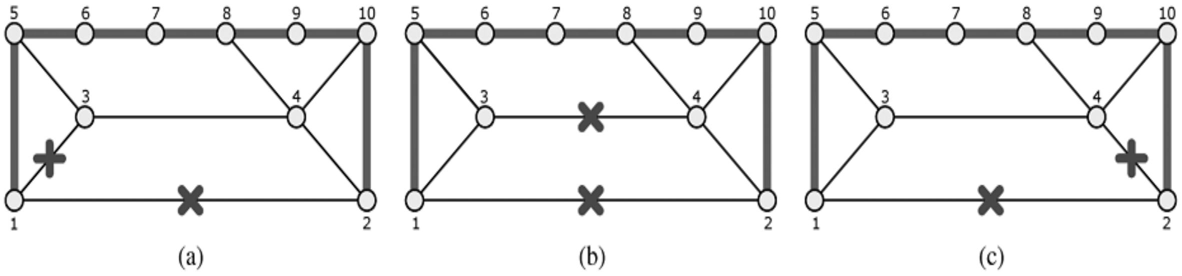

Figure.2. Dual-link failure resiliency using LP-FIP. Backup path after the second failure remains the same irrespective of the failure.

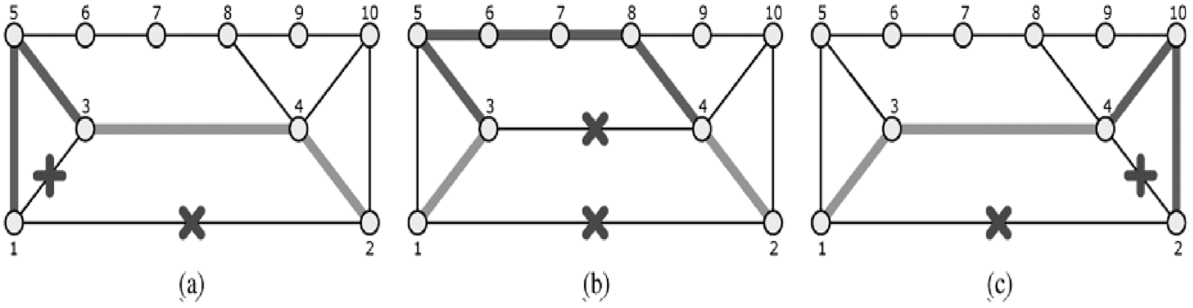

Figure.3. Dual-link failure resiliency using LP-FDP.

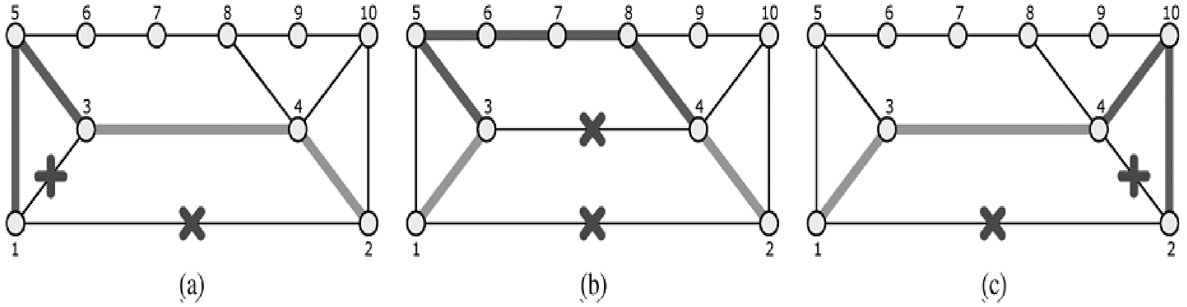

Figure.4. Dual-link failure resiliency using LP-LP

IV. PROPOSED SYSTEM ARCHITECTURE

Fig.5 shows Class Diagram of Dual Link failure resiliency system. Aim of our system is to control dual link failures in a network and provide reliable service. Fig 6 shows Sequence of Dual Link failure resiliency system.

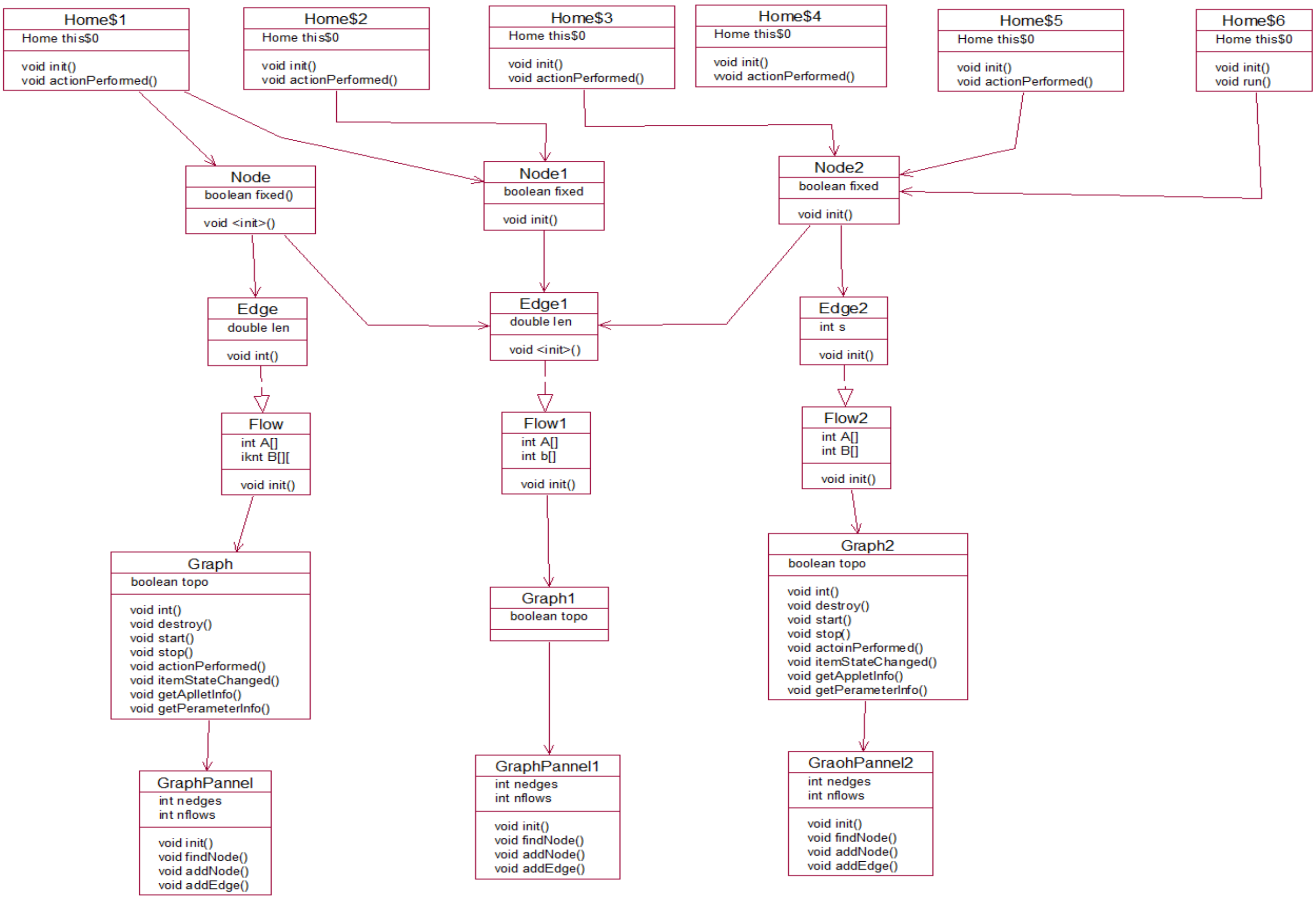

Figure 5: Class Diagram of Dual Link failure resiliency system

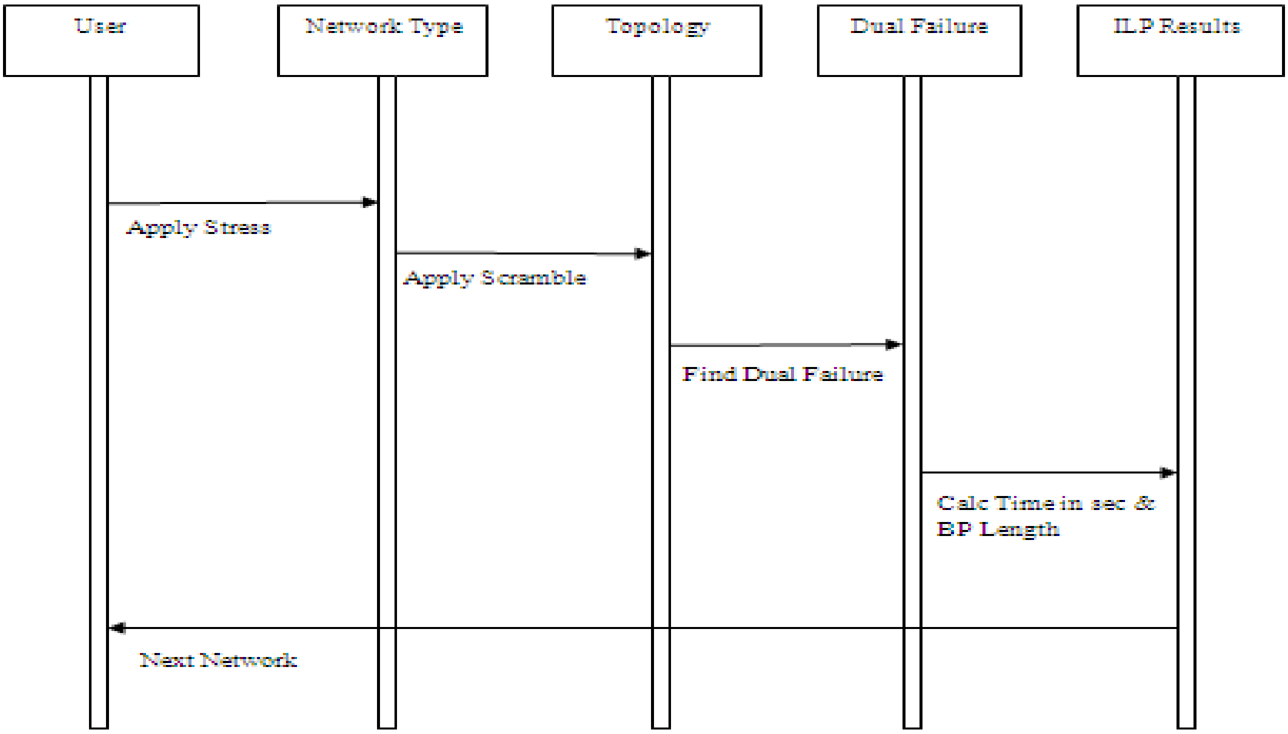

Figure 6 Sequence of Dual Link failure resiliency system.

V. MODULE DESCRIPTION

A. Module 1:

Containment hierarchy is a tree of components that has a top-A level container as its root. Each GUI component can be contained only once. If a component is already in a container and try to add it to another container, the component will be removed from the first container and then added to the second.

B. Module 2:

Application of events and positioning of components

a. Create the nodes in different positions and apply different colors.
b. Create the distance between the nodes by applying stress.
c. Apply different mouse events to the nodes.
d. Using grouplayout of JFreechart all the nodes are positioned. By using virtual and horizontal position and parallel group, all the nodes are positioned.

C. Module 3:

Calculating TIS and BP length

a. By clicking any node, the data transfer to the next node and dual failure are shown in red color.
b. For this dual failure, the backup path is shown in red color.
c. The Time in seconds and BP length is displayed.

In this paper heuristic are applied only for six different types of networks [11] that are shown here.

VI. SAMLE CODE

```
  Class GraphPanel2 extends Panel implements Runnable,
MouseListener, MouseMotionListener
  {
   Graph2 graph;
   int nnodes;
   Node2 nodes[] = new Node2[60];
   int nedges;
   Edge2 edges[] = new Edge2[100];
   int nflows = 0;
   Flow2 flows[] = new Flow2[3540]; // =60*59
   int C[][] = new int[200][200];
   int dmdm[][] = new int[60][60];
   Thread relaxer;
   boolean stress;
   boolean random
 GraphPanel2(Graph2 graph)
 {
   this.graph = graph;
   addMouseListener(this);
 }
 int findNode(String lbl)
 {
   for (int i = 0 ; i < nnodes ; i++)
    {
   if (nodes[i]!=null)
    {
        if (nodes[i].lbl.equals(lbl))
        {
         return i;
        }
    }
  }
  return addNode(lbl);
 }
 Int  addNode(String lbl)
 {
   Node2 n = new Node2();
 n.x = 10 + 380*Math.random();
   n.y = 10 + 380*Math.random();
   n.lbl = lbl;
  nodes[nnodes] = n;
  return nnodes++;
      }
void addEdge(String from, String to, int len)
{
  Edge2 e = new Edge2();
  e.from = findNode(from);
  e.to = findNode(to);
          e.len = len;
          edges[nedges++] = e;
     }
  void addFlow(int from, int to, String str1, String str2)
  {
         Flow2 f = new Flow2();
          f.A = new int[nedges];
          f.B = new int[nedges];
          f.from = from;
          f.to = to;
          dmdm[from][to] = nflows;
          dmdm[to][from] = nflows;
          for (int k=0; k<nedges; k++)
          {
           f.A[k] = Integer.valueOf(str1.substring(k,k+1)).
  intValue();
f.hopA += f.A[k];
  f.B[k] =
  Integer.valueOf(str2.substring(k,k+1)).intValue();
  f.hopB += f.B[k];
  }
  flows[nflows] = f;
  nflows ++;
  }
  void showFlow(Node2 n1, Node2 n2)
  {
    int i1 = findNode(n1.lbl);
    int i2 = findNode(n2.lbl);
    int fn = dmdm[i1][i2];
    Flow2 f = flows[fn];
    for (int i = 0 ; i < nedges ; i++)
     {
           if (f.A[i]==1 )
           edges[i].color = workColor;
           else if (f.B[i]==1)
          edges[i].color = backupColor;
          else
          edges[i].color = arcColor1;
      }
          offgraphics.drawImage(offscreen, 0, 0, null);
         double m=Math.random()/10;
          String s=Double.toString(m);
          String s1=s.substring(0,6);
              graph.t1.setText(s1);
              String s2=s1.substring(4,5);
              int leng=Integer.parseInt(s2);
              leng=leng*50;
              graph.t2.setText(""+leng);
          }
  public void run()
  {
        Thread me = Thread.currentThread();
         while (relaxer == me)
        {
          relax();
                  if (random && (Math.random() < 0.03))
                 Node2 n = nodes[(int)(Math.random() *
      nnodes)];
                  if (!n.fixed)
                  {
                    n.x += 100*Math.random() - 50;
                 n.y += 100*Math.random() - 50;
            }
                 }
                 try
                {
                   Thread.sleep(100);
                 }
               catch (InterruptedException e)
               {
                 break;
               }
            }
      }
```

VII. CASE STUDY

To demonstrate the efficiency of the pattern we took the profiling values using the Netbeans IDE and plotted a graph that shows the profiling statistics when the pattern is applied and when pattern is not applied. This is shown in figure 7.Here X-axis represents the runs and Y-axis represents the time intervals in milliseconds. Below simulation shows the graphs based on the performance of the system if the pattern is applied then the system performance is high as compared to the pattern is not applied.

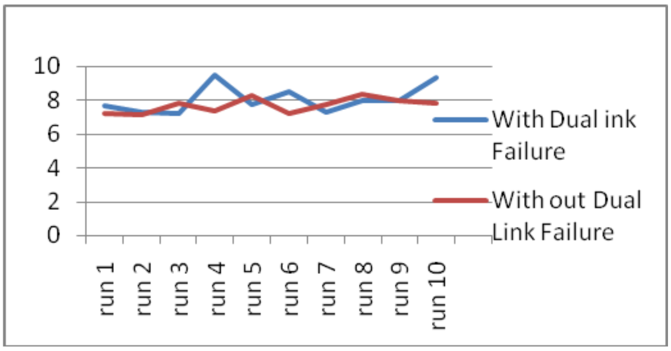

Figure 7: Profiling statistics with dual link failure and with out dual link failure

VIII. CONCLUSION

This paper focuses on the approaches for providing dual-link failure resiliency. Recovery from a dual-link failure using an extension of link protection for single link failure results in a constraint, referred to as BLME constraint, whose satisfiability allows the network to recover from dual-link failures without the need for broadcasting the failure location to all nodes. This paper develops the necessary theory for deriving the sufficiency condition for a solution to exist, formulates the problem of finding backup paths for links satisfying the BLME constraint as an ILP, and further develops a polynomial time heuristic algorithm. The formulation and heuristic are applied to six different networks and the results are compared. The heuristic is shown to obtain a solution for most scenarios with a high failure recovery guarantee, although such a solution may have longer average hop lengths compared with the optimal values.

The heuristic produces a solution in relatively less number of iterations for five of the six scenarios. A maximum of 30 iterations were performed. While the objective of the heuristic is to obtain a feasible solution, it is not guaranteed to find a solution (as seen in the Node-28 network scenario for any arbitrary two link failure scenario). The number of iterations required to arrive at the solution depends on a lot of parameters, specifically the order in which the auxiliary graphs are considered and the weights employed. Comparing the results of the heuristic to that of the ILP, it is observed that the heuristic can be as bad as 60% above optimal for average backup path lengths.

IX. REFERENCES

[1] A. Chandak and S. Ramasubramanian, “Dual-link Failure Resiliency through Backup Link Mutual Exclusion,” in Proc. IEEE Int. Conf. Broadband Networks, Boston, MA, Oct. 2008, pp. 258–267.

[2] Ashok Kumar Velpuri, Sreenivas Velagapudi, “A Novel Approach for Link/Path Protection in Dual-Link Failures”,IJMER,Vol2 Issue1,Feb-2012.

[3] J. Doucette and W. D. Grover, “Comparison of Mesh Protection and Restoration Schemes and the Dependency on Graph Connectivity,” Hungary, Oct. 2001, pp. 121–128.

[4] M. Medard, S. G. Finn, and R. A. Barry, “WDM Loopback Recovery in Mesh Networks,” in Proc. IEEE INFOCOM, 2008, pp. 752–759.

[5] S. S. Lumetta, M. Medard, and Y. C. Tseng, “Capacity versus Robustness: A Tradeoff for Link Restoration in Mesh Networks,” J. Lightw. Technol., 18, No. 12, pp. 1765–1775, Dec. 2000.

[6] G. Ellinas, G. Halemariam, and T. Stern, “Protection Cycles in WDM Networks,” IEEE J. Sel. Areas Commun., 8, No. 10, pp. 1924–1937, Oct. 2000.

[7] W. D. Grover, Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking. Upper Saddle River, NJ: Prentice-Hall, 2007.

[7] M. Fredrick, P. Datta, “Sub-graph Routing: A Novel Fault-tolerant Architecture for Sharedrisk Link Group Failures.

[8] M. Clouqueur and W. D. Grover, “Mesh-restorable Networks with Complete Dual-failure Restorability and with Selectively Enhanced Dual-failure Restorability Properties,” in Proc. OPTICOMM, 2002, pp. 1–12.

[9] J. Doucette and W. D. Grover, “Shared-risk Logical San Groups in Span-restorable Optical Networks:Analysis and Capacity Planning Model,” Photon. Netw. Commun., 9, No. 1, pp. 35–53, Jan. 2005.

[10] J. A. Bondy and U. S. R. Murthy, Graph Theory With Applications. New York: Elsevier, 2008.

[11] H. Choi, S. Subramaniam, and H. Choi, “On Double-link Failure Recovery in WDM Optical Networks,” in Proc. IEEE INFOCOM, 2007, pp. 808–816.

[12] CPLEXSolver.[Online].Available:http://www.cplex.com

[13] H. Choi, S. Subramaniam, and H. Choi, “Loopback Recovery from Double-link Failures in Optical Mesh Networks,” IEEE/ACM Trans. Netw., 12, No. 6, pp. 1119–1130, Dec. 2004.

[14] H. Choi, S. Subramaniam, and H.-A. Choi, “Loopback Recovery from Neighboring Double-link Failures in WDM Mesh Networks,” Inf. Sci. J., 149, No. 1, pp. 197– 209, Jan. 2003