Analysis of Traffic Engineering Deployment Strategies in Core IP/MPLS Networks

By
Mohamed Hasan Omar

A project report submitted in partial fulfillment of the requirements for the degree of

Master of Science in Internetworking
(Department of Computer Science)

University of Alberta
April 2008

Abstract

Internet Traffic is growing tremendously over the last few years. Although the long-term market behavior of the Internet is difficult to forecast, Internet traffic is clearly growing in a phenomenal progression.

Traffic Engineering is deployed by large carriers to address the Internet growth challenge. It's responsible for link bandwidth and for the size of the traffic flow when determining explicit routes across the backbone. Traffic engineering enables ISPs to route network traffic in such a way that they can offer the best service to their users in terms of throughput and delay.

As conventional IP technologies have limited functional capabilities. One particular shortcoming of conventional IP systems is the inadequacy of measurement functions. For example, a traffic matrix, which is a basic data set needed for traffic engineering, is difficult to estimate from interface statistics on IP routers. The limitations of intra-domain routing control functions are another issue with conventional IP systems. Interior gateway protocols (IGPs), such as Open Shortest Path First (OSPF), commonly used to route traffic within autonomous systems in the Internet, are topology-driven and employ per-packet progressive connection control. Each router makes independent routing decisions using a local instantiation of a synchronized routing area link state database. Route selection is based on shortest path computations using simple additive link metrics. This approach is highly distributed and scalable, but these protocols do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions, which results in some links of the network become congested, while other resources along alternate paths remain underutilized. This type of congestion problem is a symptom of poor resource allocation, and is an issue that traffic engineering specifically attempts to redress.

Recent developments in multiprotocol label switching (MPLS) open new possibilities to address some of the limitations of IP systems concerning traffic engineering.

Although MPLS is a relatively simple technology (based on the classical label swapping mechanism), it enables the introduction of sophisticated control capabilities that advance the traffic engineering function in IP networks. Particularly interesting aspects of MPLS are that it efficiently supports:

- Traffic Engineering
- Quality of Service (QOS)
- Fast Reroute (Link, node Protection)

This project first analyses how MPLS can significantly improves the performance and scalability of service providers and carrier backbone networks. Also, it discusses the applications of MPLS to traffic engineering in IP networks, focusing specifically on service provider networks. It presents a comparative analysis of MPLS and non-MPLS network and shows the MPLS traffic engineering capabilities to improve network performance for different application types in heavy loaded traffic environments. Traffic engineering is the main strength of MPLS. Where an IP-based network is connectionless, MPLS-enabled networks defines specific paths for network traffic. MPLS TE also supports explicit routing. By using explicitly routed LSPs, all paths in MPLS can be utilized and controlled for sending packets. Thus explicit routing can optimize the utilization of network resources and enhance traffic performance characteristics.

Acknowledgments

This Master's Project was developed at the state-of-the-art MINT lab at the University of Alberta. I would like to thank my supervisor Dr. Mike McGregor for his support and inspiration. Without his guidance and encouragement this project would never have been done.

I would like to express my gratitude to the MINT program as I've learnt various cutting edge technologies and to go beyond the limitations of some of the current technologies. Also, in the MINT program, I have had the opportunity to learn from highly experienced professors and to work in teams with many interesting persons.

Finally, I would like to express my sincere gratitude to my parents and my fiancée, for their patience and support during my MSc. studies.

Table of Contents

Chapter 1
Introduction

Services offered on today's networks are rapidly evolving in scope and availability as new access technologies is coming into view like multimedia interactive distance learning, video conferencing applications. Hence, there are lots of effort is spent towards the convergence of these network services over a robust multi-service network is driving the Internet to cope with new network utilization challenges.

The Problem

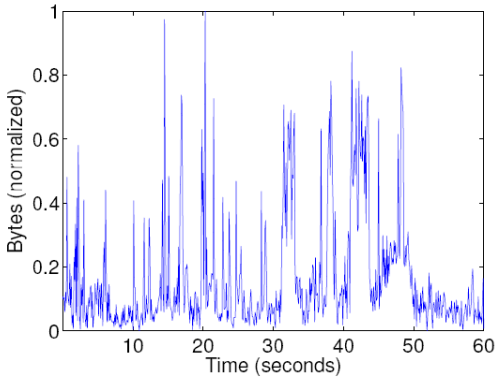


Figure 1.1 : Bytes per 100 millisecond sent on a link close to the edge of the Internet during one minute

Internet traffic has various characteristics depending on location in the network and at what time scale the traffic is observed. Figure 1.1 shows the number of bytes during each 100 millisecond interval of one minute over a link close to the edge of Internet. The plot reveals a clear bursty behavior with periods with large amounts of bytes transmitted interchanged with periods with low traffic intensity.
However, it is desirable for a network operator to keep the routing stable in order to avoid oscillatory behavior of the traffic, minimize routing signaling traffic and avoid instability in the routing system.
Traditional IP networks offer little predictability of service, which is often unacceptable for applications such as telephony, as well as for emerging and future real-time application, such as telemedicine. Also, the rapid growth of Internet users and the fact that the current basic traffic forwarding paradigm cannot support traffic engineering, create additional weaknesses.
Traffic engineering is preferably performed for a stable traffic situation.

Motivations

The current IP routing technologies deployed in conventional network utilize the best available path information based on the destination address, and the network capacity utilization is not considered. As the network grows, routers will need to handle even higher volumes of information, besides making forwarding decisions at each hop, insuring scalability and performance.

MPLS provides for the possibility to differentiate between the paths certain types of traffic follow from a source to a destination. With MPLS technology, we can provide a mechanism to dynamically define the path of certain "mission critical" traffic that has specific QoS requirements. For example, voice and video have become of the most important types of traffic carried on today's networks.

Unlike data traffic, real-time traffic needs to travel through the network without being subject to delay or packet re-ordering. A standard IP network, operating on a best effort basis, is unable to guarantee such preferred treatment.

MPLS is gaining significant attention as it's a transport networking technology in the future Next Generation Internet due to its easy implementation of efficient traffic engineering. TE aims at the ability to efficiently map available traffic onto existing network topology in a way that optimizes the utilization of network resources, and ensures QoS constraints are met. MPLS supports TE by allowing the node at the network ingress to specify the path that a LSP will take using explicit routing (ER) features. An MPLS-enabled network is able to provide low latency and guaranteed traffic paths for real-time traffic.

The Solution : Multi-Protocol Label Switching (MPLS)

A brief history of MPLS

MPLS evolved from several similar technologies that were invented in the mid-1990s. Several approaches were published, notably Cisco's Tag Switching and IBM's Aggregate Route-based IP Switching (ARB).
These approaches had a number of characteristics in common; they were not interoperable because each relied on Merent technologies to combine IP routing and ATM switching into an integrated solution. However, by early 1997, many in the Internet community were so impressed with the simplicity and elegance of these solutions; they began to view multilayer switching as the next logical evolutionary step for the design of large ISP backbone networks.
MPLS technology is now well on its way to becoming an industry standard. The IETF draft framework document says that MPLS as a "base technology (label swapping) is expected to improve the price/performance of the network layer routing improve the scalability of the network layer, and provide greater flexibility in the delivery of (new) routing services.

THE MULTI-PROTOCOL LABEL SWITCHING (MPLS) MECHANISM [17]

An IP router implements both control and forwarding components. The control component consists of routing protocols, such as the open shortest path first (OSPF), the border gateway protocol (BGP), used to construct routes and exchange routing information between IP routers. This information is used by the IP routers to construct the forwarding routing table, referred to as the forwarding information base (FIB). The forwarding component consists of procedures that a router uses to make a forwarding decision on an IP packet. For instance, in unicast forwarding, the router uses the destination IP address to find an entry in the FIB, using the longest match algorithm. The result of this table look-up is an interface number, which is the output port connecting the router to the next hop router, to which the IP packet should be sent.

A router forwards an IP packet according to its prefix. In a given router, the set of all addresses that have the same prefix, is referred to as the forwarding equivalent class (FEC). IP packets belonging to the same

FEC have the same output interface. In MPLS, each FEC is associated with a different label. This label is used to determine the output interface of an IP packet without having to look-up its address in the FIB. A label is a short fixed-length identifier that has local significance. That is, it is valid on a single hop interconnecting two routers. A label is similar in functionality to the VPI/VCI value associated with an ATM cell.

In IPv4, there is no space for such a label in the IP header. If the IP network runs on top of an ATM network, then the label is carried in the VPI/VCI field of an ATM cell. If it is running over frame relay, the label is carried in the DLCI field. For Ethernet, token ring, and point-to-point connections that run a link layer protocol like PPP, the label is encapsulated and inserted between the LLC header and the IP header. The first field of the label encapsulation is a 20-bit field used to carry the label. The second field is a 3-bit field used for experimental purposes. It can for instance carry a class-of-service (CoS) indication, which can be used to determine the order in which IP packets will be transmitted out of an interface. The S field is used in conjunction with the label stack, which will be discussed in detail later on in this chapter. Finally, the time-to-live (TTL) field is similar to the TTL field in the IP header.

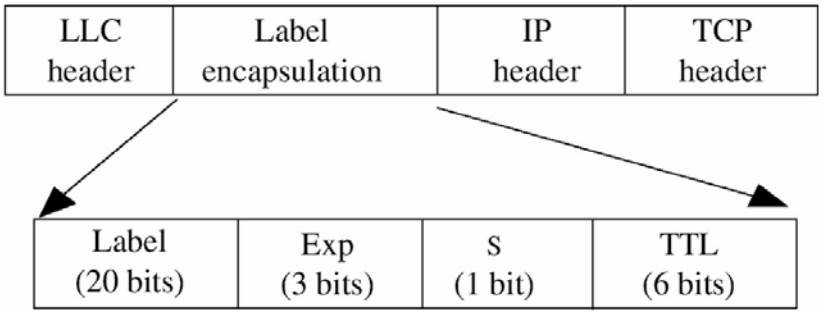


Figure 1.2 : Label encapsulation.

An MPLS network consists of label switching routers (LSR) and MPLS nodes. An LSR is an IP router that runs the MPLS protocol. It can bind labels to FECs, forward IP packets based on their labels, and carry the customary IP forwarding decision by carrying out a table look-up in the FIB using a prefix. An MPLS node is an LSR, except that it does not necessarily have the capability to forward IP packets based on prefixes.

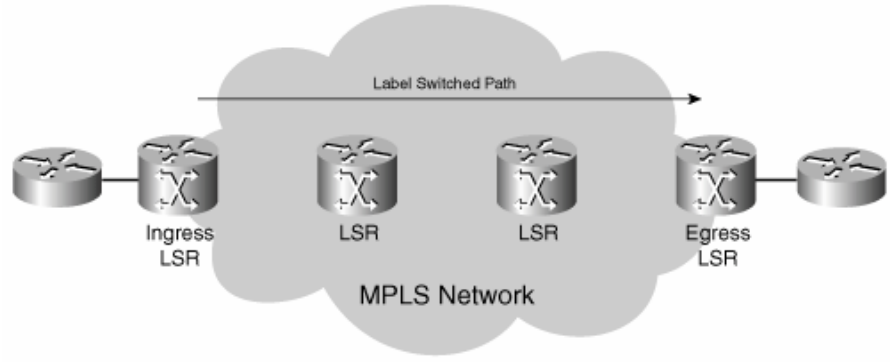
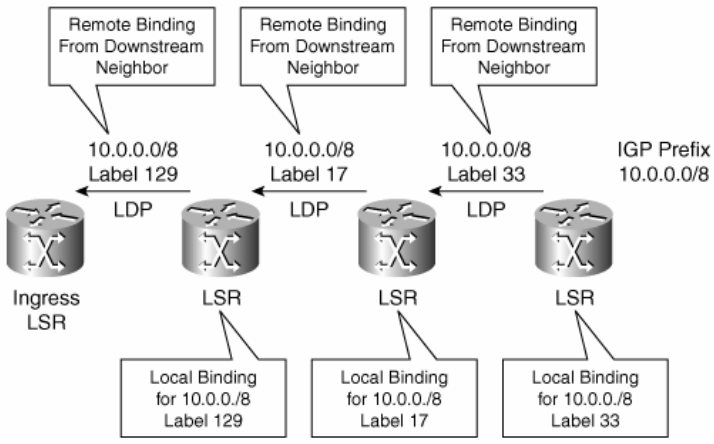


Figure 1.3 : An LSP Through an MPLS Network

Forwarding Equivalence Class

A Forwarding Equivalence Class (FEC) is a group or flow of packets that are forwarded along the same path and are treated the same with regard to the forwarding treatment. All 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 might differ; the forwarding treatment could be different, and they could belong to a different FEC. The ingress LSR decides which packets belong to which FEC. This is logical because the ingress LSR classifies and labels the packets.

The destination IP address of all IP packets entering the ingress LSR will be looked up in the IP forwarding table. All these addresses belong to a set of prefixes that are known in the routing table as BGP prefixes. Many BGP prefixes in the routing table have the same BGP next-hop address, namely one egress LSR. All packets with a destination IP address for which the IP lookup in the routing table recurses to the same BGP next-hop address will be mapped to the same FEC. All packets that belong to the same FEC get the same label imposed by the ingress LSR.



Figue 1.4 : An IPv4-over-MPLS Network Running LDP

```
2800-2#sh mpls ip binding
  10.1.4.0/24
        in label: imp-null
  10.1.20.0/24
        in label:     17
  10.1.30.0/24
        in label:     16
```

```
low-1#sh mpls forwarding-table
Local  Outgoing    Prefix           Bytes tag  Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched   interface
16     Untagged[T] 10.1.4.0/24      0          Tu0         point2point
17     18          10.1.20.0/24     0          Se0/0       point2point
18     19          150.1.23.0/24    0          Se0/0       point2point
19     Pop tag     150.1.22.0/24    0          Se0/0       point2point
20     20          150.1.21.0/24    0          Se0/0       point2point
21     Pop tag     150.1.26.0/24    0          Se0/0       point2point
22     21          150.1.25.0/24    0          Se0/0       point2point
23     22          150.1.24.0/24    0          Se0/0       point2point
24     23          150.3.3.3/32     0          Se0/0       point2point
```

In Figure 1.4, it's shown the labels allocated by the LSRs. They have local significance; that is, each label is valid only for one link.

```
2800-2#traceroute 150.7.7.7

Type escape sequence to abort.
Tracing the route to 150.7.7.7

  1 150.1.21.2 [MPLS: Label 27 Exp 0] 120 msec 120 msec 120 msec
  2 150.1.22.2 [MPLS: Label 26 Exp 0] 56 msec 56 msec 56 msec
  3 150.1.27.2 44 msec 44 msec *
```

The Next Hop Label Forwarding Entry (NHLFE)

So far, for presentation purposes we have assumed that an LSR maintains a single entry for each incoming label. In this entry, it binds the incoming label with an outgoing label and it provides information regarding the next hop, such as the next LSR and the output interface.

The MPLS architecture permits an LSR to maintain multiples entries for each incoming label. Each entry is known as the next hop label forwarding entry (NHLFE), and it provides the following information: the packet's next hop, and the operation to be performed on the packet's label. Each NHLFE entry can also contain additional information necessary in order to properly dispose of the packet.

MPLS permits a packet to carry multiple labels which are organized as a stack. An example of the label stack is given in Figure 1.5. Each row contains a different label encapsulation.

Figure 1.5: The label stack.

The following three operations can be performed on the packet 's label:

- Replace the label at the top on the packet's label stack with a new label.
- Pop the label stack.
- Replace the label at the top of the packet's label stack with a new label, and then push one or more new labels on to the stack.

In the case where the next hop of an LSR is the LSR itself, the LSR pops the top level label, and the resulting packet is forwarded based on whatever remains after the label stack was popped. This might still be a labeled packet or it might be a native IP packet that has to be forwarded based on its prefix.

The incoming label map (ILM) maps an incoming label to a set of NHLFEs associated with the incoming label. Having multiple entries for each incoming label can be useful because that allows multi-pathing for load balance and protection to be implemented.

Finally, there is the FEC-to-NHLFE map (FTN), which is used to map a FEC to a set of NHLFEs. This is used when a packet arrives unlabeled, and it has to be labeled before it is forwarded. As in the case of the ILM, if the FTN maps a FEC to multiple NHLFEs, a procedure is required to select one of them. Such a procedure is not defined in the MPLS architecture.

Explicit Routing

An IP router makes a forwarding decision by using the destination IP address of a packet in its FIB in order to determine the next hop IP router. When using a link-state protocol such as OSPF, each IP router learns about the topology of its domain by exchanging information with the other IP routers. It then calculates the next hop IP router for each destination using the shortest path algorithm. This next hop is stored in its FIB. MPLS uses the same next hop information in order to set up an LSP. In view of this, this type of routing is known hop-by-hop routing.

In addition to the hop-by-hop LSPs the MPLS architecture permits the creation of an LSP that follows an explicit route through a network which might not necessarily correspond to the hop-by-hop path. This type of routing is referred to as explicit routing. An explicitly routed LSP in MPLS is the equivalent of a point-to-point connection in ATM networks. An explicit route might be set up to satisfy a QoS criterion, such as minimizing the total end-to-end delay and maximizing throughput. Such a QoS criterion might not be necessarily satisfied by the hop-by-hop routing, which in general strives to minimize the number of

hops only. Also, explicit routing can be used to provide load- balancing, by forcing some of the traffic to follow different paths through a network, so that the utilization of the network links is as even as possible. Finally, explicit routing can be used to set up MPLS-based tunnels and virtual private networks (VPN).

An explicit route can be strictly explicitly routed or loosely explicitly routed. In the strictly explicitly routed case, the path for the ingress LSR to the egress LSR is defined precisely. That is, all of the LSRs through which the path will pass are explicitly specified. In the loosely explicitly routed case, not all of the LSRs through which the path will pass are specified. For instance, if a path has to go through several domains, the actual path through a domain might not be defined precisely. In this case, the MPLS edge LSR will calculate a path through its domain.

Schemes for Setting up an LSP

In the MPLS architecture it is possible to force an LSP to be set up through LSRs in a particular order. Specifically, the following two schemes can be used to set up an LSP: independent LSP control, and ordered LSP control. In the independent LSP control scheme, each LSR binds a label to a FEC and advertises the binding to its neighbors as soon as it recognizes a new FEC. In the ordered control case, the allocation of labels proceeds backwards from the egress LSR. The following rules are used: an LSR only binds a label to a FEC if it is the egress LSR for that FEC, or if it has already received a label binding for that FEC from its next hop LSR. The ordered LSP control scheme is used to set up an explicit route.

Within an MPLS domain, it is possible that IP packets belonging to two or more different FECs follow the same route. This can happen when these FECs have the same egress node. In this case, it is possible to aggregate these FECs in to one or more FECs, or to not aggregate them at all and simply keep them separate.

MPLS Traffic Engineering

The overlay model in which IP is run over an ATM or Frame Relay network which results in distinct Layer 2 and Layer 3 networks. The IP network operates over a virtual topology in which every other router is one hop away. This causes difficulties and slows the network's responses to events such as link or node failures. MPLS allows the elements of traffic engineering to be completely under the control of IP. This results in a one-tier network that can offer IP services that now can be achieved only by overlaying a Layer 3 network on a Layer 2 network. This provides a way to achieve the same traffic engineering benefits of the overlay model without needing to run a separate network and without needing a non-scalable full mesh of router interconnects.

MPLS traffic engineering uses Resource Reservation Protocol (RSVP) to automatically establish and maintain a tunnel across the backbone. The path used by a given tunnel at any point in time is determined based on the tunnel resource requirements and network resources, such as bandwidth. Available resource information is flooded via extensions to a link–state–based IGP such as OSPF or IS-IS.

Tunnel paths are calculated at the tunnel head (source router) based on a fit between required and available resources (constraint-based routing). The IGP automatically routes the traffic into these tunnels. Typically, a packet crossing the MPLS traffic-engineering backbone travels on a single tunnel that connects the ingress point to the egress point.

MPLS Traffic Engineering works by using OSPF or IS-IS to distribute information about available resources in your network. Three major pieces of information are distributed:

- Available bandwidth information per interface, broken out by priority to allow some tunnels to preempt others
- Attribute flags per interface
- Administrative weight per interface

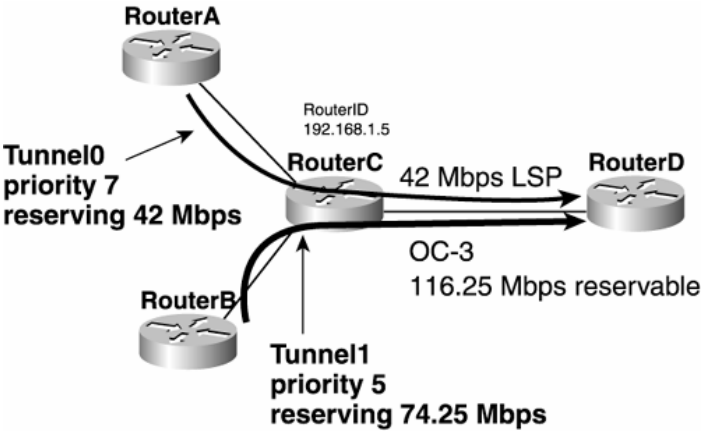


Figure 1.6 : Two Tunnels of Different Priorities

When Information Is Distributed the IGP floods information about a link in three cases:

- When a link goes up or down
- When a link's configuration is changed (when link cost is modified, for example)
- When it's time to periodically reflood the router's IGP information

All sorts of timers are associated with these actions. They differ depending on which IGP you use.

However, MPLS Traffic Engineering adds another reason to flood information—when link bandwidth changes significantly.

Resource Reservation Protocol (RSVP)

RSVP Basics

RSVP is a signaling mechanism used to reserve resources throughout a network. It has its own protocol type (46), although it is possible to encapsulate RSVP in UDP. MPLS TE never encapsulates RSVP in UDP, so that isn't discussed further.

RSVP is not a routing protocol. Any routing decisions are made by the IGP (including TE extensions) and CSPF. RSVP's only job is to signal and maintain resource reservations across a network. In MPLS TE, RSVP reserves bandwidth at the control-plane layer; there is no forwarding-plane policing of traffic. When

used for other purposes (such as VoIP or DLSW+ reservations), RSVP can be used to reserve Weighted Fair Queuing (WFQ) space or build ATM SVCs. Those uses are not discussed here.

RSVP has three basic functions:

- Path setup and maintenance
- Path teardown
- Error signaling

RSVP is a soft-state protocol. This means that it needs to periodically refresh its reservations in the network by resignalling them. This is different from a hard-state protocol, which signals its request once and then assumes that the request is up until it is explicitly taken down. With RSVP, a request goes away either if it is explicitly removed from the network by RSVP or if the reservation times out.

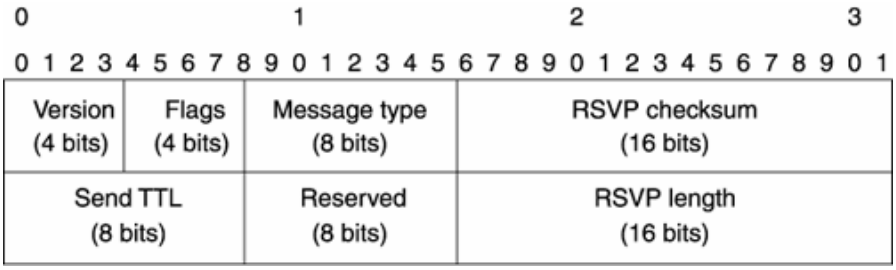


Figure 1.7 : RSVP Common Header Format

Fast Rerouting

Traffic-engineered networks must be able to respond to changes in network topology and maintain stability. Any link or node failure should not disrupt high-priority network services, especially the higher classes of service. Fast rerouting is a mechanism that minimizes service disruptions for traffic flows affected by an outage, and optimized rerouting reoptimizes traffic flows affected by a change in topology. In MPLS, splicing and stacking techniques are utilized to enable local repair of LSP tunnels.

Protection can be broken into

- Path protection (sometimes called end-to-end protection)
- Local protection, which can be broken into two types:
    - Link protection
    - Node protection

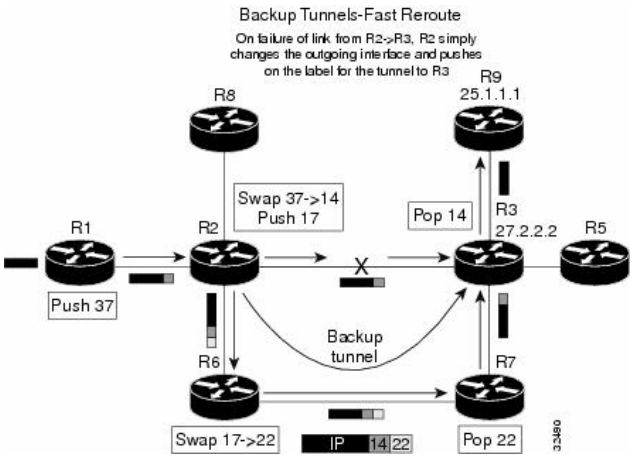


Figure 1.8 : Backup tunnel

Path Protection

Path protection is essentially the establishment of an additional LSP in parallel with an existing LSP, where the additional LSP is used only in case of failure. This LSP is sometimes called the backup, secondary, or standby LSP. The backup LSP is not used to carry traffic except during a failure condition—hence, the term standby.

The backup LSP is built along paths that are as diverse as possible from the LSP they're protecting. This ensures that a failure along the path of the primary LSP does not also affect the backup LSP. Path protection is simple in concept. Each primary LSP is backed up by a standby LSP. Both the primary and backup LSPs are configured at the headend. Both are signalled ahead of time in the control plane.

The primary and backup LSPs might have the same constraints. If the primary LSP has a bandwidth reservation of 100 Mbps, the backup LSP can also reserve 100 Mbps. This way, the end-to-end characteristics essentially remain the same, no matter whether the LSP used to carry traffic is the primary LSP or the protection LSP.

simply having a second path option under the tunnel interface does not make it path protection—it would be an LSP reroute. Path protection has better convergence than IGP convergence in an IP network or MPLS TE LSP reroute, because it makes use of a presignalled LSP that is ready to go in case the primary LSP fails. With path protection, the relationship between the backup LSP and the number of primary LSPs it is protecting is 1:1. This makes the path protection scheme less scalable.

In other words, for every LSP you want to protect, you have to signal another LSP. If you want the primary and backup LSPs to share the same bandwidth characteristics, they need to reserve the same amount of bandwidth. Protection LSPs kick in only when there's a failure, and hopefully your network failure rate is far less than 50 percent, so you end up reserving backup bandwidth that you won't use most of the time and keeping other LSPs in the network from being able to use that bandwidth. Path protection is not currently available on Cisco routers.

Link Protection Overview

In many networks that are deployed today, it is common to see high-bandwidth links carrying traffic belonging to "important" flows and other flows that are not so important. If MPLS TE is deployed in such networks, "important flows" translates to "important LSPs." These LSPs might be carrying critical information or time-sensitive data that requires a real-time response. In such cases, it would be nice if all the "important LSPs" could be protected while ignoring the less-important LSPs. FRR allows you to protect some of your TE tunnels (just the ones you deem important) or all of your TE tunnels. With link protection, you can protect links that are carrying these important LSPs by using presignalled backup tunnels that bypass the protected link.

Node Protection Overview

What if the node that is downstream of the protected link goes down, causing the protected link to fail? If this happens, it does you no good to try to deliver the packets to the node that just went down. In this case, you might want to safeguard the primary LSP from node failure on the other end of a link in addition to protecting the link itself.

Looking at this slightly differently, if you protect against failure of the downstream node, you have automatically protected against failure of the downstream link as well. Node protection uses NNHop backup tunnels instead of NHop tunnels because it needs to protect against a failure of the NHop.

Optimized Rerouting

Fast rerouting can result in suboptimal traffic-engineered paths. The key is to dynamically respond to failure as well as to new or restored paths. Thus, when a failure is detected, it is necessary to also notify the headend of the LSP tunnel. The headend can then compute a more optimal path. Traffic can then be diverted to the new LSP tunnel. This can be done without further disruption.

Often missing from Layer 2 networks is a feature called bridge-and-roll or make-before-break. This is the capability to always set up a new VC while maintaining the current VC. The problem to overcome is this: Suppose the new and existing paths for a tunnel require resources from common links. However, one or more of these links does not have sufficient capacity to admit the second path. The tunnel must first be torn down and then reestablished on the new path. However, if the links can recognize the second path as a replacement for the existing path, the path can be admitted.

RSVP has a reservation style called shared explicit. This instructs network elements to use the same capacity to service multiple explicitly named sources. In traffic engineering's use of RSVP, a second path for a tunnel is represented as a different source by carrying a path ID as part of the source identification. When a source (the tunnel's headend) wants to reroute, it sends a path message just as it would for a new tunnel. This message names the same tunnel, but with a new path ID. For links not in common, this appears as a new request. For links that are in common, no new resources need to be allocated. The tail end then sends a reserve message for both paths (senders) using the shared explicit style. The two sender objects are included, and separate label operations are associated with each. As soon as the new path is created, updating the forwarding table diverts traffic. This occurs without service disruption. The old path can then be removed. The presence of the second path message on shared links prevents the cleanup process from removing resources used by the new path.

Chapter 2
Literature Survey

This chapter summarizes the interesting points in some papers that examine MPLS traffic engineering capabilities.

Paper-A: Study of Traffic Engineering Capabilities of MPLS Networks [1]

This study examines the Traffic Engineering (TE) capabilities of Multiprotocol Label Switching (MPLS) networks. The study is carried on Pentium PCs running Linux and using open software.
The main strength of MPLS is its TE capabilities. Traffic Engineering refers to performance optimization of operational networks at both the traffic and the resource levels. The use of explicit routes, for example, gives the ability to manage the network resources efficiently and support new services. The basic element of TE over MPLS is LSP tunnel that consists of the traffic belonging to the same class and is routed along the same path. All traffic inside an LSP trunk has the same label and the same 3-bit class of service. In an MPLS domain, LSPs can be established between any two nodes and the same egress node can have multiple parallel trunks. Traffic LSPs attributes associated with path selection, traffic, priority, preemption, resilience and policing provide the ability to describe the characteristics of traffic trunks by the network operator. Trunking results in the separation of competing traffic flows and automatically leads to a traffic engineering ability and better quality of service.

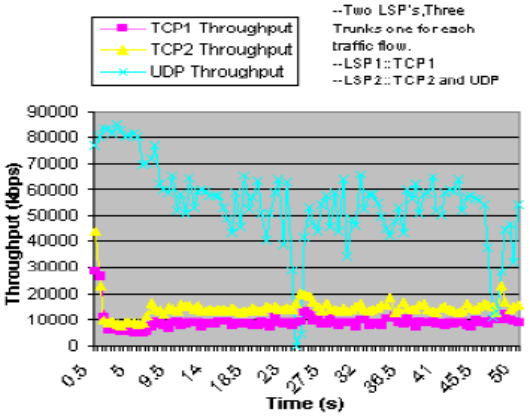


Figure 2.1 : Throughput under experiment – 1

In case-1, the authors maps TCP1 onto LSP1, and TCP2 and UDP onto LSP2. The purpose of this experiment was to analyze how TCP and UDP flows behave if they interfere with each other only for a part of the LSP.

The results indicate that the network utilization increased with the inclusion of the paths other than the shortest path when compared to the best-effort IP network case above. The throughput achieved for both the TCP flows has also increased in comparison with the IP case.

In case-2, the authors mapped both TCP1 and TCP2 to LSP1, and UDP to LSP2. They separated real time traffic from TCP. So, they separated the UDP flow from the TCP flows. The results shows that an increase in the UDP source rate does not affect the TCP sources as it does in the above two cases. The

TCP sources get a fairly constant throughput. So, by isolating the various flows, we can guarantee a given quality of service to sources, which are responsive to congestion control also.
These two scenarios are important to understand the significance of separation of TCP traffic from the real-time traffic to achieve a better performance.

Paper-B: An Optimal MPLS-TE Solution to Route Selection and Redistribution on Congested Networks [2]

This paper reviews both static and dynamic MPLS load balance algorithm. The authors propose an optimal dynamic load balance (ODLB) solution.
IGPs, such as OSPF, use the shortest path or least cost metric to determine the best route. That approach causes that only one route will be considered "best" route in the routing table which leads t congestion o that link and packet loss will occur.

Recently, the following mechanisms appeared as a solution:

1. Static load balance algorithm
    - topology-based static load balance (TSLB)
    - resource-based static load balance (RBLB)
2. Dynamic load balance algorithm

All these algorithms are based on MPLS forwarding scheme. A portion of the traffic is rerouted to other available routes through established explicit routing label switched path (ERLSP).

TSLB sorts available source routes based on the shortest path and push the shortest path to the top of the queue. While, RSLB sorts available source routes based on the bandwidth. The coming traffic takes the shortest hop route (in TSLB) or the best-fit bandwidth route (in RSLB) based on the order they enter the LSR. Traffic will be dropped if there are insufficient resources in the available routes.
TSLB and RSLB use more available resources than the IGP algorithm. The order of the coming traffic is the determinant factor on the route assignment. Sometimes, a small traffic source may be assigned to the route with the biggest bandwidth.

To fix the problem, DLB algorithm was introduced in Long's study. The benefit of the algorithm is that it assigns a better fit traffic to take the bandwidth which was originally taken by a smaller traffic. However, it puts that replaced smaller traffic to search for a fit for itself, which may induce a series of subsequent searching. If the bandwidth of the coming traffic is incremental, the search may affect all the on-going traffic, causing rerouting all the time. Frequent rerouting may lower the QoS and make the network unstable. One more thing to be noticed is that the resulted load balance may not be the optimal solution. In most cases, the solution from DLB is not optimal because the algorithm was not designed in such way.

Optimal dynamic load balance algorithm (ODLB) algorithm intends to solve the problems presented in DLB and it is guaranteed to be the optimal load balance solution.

Results and conclusion

Algorithm	BW ratio	Traffic Rerouted	Traffic dropped
TSLB	0.68	N/A	7Mbps (Src.3)
RSLB	0.77	N/A	5Mbps (Src.4)
DLB	0.82	2	4Mbps (Src.1)
ODLB	**1.00**	**1**	**0Mbps**

ODLB gives the optimal solution to load balancing problems on the parallel routes from the same source to the destination. It exhaustively searches all the routes to guarantees the optimal route with fast speed when the number of the parallel routes and the number of traffic are not large.
The ODLB algorithm combining exhaustive search is a very practical solution to tackle the real world TE problems.
ODLB reduces the frequency of rerouting ongoing traffic because new routes are calculated only when congestion occurs. During congestion, calculation is performed simultaneously with the routing of the ongoing traffic. It may cause some delay of transporting new traffic, but the old traffic remains on the previous calculated routes without interruption.

Paper-C: IETF RFC 4972 - Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router (LSR) Traffic Engineering (TE) Mesh Membership [3]

There are two basic types of TE network design:

The tactical approach, or as needed, is an approach to handle unexpected congestion. This is known as Rather than building a full mesh of TE-LSPs between a set of routers ahead of time, the tactical approach let the IGP to forward traffic as it will, and building TE-LSPs only after congestion is discovered. This allows for keeping most of the network routing on IGP routing only. This might be simpler than a full mesh of TE-LSPs, but it also lets you work around network congestion as it happens. If you have a major network event (a large outage, an unexpectedly popular new web site or service, or some other event that dramatically changes your traffic pattern) that congests some network links while leaving others empty, you can deploy MPLS TE tunnels as you see fit, to remove some of the traffic from the congested links and put it on uncongested paths that the IGP wouldn't have chosen.

Another valid way to deploy MPLS TE is Optimizing network utilization during deploying MPLS TE is sometimes called the strategic method or the full-mesh approach. If we build a full mesh of MPLS TE-LSPs between a given set of routers, size those LSPs according to how much bandwidth is going between a pair of routers, and let the LSPs find the best path in the network that meets their bandwidth demands. Building this full mesh of TE-LSPs in the network helps in avoiding congestion as much as possible by spreading LSPs across your network along bandwidth-aware paths.

A fully meshed LSPs requires {n * (n – 1)} LSPs, which potentially could be a very large number of TE LSPs. That requires the configuration of a large number of TE LSPs which could be an administratively risky and time consuming to the service provider. So, there is a need for automatic mechanism to create fully-meshed TE LSPs. That automatic mechanism can be through routing extensions that can automatically discover the members of a mesh, also referred to as a "TE mesh-group".

But using extensions to the existing IGPs, should be done with care as it may lead to unstable routing.

TE-MESH-GROUP TLV Formats

OSPF TE-MESH-GROUP TLV Format

The TE-MESH-GROUP TLV is used to advertise the desire of an LSR to join/leave a given TE mesh-group. No sub-TLV is currently defined for the TE-MESH-GROUP TLV.

The OSPF TE-MESH-GROUP TLV has the following format:

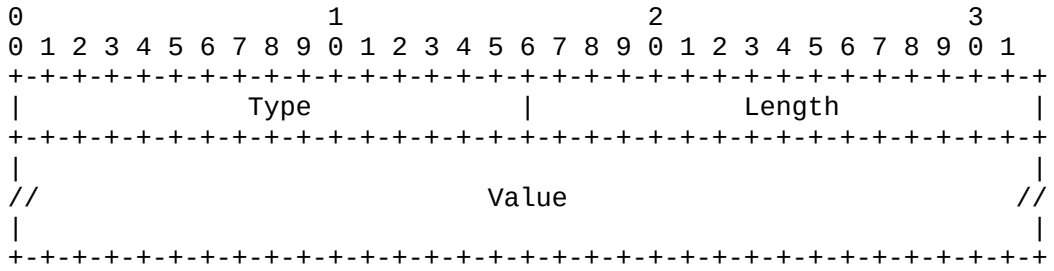


Figure 2.2 : OSPF TE-MESH-GROUP TLV format

Type: identifies the TLV type
Length: the length of the value field in octets

The format of the OSPF TE-MESH-GROUP TLV is the same as the TLV format used by the Traffic Engineering Extensions to OSPF.

Elements of Procedure

The OSPF TE-MESH-GROUP TLV is carried within the OSPF Routing Information LSA. So, a router MUST originate a new LSA/LSP whenever the content of this information changes, or whenever required by regular routing updates.

The TE-MESH-GROUP TLV is advertised within an OSPF Router Information opaque LSA (opaque type of 4, opaque ID of 0) for OSPFv2.
A router MUST originate a new OSPF router information LSA whenever the content of any of the advertised TLV changes or whenever required by the regular OSPF procedure (LSA update (every LSRefreshTime)). An implementation SHOULD be able to detect any change to a previously received TE-MESH-GROUP TLV from a specific LSR.

For OSPFv2 Router Information opaque LSA:

- Link-local scope: type 9.
- Area-local scope: type 10.

- Routing-domain scope: type 11. In this case, the flooding scope is equivalent to the Type 5 LSA flooding scope.

The TE-MESH-GROUP TLV may be advertised within an Area-local or Routing-domain scope Router Information LSA, depending on the MPLS TE mesh group profile:

Backward Compatibility

The TE-MESH-GROUP TLVs defined in this document do not introduce any interoperability issue. For OSPF, a router not supporting the TE-MESH-GROUP TLV SHOULD just silently ignores the TLV.

Paper -D: MPLS Protection Switching Versus OSPF Rerouting : A Simulative Comparison [4]

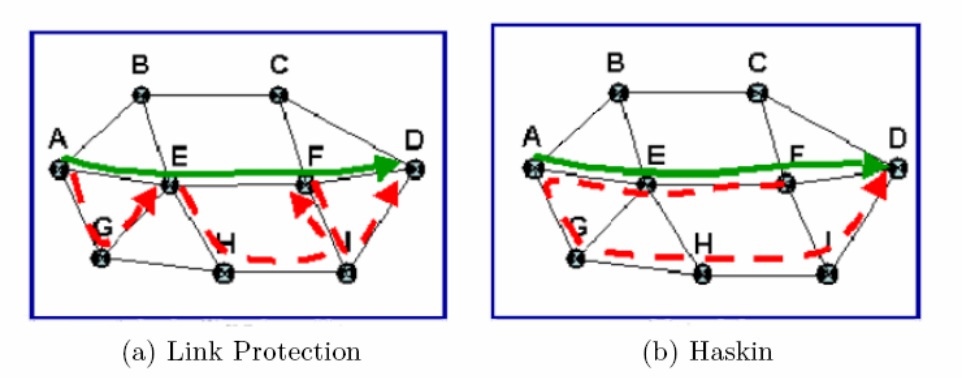
IP routing had been designed to be able to reestablish connectivity after almost any failure of network elements. However, the multimedia and conferencing applications only allow service interruptions on the order of a few hundred milliseconds - a time frame that cannot be reached by today's robust routing protocols. Therefore, network operators deploy an MPLS layer below the IP layer having its own rather fast recovery mechanisms and providing failure-proof virtual links to the IP layer. The most important aspect in the comparison of all these approaches is the resulting recovery speed.

Resilience Mechanisms :

1. Multiprotocol Label Switching (MPLS)

MPLS Recovery.
MPLS Recovery methods provide alternative LSPs to which the traffic can be switched in case of a failure. We must distinguish two types of recovery mechanisms: protection Switching and Restoration.

This paper is focusing on Protection Switching schemes. Link Protection, similar to Cisco's Fast Reroute, and the mechanism introduced by Haskin are considered further. Link Protection provides a shortest backup path for each link of the primary LSP. When a failure occurs on a protected link, the backup path replaces the failed link in the LSP: the upstream router redirects incoming traffic onto the backup path and as soon as traffic arrives on the router downstream of the failed link it will use the primary LSP again. The Haskin scheme uses a global backup path for the LSP from ingress to egress router. When a failure occurs on a protected link the upstream router redirects incoming traffic back to the ingress router, which will be advertised that a failure has occurred. Then these packets are forwarded on the backup path and reach the egress router.



(a) Link Protection (b) Haskin

Figure 2.3 : Link & Node Protection

Routes distribution. There are several possible algorithms to distribute labels through the network such as the Label Distribution Protocol (LDP). Another way is to distribute labels by piggybacking them onto other protocols, in particular the Reservation Protocol (RSVP) and its Traffic Engineering extension (RSVP-TE).

Basic OSPF mechanisms. The Hello protocol is used for the detection of topology changes. Each router periodically emits Hello packets on all its outgoing interfaces. If a router has not received Hello packets from an adjacent router within the "Router Dead Interval", the link between the two routers is considered down. When a topology change is detected, the information is broadcasted to neighbours via Link State Advertisements (LSA). After each computation of routes, the FIB must be reconfigured.

Main time constants. Considering the previous mechanisms, the convergence behaviour of OSPF in case of a failure can be divided into steps as follows : detection of the failure, then flooding of LSAs and - at the same time – scheduling of a SPF calculation, and launching a FIB update.

Proposed extensions to OSPF. Considering the standardized values, the OSPF protocol needs at least a few seconds to converge. To accelerate the convergence time, it is proposed to investigate the following two options: reduce delays, and associate multipath routing with local failure reaction. In the last years, there were several proposals to accelerate OSPF convergence time by reducing the main timers : TspfDelay and TspfHold set to 0, and sub-second THello or hardware failure detection. These accelerated variants of OSPF, the author refered to it as OSPFacc. hello when only sub-second hellos are used, and OSPFacc.hard when hardware detection is enabled in addition. [15] presents a new routing scheme which provides each node in the network with two or more outgoing links towards every destination. Two or more possible next hops are then used at each router towards any destination instead of OSPF's single next hop.

The routing algorithms for calculating the hammocks indicates that if a router detects a link or port failure can react locally, immediately rerouting the affected traffic over the remaining next hops. This local mechanism avoids the time-consuming SPF calculation and flooding of LSAs in the entire area in the case of a single link failure. However, if multiple link failures occur and there is no remaining alternative link at a router, the local reaction will trigger a standard OSPF reaction. This multipath variant of OSPF will be refered to in the following sections as OSPFhammock hello and OSPFhammock hard , depending on which detection mechanism is used.

Measurements and Results

The focus of the investigations was on the speed of the traffic restoration after a failure. As a main sample network, the Pan-European optical network from the COST 239 project was chosen because of its widespread use for network investigations.

After starting the sources, a link failure is simulated triggering failure detection, dynamic route calculation, if necessary, and switching to alternative routes. To get rid of synchronization effects of hello timers with failure times, the simulations are repeated with different periods of time between the simulation start and the failure time. The simulation is also repeated for all possible link failures, to average over the effect of different failure locations. To characterize the effect of the failure, the sum of the rates of all traffic received at sinks in the network is considered over the time.

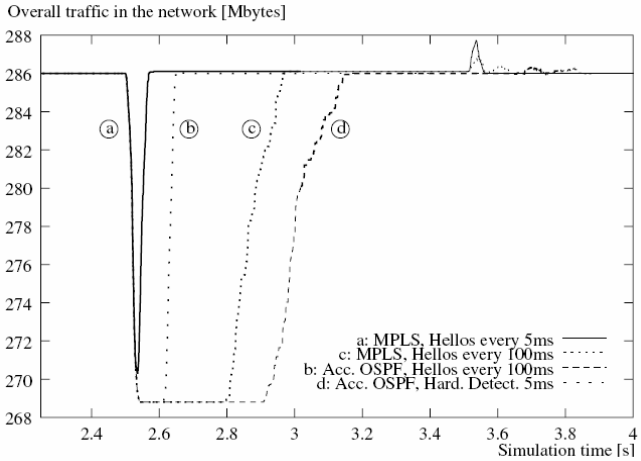


Figure 2.4 : Restoration time

It can be noticed that standard MPLS protection switching, a_, is much faster than both OSPF mechanisms. Even MPLS c_, with the same THello and TDead timers as OSPFacc. hello is still faster, in the order of 100ms. This results from the computational effort, the signaling delay and mostly from the update of the FIBs, which is more time consuming for the larger tables of OSPF – compared to MPLS.

It can be concluded that there are two major points to be addressed in order to improve the restoration speed of OSPF re-routing: speed-up of failure detection (hardware failure detection and fast hello protocols) and acceleration of forwarding information base (FIB) update (the internal router architectures have to be improved).

Chapter 3

Network Modeling and Case Studies

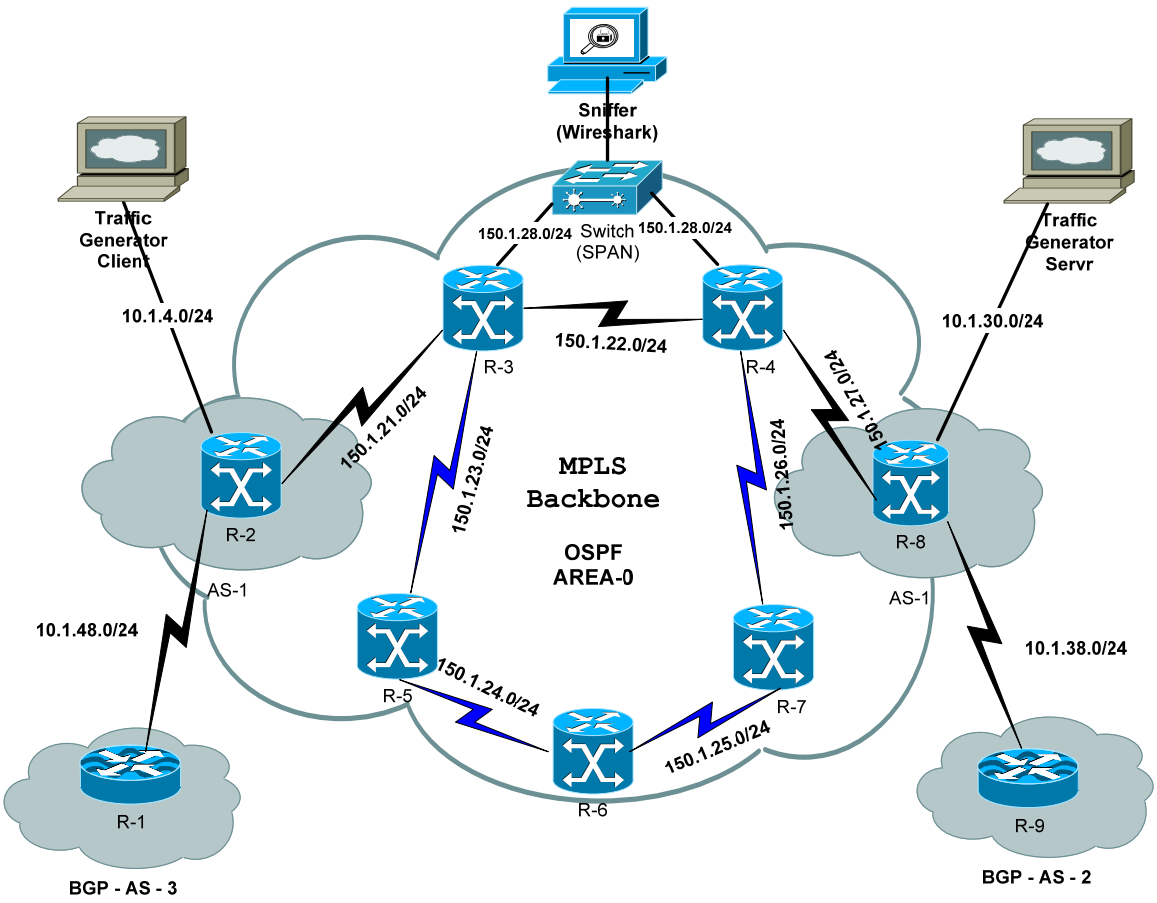


Figure 3.1 : Experimental testbed Network Configuration

The Experimental testbed network consists of the following equipment:

Device	Software	Interfaces
Cisco 2600 Series Routers	Cisco IOS Version 12.2(15)T7	Fast Ethernet & Serial Interfaces
Cisco 2800 Series Routers	Cisco IOS version 12.4	GigabitEthernet, Fast Ethernet and Serial Interfaces
Cisco 3600 Series Routers	Cisco IOS Version 12.2(32)	Fast Ethernet, ATM & Serial Interfaces
Cisco 3750 Catalyst Switch	Native IOS version 12.2 running SPAN	24 Gigabit Ethernet ports

Sun Server	DITG (Traffic Generator) running on Red Hat Linux Enterprise	Fast Ethernet
End host	Wireshark (Packet Analyser) running on Ubuntu Linux	Fast Ethernet

The testbed network consists of 3 ASs, where Router-1 and Router-2 are peeing using EBGP, as well as, Router-8 and Router-9 are peering using EBGP. Whereas, Router-2 and Router-8 are peering using IBGP.

The DITG client generates traffic through the backbone network to the DITG server. The output trace file from each test simulation is used to measure the performances of the network such as the TCP and UDP throughput, packet loss, delay and the total number of packets received by the server.
These test scenarios were chosen to demonstrate the capabilities of MPLS Traffic Engineering and its various path designs versus a normal IP network that runs IGP without MPLS TE.

Experiment Scenario –1
This test presents a comparative analysis of MPLS and non-MPLS networks.
First, the IP network topology runs OSPF and the network was setup with two paths:

A. Path 2-3-4-8.
B. Path 2-3-5-6-7-4-8.

As the network runs OSPF, the default routing would be along the shortest path 2-3-4-8.

Then, MPLS was enabled on all backbone routers and an MPLS LSP tunnel was created through 2-3-4-8 path. After that the DITG client generates TCP traffic with different packet sizes through the backbone network to the DITG server.

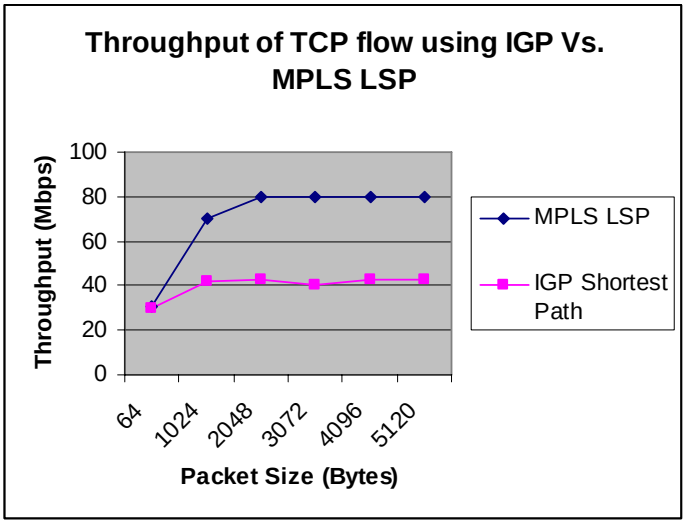


Figure 3.2 : IGP Vs. MPLS throughput

The test has run again with a mix of one TCP and one UDP streams running through the IP network and then through the MPLS-enabled network in the same tunnel 2-3-4-8.

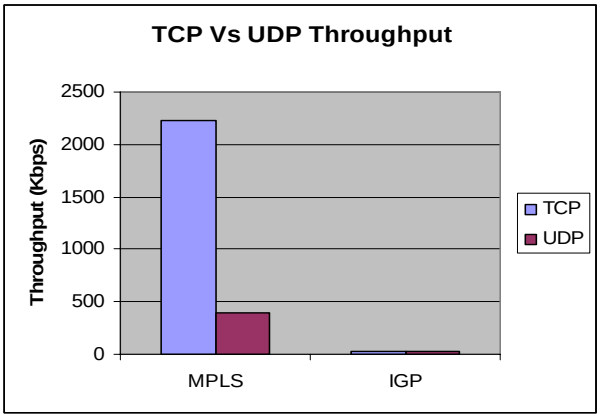


Figure 3.3 : TCP Vs. UDP throughput in MPLS & IGP networks

After analyzing the trace file generated, we can notice the following:

- In the IP network, as TCP has a congestion control mechanism called "slow start" which reduces its traffic in response to packet loss whereas, UDP has no congestion control and doesn't respond to losses. That's why UDP has 34.26 % packet loss while TCP has none. UDP starves the TCP flows.

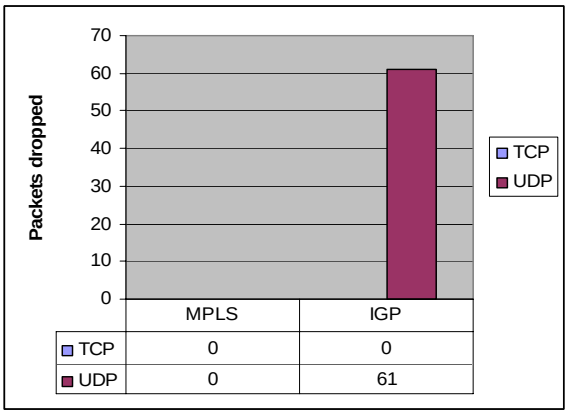


Figure 3.4 TCP Vs. UDP packetsdropped in MPLS & IGP networks

After enabling MPLS TE, we can find that TCP and UDP has 0% packet loss
From the performance chart, there is a clear indication that MPLS TE network has significantly improved the performance and throughput of the network.

Experiment Scenario –2 : TCP competing with UDP in MPSL TE tunnels

In that test, two tunnels were created as follows:

A. tunnel-1 through (2-3-4-8)
B. tunnel-0 through (2-3-5-6-7-4-8)

TCP and UDP flow in tunnel-1 and another TCP flows through tunnel-0.

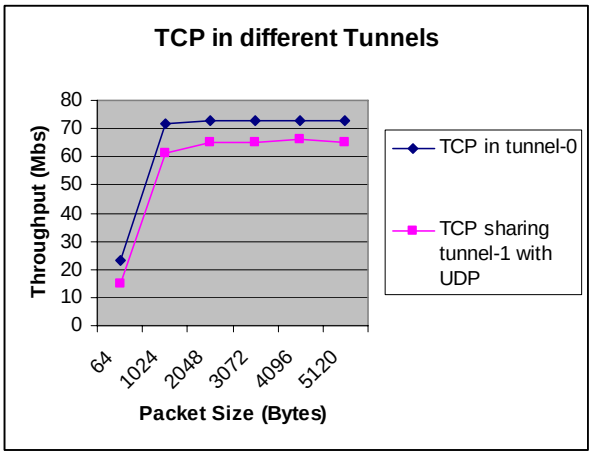


Figure 3.5 : TCP in different LSPs

From the graph, it should be noted that although the explicit tunnel-0 has three additional hops than tunnel-1, it still provides much better performance over the traditional shortest-path IGP route, which is competed by another flow.
In addition to allowing us to get an improved performance for the traffic flow between the client and the server, forcing the traffic into an alternate path, frees bandwidth for another flow on the congested link.

Experiment Scenario –3 : MPLS with Qos

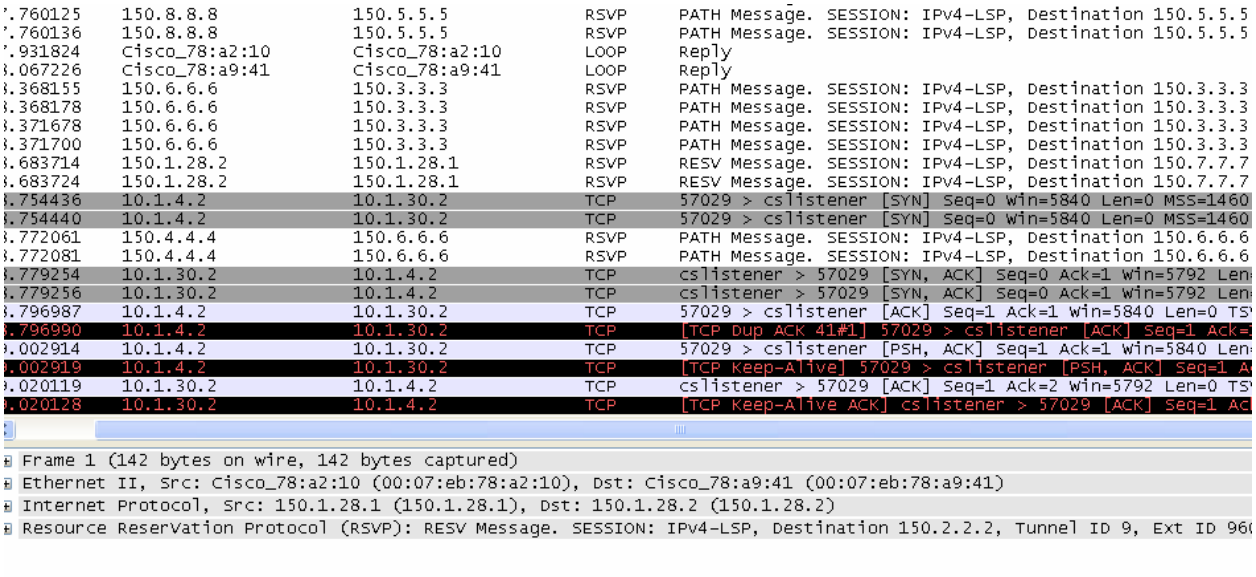


Figure 3.6 : Traffic capture using Wireshark

In this test, each traffic flow is mapped onto a separate MPLS Label Switched Path (LSP) that extends through the MPLS backbone network. In addition, each LSP is characterized with a certain reserved bandwidth across the MPLS network, as well as with different CoS values. This allows us to provide guaranteed bandwidth and different levels of service for the two flows.

Router-2 and Router-8 are configured so that all VoIP traffic received from the clients is marked with a qos-group of 5. By asuming that VoIP traffic uses UDP destination ports in the range of 16384-32767 and has a DSCP value of EF. All other traffic received from SW1 and SW2 should be marked with a qosgroup of 1. Traffic from qos-group 5 on Router-2 and Router-8 should be mapped to MPLS EXP 5 throughout the provider network. Voice traffic should be guaranteed a maximum of 640Kbps of priority as it transits the backbone network. All the backbone links is configured with bandwidth of DS-3 (45Mbps).

Traffic from qos-group 1 on Router-2 and Router-8 should be mapped to MPLS EXP 1 throughout the provider network. This traffic should be guaranteed a minimum bandwidth of 1Mbps.
The CoS value is an important parameter that affects both throughput and latency performance of the two flows. So, the traffic flow with higher priority class receives better treatment than that of with low priority class.

```
R2#sh policy-map int fastethernet0/0 out
Fastethernet0/0
Service-policy output: TO_CE
Class-map: QOS_GROUP_5 (match-all)
10 packets, 580 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: qos-group 5
Queueing
Strict Priority
Output Queue: Conversation 264
Bandwidth 640 (kbps) Burst 16000 (Bytes)
(pkts matched/bytes matched) 0/0
(total drops/bytes drops) 0/0
Class-map: QOS_GROUP_1 (match-all)
20 packets, 2280 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: qos-group 1
Queueing
Output Queue: Conversation 265
Bandwidth 1000 (kbps)
(pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0
exponential weight: 9
mean queue depth: 0
class  Transmitted  Random drop  Tail drop  Minimum  Maximum  Mark
       pkts/bytes   pkts/bytes   pkts/bytes thresh   thresh   prob
0      20/2280      0/0          0/0        20       40       1/10
1      0/0          0/0          0/0        22       40       1/10
2      0/0          0/0          0/0        24       40       1/10
3      0/0          0/0          0/0        26       40       1/10
4      0/0          0/0          0/0        28       40       1/10
5      0/0          0/0          0/0        30       40       1/10
6      0/0          0/0          0/0        32       40       1/10
7      0/0          0/0          0/0        34       40       1/10
rsvp   0/0          0/0          0/0        36       40       1/10

Class-map: class-default (match-any)
7 packets, 716 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

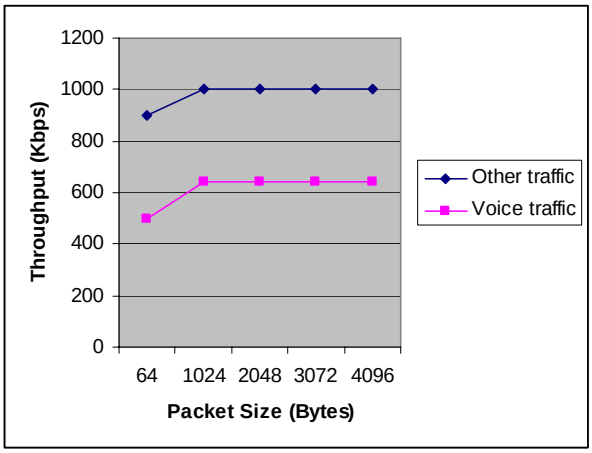


Figure 3.7 : Traffic prioritization based on CoS.

The results make it clear that service differentiation using MPLS CoS value has a significant effect on the performance of applications. And the performance effect of CoS is even more significant, especially when the network is congested. Whatever the case, the flow with higher priority class always receives better treatment than the flow with lower priority.

This shows that LSRs in an MPLS network can effectively prioritize packets based on their classes, and give the appropriate treatments to time critical traffic such as VoIP and video streaming, which are extremely latency dependent.

All the results presented here, demonstrate the effectiveness of MPLS traffic engineering and QoS in IP networks in order to achieve highest performance.

Experiment Scenario – 4 : Load-Balancing in OSPF Vs MPLS TE

When a router learns multiple routes to a specific network via multiple routing protocols, it installs the route with the lowest administrative distance in the routing table. Sometimes the router must select a route from among many learned via the same routing process with the same administrative distance. In this case, the router chooses the path with the lowest cost (or metric) to the destination which may results in under and/or over utilized links. Each routing process calculates its cost differently and the costs may need to be manipulated in order to achieve load-balancing.

If the router receives and installs multiple paths with the same administrative distance and cost to a destination, load-balancing can occur. The number of paths used is limited by the number of entries the routing protocol puts in the routing table. Four entries is the default in IOS for most IP routing protocols with the exception of Border Gateway Protocol (BGP), where one entry is the default. Six different paths configured is the maximum number.

Load Balancing with OSPF

The Cisco implementation of OSPF can support up to four equal-cost routes to a destination. If one route fails, OSPF uses the remaining paths as alternates. OSPF load balancing allows equal cost by default paths.

The cost associated is determined by the interface bandwidth statement unless otherwise configured to maximize multiple-path routing.

Unequal-Cost Load Balancing via Metric Manipulation

Unequal-cost load balancing is a concept that allows routers to take advantage of load sharing over multiple unequal-cost paths to a given destination. This can be achieved by manipulating the parameters that determine the routing metrics for protocols such as OSPF, IS-IS, and EIGRP.

OSPF Unequal-Cost Load Balancing

In order to enable OSPF unequal-cost load balancing, you use the bandwidth command on the interface. This command might not represent the actual speed of the link, so it can be used to manipulate how data is load-shared over different links with varying speeds. For OSPF to load-share across links with varying speeds, the bandwidth command can be used to set the same value (in bps) across these links. The physical throughput, however, is unchanged, and the command is used only to represent or manipulate the link speed.

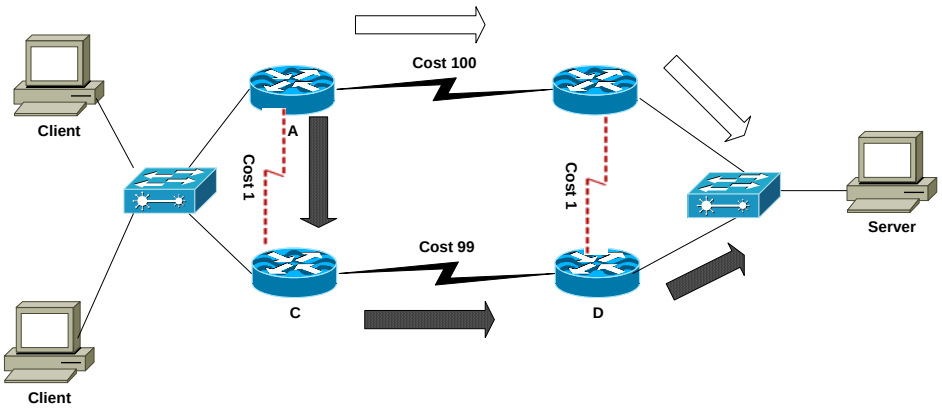


Figure 3.8 : Manipulating OSPF costs

In that Scenario, If all the clients having Router A as their default gateway, and we want to load balance the traffic on Router A between A-C and A-B links.

When using OSPF as the routing protocol, I specified manual costs of the WAN links with the ip ospf cost interface configuration command. With the reduced cost of the C-D link, Router A would find two equal-cost paths to the destination network, performing load-balancing between them.

Load-balancing on Router A

```
A#show ip route 150.1.0.0

Routing entry for 150.1.0.0 255.255.255.0

  Known via "ospf 0", distance 110, metric 101, type intra area

  Last update from 150.2.0.2 on FastEthernet0/0, 00:00:03 ago

  Routing Descriptor Blocks:

  * 192.168.1.2, from 192.168.0.3, 00:00:03 ago, via Serial0/0

      Route metric is 101, traffic share count is 1

   150.2.0.2, from 192.168.0.3, 00:00:03 ago, via FastEthernet0/0

      Route metric is 101, traffic share count is 1
```

It's obvious that the routing tricks can solve load-balancing problems only in a very tightly controlled environment (the example solution was used on a firewall-to-firewall connection between two corporations with the routing protocol not being extended beyond the four routers described in this scenario). Additional routers participating in the same routing protocol or more complex paths between the LANs make this solution extremely complex and thus highly unusable.

When changing the path cost using ip ospf cost command, you must be careful that the cost value set conforms to the lowest-speed link. If the value is set according to the highest-speed link, traffic flow will overwhelm the slow links.

Case-2 : Load-Balancing with MPLS Traffic Engineering

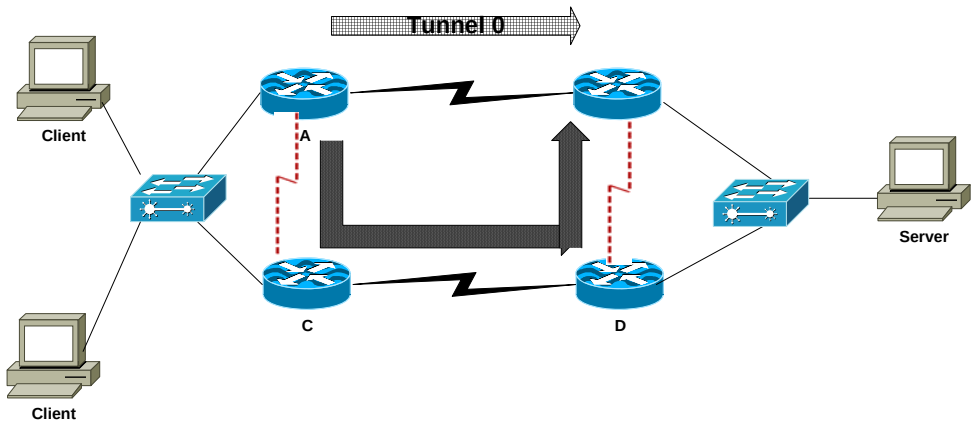


Figure 3.9 : Load balancing using MPLS LSPs

One of the design goals of the Traffic Engineering module of the Multi-Protocol Label Switching (MPLS-TE) was to enable optimum network utilization beyond the equal-cost load-balancing permitted by the IP routing. The MPLS TE is thus the ideal candidate technology in optimal load-balancing designs. A TE tunnel between A1 and B1 going through A2 and B2 would give us a second direct path between A1 and B1. As the routing protocol cost of a TE tunnel is by default equal to the IP routing cost between the tunnel endpoints, the A1-A2-B2-B1 tunnel would be the second equal-cost path between A1 and B1 (Figure 5).

However, Cisco IOS does not permit load-balancing between an MPLS TE interface and a regular IP interface. It's thus necessary to establish a second MPLS TE tunnel directly between A1 and B1 (Figure 6). With the two tunnels in place, A1 performs equal-cost load-balancing between the two tunnels (Listing 3).

Load-balancing over MPLS TE tunnels

```
A#show ip route

… output ommited …

O    150.1.0.0 255.255.255.0 [110/101] via 0.0.0.0, 00:00:31, Tunnel1
                             [110/101] via 0.0.0.0, 00:00:31, Tunnel0
```

```
A#show ip cef 150.1.0.0

150.1.0.0 /24, version 12, epoch 0, per-packet sharing

0 packets, 0 bytes

  tag information set

    local tag: tunnel-head

  via 0.0.0.0, Tunnel1, 0 dependencies

    traffic share 1, current path

    next hop 0.0.0.0, Tunnel1

    valid adjacency

    tag rewrite with Tu1, point2point, tags imposed: {}

  via 0.0.0.0, Tunnel0, 0 dependencies

    traffic share 1

    next hop 0.0.0.0, Tunnel0
```

```
  valid adjacency

  tag rewrite with Tu0, point2point, tags imposed: {12}

0 packets, 0 bytes switched through the prefix

tmstats: external 0 packets, 0 bytes

        internal 0 packets, 0 bytes
```

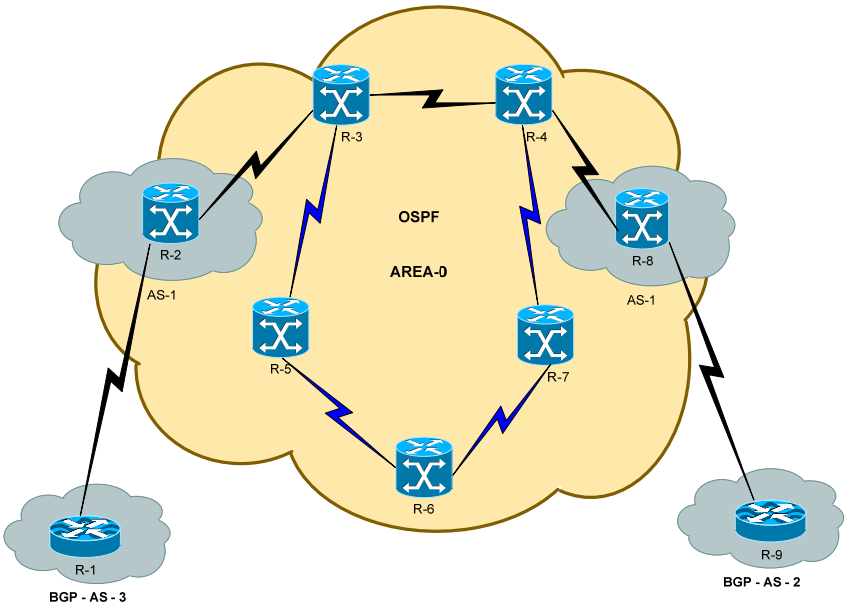
Metric Manipulation Versus MPLS Traffic Engineering

The only mechanism for redirecting traffic in IP networks is to change the link metrics presented to a link-state IGP such as OSPF. However, changing a link's metric can change the path of all packets traversing the link. Also, these methods do not provide dynamic redundancy and do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions.

In an MPLS traffic-engineered network, any Label-Switched Path (LSP) can be dynamically shifted from a congested path to an alternative path. This represents an efficiency improvement over the traditional operational methods for IP networks, because the network designers can run their networks at much higher capacity under normal circumstances, secure in the knowledge that before congestion occurs, some of the traffic can easily be shifted away from the congestion point. Furthermore, network designers can make use of global optimization algorithms that provide a mapping from the traffic demand to the physical links that could not otherwise be achieved using only local optimization.

Traffic engineering can also perform Cisco Express Forwarding (CEF)-based unequal-cost load balancing across tunnels. This combination of manual automatic tuning helps realize the goals of capacity planning and helps optimize network utilization on backbone trunks.

Experiment Scenario –5 : MPLS TE Tactical design study



In the tactical model, we usually create TE LSPs for the following reasons:

1. A link failure somewhere else in the network (probably either Router-3 t Router-4) pushes more traffic than we have planned for onto the 3-5-6-7-4 links. The link failure can be short (a line card crashed), or it can be long (a fiber cut that will take days to fix).
2. Something on the other side of that link becomes a major traffic draw. Perhaps a customer who has just turned up a major streaming media service, and they're sending a lot of traffic from Router-3 to Router-4. Maybe there's major breaking news. Maybe a big Denial of Service attack is headed toward Router-3 or Router-4.

Suppose a large amount of traffic is going from Router-3 t Router-4, as shown in Figure 9-5. It is routed across the Router-3 t Router-4 link even if the traffic being sent down the link exceeds the link capacity. Then the link will start queuing and dropping packets. So if we have a significant traffic disruption for a long time, we should consider using TE-LSPs to temporarily clear up the problem.

```
2800-2#sh mpls traffic-eng tunnels br
Signalling Summary:
    LSP Tunnels Process:            running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 89 seconds
    Periodic auto-bw collection:    disabled
TUNNEL NAME                 DESTINATION     UP IF     DOWN IF   STATE/PROT
Tunnel0                       150.8.8.8       -         Se0/0/0     up/up
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 1 (of 1) tails
```

We can create temporarily TE-LSP tunnel through 3-5-6-7-4 links and push the traffic down that link.

So, we built a new TE LSP to work around the problem.

In fact, two cases when you should consider removing TE LSPs:

- When they're no longer needed as the problem they're solving doesn't exist anymore.

A. You need to constantly monitor the network to see which new TE LSPs have been installed and whether existing TE LSPs are still useful.
B. We should know the complete purpose of the TE-LSP.

- When the TE-LSPs are causing problems— A traffic spike somewhere else in the network collides with traffic in a TE LSP. If we discover a congested link, the first thing we need to do is check to see if any TE LSPs are crossing that link, and if so, how much bandwidth they're consuming.

```
3600-2#sh mpls traffic-eng tunne de 150.8.8.8

LSP Tunnel low-1_t0 is signalled, connection is up
  InLabel  : Serial0/2, 30
  OutLabel : Serial0/0, implicit-null
  RSVP Signalling Info:
       Src 150.7.7.7, Dst 150.8.8.8, Tun_Id 0, Tun_Instance 62
    RSVP Path Info:
      My Address: 150.1.22.1
      Explicit Route: 150.1.21.1 150.8.8.8
      Record   Route:  NONE
      Tspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
    RSVP Resv Info:
      Record   Route:  NONE
      Fspec: ave rate=100 kbits, burst=1000 bytes, peak rate=100 kbits
```

We should start applying tactical TE-LSPs to failures that lasts long times. Being too responsive to short-term problems can create administrative burden.

Experiment Scenario –6 : Strategic TE Design

```
top-1# sh mpls traffic-eng link-management sum
System Information::
    Links Count:          2
    Flooding System:      enabled
IGP Area ID::  ospf area 0
    Flooding Protocol:    OSPF
    Flooding Status:      data flooded
    Periodic Flooding:    enabled (every 180 seconds)
    Flooded Links:        2
    IGP System ID:        150.4.4.4
    MPLS TE Router ID:    150.4.4.4
    IGP Neighbors:        2
```

```
Link ID:: Fa0/0 (150.1.24.2)
   Link Status:
  Physical Bandwidth:   155000 kbits/sec
     Max Res Global BW:    116250 kbits/sec (reserved: 0% in, 43% out)
     Max Res Sub BW:       0 kbits/sec (reserved: 100% in, 100% out)
     MPLS TE Link State:   MPLS TE on, RSVP on, admin-up, flooded
     Inbound Admission:    reject-huge
     Outbound Admission:   allow-if-room
     Admin. Weight:        1 (IGP)
     IGP Neighbor Count:   1
Link ID:: Se0/0 (150.1.25.1)
   Link Status:
     Physical Bandwidth:   1544 kbits/sec
     Max Res Global BW:    256 kbits/sec (reserved: 0% in, 0% out)
     Max Res Sub BW:       0 kbits/sec (reserved: 100% in, 100% out)
     MPLS TE Link State:   MPLS TE on, RSVP on, admin-up, flooded
     Inbound Admission:    allow-all
     Outbound Admission:   allow-if-room
     Admin. Weight:        64 (IGP)
     IGP Neighbor Count:   1
```

In this model, we can decide where the boundaries of the backbone TE cloud are, and we build a full mesh of TE LSPs. These TE-LSPs reserve bandwidth commensurate with what they're actually carrying, rather than reserving only enough bandwidth to have the proper forwarding ratios. The actual bandwidth value is reserved because it makes things simple. As traffic demands between routers change, the traffic demands can be measured and the tunnels changed accordingly. Periodically, these TE-LSPs are resized to account for the amount of traffic they're actually carrying.

The strategic model has some advantages over the tactical model. Because we establish LSPs between every node at the edge of the cloud, we won't be constantly examining across LSPs that we didn't expect to find. Also, full-mesh models tend to make more optimal use of the bandwidth than tactical ones, which can save you more money.

```
top-1#sh mpls traffic-engineering tunnels brief
Signalling Summary:
    LSP Tunnels Process:            running
    RSVP Process:                   running
    Forwarding:                     enabled
    Periodic reoptimization:        every 3600 seconds, next in 1818 seconds
    Periodic auto-bw collection:    disabled
TUNNEL NAME                 DESTINATION     UP IF     DOWN IF   STATE/PROT
top-1_t2                    150.8.8.8       -         Fa0/0     up/up
top-1_t8                    150.2.2.2       -         Fa0/0     up/up
top-1_t12                   150.6.6.6       -         Fa0/0     up/up
top-1_t15                   150.3.3.3       -         Fa0/0     up/up
top-1_t18                   150.5.5.5       -         Se0/0     up/up
top-1_t19                   150.7.7.7       -         Fa0/0     up/up
3600-2_t8                   150.4.4.4       Fa0/0     -         up/up
top-2_t15                   150.4.4.4       Fa0/0     -         up/up
top-2_t16                   150.5.5.5       Fa0/0     Se0/0     up/up
mid-1_t16                   150.3.3.3       Se0/0     Fa0/0     up/up
mid-1_t20                   150.4.4.4       Se0/0     -         up/up
3600-1_t12                  150.4.4.4       Fa0/0     -         up/up
```

```
low-1_t19                      150.4.4.4       Fa0/0       -          up/up
2800-2_t2                      150.4.4.4       Fa0/0       -          up/up
Displayed 6 (of 6) heads, 2 (of 2) midpoints, 6 (of 6) tails
```

After creating strategic full-mesh design, we can have large number of tunnels.

We can derive the differences between the tactical and the full-mesh design as follows:

Tactical	Strategic
Reserves bandwidth as necessary to affect unequal-cost load balancing	Reserves bandwidth that matches the actual traffic sent
Small number of tunnels	Larger number of tunnels
Difficult to track what tunnels are where and why they're there	Easier to track, because you know how many tunnels you have and where they go

Conclusion

Through that project and after analyzing the TCP and UDP throughput through the IP/MPLS backbone network, I can conclude the results in the following points:

- The MPLS TE provides better resource utilization and throughput than original IGP networks.
- MPLS has faster recovery and restoration time which is essential for any carrier backbone network.
- Providing QoS and traffic engineering capabilities in the Internet is very essential.
- MPLS will play a key role in future service providers and carriers IP backbone networks.
- The use of MPLS in IP backbone networks will facilitate the development of new services such as real-time applications in the Internet.

Appendices

Bibliography

[1]Study of traffic engineering capabilities of MPLS networks, Alam, M.; Bethoju, P.; Song, M. Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on Volume 2, Issue, 4-6 April 2005 Page(s): 14 - 15 Vol. 2.

[2] An Optimal MPLS-TE Solution to Route Selection and Redistribution on Congested Networks Li, Chengcheng; Li, Peng; Mohammed, Tijjani, Networking, Architecture, and Storage, 2007. NAS 2007. International Conference on Volume , Issue , 29-31 July 2007 Page(s):69 – 76.

[3] RFC 4972 : Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router (LSR) Traffic Engineering (TE) Mesh Membership, P. Vasseur, Ed., L. Leroux, Ed., S. Yasukawa, S. Previdi, P. Psenak, P. Mabbey, July 2007.

[4] MPLS Protection Switching Versus OSPF Rerouting : A Simulative Comparison, Sandrine Pasqualini[1] , Andreas Iselt, Andreas Kirstädter and Antoine Frot, Springer Berlin / Heidelberg , Volume 3266/2004

[5] D. Haskin and R. Krishnan, "A method for setting an alternative label switched paths to handle fast reroute," IETF," Internet Draft.

[6]Traffic engineering and QoS optimization of integrated voice & data networks, Ash, Gerald R., Elsevier/Morgan Kaufmann Publishers, c2007.

[7]Designing multiprotocol label switching networks, Lawrence, J., Communications Magazine, IEEE, Jul 2001, Volume: 39, Issue: 7

[8]Traffic engineering with MPLS in the Internet, Xipeng Xiao; Hannan, A.; Bailey, B.; Ni, L.M. Network, IEEE, Volume 14, Issue 2, Mar/Apr 2000 Page(s):28 – 33

[9]HaoWang, Haiyong Xie, Lili Qiu, Yang Richard Yang, Yin Zhang, and Albert Greenberg. Cope: traffic engineering in dynamic networks. In SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pages 99.110, New York, NY, USA, 2006. ACM Press.

[10]MPLS Configuration on Cisco IOS Softwareby Lancy Lobo, - CCIE No. 4690; Umesh Lakshman, October 17, 2005

[11]MPLS Fundamentals by Luc De Ghein - CCIE No. 1897, November 21, 2006

[12]Definitive MPLS Network Designsby Jim Guichard; François Le Faucheur; Jean-Philippe Vasseur, March 14, 2005

[13] Cisco Systems ; www.cico.com

[14] S. Halabi and D. McPherson. Internet Routing Archtectures. Cisco Press, 2001.

[15]G. Schollmeier et al., "Improving the resilience in IP networks," in *HPSR 2003*, jun 2003.

[16] MPLS and Traffic Engineering in IP Networks, Daniel O. Awduche, UUNET (MCI Worldcom), December 1999.

[17] Connection-Oriented Networks, Harry G. Perros, 31 Oct 2005.

APPENDIX II
Routers Configuration

```
R-1#sh run
Building configuration...

Current configuration : 735 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname mid-2
!
!
ip subnet-zero
!
call rsvp-sync
!
!
interface Loopback0
 ip address 10.1.7.7 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.1.48.2 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0
 no ip address
 shutdown
 no fair-queue
!
interface Serial0/1
 no ip address
 shutdown
!
router bgp 3
 bgp log-neighbor-changes
 network 10.1.7.0 mask 255.255.255.0
 neighbor 10.1.48.1 remote-as 1
!
ip classless
ip http server
!
!
voice-port 1/0/0
!
voice-port 1/0/1
!
```

```
R-2#sh run
Building configuration...

Current configuration : 3180 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname 2800-2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
!
resource policy
!
memory-size iomem 10
ip subnet-zero
!
!
ip cef
! class-map match-all VoIP
match access-group name VoIP
class-map match-all QOS_GROUP_1
match qos-group 1
class-map match-all QOS_GROUP_5
match qos-group 5
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map FROM_P
class MPLS_EXP_5
set qos-group 5
class MPLS_EXP_1
set qos-group 1
policy-map TO_CE
class QOS_GROUP_5
priority 640
class QOS_GROUP_1
bandwidth 1000
policy-map FROM_CE
class VoIP
set qos-group 5
set mpls experimental imposition 5
```

```
class class-default
set qos-group 1
set mpls experimental imposition 1

!
!
mpls traffic-eng tunnels
!
voice-card 0
 no dspfarm
!
!
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 150.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
 no routing dynamic
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 150.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
 no routing dynamic
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 150.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
 no routing dynamic
!
interface Tunnel3
 ip unnumbered Loopback0
 tunnel destination 150.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
 no routing dynamic
!
interface Tunnel4
 ip unnumbered Loopback0
 tunnel destination 150.6.6.6
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
 no routing dynamic
!
interface Tunnel5
 ip unnumbered Loopback0
 tunnel destination 150.7.7.7
 tunnel mode mpls traffic-eng
```

```
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
 no routing dynamic
!
interface Loopback0
 ip address 150.8.8.8 255.255.255.0
!
interface GigabitEthernet0/0
 ip address 10.1.48.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/0.2
 encapsulation dot1Q 2
 ip address 10.10.2.1 255.255.255.0
 no snmp trap link-status
!
interface GigabitEthernet0/0.3
 encapsulation dot1Q 3
 ip address 10.10.3.1 255.255.255.0
 no snmp trap link-status
!
interface GigabitEthernet0/1
 ip address 10.1.4.1 255.255.255.0
service-policy input FROM_CE
service-policy output TO_CE
 duplex auto
 speed auto
!
interface FastEthernet0/1/0
!
interface FastEthernet0/1/1
!
interface FastEthernet0/1/2
!
interface FastEthernet0/1/3
!
interface Serial0/0/0
 ip address 150.1.21.1 255.255.255.0
service-policy input FROM_P
service-policy output TO_P
 mpls traffic-eng tunnels
 ip rsvp bandwidth 256 256
!
interface Vlan1
 no ip address
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 10.1.4.0 0.0.0.255 area 0
 network 150.1.21.0 0.0.0.255 area 0
 network 150.8.8.8 0.0.0.0 area 0
!
router bgp 1
 no synchronization
```

```
 bgp log-neighbor-changes
 network 150.8.8.0 mask 255.255.255.0
 neighbor 10.1.48.2 remote-as 3
 neighbor 150.7.7.7 remote-as 1
 neighbor 150.7.7.7 update-source Loopback0
 neighbor 150.7.7.7 next-hop-self
 no auto-summary
!
ip classless
ip route 10.10.2.0 255.255.255.0 GigabitEthernet0/0.2
ip route 10.10.3.0 255.255.255.0 GigabitEthernet0/0.3
!
ip access-list extended VoIP
permit udp any any range 16384 32767 dscp ef
!
ip http server
no ip http secure-server
!
!
control-plane
!
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
scheduler allocate 20000 1000
!
end
```

```
R-3#sh run
Building configuration...

Current configuration : 2662 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 3600-2
!
!
ip subnet-zero
!
!
!
ip cef
mpls traffic-eng tunnels
call rsvp-sync
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
```

```
!
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map TO_PE
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
!
interface Loopback0
 ip address 150.2.2.2 255.255.255.0
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 150.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel6
 ip unnumbered Loopback0
 tunnel destination 150.6.6.6
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
!
interface Tunnel7
 ip unnumbered Loopback0
 tunnel destination 150.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
!
interface Tunnel8
 ip unnumbered Loopback0
 tunnel destination 150.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
!
interface Tunnel9
 ip unnumbered Loopback0
 tunnel destination 150.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
!
interface Tunnel10
 ip unnumbered Loopback0
 tunnel destination 150.7.7.7
 tunnel mode mpls traffic-eng
```

```
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
!
interface Serial0/0
 ip address 150.1.21.2 255.255.255.0
 mpls traffic-eng tunnels
 clock rate 64000
 ip rsvp bandwidth 256 256
!
interface Serial0/1
 ip address 150.1.23.1 255.255.255.0
 mpls traffic-eng tunnels
 fair-queue 64 32 1000
 ip rsvp bandwidth 256 256
!
interface Serial0/2
 no ip address
 shutdown
!
interface Serial0/3
 no ip address
 shutdown
!
interface Serial0/4
 no ip address
 shutdown
!
interface Serial0/5
 no ip address
 shutdown
!
interface Serial0/6
 no ip address
 shutdown
!
interface Serial0/7
 no ip address
 shutdown
!
interface Ethernet1/0
 ip address 150.1.28.1 255.255.255.0
 half-duplex
 mpls traffic-eng tunnels
 ip rsvp bandwidth 256 256
!
interface ATM2/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface Ethernet3/0
 no ip address
 shutdown
 half-duplex
!
router ospf 1
 log-adjacency-changes
```

```
 network 150.1.21.0 0.0.0.255 area 0
 network 150.1.22.0 0.0.0.255 area 0
 network 150.1.23.0 0.0.0.255 area 0
 network 150.1.28.0 0.0.0.255 area 0
 network 150.2.2.2 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip classless
ip http server
!
!
!
dial-peer cor custom
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
!
end
```

```
R-4#sh run
Building configuration...

Current configuration : 2715 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 3600-1
!
!
ip subnet-zero
!
!
!
ip cef
mpls traffic-eng tunnels
call rsvp-sync
!
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map TO_PE
```

```
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect

!
interface Loopback0
 ip address 150.6.6.6 255.255.255.0
!
interface Tunnel4
 ip unnumbered Loopback0
 tunnel destination 150.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel6
 ip unnumbered Loopback0
 tunnel destination 150.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
!
interface Tunnel11
 ip unnumbered Loopback0
 tunnel destination 150.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
!
interface Tunnel12
 ip unnumbered Loopback0
 tunnel destination 150.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
!
interface Tunnel13
 ip unnumbered Loopback0
 tunnel destination 150.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
!
interface Tunnel14
 ip unnumbered Loopback0
 tunnel destination 150.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
!
interface Ethernet0/0
 ip address 150.1.28.2 255.255.255.0
 half-duplex
 mpls traffic-eng tunnels
 ip rsvp bandwidth 256 256
```

```
!
interface Serial1/0
 ip address 150.1.26.1 255.255.255.0
 mpls traffic-eng tunnels
 no fair-queue
 ip rsvp bandwidth 256 256
!
interface Serial1/1
 ip address 150.1.27.1 255.255.255.0
 mpls traffic-eng tunnels
 fair-queue 64 32 37
 clock rate 64000
 ip rsvp bandwidth 256 256
!
interface Serial1/2
 no ip address
 shutdown
!
interface Serial1/3
 no ip address
 shutdown
!
interface Serial1/4
 no ip address
 shutdown
!
interface Serial1/5
 no ip address
 shutdown
!
interface Serial1/6
 no ip address
 shutdown
!
interface Serial1/7
 no ip address
 shutdown
!
interface ATM2/0
 no ip address
 shutdown
 no atm ilmi-keepalive
!
interface Ethernet3/0
 no ip address
 shutdown
 half-duplex
!
router ospf 1
 log-adjacency-changes
 network 150.1.22.0 0.0.0.255 area 0
 network 150.1.26.0 0.0.0.255 area 0
 network 150.1.27.0 0.0.0.255 area 0
 network 150.1.28.0 0.0.0.255 area 0
 network 150.1.29.0 0.0.0.255 area 0
 network 150.6.6.6 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
```

```
 mpls traffic-eng area 0
!
ip classless
ip http server
!
!
!
dial-peer cor custom
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
!
end
```

```
R-5#sh run
Building configuration...

Current configuration : 2284 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname top-2
!
!
memory-size iomem 10
ip subnet-zero
!
!
!
ip cef
mpls traffic-eng tunnels
call rsvp-sync
!
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map TO_PE
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
```

```
!
interface Loopback0
 ip address 150.3.3.3 255.255.255.0
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 150.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel7
 ip unnumbered Loopback0
 tunnel destination 150.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
!
interface Tunnel11
 ip unnumbered Loopback0
 tunnel destination 150.6.6.6
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
!
interface Tunnel15
 ip unnumbered Loopback0
 tunnel destination 150.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
!
interface Tunnel16
 ip unnumbered Loopback0
 tunnel destination 150.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
!
interface Tunnel17
 ip unnumbered Loopback0
 tunnel destination 150.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
!
interface FastEthernet0/0
 ip address 150.1.24.1 255.255.255.0
 duplex auto
 speed auto
 mpls traffic-eng tunnels
 ip rsvp bandwidth 256 256
!
interface Serial0/0
 ip address 150.1.23.2 255.255.255.0
 mpls traffic-eng tunnels
 no fair-queue
```

```
 clock rate 64000
 ip rsvp bandwidth 256 256
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 log-adjacency-changes
 network 150.1.23.0 0.0.0.255 area 0
 network 150.1.24.0 0.0.0.255 area 0
 network 150.3.3.3 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
ip classless
ip http server
!
!
voice-port 1/0/0
!
voice-port 1/0/1
!
dial-peer cor custom
!
!
gatekeeper
 shutdown
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
!
end
```

```
R-6#sh run
Building configuration...

Current configuration : 2466 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname top-1
!
logging queue-limit 100
```

```
!
ip subnet-zero
!
!
ip cef
mpls ldp logging neighbor-changes
mpls traffic-eng tunnels
!
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map TO_PE
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
interface Loopback0
 ip address 150.4.4.4 255.255.255.0
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 150.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel8
 ip unnumbered Loopback0
 tunnel destination 150.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
!
interface Tunnel12
 ip unnumbered Loopback0
 tunnel destination 150.6.6.6
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
!
```

```
interface Tunnel15
 ip unnumbered Loopback0
 tunnel destination 150.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
!
interface Tunnel18
 ip unnumbered Loopback0
 tunnel destination 150.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
!
interface Tunnel19
 ip unnumbered Loopback0
 tunnel destination 150.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
!
interface FastEthernet0/0
 ip address 150.1.24.2 255.255.255.0
 duplex auto
 speed auto
 mpls traffic-eng tunnels
 ip rsvp bandwidth 256 256
!
interface Serial0/0
 ip address 150.1.25.1 255.255.255.0
 mpls traffic-eng tunnels
 ip rsvp bandwidth 256 256
!
interface FastEthernet0/1
 ip address 10.1.20.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 10.1.20.0 0.0.0.255 area 0
 network 150.1.24.0 0.0.0.255 area 0
 network 150.1.25.0 0.0.0.255 area 0
 network 150.4.4.4 0.0.0.0 area 0
!
ip http server
ip classless
!
!
call rsvp-sync
!
```

```
voice-port 1/0/0
!
voice-port 1/0/1
!
!
mgcp profile default
!
dial-peer cor custom
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
!
!
end
```

```
R-7#sh run
Building configuration...

Current configuration : 2433 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname mid-1
!
logging queue-limit 100
!
memory-size iomem 10
ip subnet-zero
!
!
ip cef
mpls ldp logging neighbor-changes
mpls traffic-eng tunnels
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map TO_PE
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
```

```
!
!
no voice hpi capture buffer
no voice hpi capture destination
!
!
mta receive maximum-recipients 0
!
!
!
!
interface Loopback0
 ip address 150.5.5.5 255.255.255.0
!
interface Tunnel3
 ip unnumbered Loopback0
 tunnel destination 150.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel9
 ip unnumbered Loopback0
 tunnel destination 150.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
!
interface Tunnel13
 ip unnumbered Loopback0
 tunnel destination 150.6.6.6
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
!
interface Tunnel16
 ip unnumbered Loopback0
 tunnel destination 150.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
!
interface Tunnel20
 ip unnumbered Loopback0
 tunnel destination 150.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
!
interface Tunnel21
 ip unnumbered Loopback0
 tunnel destination 150.7.7.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
!
interface FastEthernet0/0
```

```
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 150.1.26.2 255.255.255.0
 mpls traffic-eng tunnels
 clockrate 64000
 no fair-queue
 ip rsvp bandwidth 256 256
!
interface Serial0/1
 ip address 150.1.25.2 255.255.255.0
 mpls traffic-eng tunnels
 clockrate 64000
 fair-queue 64 256 37
 ip rsvp bandwidth 256 256
!
router ospf 1
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
 log-adjacency-changes
 network 150.1.25.0 0.0.0.255 area 0
 network 150.1.26.0 0.0.0.255 area 0
 network 150.5.5.5 0.0.0.0 area 0
!
ip http server
ip classless
!
!
call rsvp-sync
!
voice-port 1/0/0
!
voice-port 1/0/1
!
!
mgcp profile default
!
dial-peer cor custom
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
!
!
end
```

```
R-8#sh run
Building configuration...

Current configuration : 2368 bytes
!
version 12.2
```

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname low-1
!
!
ip subnet-zero
!
!
!
ip cef
mpls traffic-eng tunnels
call rsvp-sync
!
class-map match-all VoIP
match access-group name VoIP
class-map match-all QOS_GROUP_1
match qos-group 1
class-map match-all QOS_GROUP_5
match qos-group 5
class-map match-all MPLS_EXP_5
match mpls experimental topmost 5
class-map match-all MPLS_EXP_1
match mpls experimental topmost 1
!
!
policy-map TO_P
class MPLS_EXP_5
priority 640
class MPLS_EXP_1
bandwidth 1000
random-detect
policy-map FROM_P
class MPLS_EXP_5
set qos-group 5
class MPLS_EXP_1
set qos-group 1
policy-map TO_CE
class QOS_GROUP_5
priority 640
class QOS_GROUP_1
bandwidth 1000
random-detect
policy-map FROM_CE
class VoIP
set qos-group 5
set mpls experimental imposition 5
class class-default
set qos-group 1
set mpls experimental imposition 1
!
!
interface Loopback0
 ip address 150.7.7.7 255.255.255.0
!
interface Tunnel5
```

```
 ip unnumbered Loopback0
 tunnel destination 150.8.8.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 1 dynamic
!
interface Tunnel10
 ip unnumbered Loopback0
 tunnel destination 150.2.2.2
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 2 dynamic
!
interface Tunnel14
 ip unnumbered Loopback0
 tunnel destination 150.6.6.6
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 3 dynamic
!
interface Tunnel17
 ip unnumbered Loopback0
 tunnel destination 150.3.3.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 4 dynamic
!
interface Tunnel19
 ip unnumbered Loopback0
 tunnel destination 150.4.4.4
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 5 dynamic
!
interface Tunnel21
 ip unnumbered Loopback0
 tunnel destination 150.5.5.5
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng path-option 6 dynamic
!
interface FastEthernet0/0
 ip address 10.1.30.1 255.255.255.0
service-policy input FROM_P
service-policy output TO_P
 duplex auto
 speed auto
!
interface Serial0/0
 ip address 150.1.27.2 255.255.255.0
service-policy input FROM_P
service-policy output TO_P
 mpls traffic-eng tunnels
 no fair-queue
 ip rsvp bandwidth 256 256
!
interface Serial0/1
```

```
 ip address 10.1.38.1 255.255.255.0
 no ip route-cache cef
!
router ospf 1
 log-adjacency-changes
 network 10.1.30.0 0.0.0.255 area 0
 network 150.1.27.0 0.0.0.255 area 0
 network 150.7.7.7 0.0.0.0 area 0
 mpls traffic-eng router-id Loopback0
 mpls traffic-eng area 0
!
router bgp 1
 bgp log-neighbor-changes
 network 150.7.7.0 mask 255.255.255.0
 neighbor 10.1.38.2 remote-as 2
 neighbor 150.8.8.8 remote-as 1
 neighbor 150.8.8.8 update-source Loopback0
 neighbor 150.8.8.8 next-hop-self
!
ip access-list extended VoIP
permit udp any any range 16384 32767 dscp ef
!
ip classless
ip http server
!
!
!
voice-port 1/0/0
!
voice-port 1/0/1
!
dial-peer cor custom
!
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
!
end
```

```
Switch-3750#sh run
Building configuration...

Current configuration : 1628 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
!
hostname Switch
!
!
no aaa new-model
switch 1 provision ws-c3750g-24ps
ip subnet-zero
!
!
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
interface GigabitEthernet1/0/1
!
interface GigabitEthernet1/0/2
 bandwidth 45000
!
interface GigabitEthernet1/0/3
 bandwidth 45000
!
interface GigabitEthernet1/0/4
!
interface GigabitEthernet1/0/5
!
interface GigabitEthernet1/0/6
!
interface GigabitEthernet1/0/7
!
interface GigabitEthernet1/0/8
!
interface GigabitEthernet1/0/9
!
interface GigabitEthernet1/0/10
!
interface GigabitEthernet1/0/11
!
interface GigabitEthernet1/0/12
!
interface GigabitEthernet1/0/13
!
interface GigabitEthernet1/0/14
!
interface GigabitEthernet1/0/15
!
interface GigabitEthernet1/0/16
!
interface GigabitEthernet1/0/17
!
interface GigabitEthernet1/0/18
!
interface GigabitEthernet1/0/19
!
```

```
interface GigabitEthernet1/0/20
!
interface GigabitEthernet1/0/21
!
interface GigabitEthernet1/0/22
!
interface GigabitEthernet1/0/23
!
interface GigabitEthernet1/0/24
!
interface GigabitEthernet1/0/25
!
interface GigabitEthernet1/0/26
!
interface GigabitEthernet1/0/27
!
interface GigabitEthernet1/0/28
!
interface Vlan1
 ip address 150.1.28.3 255.255.255.0
 no ip route-cache
!
ip classless
ip http server
!
control-plane
!
line con 0
 logging synchronous
line vty 0 4
 no login
line vty 5 15
 no login
!
monitor session 1 source interface Gi1/0/2 - 3
monitor session 1 destination interface Gi1/0/1
```