Multiple Fault Tolerance in MPLS Network using Open Source Network Simulator

Muhammad Kamran[1] and Adnan Noor Mian[2]
Department of Computer Sciences,
FAST- National University of Computer & Emerging Sciences, Lahore, Pakistan.
[1]kamran@inbox.pk, [2]adnan.noor@nu.edu.pk

Abstract

Multiprotocol Label Switching (MPLS) is a differentiated and scalable framework introduced by IETF, which uses simple configuration & management to deliver end-to-end IP services. It enables the network devices to specify path based on Quality of Service (QoS) and bandwidth requested for those applications by changing the hop-by-hop paradigm. Fault tolerance is the ability of the system to respond gracefully to an unexpected hardware or software failure. By using fault recovery techniques we can make MPLS network fault tolerant. To simulate fault recovery techniques in MPLS network, an open source simulator Network Simulator (NS) often called as NS2 is used. We used NS2 for implementing multiple fault tolerance using protection switching domain in MPLS network. We developed a new fault recovery protocol based on protection switching domain and compared it with the NS2 rerouting fault recovery protocol. We found that proposed fault recovery protocol is more reliable and has less recovery time in single fault and multiple faults as compare to NS2 rerouting fault recovery protocol.

Keywords

MPLS, Rerouting, Protection switching, Open source simulator, NS2 and Fault tolerance.

1. Introduction

Multi-protocol Label Switching (MPLS) is a reliable broad band technique used to strength the IP networks. Packets enter the MPLS network through a router called Label Edge Router (LER) or often called Ingress router. This router is responsible for adding a label on the packet for further transmission. Functionally label is a short fixed length identifier that is used to forward the packets. Within the network the labels are used to route the packets without regard to the original packets header information. The label is based on certain criteria like IP address of the recipient and is used to route the packet to the next router called Label Switch Routers (LSRs) [5, 8]. The path through which the labelled packets are routed is called Label Switch Path (LSP). The last router in the LSP is responsible for removing the label from the packet. That router is called Label Edge Router (LER) or Egress router. Like IP or ATM networks, faults may occur in the MPLS network. For such link failure or particular node failure, for this situation there should be a specific mechanism for resolving faults. MPLS offers several recovery techniques mainly divided in to Protection Switching and Rerouting domains [2, 4]. Protection switching is a technique where alternative or backup paths are pre-computed/pre-established. Rerouting deals with computing/establishing path or path segment on demand after the occurrence of the fault.

The fault recovery techniques can be simulated using network simulators like Tool Box For Traffic Engineering Methods (TOTEM) [17],Optimized Network Evaluation Toil (OPNET) [18], Graphical Network Simulator (GNS) [16], Objective Modular Network Testbed in C++ (OMNeT++) [13] and Network Simulator (NS) [15]. We used Network Simulator known as NS2 to simulate the proposed fault recovery protocol. It is an open source simulator that uses two different languages because simulator requires two different kinds of functions to perform. Firstly the detailed simulations of protocols that requires a system programming language which can efficiently manipulate bytes, packet headers and implement algorithms that run over large data sets. Secondly a large part of network research involves slightly varying parameters or configurations. In these cases iteration time, change the basic model and re-run that model is more important. NS2 meets both of these needs with two languages C++ and OTcl. C++ is fast to run but requires more time to change for making it suitable for detailed protocol implementation. OTcl runs much slower but can be changed very quickly and interactively, that makes it ideal for simulation configuration. Furthermore ns2 is the best showing thing visually and it also supports trace files that traces and save everything happening against the network, like total number of packet sent in a particular time simulation, number of packets dropped, at specific time which router sends packet to which router, which link was went down at what time etc.

In this paper we have proposed a new algorithm that belongs to protection switching domain and using explicit routing and TCL we have simulated it in network simulator. We compared it with the rerouting fault recovery algorithm focusing on fault recovery time and multiple fault tolerance. Fault recovery time is the most important consideration in fault tolerance. If the taken by an algorithm to recover or send the traffic on the backup path the packet loss will be increased, which is not desirable. Here by saying Multiple Fault Tolerance we mean that first fault occurs in the working or primary label switch path and proposed algorithm switches over the traffic on the alternative path. Then second fault

occurs in the alternative path meanwhile the primary path was not restored yet, this is multiple fault tolerance. Some authors/reviewers may take this as multilevel fault tolerance. We found that proposed fault recovery protocol performs well in terms of fault recovery time as compare to rerouting algorithm though both of them provide multiple fault tolerance.

2. Background

2.1. Multiprotocol Label Switching (MPLS)

Multi-protocol Label Switching (MPLS) is a framework defined by IETF for fast packet switching and routing. It uses specific labels to forward the packets with in MPLS network. More specifically, MPLS has mechanisms to manage traffic flows of various granularities [5, 7]. It is independent of the layer-2 and layer-3 protocols such as ATM and IP. It provides a means to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies. MPLS interfaces to existing protocols such as IP, ATM, Frame Relay, Resource Reservation Protocol (RSVP) Label Distribution Protocol (LDP) and Open Shortest Path First (OSPF), etc. There are some specific terms related to the MPLS network [4, 6, 9, 10] defined as;

Label: A header created by an edge label switch router (edge LSR) and used by label switch routers (LSR) to forward packets. Label specifically identifies the path on which a packet should traverse.

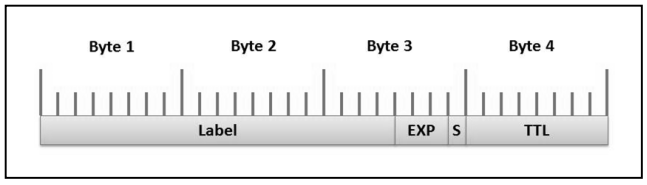

Figure 1: MPLS Header / Label.

Label forwarding information base: A table created by a label switch-capable device (LSR) that indicates where and how to forward frames with specific label values. Contents of the table identify the mapping between the label and the FEC.

Forward Equivalence Class (FEC): Having group of packets that share same requirements. A path is the representation of the Forward Equivalence Class.

Downstream and Upstream LSR: Downstream and upstream are important terms in MPLS domain. Router sending the traffic is called upstream to the other router and the router receiving the data is called as downstream LSR.

Edge label switch router (edge LSR): The device that initially adds or ultimately removes the label from the packet. It operates at the edge of the access and MPLS network.

Label switched: When an LSR makes forwarding decision based on the presence of a label in the frame/cell.

Label-switched path (LSP): The path defined by the labels through LSRs between end points, from source edge router to destination edge router. There are two options to set a LSP, hop-by-hop routing and explicit routing.

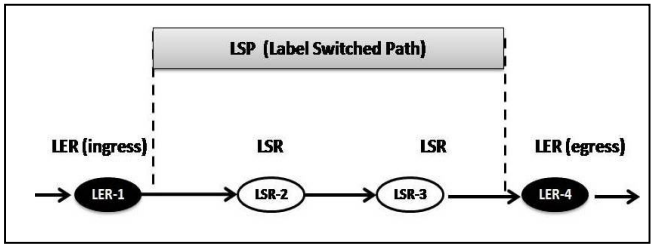

Figure 2: Label Switch Path (LSP).

Label distribution protocol (LDP): A set of messages defined to distribute label information among LSRs [6].

Label switch router (LSR): A device such as a switch or a router that forwards labeled entities based upon the label value. Each LSR, also known as an MPLS node, must have the following:

- At least one layer 3 routing protocol.
- A label distribution protocol.
- The ability to forward packets based on their labels.

The router can use Border Gateway Protocol (BGP), Intermediate System to Intermediate System (IS-IS) or Open Shortest Path First (OSPF) as its layer 3 routing protocol and BGP, LDP or Resource Reservation Protocol (RSVP) as its label distribution protocol. Figure3 shoes an example of MPLS based network.

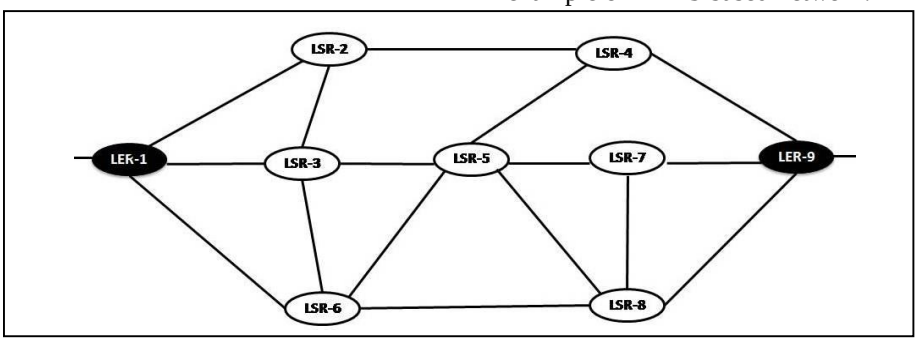

Figure 3: MPLS Network

2.2. Network Simulator

In 1995 Lawrence Berkeley National Laboratory (LBNL) developed Network Simulator with support of Defense Advanced Research Projects Agency (DARPA). Later NS2 was extended and distributed by Virtual InterNetwork Testbed (VINT) [19]. Latest version of NS2 (ns-2.34) was released in June 2009. NS2 supports different platforms like Linux, Windows. NS2 has different components; NS for Simulation, Network AniMator (NAM) for visual demonstration of NS output, pre-processing for hand written TCL or topology generator and Post analysis for trace file using Perl, Tcl, AWK and MATLAB [15].

NS2 is used in the simulation of routing and multicast protocols and is mostly used in ad-hoc networking research. NS2 uses two types of languages system level languages C, C++ and scripting language TCL (OTcl). C++ is used for creation of objects because of its speed and efficiency. OTcl is used as a front-end to setup the simulator configures objects and schedule events.

3. Related Work

For fault tolerance in MPLS networks there are several schemes and algorithms are developed, some of them are related to the domain of "Protection Switching" and some of them are "Rerouting". They are generally tested on major criteria's like, Recovery Time, Packet Loss and Multiple Fault Tolerance [1]. Most important criteria we focused are multiple fault tolerance though it is very rare but has strong impact on the reliability of the network. The two major types of recovery schemes that are used for MPLS recovery are Protection Switching and Rerouting are defined under.

3.1. Protection Switching

Protection switching is a recovery scheme in which recovery label switch path(s) are pre-computed or pre-established before a failure occurs on the working label switch path. When the fault occurs and Path Switch LSR (PSL) receives the Fault Identification Signal (FIS) it switches the traffic to the pre-established recovery path. As the recovery paths are pre-established so PSL immediately transfers the traffic on the backup path after receiving the FIS this makes protection switching faster than rerouting [1,2,4,11]. Resources required for the establishment of recovery path are reserved. Protection switching pre-establishes a recovery path or path segment based on network routing policies, the restoration requirements of the traffic on the working path and administrative considerations [1, 8].

3.2. Rerouting

Rerouting, a fault recovery technique where a recovery path is established on demand after a fault occurs. The recovery path can be based on fault information, network routing policies and network topology information [1, 3, 14]. An advantage of fault recovery by rerouting is that it does not take up any backup resources in the network before the recovery path is signaled. The new paths may be based upon fault information, network routing policies, pre-defined configurations and network topology information. Thus, on detecting a fault, paths or path segments to bypass the fault are established using signaling. On the other hand rerouting has the disadvantage that resources may not be available at the time of computing recovery path that may leads to major failures [4, 8].

3.3. Placement of Recovery Path

After the computation of recovery path or if the path is pre-computed by protection switching technique, path can be place locally or globally.

*Local Repair***,** in local recovery, the recovery path selection or switching is done by a label switch router (LSR), which is nearest to the failed router or link. The main function of local repair is to fix the problem at the point of failure or within a very short distance from the failure for minimizing total packet loss and recovery time. In other words local repair aims to protect against a link failure or neighbour node failure and to minimize the amount of time required for propagation of failure signal [8, 14]. If a repair can be performed local to the device that detects the failure, restoration can be achieved faster. In local repair, the immediate upstream LSR of the failure is the LSR that initiates the recovery operation.

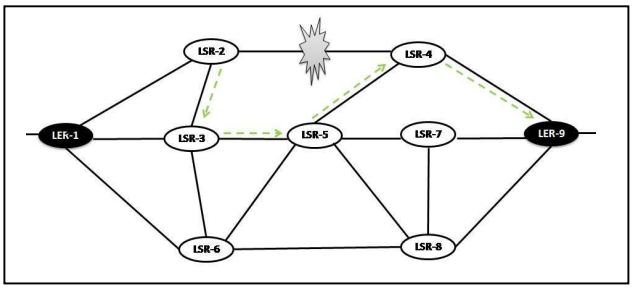

Figure 4: Local Repair.

*Global Repair***,** in global recovery the alternative backup path selection is done by Protection Switch LSR. There is an alternative LSP that is pre-established or computed dynamically from ingress to egress routers. Ingress router is the entry point of MPLS network and Egress router the end point of MPLS Network. In other words global repair protect against any link or node failure on a path or on a segment of a path. In global repair the Point of Repair (POR) is distant from the failure and needs to be notified by a FIS [6, 12]. Recovery path is completely disjoint from the working path. This has the advantage that all links and nodes on the working path are protected by a single recovery path and having the disadvantage that a FIS has to be propagated all the way back to the ingress LSR before recovery can start.

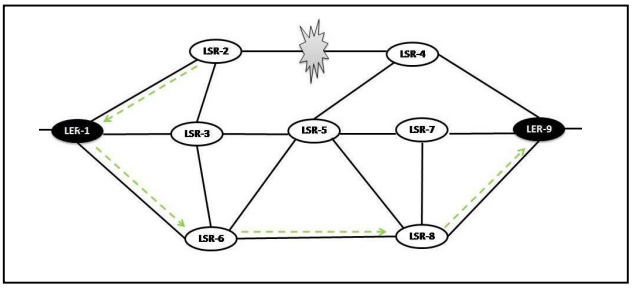

Figure 5: Global Repair.

4. Simulating MPLS using NS2

This section describes the implementation of MPLS Network through Network Simulator (NS2). We compared the already implemented rerouting fault recovery protocol in NS2 with our proposed fault recovery protocol. Rerouting fault recovery protocol uses rerouting scheme in which LSP is computed after the occurrence of fault. Furthermore rerouting scheme recovers faults in two steps. Firstly time taken to send FIS to the ingress in global recovery and to upstream LSR in case of local recovery plus the time for computing new LSP and transferring traffic on the backup path. Whereas in case of our proposed protocol that is protection switching domain requires time to send FIS to ingress or upstream LSR. Because backup paths are already computed so when ingress LSR receives the FIS it directly transfers the traffic to backup path.

Considering the space complexity of the proposed and rerouting fault recovery algorithms, increasing the number of nodes in the network the proposed protection switching algorithm will take more space as compare to rerouting algorithm. Proposed algorithm will store all paths in advance so more space will be required but the computation of recovery path and storing them in advance will only be done once.

Algorithm 1: Rerouting Fault Recovery Protocol

1. **Upon** reception of FIS
2. calculate backup path using SPF **then**
3. **send** traffic to backup path
4. **if** FIS received through backup path **then**
5. calculate the shortest path using SPF
6. **send** traffic to computed backup path
7. **if** original working path restored
8. **switch** traffic to it

On the other hand rerouting algorithm though it will not require pre-allocated space but the time of computing backup path will require more time and more resources as the number of nodes are increased. It may be possible that the required resources are not available so that rerouting algorithm has to wait till the resources are available completely resulting large recovery time and packet lost.

The protocol in algorithm 1 is rerouting fault recovery protocol. When ingress LSR receives the FIS from the core LSR then it computes the backup path (path on demand) and transfers the traffic to it. If fault occurs in the backup path as well on FIS request ingress again computes the second backup path and transfers the traffic to it. When the original link is restored then it shifts the traffic to original working path.

Algorithm 2: Proposed Fault Recovery Protocol.

Protocol running on Ingress LSR,

1. **Upon** reception of FIS
2. **if** path Not Found against failure Link **then**
3. terminate algorithm
4. **else**
5. **switch** traffic to backup path
6. **if** FIS received through backup path **then**
7. **switch** traffic to second backup path
8. **if** original working path restored
9. **switch** traffic to it

Protocol running on Core LSR,

1. **send** Keep alive messages
2. **if** no acknowledgement **then**
3. **send** FIS to Ingress

In proposed fault recovery protocols running on ingress LSR, upon reception of FIS from core routers ingress router will check the backup path against the link failure if it finds the path against that link it will transfer the traffic to it else it will terminate. Such termination will occur only when no protected path will be available. If fault or link failure occurs in the backup path then the core routers of that backup path will send FIS to ingress. After that when ingress will receive the FIS it will again follow the same procedure and traffic will be sent to the second backup path.

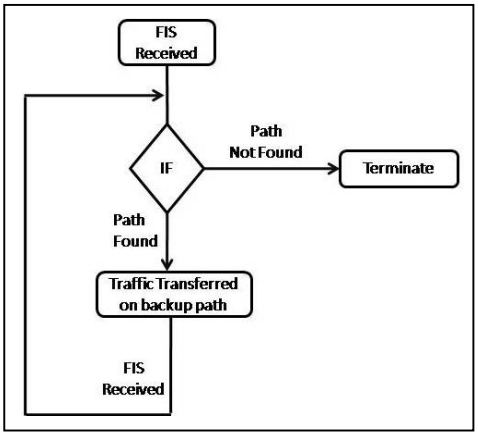

Figure 6: Flow Chart diagram of Proposed Algorithm.

Meanwhile all MPLS routers will send Keepalive messages. If the original link is restored then traffic will be sent to it. Flow chart of proposed algorithm is shown in figure 6.

For implementing multiple fault tolerance in MPLS based network and simulating it in an open source simulator i.e. Network Simulator we have created a topology that have nine MPLS nodes and two non-MPLS nodes as shown in figure 7. First non-MPLS node is attached with ingress LSR and the second one is attached with the egress LSR. All the MPLS nodes excluding ingress and egress are called as core MPLS nodes or Label Switch Routers.

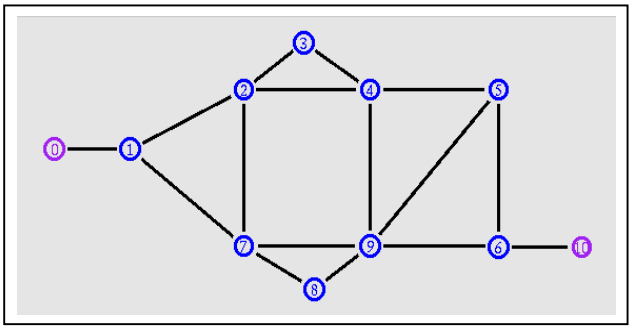

Figure 7: Network Topology used in NS.

Tcl code used for creating MPLS and non-MPLS nodes is as,

```
set node0 [$ns node]
$node0 color "purple"
$ns node-config -MPLS ON
set LSR1 [$ns node]
:     :     :     :
:     :     :     :
:     :     :     :
set LSR9 [$ns node]
$LSR9color "blue"
$ns node-config -MPLS OFF
set node10 [$ns node]
$node10 color "purple"
```

LDP is configured by using [9],

```
for {set i 1} {$i < 10} {incr i}
       {
       set a LSR$i
for {set j [expr $i+1]} {$j < 10}
{incr j}{
       set b LSR$j
       eval $ns LDP-peer $$a $$b}
 set m [eval $$a get-module "MPLS"]
       $m enable-reroute "new"
 }}
```

Links between nodes (MPLS/non-MPLS) having bandwidth 1Mb, delay 10ms and Queue type DropTail are created through,

```
$ns  duplex-link  $node0  $LSR1  1Mb
10ms DropTail
:     :     :     :     :
:     :     :     :     :
$ns  duplex-link  $LSR6  $node10  1Mb
10ms DropTail
```

Here in this part of Tcl file we are defining the way to create paths and store them in ingress for use against the any link failure.

For first link failure,

```
$ns at 0.0 "[$LSR1 get-module MPLS]
make-explicit-route 6 1_2_4_5_6 2000
-1"
$ns at 0.4 "[$LSR1 get-module MPLS]
flow-erlsp-install   10 -1   2000"
```

For second link failure,

```
$ns at 0.0 "[$LSR1 get-module MPLS]
make-explicit-route   6   1_2_3_4_5_6
2500 -1"
$ns at 1.0 "[$LSR1 get-module MPLS]
flow-erlsp-install   10 -1   2500"
```

For third link failure,

```
$ns at 0.0 "[$LSR1 get-module MPLS]
make-explicit-route 6 1_7_8_9_6 3000
-1"
$ns at 1.5 "[$LSR1 get-module MPLS]
flow-erlsp-install   10 -1   3000"
```

After running TCL code, LDP message exchange between LSR occurs as shown in figure 8.

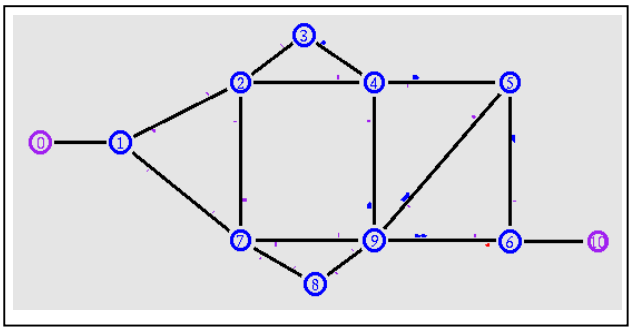

Figure 8: LDP message exchange.

Figure 9 shows the first working path, that is LSR1-LSR7-LSR9-LSR6. LSR1 is ingress and LSR6 is the egress LSR through the core routers LSR7 and LSR9. Figure 9 explains the first link failure between LSR7 and LSR9.

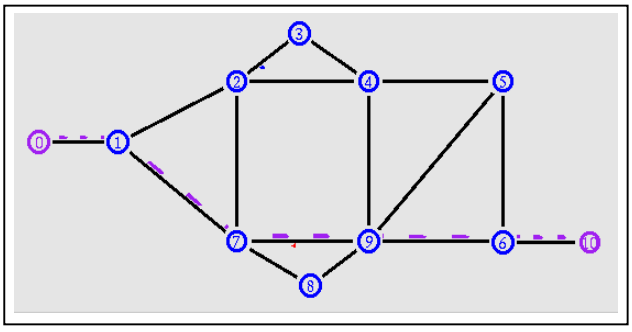

Figure 9: First working label switched path.

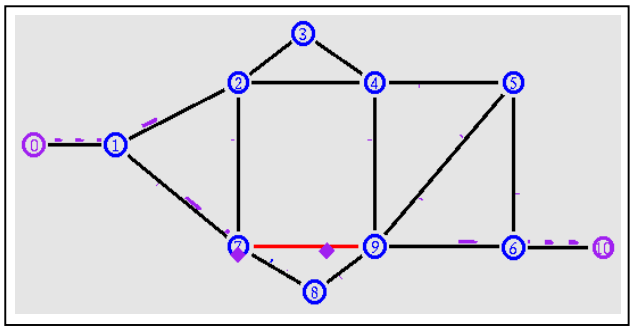

Figure 10: First link failure in working LSP.

When the link failure occurs in case of rerouting fault recovery protocol it switches the traffic to the backup path LSR1-LSR7-LSR8-LSR9-LSR6 in 0.04ms whereas our proposed fault recovery protocol switches over the traffic to label switch path LSR1-LSR2-LSR4-LSR5-LSR6 on 0.01ms. Thus we observed that proposed fault recovery protocol takes less time as compare to rerouting because of protection switching in which backup paths are pre-computed whereas rerouting fault recovery protocol computes backup path after the occurrence of fault. Figure 11 shows the rerouting fault recovery protocol backup path and figure 12 shows the proposed fault recovery protocol backup path.

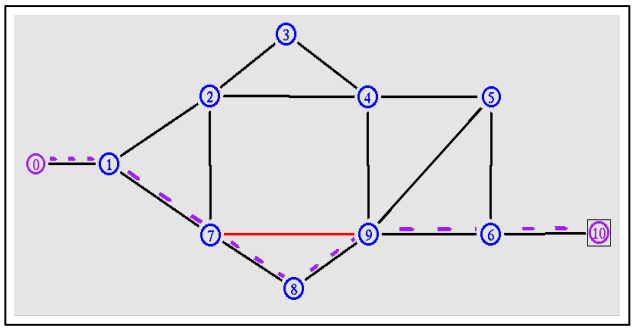

Figure 11: Traffic transferred to backup path using rerouting fault recovery protocol.

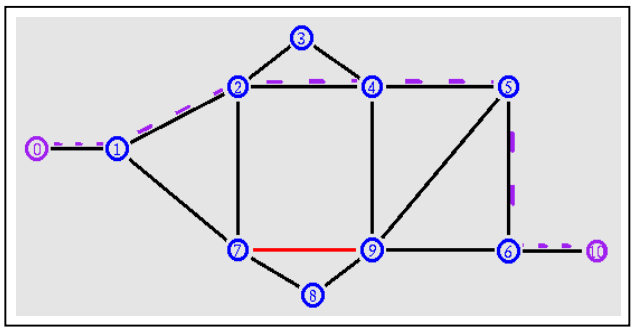

Figure 12: Traffic transferred to backup path using proposed fault recovery protocol.

In case of second fault that means fault in the backup path when link was down between LSR1 and LSR7, rerouting fault recovery protocol switches the traffic to new path LSR1-LSR2-LSR4-LSR5-LSR6 in 0.07ms. On the other hand our protocol switches the traffic to pre-computed path LSR1-LSR2-LSR3-LSR4-LSR5-LSR6 in 0.02ms.

Figure 13 and 14 shows the rerouting and proposed protocol multiple fault tolerance respectively.

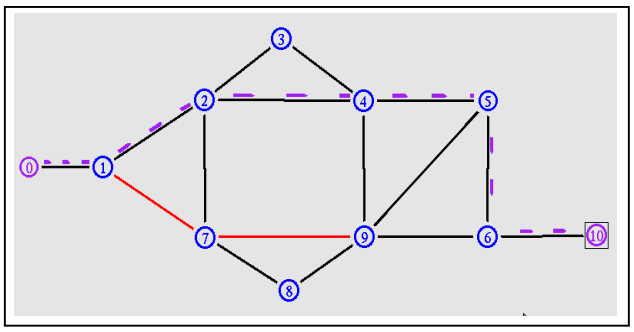

Figure 13: Multiple fault tolerance in rerouting protocol.

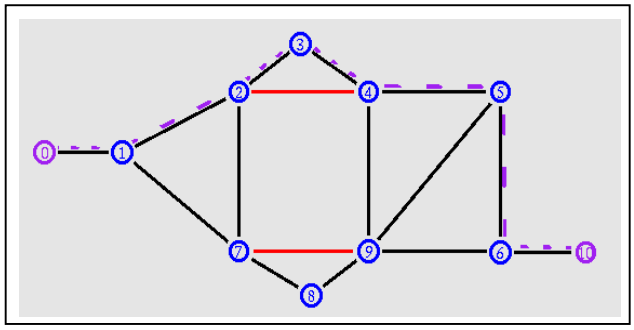

Figure 14: Multiple fault tolerance in proposed protocol.

For the third fault rerouting fault recovery protocol switches the traffic to path LSR1-LSR2-LSR7-LSR8-LSR9-LSR6 in 0.02ms and proposed fault recovery protocol switches the traffic to path LSR1-LSR7-LSR8-LSR9-LSR6 in 0.01ms.

Figure15 shows the rerouting fault recovery protocol and figure16 shows proposed protocol path switching.

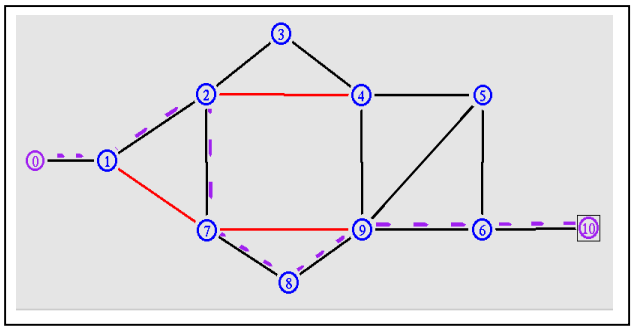

Figure 15: Third Fault in rerouting fault recovery protocol.

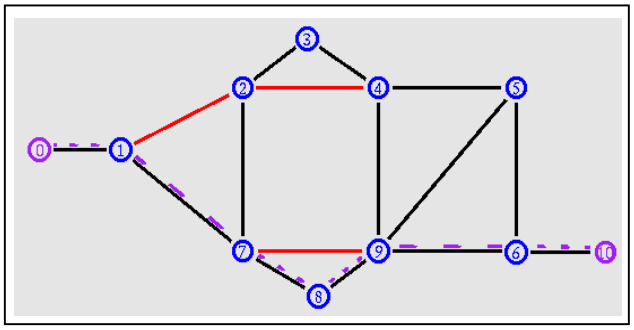

Figure 16: Third Fault recovery in proposed fault recovery protocol.

After comparing results we found that protection switching protocol is more reliable then rerouting protocol. Though protection switching requires some extra resources like space but it perform well in single or multiple faults. On the other hand rerouting protocol takes some additional time to switch over the traffic on the alternative path. Figure17 shows the recovery time comparison graph between rerouting fault recovery protocol and proposed fault recovery protocol.

5. Conclusion

In this paper we focused on multiple fault tolerance in MPLS network. We created a network topology and implemented proposed protocol on it and compare it with rerouting fault recovery protocol. Rerouting fault recovery protocol uses rerouting domain for fault tolerance and it computes recovery path on demand after the occurrence of the fault whereas proposed fault recovery algorithm is from protection switching domain in which recovery paths are pre-computed. We observed proposed fault recovery protocol took less time to switch over the traffic to the recovery path as compare to rerouting fault recovery protocol. Total time that rerouting fault recovery protocol took to recover from a particular fault is time for sending FIS to the ingress plus time to computing backup path and sending traffic on it. In proposed fault recovery protocol total time is just to send FIS to ingress then ingress will automatically transfer the traffic to the backup path because it has already stored backup paths. Network Simulator NS2 being an open source simulator has the advantage over other simulators like Graphical Network Simulator (GNS) that its files are easy to modify and source files are freely available. We can implement new protocol in NS2 easily. GNS require extra files like .iso files for specific router. We cannot modify files in GNS like we can in NS2. For network researchers and students NS2 is the best available simulation tool.

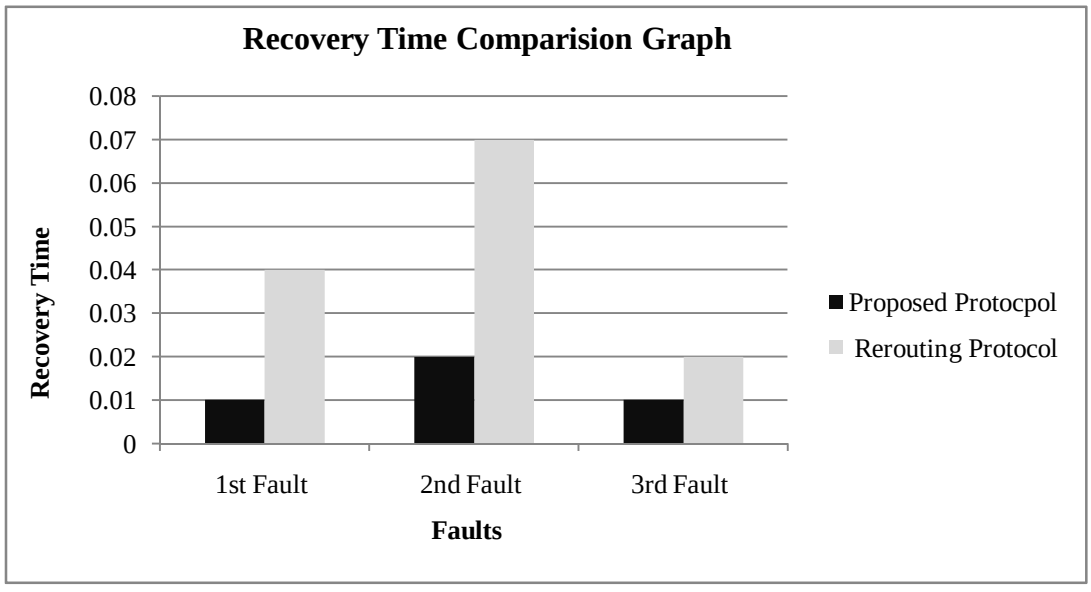

Figure 17: Recovery time comparison graph.

6. References

[1] M.Had, C.Geo, M.Pap, V.Vass. "A Haybrid Fault-Tolerant Algorithm for MPLS Networks", *WWIC 2008, LNCS 5031*, pp. 41-52, 2008.
[2] Jack Foo. "A Survey of Service Restoration Techniques in MPLS Networks".
[3] V.Alar, Y.L.Tak, J.C.Mar, L.G.Gue, "MPLS/IP Analysis and Simulation for the Implementation of Path Restoration Schemes".
[4] Johan Martin. Thesis on "MPLS Based Recovery Mechanisms", 2005.
[5] CISCO. Multiprotocol Label Switching CICSO.
[6] G.Ahn, W.Chun, "Design and Implementation of MPLS Network Simulator Supporting LDP and CR-LDP". The *Proceedings of the IEEE International Conference on Networks (ICON'00).*
[7] G.Kaur, D.Kumar, "MPLS Technology on IP Backbone Network", *International Journal of Computer Applications (0975-8887)*, 2010.
[8] V.Sharma. "Framework for Multi-Protocol Label Switching (MPLS)-based *Recovery", RFC-3469*, Ferruary 2003.
[9] L.Andersson. "LDP Specification", *RFC-5036*, October 2007.
[10] E.Rosen. "Multiprotocol Label Switching Architecture", *RFC-3031*, January 2001.
[11] S.Shew. "Fast Restoration of MPLS Label Switched Paths", *draft-shew-lsp-restoration-00*, October 1999.
[12] Y.Qiu, J.Che, J.Gu, X.Xin. "A Simulation for MPLS Global Recovery Model", *First International Conference on Intelligent Networks and Intelligent Systems, IEEE* 2008.
[13] http://www.omnetpp.org
[14] S.Alo, A.Aga, A.Nou. "A Novel Approach for Fault Tolerance in MPLS Networks", IEEE 2006.
[15] http://www.isi.edu/nsnam/ns/
[16] http://www.gns3.net
[17] http://totem.run.montefiore.ulg.ac.be
[18] http://www.opnet.com/solutions/network_rd/modeler
[19] http://www.isi.edu/nsnam/vint/