CHAPTER 2

LITERATURE REVIEW

This Chapter presents a brief introduction of Multi Protocol Label Switching (MPLS) techniques. Also, it narrates the research efforts concerned to load balancing schemes and restoration techniques used in the MPLS networks.

2.1 MPLS ENABLED INTERNET PROTOCOL NETWORKS


A MPLS enabled Internet consists of a set of nodes called Label Switch Routers (LSRs) and Label Edge Routers (LERs) as depicted in Figure 2.1. The LERs and LSRs are capable of switching and routing packets through multiple paths called Label Switched Paths (LSPs). LSPs are formed on the basis of a label which has been appended to each packet instead of the Internet Protocol (IP) address. The LERs present at the beginning edge of the MPLS domain are called ingress nodes and the LERs at ending edges of the MPLS domain are called ingress nodes. Load balancing is realized in MPLS networks by redistributing the traffic over two or more LSPs, which connect the same Ingress and Egress (IE) node pair.
2.1.1 Advantages of MPLS

MPLS has a number of advantages over conventional network layer forwarding. Some of the advantages are as follows:

- MPLS forwarding can be done by switches which are capable of doing label look-up and replacement.
- Depending upon the types of service required the packets are assigned to different Forward Equivalence Classes (FECs) when they enter the network.
- A packet that enters the network at a particular router can be labeled differently than the same packet entering the network at a different router. As a result, forwarding decisions that depend on the ingress router can be easily made. This cannot be done by conventional forwarding.
- The considerations that determine how a packet is assigned to a FEC can become even more complicated, without any impact at all on the routers that merely forward labeled packets.

- Sometimes it is desirable to force a packet to follow a particular route which is explicitly chosen at or before the time the packet enters the network, rather than being chosen by the normal dynamic routing algorithm as the packet travels through the network. This may be done as a matter of policy, or to support traffic engineering. In conventional forwarding, this requires the packet to carry an encoding of its route along with it (source routing). In MPLS, a label can be used to represent the route, so that the identity of the explicit route need not be carried with the packet.

### 2.1.2 MPLS Label

A label is a short, fixed length, locally significant identifier which is used to identify a FEC. The label which is put on a particular packet represents the FEC to which that packet is assigned. An MPLS label is a 32-bit field, consisting of the following elements as shown in Figure 2.2.

**Label Value:** The first 20 bits are the label value. This value can be between 0 and 1,048,575.

**EXP (Experimental):** There are three bits reserved for experimental use; for example, these bits could communicate Differentiated Service (DS) information or Per Hop Behavior (PHB) guidance. These bits are also used for Quality of Service (QoS).
S (Bottom of Stack): This bit is ‘0’, unless this is the bottom label in the stack. If so, the Bottom Of Stack (BOS) bit is set to 1. The stack is the collection of labels that are found on top of the packet. The stack can consist of just one label, or it might have more. The number of labels (that is, the 32-bit field) that are found in the stack is limitless, but it is seldom to notice a stack that consists of four or more labels.

MPLS capable routers may need more than one label on top of the packet to route that packet through the MPLS network. This is done by packing the labels into a stack. The first label in the stack is called the top label, and the last label is called the bottom label. Between these two labels, any number of labels may be present. Table 2.1 presents the structure of the label stack. The label stack shows that the BOS bit is 0 for all the labels, except for the bottom label. For the bottom label, the BOS bit is set to 1.
Table 2.1 Structure of the label stack

<table>
<thead>
<tr>
<th>Label</th>
<th>EXP</th>
<th>0</th>
<th>TTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>EXP</td>
<td>0</td>
<td>TTL</td>
</tr>
<tr>
<td>......</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Label</td>
<td>EXP</td>
<td>1</td>
<td>TTL</td>
</tr>
</tbody>
</table>

**Time To Live (TTL):** This TTL (8 bits) is simply decreased by 1 at each hop, and its main function is to avoid a packet being stuck in a routing loop. If a routing loop occurs and no TTL is present, the packet loops forever. If the TTL of the label reaches 0, the packet is discarded. Yoshihiro Ohba (1999) examined the issues on loop prevention in MPLS networks.

2.1.3 **Forwarding Equivalence Class**

A FEC is a group or flow of packets that are forwarded along the same path and are treated the same way, with regard to the forwarding treatment. All the packets belonging to the same FEC have the same label. However, not all packets that have the same label belong to the same FEC, because their EXP values may differ. Their forwarding treatment can be different, and they can be even belonging to a different FEC.

2.1.4 **Label Switch Router**

A LSR is a router that supports MPLS. It is capable of understanding MPLS labels and of receiving and transmitting a labeled packet on a data link. Three kinds of LSRs exist in MPLS network:
• **Ingress LSRs**—Ingress LSRs (called ingress LERs) receive a packet that is not labeled yet, insert a label (stack) in front of the packet, and send it on a data link.

• **Egress LSRs**—Egress LSRs (called egress LERs) receive labeled packets, remove the label(s), and send them on a data link. Ingress and Egress LSRs are edge LSRs.

• **Intermediate LSRs**—Intermediate LSRs receive an incoming labeled packet, switch the packet, and send the packet on the correct data link.

2.1.5 **Label Switched Path**

A Label Switched Path (LSP) is a sequence of LSRs that switch a labeled packet through an MPLS network or part of an MPLS network. The ingress LER of an LSP is not necessarily the first router to label the packet. The packet might have already been labeled by a preceding LSR. Such a case would be a nested LSP. It is in fact an LSP with another LSP inside.

Figure 2.3 shows an example of MPLS enabled Internet Protocol version 4 (IPv4) network. It is a network that consists of LSRs that run an IPv4 Interior Gateway Protocol (IGP). The ingress LER looks up the destination IPv4 layer 3 address of the packet, imposes a label, and forwards the packet (layer 3 routing). The next LSR (and any other intermediate LSR) receives the labeled packet, swaps the incoming label with an outgoing label, and forwards the packet (layer 2 forwarding). The egress LER pops the label and forwards the IPv4 packet without labels on the outgoing link (layer 3 routing). For this to work, the adjacent LSRs must agree on which label to use for each IGP prefix. Jeremy Lawrence (2001) discussed IP routing issues specific to MPLS networks.
2.1.6 Label Distribution with LDP

The Label Distribution Protocol (LDP) distributes labels between LSRs. As a protocol, the activities of LDP are separated into four categories:

1. Discovery of LDP-capable LSRs those are adjacent and connected by a logical or physical link.
2. Establishment of a control conversation between adjacent LSRs, and negotiation of capabilities and options
3. Advertisement of labels
4. Withdrawal of labels

LDP exchanges messages between LSRs as indicted in Figure 2.4.
Figure 2.4 Label distribution

Table 2.2 Label Information Base (LIB) for LER – 1

<table>
<thead>
<tr>
<th>Destination</th>
<th>Label</th>
<th>Out port</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.7</td>
<td>85</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2.3 Label Information Base (LIB) for LSR – 1

<table>
<thead>
<tr>
<th>IN port</th>
<th>Label IN</th>
<th>Destination</th>
<th>Out port</th>
<th>Label out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>85</td>
<td>128.7</td>
<td>3</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 2.4 Label Information Base (LIB) for LSR – 2

<table>
<thead>
<tr>
<th>IN port</th>
<th>Label IN</th>
<th>Destination</th>
<th>Out port</th>
<th>Label OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39</td>
<td>128.7</td>
<td>4</td>
<td>56</td>
</tr>
</tbody>
</table>

Table 2.5 Label Information Base (LIB) for LER – 2

<table>
<thead>
<tr>
<th>Destination</th>
<th>Label</th>
<th>Out port</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.7</td>
<td>56</td>
<td>3</td>
</tr>
</tbody>
</table>
The ingress router forms the LSP using the layer 3 protocol by distributing request signal to the LSRs. Each LSR binds a label to the IPv4 address and the downstream node distributes the label to its upstream node. LDP has two modes of label distribution: 1) Downstream on demand 2) Downstream unsolicited. In both cases, the downstream node is responsible for distributing the labels. However, in the first case, it does so only in response to a specific request from upstream, whereas in the second case, the downstream node distributes labels whenever it can. This process is called label mapping. The neighbours then store these remote and local bindings in a special table called Label Information Base (LIB). The LIB tables of LER 1, LSR 1, LSR 2 and LER 2 are represented in Tables 2.2, 2.3, 2.4 and 2.5 respectively.

For example, the LIB of LSR 1 in Table 2.3 shows that a packet with a label 85 enters the LSR through port 1. The LSR swaps the label with the help of its LIB (LIB-2) and sends the packet with a new label (39) to the next LSR through port 3. The egress LER (LER 2) removes the label and forwards the packet to the destination using IP address.

2.2 LOAD BALANCING

The characteristics of internet traffic have been studied at length by Paxson V and Floyd S (1995) and Crovella M and Bestavros A (1997). The traffic behaviour is independent of the time scale and sophisticated load balancing techniques are required to manage it.

Generally, load balancing is classified as static version as well as dynamic version. Bertsekas D and Gallager R (1992) discussed static version of the load balancing method. In static version, the traffic demand and network topology are assumed as known. As the static problem is formulated
and solved with precise information of the traffic demands, it delivers optimal result in load balancing. Hence it is used as a reference for the other load balancing schemes.

In dynamic load balancing, the traffic demands are unknown and the link loads are measured periodically at times \( t_i \). These measured link loads \( y_i(i) \) are the time-averages over the whole measurement period \( (t_{i-1} \rightarrow t_i) \). Dynamic load balancing iteratively makes incremental changes in the splitting ratio to balance the load among the paths. At every step, it splits a part of the load from the maximum loaded path into minimum loaded path. In dynamic load balancing, the total path cost is used to formulate the dynamic load balancing algorithm.

There are various dynamic load balancing algorithms available to balance the load and also increase the performance of MPLS enabled Internet protocol (IP) networks.

Anoop Ghanwani et al (1999) enunciated the traffic engineering standards in MPLS enabled IP networks. They pointed out the direction taken by the Internet Engineering Task Force (IETF) and the recent standardization efforts for traffic engineering using multi-protocol label switching. The traffic engineering approach to address the shortcomings of the traffic handling in IP networks using MPLS and the signaling protocols used for MPLS-based traffic engineering are also discussed by them.

In load balancing, performance optimization of networks is actually a control problem. Traffic engineering should provide sufficient control for an adaptive feedback control system. The control capabilities offered by existing IGP are not adequate for traffic engineering. The tasks of a controller consist of modification of traffic management parameters, modification of routing
parameters and modifications of resource attributes. Awduche D et al (1999) concerned with these considerations in RFC-2702. This document also presents a set of requirements for traffic engineering over MPLS. It identifies the functional capabilities required to implement policies that facilitate efficient and reliable network operations in an MPLS domain.

George Swallow (1999) focused on the advantages of MPLS for load balancing. Important features of MPLS, according to the author, are the scalability and flexibility of routing with performance, quality of service and traffic management of layer-2 switching. It is also showed that local repair is very important for sustaining voice calls and other high-priority services. Xipeng Xiao et al (2000) discussed constrain based routing to provide a background for traffic engineering with MPLS in the Internet.

Load balancing in traffic engineering encompasses many aspects of network performance. These include the provision of a guaranteed QoS. Wei Sun et al (2000), Hitomi Tamura et al (2004) and Hong D.W et al (2005) have dealt at length QoS using traffic engineering over MPLS.

Delay-based adaptive load balancing for MPLS networks proposed by Deyun Gao et al (2002) does not requires the core LSRs to perform traffic engineering. Based on measurements, the average one way delay between a pair of LSR is obtained and the traffic load is dynamically distributed among multiple LSP according to this average one way delay. The objective here is to provide a distributed control algorithm that could accommodate incoming traffic. The probe packets are used to measure the delays and delay-based heuristic balancing to shift traffic among several LSPs. The algorithm is implemented along with other MPLS modules in the network simulator NS-2.
MPLS Adaptive Traffic Engineering (MATE) introduced by Elwalid A et al (2001) is a state-dependent traffic engineering mechanism. In which the traffic load is balanced using a distributed adaptive algorithm. This mechanism adopts a simple approach in that intermediate nodes are not required to perform traffic engineering or make measurements besides normal packet forwarding. It does not impose any particular scheduling, buffer management or a priori traffic characterization on the nodes. The delay and the packet loss are measured periodically by sending probe-packets between the ingress and the egress nodes of a LSP. When a change of predefined size in these traffic conditions is perceived, the algorithm moves from the monitoring phase to the load balancing phase. In this phase, the algorithm tries to equalize congestion measures among the LSPs by approximating the gradient-projection algorithm. After the load balancing phase, the algorithm moves back to the monitoring phase. But the uses of active path measurements offer overlapping information since LSPs might use the same links. The amount of control data can be reduced by using only link load information. In the network scenario, MATE is suitable only when a few ingress-egress pairs are considered.

Keping Long et al (2001) proposed three load-balancing algorithms: 1) Topology-based static load balancing algorithm, 2) Resource-based static load balancing algorithm and 3) Dynamic load-balancing algorithm for MPLS traffic engineering. These algorithms are suitable only for non-priority point to point traffic. A data structure named route to describe a route character, which includes its resource availability and its topological character, is constructed first. Then, a data structure named Selectable Route Collection (SRC) is constructed. SRC comprises all parallel routes between ingress and egress node pair. Under light load condition, traffic flows are mapped on the short and high capacity routes. When the load increases
beyond normal, a small traffic flow is rerouted to another appropriate route so as to a reserve high capacity route for a large traffic flow.

An essential part of planning and managing operational networks is the measurement process. Butenweg S (2003) described a decentralized measurement system in MPLS networks for traffic engineering. In this system, each LSR monitors the total load of its outgoing links over a certain time period. Then it calculates the average load of each outgoing link. In addition, the traffic load of each LSP passing through the router is monitored and the average load is calculated. Information on the link load is distributed to the other routers in the network by means of flooding mechanism. The load balancing is realized by either rerouting or multi path load balancing. To prevent the system from oscillations, the distributed traffic engineering units coordinate the rebalancing actions by using routing update messages.

Load Distribution over Multi-path (LDM) is a technique which splits the traffic dynamically at flow level into multiple paths (Song J et al, 2003). The set of available LSPs is fixed. The LSP for the incoming traffic is selected based on congestion and length of the path. When a particular link becomes congested, those ingress and egress pairs have to shift the traffic to a new LSP. A new LSP is selected randomly according to probabilities that depend on the path utilizations and the path lengths. The momentary efficiency of the algorithm depends largely on the flow-level dynamics and it is difficult to keep the granularity of the traffic splitting at fine grain level to provide stable network conditions.

Traffic Engineering Automated Manager (TEAM) for Differentiated Services (DS)/MPLS networks is designed for the complete automated management of an Internet domain (Caterina Scoglio et al, 2004). TEAM is an adaptive manager that provides the required quality of service to
the users and reduces the congestion in the network. Network simulations NS-2 is used to evaluate the performance of the TEAM. Multi-path load balancing model at the flow level proposed by Zenghua Zhao et al (2004) is a heuristic but efficient mechanism. It is implemented only at the ingress LSRs and egress LSRs in the network.

Hanoch Levy et al (2004) formulated the resource allocation problem by accounting both for network utilization and for connection processing constraints in ATM and MPLS networks. An important question in the design of these networks is the amount of network resources to be dynamically allocated to and held by the virtual path agents. An allocation which is too high will result in bandwidth resource waste, whereas too low will result in heavy connection set-up and teardown processing load. The authors dealt with this problem and derived a simple operational rule to determine the amount of bandwidth resources, held by the various virtual path agents, while balancing bandwidth waste and connection processing overhead. Sudeept Bhatnagar and Samrat Ganguly (2005) discussed about creation of parallel paths using multipoint-to-point LSPs for traffic engineering.

Distributable traffic based MPLS dynamic load balancing scheme proposed by Gang Yuan et al (2005) defines a parameter called Distributable Traffic (DT) to reflect the availability of the unused bandwidth of LSPs. According to the predefined formulation of DT, the real time measurements of queuing delay and packet loss probability are necessary for the computation of DT. This is done by allowing the ingress node to transmit the probe packets periodically to the egress node, which in turn returns to the ingress node, on every parallel LSPs. Based on the information in the returning probe packets; the ingress node is able to compute the one-way LSP statistics, and packet loss probability. Probe packets are transparent for core
LSRs and most of the tasks are performed in the ingress node so as to maintain the scalability. Traffic distribution is performed depending upon the values of DT. A higher value of DT of a LSP represents the availability of more unused bandwidth. It means that more traffic can be distributed on that LSP. Only LERs are required to perform this special function in DT load distribution scheme.

2.3 RESTORATION

To deliver reliable service, MPLS requires a set of procedures to provide protection for the traffic, carried on different paths. Thomas M.C. and Tae H.O. (1999) examined the distributed methods for fast fault recovery using modified label distribution protocol messages. Tae H.O et al (2000) investigated fault restoration and spare capacity allocation with QoS constraints for MPLS Networks. They examine the distributed fault restoration techniques for MPLS to automatically reroute label switched paths in the event of link or router failures by maintaining QoS requirements.

Building reliable MPLS network using path protection mechanism (Changcheng Huang et al 2002) describes the design considerations, the communication of fault information to appropriate switching elements, and the fault detection protocol. This consists of three components: a reverse notification tree, a hello protocol, and a lightweight notification transport protocol. One of the key issues in any path protection mechanism is the delay. In this algorithm, the delay is reduced by building a fast and efficient notification tree structure.

Path restoration scheme for MPLS based network introduced by Santos Rumar Das and Venkataram P (2002) presents an algorithm of a fault management scheme for MPLS network. Authors used a backup path
restoration scheme and a dynamic restoration scheme by assigning an alternative path for a working path.

Marzo J.L et al (2003) presented QoS online routing and MPLS multilevel protection. Usually fault management methods pre-establish backup paths to recover the traffic of failure path. Author introduced several LSP backup path types and pointed out their pros and cons. As the formation of a new LSP includes QoS aspects, the backup paths also incorporate QoS aspects.

2.4 MPLS NETWORK SIMULATOR (MNS)

Many techniques have been developed to enhance the performance of the MPLS networks. It is very difficult to evaluate the performance of these techniques by deploying on real MPLS networks. Instead, those techniques can be tested in a simulated environment. Gaeil Ahn and Woojik Chun (2000) proposed the design and implementation of MPLS Network Simulator (MNS) supporting LDP and Constrain Based LDP (CR-LDP). These authors describe design, implementation, and capability of MPLS simulator, which supports label swapping operation, LDP, CR-LDP, and various sorts of label distribution function. It enables researchers to simulate how an LSP is established and terminated, and how the labeled packets act on the LSP. Besides, the authors also simulated the basic MPLS function defined in MPLS standards; label allocation scheme, LSP trigger strategy, and label distribution control mode.

Gaeil Ahn and Woojik Chun (2001) recommended a MPLS Network Simulator (MNS) built on the NS, which supports QoS. They described the design and implementation of MNS, focusing on its components: CR-LDP, MPLS Classifier, Service Classifier, Admission
Control, Resource Manager, and Packet Scheduler. To verify the accuracy and efficiency of MNS, two examples have been simulated and evaluated. One is the simulation for several kinds of traffic with different QoS per each simulation. The other is the simulation for resource preemption. Design of MNS have also been studied by Ali Boudani et al (2003).

Design and implementation of Resource Reservation Protocol with Traffic Engineering (RSVP-TE) network simulator for differentiated service aware MPLS networks (Adami D et al 2005) provides a new software module to simulate the RSVP-TE protocol in the Network Simulator 2 (NS2). This module deals a complete implementation of the control plane mechanisms needed for label distribution, label binding, DS-aware traffic engineering and end to end recovery mechanisms in MPLS networks. This module is developed, taking into account extensibility and flexibility features. Hence, enhancements to the signaling protocol are easily introduced and it is useful to speedup the design, development and deployment of MPLS networks.

Adami D et al (2006) dealt at length design, development and validation of an NS2 module for dynamic LSP rerouting. This module enhances NS2 with new fundamental functionalities, such as dynamically rerouting LSPs, taking into account traffic engineering metrics.

2.5 SUMMARY

Most of the works discussed above use path load as a control variable to balance the load. The active path load measurements will create overlapping information since LSPs might use the same links. As a result, the re-distribution of the traffic based on this overlapping information increases the packet loss and thus, decreases the throughput.
Another limitation is the number of iterations required to reach the balanced state and the magnitude of oscillations. As the dynamic load balancing is a closed loop control system, it generates oscillations due to concurrent rebalancing actions. There is a tradeoff between the number of iterations required to balance the load and the stability of the system (without oscillations). While the small (fine grain) incremental steps reduce the magnitude of oscillations, the number of iterations required to reach the balanced state becomes high. Instead, the large (course grain) incremental steps reduce the number of iterations required to reach the balanced state. However, the magnitude of oscillations becomes high and harmful. In the above load balancing algorithms, fine grain incremental steps are used to avoid the oscillations. As the fine grain incremental steps increase the number of iterations required to reach the balanced state, the throughput is reduced from its maximum value.

Hence, new algorithms to provide fast load balancing without generating the oscillation are greatly required for the IP networks. The effective utilization of resources increases the throughput of the network to the maximum level and also, decreases the end to end delay. In addition to the fast load balancing, fast restoration techniques are also required to provide protection to the traffic carried on the links of the network. As the Network Simulator NS2 is a vital tool for simulation, it can be used to evaluate the performances of the load balancing algorithms.