

A Survey of Service Restoration Techniques in MPLS Networks

Jack Foo

Department of Electrical and
Computer Systems Engineering
Monash University

Email: jack.foo@eng.monash.edu.au

I. ABSTRACT

Multi-protocol Label Switching (MPLS) has become an attractive technology of choice for Internet backbone service providers. MPLS features the ability to perform traffic engineering and provides support for Quality of Service traffic provisioning. This has resulted in increasing interest in MPLS network reliability and survivability. This article surveys the various service restoration techniques proposed for MPLS networks.

II. INTRODUCTION

As demand grows for the Internet to carry more traffic and provide support for Quality of Services (QoS), it is essential to maintain a high level of performance and efficiency. Traffic engineering is the process of optimization of the network to maximize performance and efficiency. Multi-Protocol Label Switching (MPLS) is a tool for network traffic engineering and hence is becoming the technology of choice for Internet backbone.

MPLS forward packets based on fixed size labels, the path that the packets traverse is pre-established. Incoming packet labels are examined to determine the next hop, the old label is then exchanged with a new label to be forwarded to the next hop. The path the packet traverses is called a Label Switch Path (LSP) and the Label Distribution Protocol (LDP) is used to set up the LSP. LSP can allow packets to traverse an explicit route compare with dynamic routing in IP, allowing for traffic engineering. Another key feature of MPLS is the functionality of Label Stacking, allowing a packet to have multiple labels at one time. Therefore packets can be tunneled through several LSPs.

The reservation of network resources is also possible using MPLS, thus providing QoS guarantees. Two signaling protocols have been established to provide the reservation of bandwidth for the LSP, resource reservation protocol with traffic engineering extension (RSVP-TE) and constraint based-routing label distribution protocol (CR-LDP).

There has been an increasing need for current networks to carry QoS or high priority traffic. The traditional best-effort IP networks are quite robust; if a segment of the network failed it will find alternate route if one exists. However, with today's requirements of guaranteeing bandwidth for high priority traffic, network survivability has become a complicated issue.

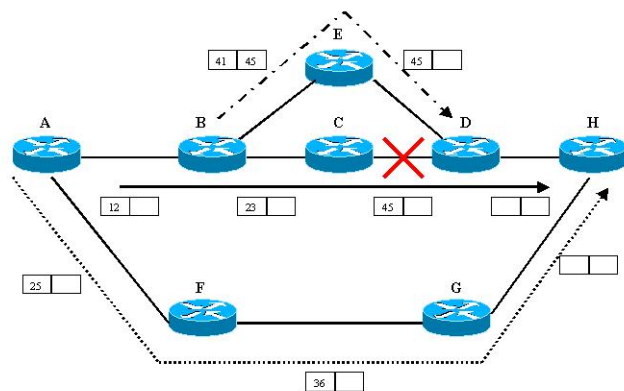


Fig. 1. Example of MPLS Recovery

In this article we survey various proposals in MPLS networks to maximize network reliability and survivability.

III. MPLS RECOVERY MECHANISM

“To deliver reliable service, MPLS requires a set of procedures to provide protection of the traffic carried on different paths. This requires that the label switching routers (LSRs) support fault detection, fault notification, and fault recovery mechanisms, and that MPLS signaling support the configuration of recovery”[1]. As mentioned earlier there are two main signaling protocols for MPLS networks which support the configuration of recovery. RSVP-TE is based on the extension of the reservation protocol to support traffic engineering. This is a soft state signaling protocol, requiring periodic refresh messages to ensure links are still active. The CR-LDP is an extension based on the new Label Distribution Protocol (LDP). CR-LDP also allows for reservation of bandwidth and is a hard state signaling protocol, ie does not require refresh messages to maintain an active link. Chung provide a comparison of the two signaling protocols [2]. Fault detection is the time required to detect a fault; the quicker the fault is detected in the network the earlier the fault can be repaired. The current method of detecting faults on a link comes from the use of periodic messages, if the messages are absent then the link is assumed to be faulty and fault notification is initiated. Fault notification is the process of informing other LSRs of the failure that

has occurred which eventually propagates to ingress LSRs to initiate recovery procedures to maintain traffic flows.

The recovery models used in an MPLS network are classified into two categories, rerouting and protection switching. The rerouting recovery model can only establish the recovery path when a fault has occurred. The new recovery path is usually created based on fault information, network topology, etc. Traffic is switched onto the recovery path temporarily until the fault is repaired. Rerouting has the advantage of using up-to-date information and is able to make efficient use of resources in creating recovery paths. The drawback is the time required to compute and setup the recovery path after failure has occurred. Figure 1 shows an example of link failure between node **C** and **D**. The rerouting recovery model sends a Fault Indicator Signal (FIS) upstream to ingress node **A**, which then finds and establishes a recovery path around the failed link, eg. from **A** \rightarrow **F** \rightarrow **G** \rightarrow **H**.

Protection switching involves pre-determining and establishing a recovery path before the fault has occurred. The recovery path is usually disjoint from the working path to prevent any fault that can affect both the working and recovery path; When failure occurs the traffic is switched from the working path to the recovery path, this is simply done by swapping the label from the working LSP with the label associated with the recovery LSP. This has the advantage of permitting fast recovery, but requires additional resources to setup the recovery path which remain unused until a fault occurs.

Protection switching can also be divided into local or end-to-end repair. Local repair is used to protect against the fault of a single link or node, it has the advantage of minimizing the time required for fault notification, as only the previous upstream node is notified of the fault. The upstream node creates a recovery path around the fault and label stacking is used to direct traffic onto the recovery path. Once traffic has been routed around the fault the top label is removed and traffic continues as normal without the ingress or egress LSRs ever knowing a fault has occurred. An end-to-end repair requires the recovery path to protect against all links and nodes used in the working path. An example of local and end-to-end repair is shown in Figure 1. If link failure occur between node **C** and **D**, protection switching using end-to-end repair would established a recovery path **A** \rightarrow **F** \rightarrow **G** \rightarrow **H** and reserve the necessary resources. Local repair shows that the upstream node **B** established a recovery path **B** \rightarrow **E** \rightarrow **D** to protect node **C**. Label stacking is used to divert traffic to the recovery path by simply adding an extra label to the packet at node **B**. Label **41** is added at node **B** hence traffic is switched to the recovery path, at node **E** the extra label is removed and traffic is switched back to the original path.

Another classification involves different levels of protection that can be specified for the protection path [3] [4]. A 1+1 path protection provides a dedicated recovery path that is established and resources are fully reserved. This provides the best protection as it provides fast restoration while preserving QoS. The 1:1 path protection provides a recovery path for each working path, the recovery path resources can be shared with low-priority traffic. When faults occur the low-priority traffic is preempted by the recovery path to carry the protected traffic.

| Recovery Model | Recovery Path Type | Recovery Path Setup Point | Recovery Time | Resource Utilization Optimization |
|----------------------|---------------------|---------------------------|---------------|-----------------------------------|
| Rerouting | Pre-Qualified | After fault | ** | *** |
| | Establish-On-Demand | | * | **** |
| Protection Switching | 1+1 | Before fault | *** | * |
| | 1:1, 1:n, m:n | | *** | ** |

TABLE I
COMPARISON OF RECOVERY MODELS [5]

The 1:N path protection uses a single protection path to protect multiple working paths. The 1:1 and 1:N path protection make more efficient uses of network resources. Table I gives a comparison of the recovery models [5].

The ability to explicitly route LSP across the MPLS network domain gives complete freedom in managing traffic flows. This resulted in a number of new proposals that aim at improving the network performance and efficiency as a whole, as well as new methods for recovery during failure.

IV. PROTECTION SWITCHING

Fast Reroute was introduced by Haskin [4] to quickly reroute traffic so that the disruption of high priority traffic flows are minimized during a fault in an MPLS network. A recovery path is pre-established that is parallel to the working path before the occurrence of a fault. When a fault occurs the traffic is switched to the recovery path at the source node. This provides a fast recovery mechanism, but a FIS is still required to traverse upstream to the source node before traffic is switched to the recovery path. This delay is further reduced with the introduction of link or node protection. The idea is to setup a recovery path to protect each physical link or node. This reduces the delay of recovery, the FIS is only required to traverse upstream one hop before fault recovery begins and reroute traffic around that failed link or node. This process provides a very quick response to failure, as the fault is repaired locally and the ingress or egress nodes will not even realize a fault has occurred. The cost for the quick recovery is a significant bandwidth requirement for the recovery path. The recovery path should have enough bandwidth reserved for all the protected LSPs, and multiple links or nodes are required to be protected. Fast reroute provides fast recovery of service but requires a large amount of network resources that remain unused in the absence of faults in the network.

A limitation of fast reroute is the requirement that the pre-established recovery path be link-disjoint from the working path, otherwise a failure in a link which is used by the working and recovery path would cause traffic flow to cease. A new approach in service restoration without the limitation of the recovery path being link-disjoint is examined by Bartos [6] [7]. The proposed scheme places two protection paths such that it can restore traffic flows from all single link failures. The use

of two protection paths allows working and protection paths to share common links, removing the restriction that links must be disjoint from the working path. The two protection paths protects all working paths ending at a common egress node, the calculation for these working paths are done simultaneously. The recovery method provides flexibility and fast restoration of service as it is based on protection switching. The method however, only assumes single link failure and does not consider restoration of QoS traffic.

Bremner-Barr shows that “After k edge failures in an unweighted network, each new shortest path is the concatenation of at most $k+1$ original shortest paths” [8]. This theory became the basis used in the restoration by path concatenation (RBPC) method. The RBPC uses the knowledge that if a link failed and for each path that is affected, if an alternative path exist, it is the concatenation of at most two original shortest paths. Recovery is implemented by first seeking two LSPs which combine to form the recovery path. MPLS label stacking is used to direct traffic flow to the recovery path, simply by using two labels for each packet. The first label is used to forward traffic through the first LSP, the top label is then removed upon reaching the egress node of the first LSP. The second label is then used to forward packet through the second LSP, reaching the destination node. The RBPC implementation provides a fast and simple recovery process but lacks consideration for QoS traffic which MPLS supports.

Protection switching provides a mechanism for fast recovery in MPLS networks, but inefficient utilization of network resources occur, as protection paths are unused in the absence of faults. Protection switching also requires protection paths to be pre-established before failures occurs. This can lead to suboptimal protection paths as the network changes over time.

V. REROUTING

The rerouting model establishes the recovery path only after a fault has occurred. Optimization of recovery paths can be achieved as the current network state is considered before recovery path selection. The rerouting model can be further divided into two types, establish-on-demand and pre-qualified. Establish-on-demand calculates and establishes a recovery path only after a fault is detected. Pre-qualified implies that a recovery path is already calculated but is only established when a fault occurs. It has the advantages of efficient resource utilization as bandwidth is not reserved and speed up recovery as path selection is completed before faults occur.

Integer Linear Programming technique was used by Yetginer and Karasan to design working and recovery paths such that it optimized the whole MPLS network [9]. They introduced four different methods for selecting working and recovery paths and these methods are each formulated as an integer linear programming problem. The restoration scheme involved off-line path calculation and rerouting as the recovery method. It assumed that the network is a 2-connected topology and can only protect against single link failure. For a given demand set, the working and recovery path of each requested connection are calculated, subject to constraints of links having available bandwidth. The traffic can be fully restored

against all possible single link failures. The goal is then to accommodate as many requests as possible while meeting the constraints. Four methods have been proposed using integer linear programming. Two of the methods involved selecting the working and recovery paths independent of each other, aiming at maximizing the residue capacity or even distribution of the load across the network. The other two methods consider selecting the working and recovery paths with consideration of each other. One method looks at maximizing residual capacity while distributing the load evenly across the network. The other method is similar except it considered the weight associated with each link and was shown to be the best method for accommodating the most requests.

Computing the recovery path beforehand has the disadvantage of being suboptimal, as network traffic flow changes over time and the recovery path may be unoptimized at the time of the fault. A new pre-qualified recovery method was introduced by [5]. They suggested that the used of pre-qualified recovery method is useful in reducing recovery time, but needs frequent update to ensure that recovery paths remain optimal. Modification of existing Interior Gateway Protocol (IGP) is used to update network information on LSRs, such as Open Shortest Path First with Traffic Engineering extension (OSPF-TE). The proposed recovery scheme involved updating the recovery paths whenever LSR receives new network topology updates, ensuring that recovery paths remain optimum when the network state changes. Simulation results suggest an improvement over existing pre-qualified recovery method, however it requires significant additional resources from LSRs such as CPU, memory, etc.

A two stage restoration technique is proposed by Otel [10], which includes establishing a temporary bypass tunnel where ingress LSRs performed recovery procedures. An Incremental Dijkstra algorithm was used to find a path for the bypass tunnel. The algorithm uses existing information to reduce propagation of failure messages and IGP updates, hence reducing the time required to setup a bypass tunnel. A bypass tunnel is established at the upstream node where a fault has occurred. The algorithm uses information such as the local node outdated shortest path tree, the failed link and the disconnected downstream node. If an alternate route is found, it is used as a bypass tunnel for forwarding traffic around the failed link. Once all LSRs have completed rerouting of traffic, the bypass tunnel can be torn down. It was suggested that it is approximately 5 times faster than local IGP re-convergence, but the recovery scheme assume only single link failures.

The rerouting recovery models are generally more resource friendly than protection switching. They utilize network resources efficiently and can service more requests while providing protection. However recovery only begins after the occurrence of a fault, hence it is slower than protection switching. In [5] they reduce the recovery time of rerouting by pre-calculating recovery path before fault but does not reserve bandwidth resources. Additionally, recovery paths are periodically updated to ensure they remain optimal.

VI. OTHER RECOVERY IMPROVEMENT SCHEMES

As mentioned above the MPLS recovery mechanism involves fault detection, fault notification and fault recovery. The main focus of many proposed methods is of the actual fault recovery, however in [3] the authors focused on the reliable, efficient and fast distribution of fault notification messages. The proposed mechanism for recovery involved components in fault detecting and distribution of fault messages, using protection switching as the recovery model. It introduced a new fault notification structure, the reverse notification tree (RNT). The RNT can perform fault notification for label merging, it propagates fault signals to ingress LSRs responsible for switching to recovery path. The RNT is a point to multi-point tree, which is the mirror image of the working paths. They have also proposed a hello protocol to increase sensitivity of fault detection and a lightweight transport protocol for rapid delivery of fault signals.

The critical link is defined as a link that is likely to be used by future requests [11] [12]. Minimum Interference Routing Algorithm (MIRA) was introduced by [11] and is an on-line routing algorithm aimed at *not interfering too much* with future requests, by avoiding critical links. The algorithm is based on the max-flow theory and tries to maximize the residual capacity between ingress-egress pairs, improving the chances of finding alternative paths during failure.

VII. CONCLUSION

There are two main methods of recovery, protection switching and rerouting. Protection switching provides fast restoration of service but lacks efficient use of network resources in MPLS networks. Rerouting has the advantage of optimizing the recovery paths, allowing for more working or recovery LSP requests. The disadvantage is slower restoration of service. Other approaches to improving service restoration include using concepts such as RNT, that aims to improve the reliability and efficiency of fault notification. Protection switching and rerouting both have their complementary advantages and disadvantages; the choice of recovery method should be based on the type of protection required.

REFERENCES

- [1] V. Sharma and F. Hellstrand, "Framework for multi-protocol label switching (mpls)-based recovery," 2003, request for comments: 3469.
- [2] J.-M. Chung, "Analysis of mpls traffic engineering," in *Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on*, vol. 2, 2000, pp. 550–553 vol.2, tY - CONF.
- [3] C. Huang, V. Sharma, K. Owens, and S. Makam, "Building reliable mpls networks using a path protection mechanism," *Communications Magazine, IEEE*, vol. 40, no. 3, pp. 156–162, 2002, tY - JOUR.
- [4] D. Haskin and R. Krishnan, "A method for setting an alternative label switched paths to handle fast reroute," 2000, draft-haskin-mpls-fast-reroute-05.txt.
- [5] S. Yoon, H. Lee, D. Choi, Y. Kim, G. Lee, and M. Lee, "An efficient recovery mechanism for mpls-based protection lsp," in *ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001. Joint 4th IEEE International Conference on*, 2001, pp. 75–79, tY - CONF.
- [6] M. Bartos, R.; Raman, "A heuristic approach to service restoration in mpls networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 1, 2001, pp. 117–121 vol.1, tY - CONF.
- [7] R. Bartos, M. Raman, and A. Gandhi, "New approaches to service restoration in mpls-based networks," in *EUROCON'2001, Trends in Communications, International Conference on.*, vol. 1, 2001, pp. 58–61 vol.1, tY - CONF.
- [8] A. Bremler-Barr, Y. Afek, H. Kaplan, E. Cohen, and M. Merritt, "Restoration by path concatenation: Fast recovery of mpls paths," 2001.
- [9] E. Yetginer, E.; Karasan, "Robust path design algorithms for traffic engineering with restoration in mpls networks," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, 2002, pp. 933–938, tY - CONF.
- [10] F.-D. Otel, "On fast computing bypass tunnel routes in mpls-based local restoration," in *High Speed Networks and Multimedia Communications 5th IEEE International Conference on*, 2002, pp. 234–238, tY - CONF.
- [11] T. Kodialam, M.; Lakshman, "Minimum interference routing with applications to mpls traffic engineering," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2000, pp. 884–893 vol.2, tY - CONF.
- [12] V. Subramanian, S.; Muthukumar, "Alternate path routing algorithm for traffic engineering in the internet," in *Information Technology: Coding and Computing [Computers and Communications], 2003. Proceedings. ITCC 2003. International Conference on*, 2003, pp. 367–372, tY - CONF.
- [13] D. Katz and D. Yeung, "Traffic engineering extensions to ospf version 2," Tech. Rep., 2002, internet Draft: draft-katz-yeung-ospf-traffic-09.txt.
- [14] E. Osborne and A. Simha, *Traffic Engineering with MPLS*. Cisco Press, 2003.
- [15] V. Alwayn, *Advanced MPLS design and implementation*. Cisco Press, 2002.
- [16] D. Awduche, L. Berger, D. Gan, T. Li, V. Sprinivasan, and G. Swallow, "Rsvp-te: Extensions to rsvp for lsp tunnels," 2001, request for Comments: 3209.
- [17] A. Farrel and B. Miller, "Surviving failure in mpls networks," Tech. Rep., 2001.
- [18] E. Rosen, A. viswanathan, and R. Callon, "Multiprotocol label switching architecture," 2001, request for Comments: 3031.