Search

Go

- Software Notes
- Documentation
- MIBS
- Riverstone Technical Assistance Center
- File Exchange
- Service Contacts
- Knowledge Base
- Software Download
- Request a Password

Note: A user ID and Password are required to access Knowledge Base and Software Download Content.

Riverstone MPLS Support Page

Multiprotocol Label Switching (MPLS) – An MPLS primer reviewing some of the key architectural components as described in RFC3031, including such discussion points as control versus data plane, MPLS labels types and label management, with a brief protocol history.

Preparing The Network For MPLS – Works through the underlying network requirements for deploying MPLS networks, including such discussion points as creating the underlying IP network, the type of signaling protocols that may be deployed and what influences the deployment decisions. Finally, this is meant to provide a step by step guide to creating a base MPLS enabled network using a dynamic signaling.

Interior Gateway Protocols in an MPLS Environment – Discusses the challenges with the base link state routing protocols and how OSPF & IS-IS extensions have been introduced to provide more information on the state of network resources. The resulting resource information flooded through link state advertisements populates the traffic engineering database. The resource information is used to optimize the selection of end-to-end paths through a network taking into consideration the specific requirements of the path and available resources.

Different Signaling Dynamics in MPLS – Discussions surround the different approaches to signaling and distributing information through an MPLS network. Three of the existing dynamic signaling protocols, LDP, RSVP-TE & CR-LDP are positioned and compared.

Resource ReSerVation Protocol – Traffic Engineering (RSVP-TE) – A very detailed discussion covering the extensions to RSVP creating the RSVP-TE protocol which includes new objects and improved scalability. It also describes the process by which a Label Switched Path, LSP, is signaled and established with end-to-end significance using the traffic engineering capabilities or following the underlying routing information. The topics here are confined to the specific protocol details and establishment of the end-to-end LSP. Step by step configurations specific to the RS platform accompany the discussion topics. Making use of an established LSP is a follow on in Layer Two Virtual Private Networks and Layer Three Encapsulation in MPLS.

Label Distribution Protocol (LDP) – Introduction to the LDP protocol accompanied by a discussion on where the protocol can is mostly likely to be deployed. With detailed discussions on the different types of peering relationships and stepping through finding a peer up to exchanging label binding to FEC information. As well as the method used to map an active next hop IP address in the forwarding table to a label in the database. The topics here are confined to the specific protocol details. Making use of an established LSP is a follow on in Layer Two Virtual Private Networks and Layer Three Encapsulation in MPLS.

Layer Two Virtual Private Networks (Martini) – A look at the ability to encapsulate layer two traffic and tunnel it across a shared MPLS network, while maintaining individual customer privacy using a virtual circuit approach. Examine the point-to-point Virtual Leased Line, VLL, service that forms the basis for many of the efforts underway in the L2 VPN space. A further effort is underway in the community looking to extend the VLL model to allow Transparent LAN Services, TLS. The TLS extensions allow the MPLS cloud to appear as a traditional learning bridge to subscribers, yet remaining a scalable manageable highly tunable MPLS network. Step by step configurations specific to the RS platform accompany the discussion topics

Layer Three Encapsulation in MPLS – Details the mapping of packets to an FEC based on layer three or four protocol information into an RSVP-TE tunnel. All layer two information is stripped from the inbound packet and the remaining native IP packet in encapsulated in an MPLS header. Local policy on the ingress router classifies the packet based on protocol header information up to and including layer four socket. Edge routers need to have enough routing table information to be able to forward the native IP packet when it arrives at the egress. The core MPLS network needs only enough routing information to be able to perform RSVP-TE control functions.

Multiprotocol Label Switching (MPLS)

Introduction to MPLS

- History of Multiprotocol Label Switching
- Important Terms and Definitions
- Separation of Control & Data Plane

Encoding MPLS Labels

- Two Types of Encoding Techniques
- The MPLS Shim
- Native Layer Two Encoding

Label Processing and Values

- MPLS Packet Walkthrough on an RS Router
- Defining the Label Space on the RS Platform

Label Distribution and Management

- Role of the MPLS Label
- Distributing MPLS Labels
- Label Retention
- Label Advertising
- Label Significance
- Demonstrating Different Label Distribution & Management

Introduction to MPLS

Introduction to Multiprotocol Label Switching (MPLS)

History Of Multiprotocol Label Switching - MPLS

Multiprotocol Label Switch, or MPLS, has its roots in many of the early tagging and label swapping protocols. To present an incomplete list, Ipsilon's IP Switching, IBM's Aggregate Route-based IP Switching and Cisco's Tag switching, all developed proprietary approaches to provide data forwarding using labels. These early developments were meant to resolve the challenges presented with overlay models, like ATM and IP. The overlay models delivered improved quality of service but incurred the N^2 meshing problem. Another goal of the first label forwarding technologies was to extend the life of software based routers that dominated large provider networks. The potential of such a promising technology could only be realized through the standards process, general technology acceptance and vendor interoperability.

The arrival of hardware based routers, capable of non-blocking wire speed performance, meant that extending the life of software based router faded into the background. However, new applications including dynamic traffic engineering and layer two and layer three virtual private networks took center stage. The destination based hop-by-hop best effort IP networks, which dominate today's landscape, are starting to realize some growth limitations and restrictions. In

traditional IP based networks, routers work independently to determine the best next hop for packet forwarding, with then as the root of a graph of routers. Robust interior gateway protocols, like OSPF and IS-IS, consult a link state database to understand the entire network topology and choose the best next hop based on the lowest cost. Ultimately, all lowest equal cost paths are selected and used to forward data. Indeed, some vendors do employ the use of a tolerance or variance in an effort to gain the use of lesser-preferred routes. However, today's complex partial and full mesh networks have a myriad of unequal cost paths that are outside of tolerances, yet can certainly service many of the data requests and still they go unused or at least under utilized. The exhaustion of a singular or multiple equal cost routes when other route are underutilized, or not used at all, is commonly referred to as hyper aggregation. The answer is to gain control of the network and map customer traffic to applicable network path or paths that meet that specific customer's requirements.

It is expected, longest prefix match, hop-by-hop, best effort networks will give way to traffic engineered, multi-service networks capable of providing virtual private networks. Among the many positive attributes MPLS brings to internetworking is the ability to provide connection-oriented services to the inherently connectionless IP networks. The label switched path, or LSP, is the establishment of a unidirectional end-to-end path forwarding data based on fixed size labels.

Riverstone Networks is committed to a leadership role in delivering MPLS enabled networks.

For a complete review of the MPLS Architecture consult RFC3031.

Important MPLS Terms and Definitions

Forwarding Equivalence Class -FEC: An FEC represents the binding a group of packets or flows that require the same handling, like class-of-service, and egress node to the same label and thus the same label switch path. In traditional IP networks requests with the same destination prefix could be mapped to the same

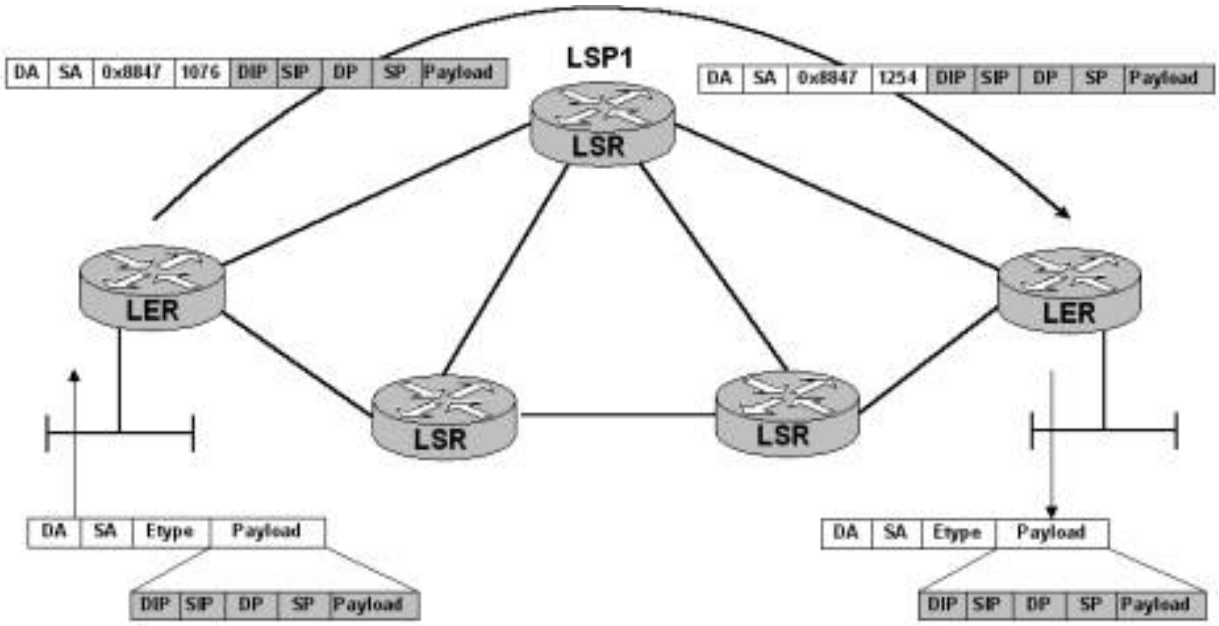
FEC, if no local policy stated otherwise.

Path: The collection of router from ingress to egress that packets will cross as they are forwarded through the network.

Label Switched Path: A complete path that has the ability to map incoming MPLS labeled packets to some outgoing action.

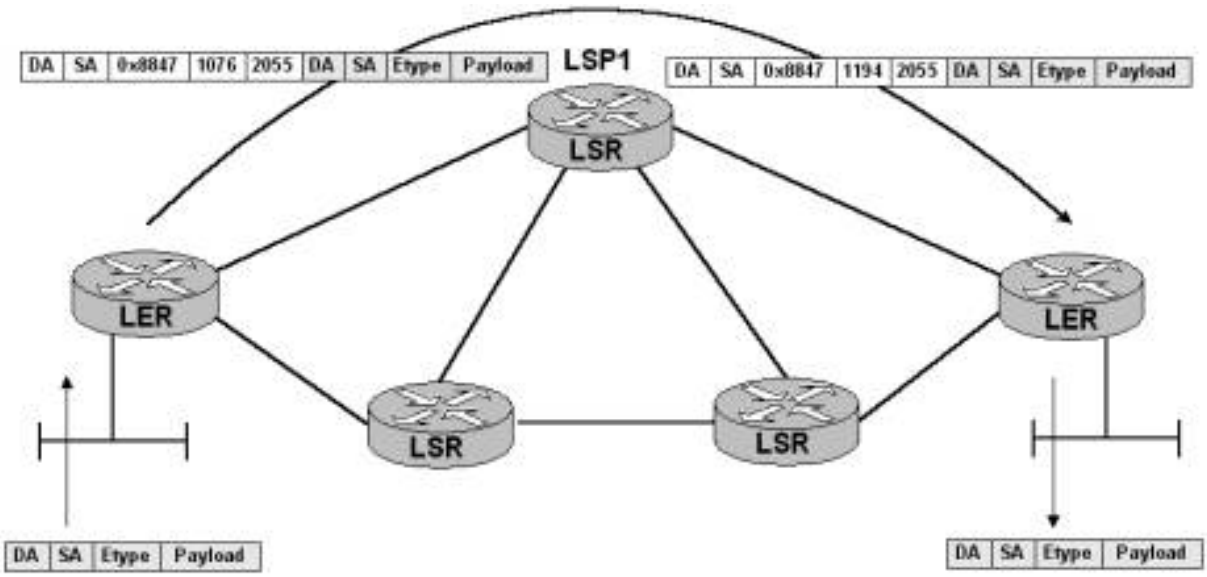
Tunnel: The encapsulating transport allowing packet movement through a network below the traditional forwarding mechanisms of the network. Simply put, MPLS creates a tunnel beneath the traditional IP forwarding component using labels between layer two addressing information and the encapsulated packet. Many independent labels sharing a single passageway through the network creates a hierarchy. The stacking of labels creates virtual channels within the tunnel and provides customer isolation.

An example of a single label encapsulating a native IP data frame would look similar to the following. This represents an IP frame encapsulated in MPLS label that is using RSVP.



An example of multiple labels and a label hierarchy would look similar to the

following. This represents the encapsulation of a layer two packet into an MPLS using RSVP-TE at the top level and LDP at the bottom of the stack.



Penultimate Hop Pop – PHP: The router preceding the final router in the label switched path removes the label and sends the native packet to the final router in the label switched path. The preceding router understands it is to perform this function when it receives the label value of "3" for the FEC.

Router Types: There are three types of *Label Switched Routers, or LSR*, in an MPLS network. The role they play depends on their geography with respect to the data they are passing. This means, even though a label switched router is geographically situated at the edge of an MPLS network it is not relegated to the tasks of an edge router. In most cases the edge MPLS router will perform the functions required at the edge. However, it may also be required to perform the duties of a transit router.

Assuming penultimate hop propping is not being preformed the following is a brief review of the functions each type of router performs, ignoring label distribution. The label to FEC binding depends largely on which type of type of label distribution and advertising methods have been deployed.

Ingress Label Switched Router – Ingress LER	The entry point into the MPLS network. • Calculating the path through the MPLS network • Instantiating label switched path • Classifying inbound traffic into L2 or L3-FEC
Transit Label Switched Router - LSR	An MPLS router somewhere along the LSP that forwards traffic based on MPLS labels that is neither the source nor the destination.
Egress Label Switched Router – Egress LER	The exit point connecting an MPLS network to a traditional network, the mapping of inbound label to the router itself, indicates the end of the tunnel. • Remove label and act on native packet

Separation of Control and Data Planes

Having seen some of the shortcomings with today's longest prefix match, hop-by-hop, best effort routed networks a closer look at the separation between the control plane and the data plane demonstrates the new perspective used to develop the various MPLS related standards and Internet-Drafts. The control plane relies heavily on the underlying IP infrastructure to disseminate the decision-making information, establish paths and maintain established paths through the MPLS network. The forwarding plane is a *tunnel* created below the IP layer carrying client data. The concept of a tunnel is key because it means the forwarding process is not IP based and classification at the ingress, or entry point to the MPLS network, is not relegated to IP only information. This enables the flexibility when defining the concept of "*Customer Identification*". Since no longer IP specific, other criteria such as physical port, VLAN ID, or the combination of VLAN ID and the physical port, at the ingress, provide the ability to tunnel any traffic over the backbone, see Layer Two Virtual Private Networks. Of course the network

protocol header information could provide layer three IP over MPLS services, mapping requests using layer three or four protocol header information to specific tunnels, see Layer Three Encapsulation in MPLS.

Information Dissemination: Control Plane – The link state protocols, specifically OSPF and IS-IS, provide the link state information that details the entire underlying IP network. This information is crucial to the path selection, path establishment and maintenance functions. Further, both OSPF and IS-IS protocols have been extended to include resource style information about all links in the specific area. Through these extensions MPLS traffic engineering becomes possible. For a discussion about the role the IGP plays in an MPLS network see Link State Protocols in an MPLS Environment.

Path Selection: Control Plane - Determine the best path through a network using either using a hop-by-hop or an explicit route methodology. The hop-by-hop method allows the path selection to follow the normal underlying IGP best path. Each node in the path is responsible for determining the best next hop based on the link state database. Alternatively, an explicit route is a path through the network that is specified by the instantiating router. The explicitly routed path has administratively configured criteria, like constraints, to influence the path selection through the underlying network. It is very possible an explicit route will deviate from a path that would have been selected using the hop-by-hop IGP method.

Path Establishment: Control Plane - Once the path has been determined, a signaling protocol is used to inform all the routers in the path that a new label switch path, or LSP, is required. The signaling protocol is responsible for indicating the specifications of the path, including the session id, resource reservations, and the like, to all other routers in the path. This process also includes the label mapping request for all data that will use the label switched path. Following the successful establishment of the path the signaling protocol is responsible for ensuring the integrity of the peering session.

Packet Forwarding: Data Plane - At the very highest level and in general terms, the data flow toward an MPLS network occurs at the ingress label switch router,

commonly referred to as ingress label edge router, or ingress LER. The ingress LER classifies a packet or a flow to a specific path label switch path and pushes the applicable label on the packet. This classification of client data to label switched path occurs only once, at the ingress router, using some policy. Routers along the label switched path perform forwarding based on the top level inbound label. The label switched path terminates at the boundary between an MPLS enabled network and traditional network. The egress label switch router, the egress LER, is responsible for removing the label from the packet and forwarding the packet based the packets original contents, using traditional means. Different approaches forwarding mechanisms will be explored as throughout these pages.

Layer Two Virtual Private Networks (Martini)

L2 VPN Introduction

Virtual Leased Lines - Martini

- **Martini Drafts**
- **Tunnel LSP Creation**
- **Signaling the VC Label**
- **Monitoring the Signaling Process**
- **Forwarding Packets**
- **Creating & Signaling the Different VC ID & Group ID**
- **L2-FEC Transport**
- **Steps to Delivering VLL Services**
- **Virtual Lease Line Examples**
- **Related Show Commands**

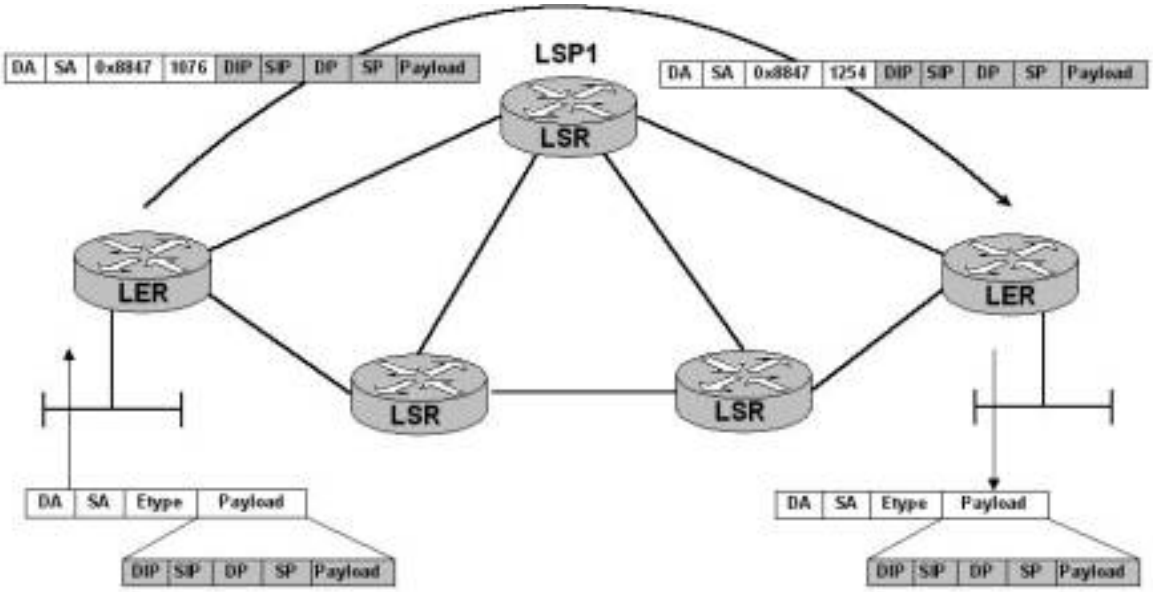
Transparent LAN Services - TLS

Layer Three Encapsulation in MPLS

Layer Three Encapsulation in MPLS

Packet Processing for L3 Encapsulation

RSVP-TE instantiates the required tunnels through the core network to interconnect the required points of presence. Using any of the methods supported by RSVP-TE, explicitly routed with resource reservations or hop-by-hop, creates these tunnels. Once these tunnels link the edge routers, policy on the ingress router is used to classify and map the inbound traffic to a specific tunnel. All layer two information is stripped and the native IP packet is encapsulated in the MPLS label for the specified tunnel and forwarded accordingly.



Creating The L3-FEC with Policy

The policy statements on the ingress MPLS router use information in the protocol header to classify inbound packets into an FEC. Policy is not limited to the ability to classify based purely on Source or Destination IP address. Classification can look deep into the header information to include Source Socket, Destination Socket, Protocol and Type of Service, as well as the base IP information. How the profile is created will depend on the end goal. For example, classifying based on prefix would map all packets from or to a prefix to a specific LSP. If more granularity is required the layer four socket information could be used to differentiate application types.

A list of the classification criteria…

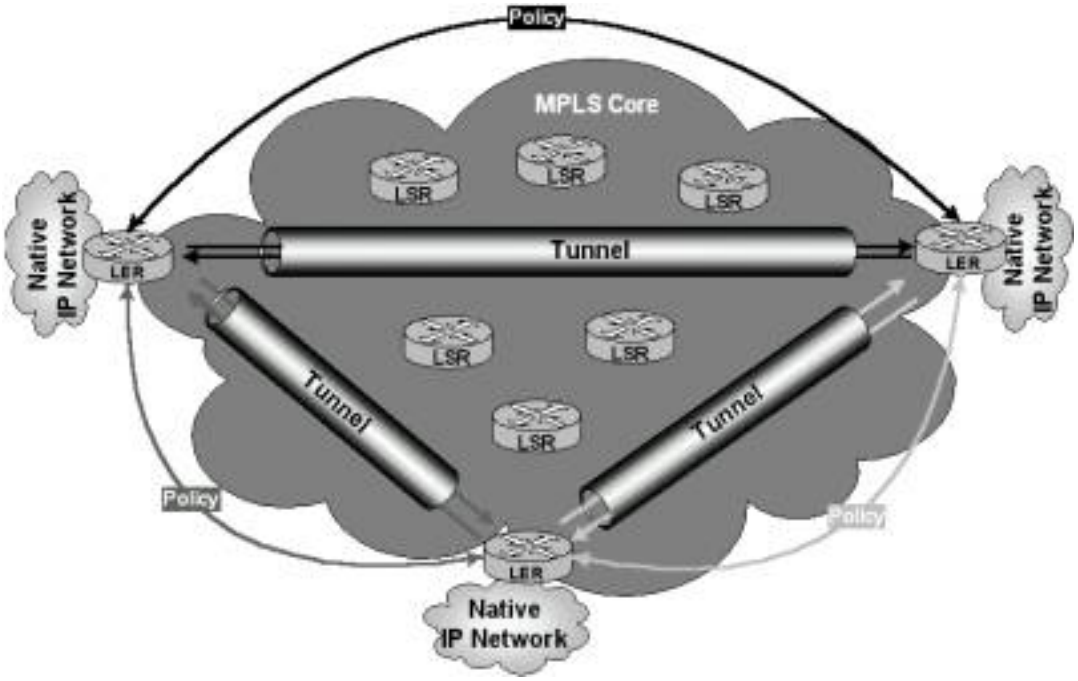
```
RS(config)# mpls create policy <name> <classification>
 dst-ipaddr-mask           - Destination IP address and mask
 dst-port                  - Destination TCP/UDP port number
 proto                     - Protocol
 src-ipaddr-mask           - Source IP address and mask
 src-port                  - Source TCP/UDP port number
 tos                       - Type of Service
```

```
 tos-mask                    - The mask used for the tos byte.
Default is 30
```

Once the policy is created it must be associated to one of the existing RSVP-TE LSPs that has been created.

RS(config)# mpls set label-switched-path <name> policy

It is important to realize this is not a Virtual Router solution. Each edge routers participate in the native IP network to which they are connected, this means the route tables must have destination based information for packets that must be forwarded from the MPLS core to the native IP network in which they reside. The core network does not have to be aware of any of the IP information that is not part of the MPLS core network; it is shielded from having to know those routes. Again, the beauty of using a dynamic signaling protocol means the configuration of the core network remains unchanged as new LSPs and policy are applied at the edge.



Steps to Delivering IP over MPLS

IP over MPLS can be summarized in three simple steps

1) Build the MPLS core network
   - IGP required to distributed reachability information with traffic engineering, optional but recommended
   - MPLS and RSVP are required on all core facing interfaces, edge and transit.

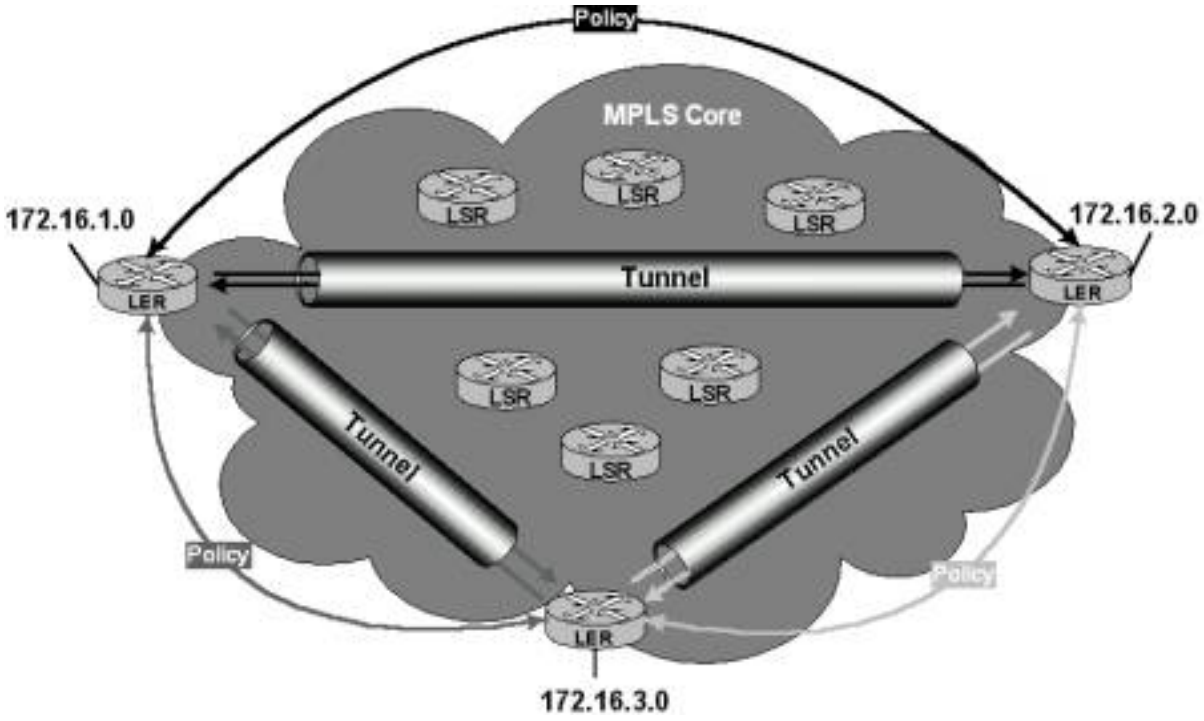
2) Edge routers instantiate the tunnels using RSVP-TE

3) Define the L3-FEC, via policy, on the edges of the network and map to appropriate LSP

Once again, it is important to remember that MPLS networks are unidirectional and need complementary label switched paths running in each direction to facilitate bi-directional communications.

IP over MPLS Example

Using this network, the key configuration components will be described using the steps presented above.



1) All core based transit label switch routers have the IP knowledge for core reachability, using either OSPF-TE or IS-IS-TE. The native IP routing information outside the MPLS core is not present in the core routing tables. MPLS must also enable on all core interfaces. A sample transit router configuration is presented below. All transit routers will follow this same basic configuration, with the obvious deviations, like IP addressing on interfaces.

```
interface create ip Core1-LER1 address-netmask 192.168.1.1/30 port
gi.3.1
interface create ip Core1-Core2 address-netmask 192.168.1.9/30
port gi.3.2
interface create ip Core1-Core3 address-netmask 192.168.1.13/30
port gi.4.2
interface add ip lo0 address-netmask 1.1.1.1/32
ip-router global set router-id 1.1.1.1
ospf create area backbone
ospf add interface Core1-LER1 to-area backbone
ospf add interface Core1-Core2 to-area backbone
ospf add interface Core1-Core3 to-area backbone
ospf add stub-host 1.1.1.1 to-area backbone cost 10
ospf start
mpls add interface all
mpls start
rsvp add interface all
rsvp start
ospf set traffic-engineering on
```

Once the core MPLS network has been established the edge routers need to be configured to participate as part of the core MPLS and native IP networks. Core facing interfaces require MPLS and RSVP-TE to be enabled. The interfaces facing the native IP network need to be created as participatory for the IP only network, no MPLS or RSVP-TE support is required. A sample configuration for one of the three edge routers is presented below.

```
interface create ip To-Core address-netmask 192.168.1.2/30 port
gi.3.2
```

```
interface create ip To-NativeIP address-netmask 172.16.1.1/24 port
gi.4.2
interface add ip lo0 address-netmask 3.3.3.3/32
ip-router global set router-id 3.3.3.3
ospf create area backbone
ospf add interface To-Core to-area backbone
ospf add stub-host 3.3.3.3 to-area backbone cost 10
ospf start
mpls add interface To-Core
mpls start
rsvp add interface To-Core
rsvp start
ospf set traffic-engineering on
```

2 & 3) The options for signaling the RSVP-TE tunnel from the edge network may include the use of traffic engineering, backup paths or simply hop by hop. The simple example below instantiates the end-to-end RSVP-TE without traffic engineering, using loopback address as tunnel start and end points. After the tunnels have been configured the policy is defined it is associated to an LSP.

Here three sites will be interconnected by LSPs and policy maps traffic to the appropriate LSP using a match of the destination IP address. The policies are written in such a way to allow any non-local traffic to be mapped to the LSP that has a connection to that remote site. None of the routes from any of the native IP networks are found in any routers, other than the edge to which they belong.

The IP addresses located at each site are important to note, to understand how this policy has been written.

Site Number	IP Address Within Site
1	172.16.1.0/24
2	172.16.2.0/24
3	172.16.3.0/24

LER1

```
mpls create label-switched-path LSP13 adaptive from 3.3.3.1 to
3.3.3.3
mpls create label-switched-path LSP12 adaptive from 3.3.3.3 to
3.3.3.2
mpls create policy Sub1Site2 dst-ipaddr-mask 172.16.2.0/24
mpls create policy Sub1Site3 dst-ipaddr-mask 172.16.3.0/24
mpls set label-switched-path LSP12 policy Sub1Site2
mpls set label-switched-path LSP13 policy Sub1Site3
```

LER2

```
mpls create label-switched-path LSP21 adaptive from 3.3.3.2 to
3.3.3.1
mpls create label-switched-path LSP23 adaptive from 3.3.3.2 to
3.3.3.3
mpls create policy Sub1Site1 dst-ipaddr-mask 172.16.1.0/24
mpls create policy Sub1Site2 dst-ipaddr-mask 172.16.2.0/24
mpls set label-switched-path LSP31 policy Sub1Site1
mpls set label-switched-path LSP32 policy Sub1Site2
```

LER3

```
mpls create label-switched-path LSP21 adaptive from 3.3.3.3 to
3.3.3.1
mpls create label-switched-path LSP23 adaptive from 3.3.3.3 to
3.3.3.2
mpls create policy Sub1Site3 dst-ipaddr-mask 172.16.3.0/24
mpls create policy Sub1Site1 dst-ipaddr-mask 172.16.1.0/24
mpls set label-switched-path LSP23 policy Sub1Site3
mpls set label-switched-path LSP21 policy
Sub1Site1
```

If an inbound packet from the native IP network does not match a policy normal routing rules apply, destination based longest prefix match. However, in the case above, since prefixes are local and not propagated to the rest of the network the packets are discarded

.

Related Show Commands

Some useful show commands are presented in this section.

An detailed look at the policies shows information about each policy, including the classification criteria and the label switched path that is used when there is a match.

RS# mpls show policy <options>

```
LER# mpls show policy verbose
Name                  : Sub1Site2
Type                  : L3
Source address        : anywhere
Source Port           : any
Destination address   : 172.16.2.0/24
Destination Port      : any
TOS                   : any
TOS Mask              :
Protocol              : IP
Used by               : LSP12

Name                  : Sub1Site3
Type                  : L3
Source address        : anywhere
Source Port           : any
Destination address   : 172.16.3.0/24
Destination Port      : any
TOS                   : any
TOS Mask              :
Protocol              : IP
Used by               :
LSP13
```

A more detailed look at the policy provides information about the match, the next hop, the label to be used, the state of the LSP, and so on.

RS# ip-policy show all

```
CPE-LER1# ip-policy show all
-------------------------------------------------------------------------
------------
```

```
IP Policy name     : MPLS_PBR_LSP12
Applied Interfaces : all-IP-interfaces local-policy
Load Policy        : first available
Health Check       : disabled

ACL          Source IP/Mask     Dest. IP/Mask      SrcPort
DstPort   TOS TOS-MASK Prot ORIG AS
---          --------------     -------------      -------   ------
-   --- -------- ---- -------
MPLS_ACL_Su  anywhere           172.16.2.0/24      any
any       any  None     IP

                                                      Next Hop
Information
                                                      -------------
-------
 Seq   Rule   ACL          Cnt      Action       Next Hop
Cnt      Last
 ---   ----   --------     ---      -----------  --------         --
-      ----
 10    permit MPLS_ACL_Su  14       Policy First 192.168.1.1
14       Up
 LSP state            : Up
 OTT index            : 1
 Label Stack Count    : 1
 Labels               : 17,

 65536 deny   deny         3141     N/A          normal fwd
N/A      N/A



------------------------------------------------------------------------
-------------
IP Policy name     : MPLS_PBR_LSP13
Applied Interfaces : all-IP-interfaces local-policy
Load Policy        : first available
Health Check       : disabled

ACL          Source IP/Mask     Dest. IP/Mask      SrcPort
DstPort   TOS TOS-MASK Prot ORIG AS
```

```
---          --------------     -------------      -------   ------
-   --- -------- ---- -------
MPLS_ACL_Su  anywhere           172.16.3.0/24      any
any       any  None     IP

                                                        Next Hop
Information
                                                        ------------
-------
 Seq   Rule   ACL           Cnt      Action        Next Hop
Cnt      Last
 ---   ----   --------      ---      -----------  --------         --
-      ----
 10    permit MPLS_ACL_Su  3        Policy First 192.168.1.1
3        Up
 LSP state             : Up
 OTT index             : 2
 Label Stack Count     : 1
 Labels                : 18,

 65536 deny   deny         3138     N/A          normal fwd
N/A      N/A
```

Link State Protocols in an MPLS Environment

Link State Protocols in an MPLS Environment

Challenges With Basic Link State Interior Gateway Protocols

Of the well know limitations of the existing link state database interior gateway protocols surrounds their inability optimize network resources. Simply put the popular link state database routing protocols, OSPF and IS-IS, in their base implementation, restrict the ability for service providers to leverage network resources to their fullest extent. Each router independently maintains the same graph of the network created by listening to link state advertisements from other routers. Each router uses the information in the link state database to determine which is the best, shortest path, through the network, based on longest prefix matches, positioning themselves at the base of the graph. Since all routers have the same view of the network it is likely that the same shortest path will be saturated while other paths through the network go underutilized or in the worst case unused.

Various approaches are available to scale oversubscribed links, including Link Aggregation 802.3ad, support for equal cost multipath and to a lesser extent allowing for slightly unequal costs that fall within a acceptable variance to be used as well. However, as the complexity of the underlying IP network grows and traffic patterns become less deterministic better methods are required to ensure all links in a network can be optimized.

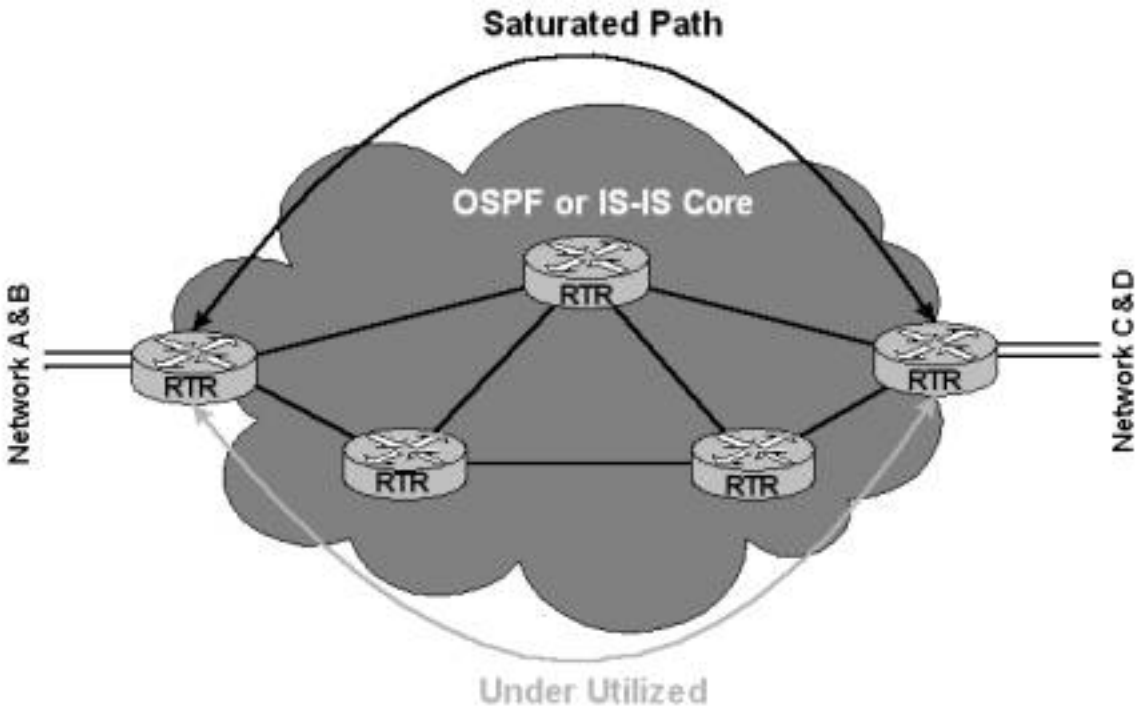
As robust as the OSPF and IS-IS are, they both suffer from a convergence issue that worsens as the network increases in size and complexity. During a steady state the routes are predictable and convergence is not present. However, in the event of failure, the network converges over time and traffic may be routed over unexpected paths. This newly converged shortest path first graph may result in the creation of oversubscribe links in different parts of the network.

For a complete understanding of Riverstone Networks support of the OSPF protocol refer to the OSPF Support Page.

The following simple network is meant to demonstrate some of the issues with the base level OSPF and IS-IS routing protocols.

In this example, there are two possible paths connecting the networks on each side of the network core. Assuming that all routers are connected using Gigabit Ethernet, the routing protocol will prefer the northerly most path because it represents the shortest path between the edge networks. By default the southerly path and the links used to partially mesh the core will go completely unused.

Should a failure occur, the network will ultimately converge and route around the failure. Although this self-healing is very welcome, the new preferred routes through the network are not deterministic and completely dependent on where the failure occurred. Now consider increasing the complexity of this simple network to add more core routers, meshing and edge networks. Things can get complicated quickly.



For a complete sample configuration, including forwarding tables click here.

MPLS, through its use of dynamic signaling protocols, such as RSVP-TE, provide the tools required to take control of the network using a deterministic approach. However, the signaling process needs to understand reachability and resource information. To this end, both OSPF and IS-IS have been extended to support traffic engineering.

Interior Gateway Protocol Traffic Engineering Extensions

The Internet draft draft-katz-yeung-ospf-traffic-06.txt defines the traffic engineering extension for OSPF.

The Internet draft draft-ietf-isis-traffic-04.txt defines the traffic engineering extension for IS-IS.

Both the drafts define are similar in nature, using the concept of "*sub-TLV*" to carry the necessary information about the resources of routed links. OSPF introduces the concept of the "*Opaque LSA – Type 10*" and IS-IS introduces changes to the "*IS Neighbor TVL and IP Reachability TVL*". These changes

allow the link state advertisements to include link resource information. The extensions move path calculation beyond the basic *Shortest Path First, or SPF*, by allowing administratively defined constrains to influence the path through the network. The resulting path calculation uses the constraints defined on the instantiating router as input into the SPF process, thus creating the concept of *Constraint-based Shortest Path First, or CSFP*.

The CSPF process follows a similar process as the SPF process. It works form the root, itself, to the destination using the shortest path that can accommodate the configured constraints.

Both IS-IS and OSPF-TE are defined as intra-area or intra-level. Link information is not carried between areas or levels.

Each of the routing protocols is extended in their own syntax to include the following extensions available for consideration when calculating the CSFP.

Traffic Engineering (Default) Metric – A value that will be considered when calculating the CSPF process. The metric is specific to the CSPF process and used to determine which link is most preferred if two possible links meet the constraints. If this value is not specified the default metric of this link is used in the calculation. Riverstone uses the default link metric, as advertised by the routing protocol.

Maximum (Link) Bandwidth –The maximum, unidirectional, bandwidth the link is capable of sending, 4 octets in length. By default this metric is based on the physical capabilities of the link. However, this default can be overridden using the following command, where the bandwidth is represented in bits per second…Link state advertisements are automatically generated for changes represents more than a 5% change in available bandwidth. This is for the benefit of the CSPF process and does not result in execution of Dijkstra's algorithm.

```
RS(config)# mpls set interface <name> bandwidth <1-1000000000>
```

Maximum Reservable (Link) Bandwidth – The maximum reservation capacity, unidirectional, of a link, 4 octets in length. By default maximum reservable bandwidth is equal to the maximum (link) bandwidth. If it is preferable to over subscribe or under subscribe a link the following command can be used, where the value of the subscription is a percentage of the maximum bandwidth… Link state advertisements are automatically generated for changes represents more than a 5% change in available bandwidth. This is for the benefit of the CSPF process and does not result in execution of Dijkstra's algorithm.

```
RS(config)# mpls set interface <name> subscription <0-64000>
```

Unreserved Bandwidth – This provides the amount of bandwidth at each of the eight, 0-7, priority levels. Currently, all the priority levels are reduced when a reservation of bandwidth is made. There is no distinction between priority levels at the current time.

Link Colors or Resource Classes (affinity) – The link attribute that defines the links membership to a group. This filed is 32 bits in length, allowing for up to 32 domain wide memberships to be configured. When a bit is in the "on" state, as represented by a one in the 32-bit field, the link is a member of that particular membership. Link state advertisements are automatically generated should an administrative value for a link be altered. Before setting the link memberships the administrative group must be created, where the group value represents the bit position in the 32 bit field.

```
RS(config)# mpls create admin-group <name> group-value <0-31>
RS(config)# mpls set interface <name> admin-group <group-name>
```

To enable traffic engineering for OSPF

```
RS(config)# ospf set traffic-engineering on
```

To enable traffic engineering for IS-IS

```
RS(config)# isis set traffic-engineering enable
```

Encoding MPLS Labels

Encoding MPLS Labels

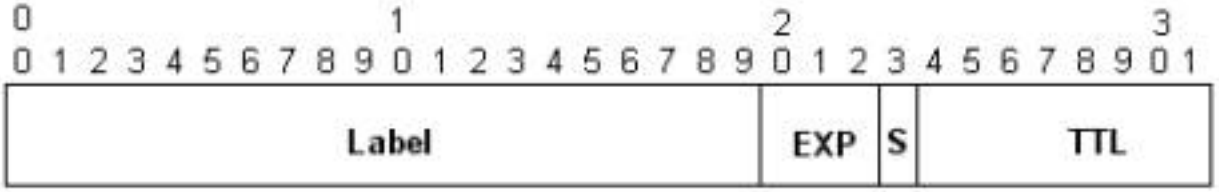
Two Types Of Encoding Techniques

MPLS labels are the key forwarding mechanisms in an MPLS network. The labels used in an MPLS network can take two different forms, the *shim*, used in packet based networks like Ethernet and Packet over Sonet, or native layer two encoding, seen with link layer technologies like Asynchronous Transfer Mode and Frame Relay.

The MPLS Shim

The shim used in packet based networks consists of a four octet fixed length data field placed between the layer two link layer information and the encapsulated packet. The layer two *EtherType 8847* immediately following the layer two source MAC address identifies the packet as containing the MPLS shim. This identifier uniquely maps the traffic to the proper label switched path throughout the MPLS enabled network.

The 32 bit label is divided out into four fields.



Label: A 20 bit field mapping the FEC to an MPLS identifier. This is the label value.

Experimental - EXP: It is generally accepted that this 3 bit field differentiates classes of service or per hop behavior for differing classes of traffic traveling within the LSP, or E-LSP. Alternatively, a LSP that carries a single traffic class, and does not use the EXP, uses the label to determine the per hop behavior of the single class using traffic traveling within the LSP, or L-LSP.

Bottom of Stack – S: This single bit field in position 23 represents the last MPLS label contained in the packet. Remember, labels can be used in a hierarchy, to deploy services like *Virtual Private Networks*. The Riverstone Networks implementation can push or pop up to three labels at a time.

Time-to-Live – TTL: The final 8 bits of the MPLS shim is analogous to the IP TTL field. By default, the Riverstone Networks implementation works as a traditional IP router with respect to the TTL field. The ingress LER decrements the IP TTL by one and copies the result to the MPLS TTL. Each Transit LSR decrements the MPLS TTL by one before forwarding the packet. Finally, the Egress LER decrements the MPLS TTL and copies the result to the IP TTL and forwards the packet natively. If a penultimate hop pop is performed at the preceding LSR then no MPLS packet is seen at the egress node and IP TTL logic takes over. Should a packet arrive with an MPLS TTL value of 0 or 1 the packet is sent to the CPU and the ICMP error message “TTL Expired” is reported using the IP interface local to the router participating in the LSP. This is useful when using *traceroute* over an MPLS network.

The following command, overrides this process and packets are silently be dropped in hardware, saving what would be otherwise wasted CPU cycles.

```
RS(config)#  mpls set global drop-zero-ttl-packets
```

It is also possible the override the default TTL process and have an MPLS router not decrement the TTL field. If the ingress LER has this command in the configuration the MPLS cloud appears as a single hop when using a traceroute command. Also, issuing a ping with a hop count limit of two will allow the packet to traverse an entire MPLS cloud and reach its destination, assuming of course the nodes is directly attached to the egress LER. However, when this command is not coded on the ingress LER and is coded on other routers in the path the traceroute function will reveal all the routers in the path, yet those routers with this command in their configurations will not decrement the TTL as the packet passes through that MPLS node.

```
RS(config)#  mpls set global no-propagate-ttl
```

Specific to an RSVP-TE signaled LSP, the same functionality can be set on a per LSP basis. When the instantiating router signals the LSP the TTL functionality is contained as parameters for that LSP. WARNING: If the LSP is already established and this command is issued, the LSP will be torn down and re-established, causing a service interruption along that LSP.

```
RS(config)#mpls set label-swicthed-path <name> no-decrement-ttl
```

Native Layer Two Encoding

For layer two technologies that cannot accommodate the use of the MPLS shim, the label information must be encoded in the link layer information. Technologies like ATM and Frame Relay have used the concept of label swapping since their inception. Where these link layer technologies have been deployed MPLS may use the existing circuit identification space as the MPLS label. ATM uses the virtual path identifier/virtual channel identifier (VPI/VPI) pair and Frame Relay networks use Data Link Control Identifier (DLCI).

Both types of labeling techniques, shim and link layer encoding, can co-exist in the same network along the same LSP, as long as the routers connecting the two disparate label capable link layers has the ability to perform the necessary conversion.

Note: The Riverstone Networks implementation of MPLS supports only packet based networks. Therefore, support for using ATM and Frame Relay in the MPLS network core is not available. However, any client facing network interfaces may be used, since the MPLS functionality is performed on the egress interface. Simply put, existing client facing interfaces need not be MPLS enabled.

Riverstone Label Processing & Values

Riverstone Label Processing & Values

MPLS Packet Walkthrough a Riverstone Router

Before a packet enters an MPLS network the ingress router has to classify the packet. How this classification happens is completely dependant on the type of service. Once the classification has been completed the *Hardware Output Tag Table, or hw-ott-tbl* is consulted to determine the actions that must occur, like the next hop, label value or values that are to be used, before forwarding the packet toward the MPLS core. The hw-ott-tbl is the equitant to the NHLFE table.

When a packet arrives at a transit LSR a DRAM lookup of the inbound label against the *Incoming Label Map, or ILM* returns an index into the hw-ott-tbl. Again, the hardware output tag table contains all the information required to forward the packet in hardware, as above. If no index is returned, then no label exists and the packet is discarded. Both the hw-ott-tbl and the ILM have 4k of table space. The ILM allocates the space in this way, 2k of label space for interface specific labels, used in RSVP-TE environments and 2k for platform wide labels, used in LDP environments. The hw-ott-tbl is a 4k storage space for labels it receives from its peers.

The same process occurs again at the egress LER. A DRAM lookup into the ILM results in an “end-of-tunnel” label. This indicates this router is the egress for the tunnel and must pop the top level label and act on either the original native packet or the lower label. If the encapsulated packet presents a native packet traditional networking, routing or switching, rules will apply. If the encapsulated packet

presents another MPLS label, the label is checked using the DRAM lookup in the ILM and the returned index provides the pointer into to hardware output tag table, where all the necessary information is found on what action to take.

Defining The Labels Space on the RS Platform

The standard has reserved labels 0-15 for specific use. Currently four have been defined.

Label Value	Defined	Meaning
0	Standard	IPv4 explicit null label. When it is the only label entry (i.e., there is no label stacking), it indicates that the label is popped upon receipt. For example, if the LSP is for IPv4 traffic only, the egress router can signal the penultimate router to use 0 as the final hop label.
1	Standard	Router alert label. Packets received with this label value are sent to the CPU for processing.
2	Standard	IPv6 explicit null label. When it is the only label entry (i.e., there is no label stacking), it indicates that the label is popped upon receipt. For example, if LSP is for IPv6 traffic only, the egress router can signal the next to last, or penultimate, router to use 2 as the final hop label.
3	Standard	Implicit null label. Used in LDP or RSVP packet to request the label be popped by the upstream router (penultimate hop label popping). This label should not appear in encapsulation and should not be used in a data packet.
4-15	Standard	Reserved but unused
16	Riverstone	End of Tunnel – Indicates that this is the end of tunnel. The tunnel label is to be removed and the lower level label is examined for further processing

17-2047	Riverstone	RSVP-TE per interface labels (re-used per interface basis for RSVP-TE)
2048-4095	Riverstone	LDP Global label space. LDP labels are tied to loopback interfaces not network prefixes.

There is not restriction on the labels a Riverstone Networks router can place in the output tag table. Nor do these Riverstone specific label definitions create any interoperability issues with any other vendors. Labels are distributed upstream, with respect to the data flow. This means that labels sent upstream only have significance for the downstream router that issued them.

When an MPLS router has a label addressed to it as the destination, label value "0", it is responsible for performing multiple lookups. Through the use if the "P*enultimate Hop Popping", or PHP*, label value 3, the requirement to perform multiple processes need not occur. When the downstream router realizes it is the final node in the label switched path it can distribute the label value of 3 to the upstream at the time the path is established. The upstream router makes the next hop determination for the packet based on the inbound label it received and forwards the packet without a label. This allows the downstream node perform a single lookup. By default, Riverstone Routers will issue a label value of 3 if they represent the end of the tunnel.

It is possible to disable the penultimate hop pop function. WARNING: If an interface is already added to MPLS the signaling protocol must be disabled to make the change from default to no-php.

```
RS(config)# mpls set interface <name> no-php
```

Label Distribution & Management

Label Distribution & Management

Role of the MPLS Label

The MPLS label uniquely identifies the FEC encapsulated within the MPLS packet. A label that is bound to the FEC is used throughout the MPLS network to quickly forward packets. In the case of L2 VPN solutions, an inner label can be used to isolate customers and determine packet handling to all the way to the egress port connecting to a traditional network. Labels have only local signifance.

Distributing MPLS Labels

Labels are distributed between pairs of routers who form a label distribution peering relationship. Depending on the label distribution protocol used, routers can form local, directly connected, or remote, nonadjacent, relationships. It is important to note, *downstream* is relative to the data flow. It is also important to point out that an LSP is unidirectional. This means, that labels can and do travel in each direction across the IP Network but always remain downstream with

relation to the data flow.

Downstream-on-demand: An explicit request for a label binding to an FEC to a next-hop. The reaction of the downstream router to this request depends on the label advertising mode supported on the next hop. This method is typically deployed in explicitly routed MPLS networks using dynamic signaling protocols like RSVP-TE.

Downstream-unsolicited – An LSR may issue a label binding to an FEC without an explicit request from an upstream router. The label binding to FEC is set to all label distribution peers. Typically this is found in environments where explicitly routed requests and end-to-end path signaling are not deployed. Simply put, this is most often seen in Label Distribution Protocol, or LDP, environments.

Label Retention

Liberal Retention – All labels binding to FEC received from label distribution peers are retained even if they does not represent the current active next hop. This method allows quick convergence around link or node failure. Packet based networks are well suited for liberal retention since the label space provides 2^{20} , or in excess of one millions potential labels. Conversely Riverstone does not support, *Conservative Retention,* which will only maintain the label bindings for valid next hops in a label switched path.

Label Advertising

Ordered Label Distribution – This method, as the name would suggest, is a distribution method with structure. A label binding to FEC will not be distributed upstream unless that router has a corresponding label binding to FEC in their table. Should a label binding exist on the downstream node it will respond to the requesting upstream with its label binding to FEC. If no such label binding exists on the downstream node that router must first make a request of their downstream

for a label binding to FEC. Labels are issued from the egress LER into the MPLS cloud in an ordered fashion, until every router has pushed a label binding to FEC back to the farthest-reaching points of the MPLS network. Riverstone supports the ordered label distribution approach since the alternative, *Independent Label Distribution*, could lead to data forwarding across partially established paths, thus creating black holes.

Label Significance

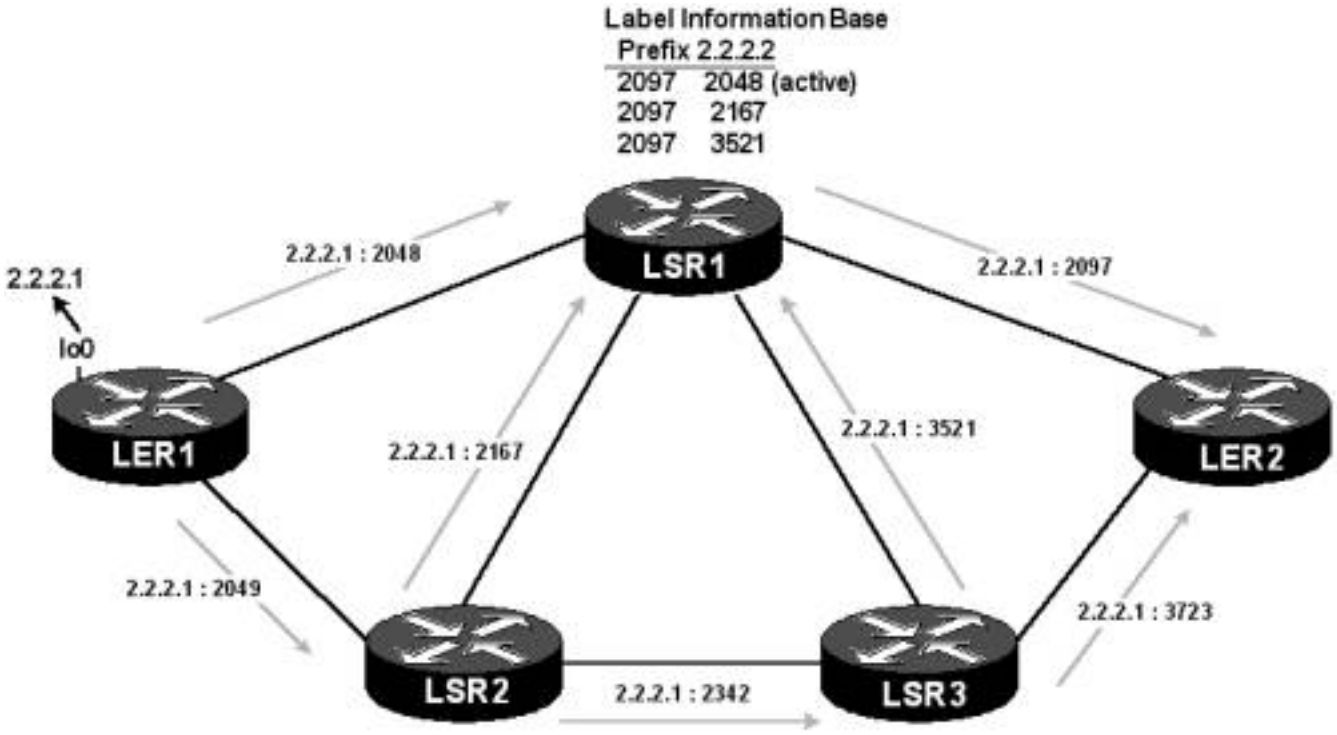
Interface Specific – For environments where it is possible to determine the uniqueness of the label by combining the inbound interface and the label value, interface specific label significance can add scale. This method allows every interface to issue label values that may be issued on other interfaces. There is never a conflict because the interface: label value pair will always result in a unique pointer into the *Next Hop Forwarding Label Entry, or NHFLE*. This approach is applicable when using RSVP-TE. In this case the label binding to FEC will always arrive on the same interface. Riverstone has chosen to implement this strategy when using RSVP-TE to establish paths through an MPLS network.

Platform Wide - For environments that cannot use the interface: label value as a unique pointer into the NHFLE, this approach must be taken. Consider a remote LDP peering session that is tunneled inside some other LSP, a tunnel hierarchy. It may be possible to use interface specific labels for the outer tunnel or trunk. However, it is not possible to use the interface: label value pair for the remote LDP peering session, simply because the remote LDP peering session is not tied to a physical interface specific. The remote LDP peering session is loopback to loopback, thus the packets can arrive on any physical interface.

Demonstrating Different Label Distribution & Management

Although incomplete, the following provides a reasonable representation of how liberal label retention combines with ordered control, downstream-unsolicited

distribution and platform wide label space at work in a basic LDP network. The owner of the loopback address creates a label binding to FEC and that information is distributed to the LDP peers. This information is proliferated though the extreme edges of the network in an ordered and unsolicited downstream fashion. Taking a look at the *Label Information Base, or LIB,* on one of the routers, all labels are retained but only the active next hop is marked as active or inuse.



Preparing The Network for MPLS

Creating The Base Network Support for MPLS

- Introduction
- Creating the Base Network

Enabling MPLS on The RS Platform

- Enabling MPLS

Enabling Dynamic Signaling Protocols

- Choosing the Signaling Protocol
- Enabling RSVP-TE for Traffic Engineering
- Enabling the LDP Protocol

Creating the Base Network Support For MPLS

Creating the Base Network Support for MPLS

Introduction

Signaling protocols are one of the main considerations in determining how to enable MPLS across a network. In order to use a dynamic signaling protocol like RSVP-TE or LDP a routing protocol is certainly required to disseminate IP reachability information, used by the control plane. However, smaller networks may not require the deployment of a routing protocol if label distribution is performed manually with static per hop label configuration, every hop. These static configurations will provide limited scale, so in most if not all networks some type of routing protocol is required. So it is safe to consider a routing protocol a prerequisite in an MPLS network. Once the reachability information is being distributed throughout the network, interfaces that are required to properly interpret and act on MPLS encapsulated packets need to have that support enabled, simply by adding the interface to MPLS. Similarly, any interface that requires support for a dynamic signaling protocol must have the specific signaling protocol added to it in the same way.

Creating The Base Network

The base IP infrastructure underpins the MPLS network. There are numerous routing protocol choices to consider. Focusing on the *Interior Gateway Protocol,*

or IGP, the choice comes down to *Distance Vector* of *Link State* protocols. The older and simpler distance vector approaches have scaling limitations and cannot be nor are they being considered to support traffic engineering extensions. This means the choice is between the two popular and equally capable link state routing protocols, *Open Shortest Path First, or OSPF*, and *Intermediate System to Intermediate System, or IS-IS*.

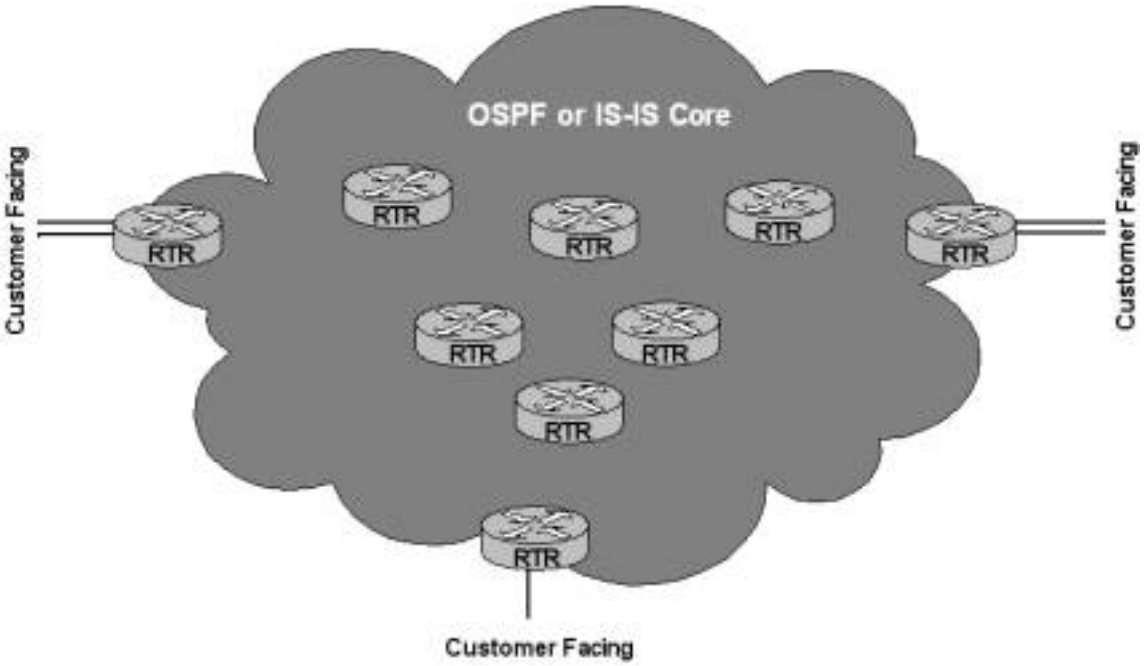
Ensure all the MPLS facing interfaces and the router loopback addresses are in each router's forwarding information base. It is important to note; the router-id must be set to of one of the loopback addresses. This is a pretty well understood and accepted routing practice. However, in the case of LDP signaling it is a requirement. Control session information like hellos and keepalives including those used to establish and maintain remote peering sessions use the router-id as the LSR Id. If no router-id is set, no LDP peering relationships of any kind can be established. If the router-id is set but is one of the loopback interfaces then establishing remote peering sessions are impossible. The following error is generated when no router-id is configured on the router.

```
%CLI-E-FAILED, Execution failed for "ldp add interface lo0"
%MPLS-E-LOADDRESSNOTSAMEASROUTERID, One of the addresses of lo0
must be same as router-id 0.0.0.0 for LDP remote peering
```

The customer specific interfaces, non-MPLS core facing interfaces, are not distributed across the core. This means that the backbone routing tables to not include the customer specific routing information. That's a very good thing! A classification function is executed on the ingress LER that encapsulates customer traffic within the tunnel. This shelters the common core from possible customer conflicts, like IP space or 802.1Q VLAN ID.

The following may represent how all routers with core facing interfaces have the complete link state database representing complete core reachability. The customer facing interfaces are not distributed to the core routers and have only local significance on the router configured with that interface.

See the OSPF Support Page for details on OSPF support.



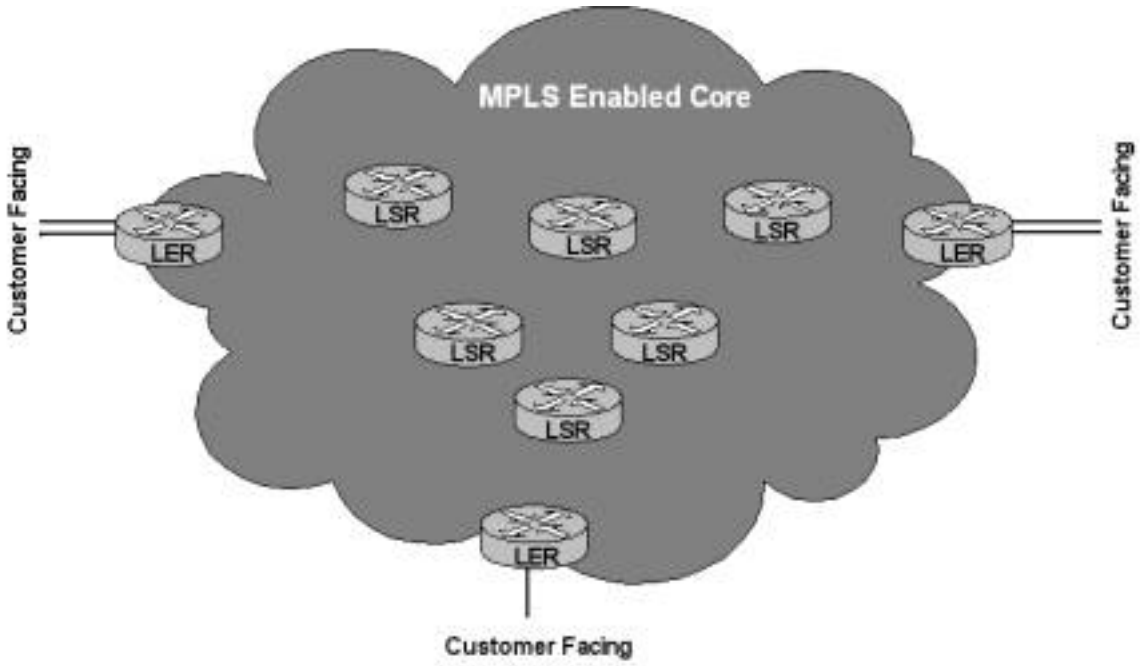
Enabling MPLS on RS Platform

Enabling MPLS on RS Platform

Enabling MPLS

Enabling MPLS

MPLS must be enabled on all interfaces that are required to properly interpret and act on MPLS encapsulated packets. This is simply accomplished by adding the required interfaces to MPLS. This does not mean MPLS should be enabled on any customer facing non-MPLS networks. Although the resources usage would not be abusive to the CPU or to the network bandwidth, enabling any functionality on any interface that will not use an enabled feature is simply a bad practice and serves no purpose. Simply put, any ingress and egress interfaces that act as gateways between an MPLS network and a traditional routed network do not need the MPLS protocol enabled on it.



By their omission, the customer facing are not added to the MPLS network. These customer facing interfaces are not expected to receive nor MPLS encapsulated packets. The MPLS processing is a function of the core facing interface, not part of the customer facing. Therefore, the core facing interface is added to MPLS. To enable the MPLS process on the router, it is required to start the task.

To add an interface to MPLS…

```
rs(config)# mpls add interfaces <name|all>
```

To start the MPLS process on the router

```
rs(config)# mpls start
```

Example configuration of an edge router (LER) connecting to both a traditional IP network and the MPLS core…

```
interface create ip Customer-Facing1 address-netmask
192.168.2.1/30 port et.3.1
interface create ip Customer-Facing1 address-netmask
192.168.2.5/30 port et.15.1
interface create ip Core-Facing address-netmask 192.168.1.1/30
port gi.4.1
interface add ip lo0 address-netmask 2.2.2.1/32
mpls add interface Core-Facing
mpls start
```

Example configuration of a transit LSR with all interfaces contained within the MPLS core…

```
interface create ip Core1-LER1 address-netmask 192.168.1.2/30
port gi.3.1
interface create ip Core1-Core2 address-netmask 192.168.1.9/30
port gi.12.2
interface create ip Core1-Core3 address-netmask 192.168.1.13/30
port gi.11.2
interface add ip lo0 address-netmask 1.1.1.1/32
mpls add interface all
```

mpls start

To display the information specific to MPLS enabled interfaces, including information about state and interface memberships (*AdminGroups*)….

rs(config)# mpls show interfaces <name|all> <options>

```
LSR1# mpls show interfaces all
Interface         State       AdminGroups
lo0                Up          <none>
Core1-LER1         Up          <none>
Core1-Core2        Up          <none>
Core1-Core3        Up          <none>
```

Note: The *lo0* interface, the loopback, is added to MPLS, even without explicit inclusion. As soon as the mpls start command as part of the configuration *lo0* becomes an MPLS enabled interface.

```
LER1# mpls show interface Core1-Core2 verbose
Interface: <Core1-LER1>
    Index: 3 Vlan: 2 Address 802.2 0:0:1d:a3:4e:97  Change: <>
    State: <Up Broadcast Multicast Simplex>
    Refcount: 2     Up-down transitions: 0
     192.168.1.2
            Metric: 32      MTU: 1436
            Refcount: 1     Preference: 0   Down: 120
            Broadcast Address:   192.168.1.3
            Subnet Number: 192.168.1        Subnet Mask:
255.255.255.252

    MPLS active configuration:
    --->proto: <static>     end-of-tunnel-label: 16
        flags: <>
    MPLS saved configuration:
    --->proto: <static>     end-of-tunnel-label: 0
        flags: <>
    Label-map [Config]:
        in-label:
    Label-map [Active]:
        in-label:
```

```
Subscription 100
Incoming Direction:
StaticBW 1000000000bps, AvailableBW 1000000000bps
ReservedBW [0]        0bps [1]        0bps [2]        0bps
[3]        0bps
           [4]        0bps [5]        0bps [6]        0bps
[7]        0bps
Outgoing Direction:
StaticBW 1000000000bps, AvailableBW 1000000000bps
ReservedBW [0]        0bps [1]        0bps [2]        0bps
[3]        0bps
           [4]        0bps [5]        0bps [6]        0bps
[7]        0bps
```

Enabling Dynamic Signaling Protocols

Enabling Dynamic Signaling Protocols

Choosing the Signaling Protocol

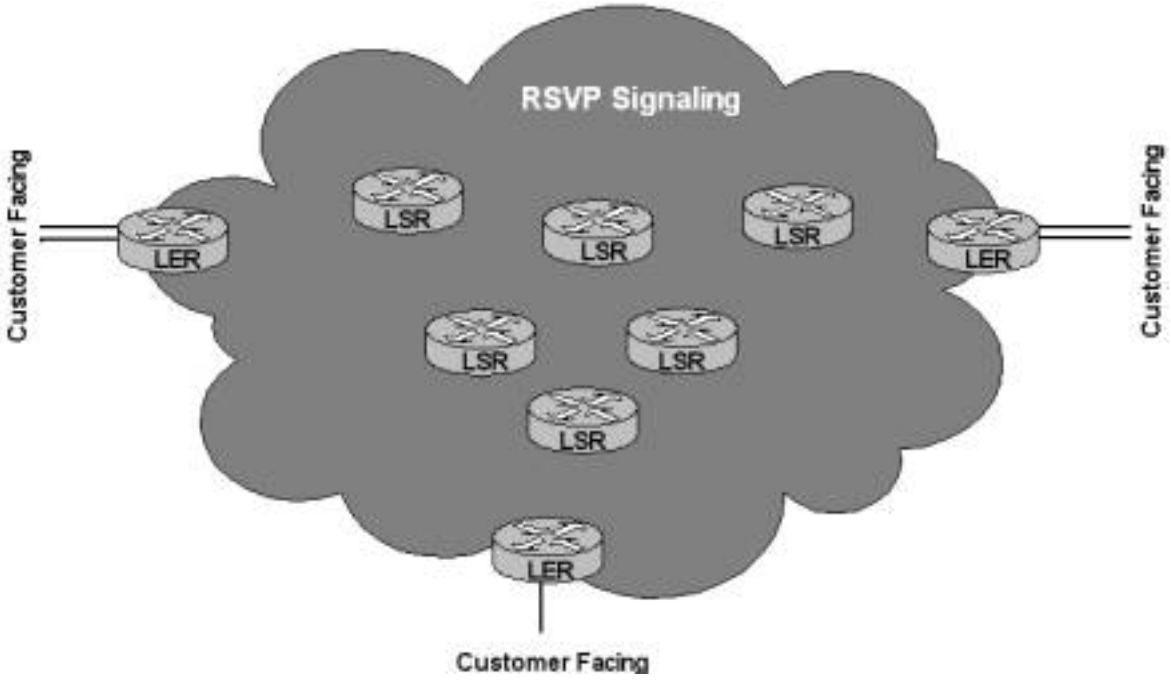
If the environment will use only static label switched paths, these are manually defined paths through the MPLS network complete with label mappings and actions configured at each hop, a signaling protocol is not required, nor is a routing protocol for that matter. However, if dynamic signaling of an LSP is more to your liking one of the two supported signaling protocols is required on all the interfaces that will support the dynamic creation of the LSP. As is true with enabling the MPLS protocol, only enable a signaling protocol on the links that require the functionality.

Before enabling the dynamic signaling protocol on any interface it must be added to MPLS, else the following error message will be displayed on the console and the line in the configuration will error out.

```
%MPLS-E-IFLDPNOTADDED, Interface Core1-Core7 is not added to LDP
%CLI-E-FAILED, Execution failed for "ldp add interface Core1-
Core7"
%MPLS-E-IFMPLSNOTADDED, Interface Core1-Core7 is not added to
MPLS.
```

Note: The RSVP and LDP protocols may NOT be enabled on the same interfaces.

Enabling RSVP-TE For Traffic Engineering



To enable RSVP-TE capabilities on an interface simply add it to the RSVP process..

```
rs(config)# rsvp add interface <name|all>
```

To start the RVSP-TE process on the router…

```
rs(config)# rsvp start
```

Example configuration of an edge router (LER) connecting to both a traditional IP network and the MPLS core…

```
interface create ip Customer-Facing1 address-netmask
192.168.2.1/30 port et.3.1
interface create ip Customer-Facing1 address-netmask
192.168.2.5/30 port et.15.1
interface create ip Core-Facing address-netmask 192.168.1.1/30
port gi.4.1
```

```
interface add ip lo0 address-netmask 2.2.2.1/32
mpls add interface Core-Facing
mpls start
rsvp add interface Core-Facing
rsvp start
```

Example configuration of a transit LSR with all interfaces contained within the MPLS core…

```
interface create ip Core1-LER1 address-netmask 192.168.1.2/30
port gi.3.1
interface create ip Core1-Core2 address-netmask 192.168.1.9/30
port gi.12.2
interface create ip Core1-Core3 address-netmask 192.168.1.13/30
port gi.11.2
interface add ip lo0 address-netmask 1.1.1.1/32
mpls add interface all
mpls start
rsvp add interface all
rsvp start
```

To display the information specific to RSVP-TE enabled interfaces, including information about type, attributes and MTU…

```
rs(config)# rsvp show interfaces <name|all> <options>
```

```
LSR1# rsvp show interface all
Interface              Type       Attrib   Path-MTU
-----------------------------------------------------
Core1-LER1             Enet/POS      <>      1436
Core1-Core2            Enet/POS      <>      1436
Core1-Core3            Enet/POS      <>      1436
```

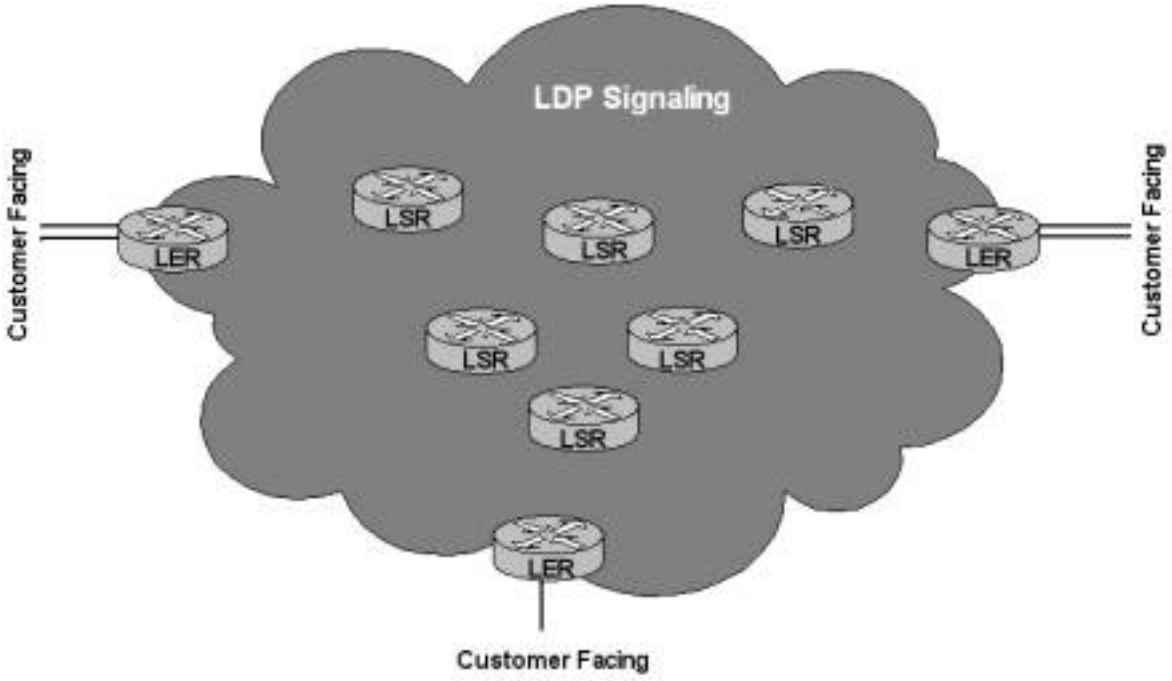
Notice interface lo0, the loopback interface, is not added to the list of RSVP capable interfaces, nor can it be. There is no need to include the loopback interface as an RSVP capable interface because RSVP label peering relationships are formed using physically adjacent interfaces, not loopback interfaces.

A more detailed view of an mpls interface is available by including the option "*verbose*".

```
LER1# rsvp show interface Core1-LER1 verbose
RSVP Interface Configuration:
------------------------------
To-LER1
    type:                       Enet/POS
    attributes:                 <>
    path-mtu:                   1436
    path-vector-limit:          8
    hop-count-limit:            255
    rapid-retransmit-interval: 1000     sec.
    rapid-retransmit-delta:    2        sec.
    rapid-retry-limit:          3
    current-msg-id:             0x0
    epoch:                      0xe51
    seq-no:                     0x0:0x8eee66e5
```

Enabling the Label Distribution Protocol

The Label Distribution Protocol, or LDP, works very differently from the RSVP-TE. In the case of LDP, simply enabling the protocol on the required interfaces will allow the routers to discover directly connected label distribution peers using multicast UDP packets and subsequently establish a peering relationship over TCP. Each router will create and distribute a label binding to FEC for each loopback interface that is defined the router. Each physical interface that is expected to interpret and function in an LDP environment must be added to it.



To enable LDP capabilities on an interface simply add it to the LDP process..

```
rs(config)# ldp add interface <name|all>
```

To start the LDP process on the router….

```
rs(config)# ldp start
```

Example configuration of an edge router (LER) connecting to both a traditional IP network and the MPLS core…

```
interface create ip Customer-Facing1 address-netmask
192.168.2.1/30 port et.3.1
interface create ip Customer-Facing1 address-netmask
192.168.2.5/30 port et.15.1
interface create ip Core-Facing address-netmask 192.168.1.1/30
port gi.4.1
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
mpls add interface Core-Facing
mpls start
ldp add interface Core-Facing
ldp add interface lo0
```

```
ldp start
```

Example configuration of a transit LSR with all interfaces contained within the MPLS core…

```
interface create ip Core1-LER1 address-netmask 192.168.1.2/30
port gi.3.1
interface create ip Core1-Core2 address-netmask 192.168.1.9/30
port gi.12.2
interface create ip Core1-Core3 address-netmask 192.168.1.13/30
port gi.11.2
interface add ip lo0 address-netmask 1.1.1.1/32
ip-router global set router-id 1.1.1.1
mpls add interface all
mpls start
ldp add interface all
ldp start
```

To display the information specific to LDP enabled interfaces, including information about the FEC and label space, number of neighbors with sessiong on that interface and the next time a hello will be generated…

```
rs(config)# rsvp show interfaces <name|all> <options>
```

```
LSR1# ldp show interface all
Interface      Label space      NbrCnt NextHello
lo             1.1.1.1:0          0       0
Core1-LER1     192.168.1.2:0      1       4
Core1-Core2    192.168.1.9:0      1       4
Core1-Core3    192.168.1.13:0     1       4
```

The loopback interface is added automatically when all interfaces are added to LDP as a group. This interface is required to establish remote LDP peering sessions and if the all option is not used to add the interfaces to LDP, the *lo0* interface must be explicitly added.

A more detailed view of an LDP interface is available by including the option

“*verbose*”.

```
LER2# ldp show interface all verbose
Interface             Label space              Nbr count   Next
hello(seconds)
lo                    1.1.1.1:0                    0          0
  Hold time: 15, Liberal label retention, Downstream unsolicited
Core1-LER1          192.168.1.2:0                  1          4
  Hold time: 15, Liberal label retention, Downstream unsolicited
Core1-Core2         192.168.1.9:0                  1          4
  Hold time: 15, Liberal label retention, Downstream unsolicited
Core1-Core3         192.168.1.13:0                 1          4
  Hold time: 15, Liberal label retention, Downstream unsolicited
```

Interior Gateway Protocols in an MPLS Environment

Link State Protocols in an MPLS Environment

- Challenges With Basic Link State Interior Gateway Protocols
- Interior Gateway Protocol Traffic Engineering Extensions

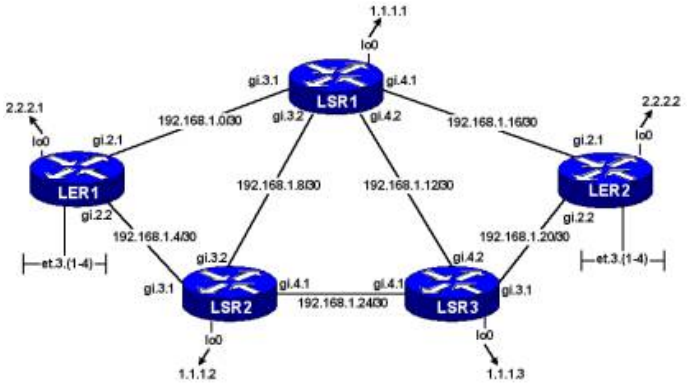
Traffic Engineering Database - Ted

- Interior Gateway Protocol Traffic Engineering Extensions

Manipulating Link Resource Information

- Over Subscription Using Ratios
- Under Subscription Using Ratios
- Manipulating Link Bandwidth Values

Poor Utilization of Network Resources



Sample Network

LER1 Configuration and Forwarding Table

```
vlan create LAN ip id 100
vlan add ports et.3.(1-4) to LAN
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port gi.2.2
interface create ip LAN address-netmask 172.16.1.1/24 vlan LAN
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
system set name LER1
```

LER1# ip show routes

Destination	Gateway	Owner	Netif
1.1.1.1	192.168.1.1	OSPF	To-LSR1
1.1.1.2	192.168.1.5	OSPF	To-LSR2
1.1.1.3	192.168.1.1	OSPF	To-LSR1
	192.168.1.5	OSPF	To-LSR2
2.2.2.1	2.2.2.1	-	lo0
2.2.2.2	192.168.1.1	OSPF	To-LSR1
127.0.0.1	127.0.0.1	-	lo0
172.16.1.0/24	directly connected	-	LAN
192.168.1.0/30	directly connected	-	To-LSR1
192.168.1.4/30	directly connected	-	To-LSR2
192.168.1.8/30	192.168.1.1	OSPF	To-LSR1
	192.168.1.5	OSPF	To-LSR2
192.168.1.12/30	192.168.1.1	OSPF	To-LSR1
192.168.1.16/30	192.168.1.1	OSPF	To-LSR1
192.168.1.20/30	192.168.1.1	OSPF	To-LSR1
	192.168.1.5	OSPF	To-LSR2
192.168.1.24/30	192.168.1.5	OSPF	To-LSR2

_

LSR1 Configuration and Forwarding Table

```
interface create ip To-LER1 address-netmask 192.168.1.1/30 port gi.3.1
interface create ip To-LER2 address-netmask 192.168.1.17/30 port gi.4.1
interface create ip To-LSR2 address-netmask 192.168.1.9/30 port gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.13/30 port gi.4.2
interface add ip lo0 address-netmask 1.1.1.1/32
ip-router global set router-id 1.1.1.1
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.1 to-area backbone cost 10
ospf start
system set name LSR1
```

LSR1# ip show routes

Destination	Gateway	Owner	Netif
1.1.1.1	1.1.1.1	-	lo0
1.1.1.2	192.168.1.10	OSPF	To-LSR2
1.1.1.3	192.168.1.14	OSPF	To-LSR3
2.2.2.1	192.168.1.2	OSPF	To-LER1
2.2.2.2	192.168.1.18	OSPF	To-LER2
127.0.0.1	127.0.0.1	-	lo0
192.168.1.0/30	directly connected	-	To-LER1
192.168.1.4/30	192.168.1.2	OSPF	To-LER1
	192.168.1.10	OSPF	To-LSR2
192.168.1.8/30	directly connected	-	To-LSR2
192.168.1.12/30	directly connected	-	To-LSR3
192.168.1.16/30	directly connected	-	To-LER2
192.168.1.20/30	192.168.1.14	OSPF	To-LSR3
	192.168.1.18	OSPF	To-LER2
192.168.1.24/30	192.168.1.10	OSPF	To-LSR2
	192.168.1.14	OSPF	To-LSR3

_
_

LSR2 Configuration and Forwarding Table

```
interface create ip To-LER1 address-netmask 192.168.1.5/30 port gi.3.1
interface create ip To-LSR1 address-netmask 192.168.1.10/30 port gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.25/30 port gi.4.1
interface add ip lo0 address-netmask 1.1.1.2/32
ip-router global set router-id 1.1.1.2
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.2 to-area backbone cost 10
ospf start
system set name LSR2
```

LSR2# ip show routes

Destination	Gateway	Owner	Netif
1.1.1.1	192.168.1.9	OSPF	To-LSR1
1.1.1.2	1.1.1.2	-	lo0
1.1.1.3	192.168.1.26	OSPF	To-LSR3
2.2.2.1	192.168.1.6	OSPF	To-LER1
2.2.2.2	192.168.1.9	OSPF	To-LSR1
	192.168.1.26	OSPF	To-LSR3
127.0.0.1	127.0.0.1	-	lo0
192.168.1.0/30	192.168.1.6	OSPF	To-LER1
	192.168.1.9	OSPF	To-LSR1
192.168.1.4/30	directly connected	-	To-LER1
192.168.1.8/30	directly connected	-	To-LSR1
192.168.1.12/30	192.168.1.9	OSPF	To-LSR1

```
                    192.168.1.26        OSPF      To-LSR3
192.168.1.16/30     192.168.1.9         OSPF      To-LSR1
192.168.1.20/30     192.168.1.26        OSPF      To-LSR3
192.168.1.24/30     directly connected  -         To-LSR3
```

_
_
_

LSR3 Configuration and Forwarding Table

```
interface create ip To-LER2 address-netmask 192.168.1.21/30 port gi.3.1
interface create ip To-LSR1 address-netmask 192.168.1.14/30 port gi.4.2
interface create ip To-LSR2 address-netmask 192.168.1.26/30 port gi.4.1
interface add ip lo0 address-netmask 1.1.1.3/32
ip-router global set router-id 1.1.1.3
ospf create area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 1.1.1.3 to-area backbone cost 10
ospf start
system set name LSR3
```

LSR3# ip show routes

```
Destination         Gateway             Owner     Netif
-----------         -------             -----     -----
1.1.1.1             192.168.1.13        OSPF      To-LSR1
1.1.1.2             192.168.1.25        OSPF      To-LSR2
1.1.1.3             1.1.1.3             -         lo0
2.2.2.1             192.168.1.13        OSPF      To-LSR1
                    192.168.1.25        OSPF      To-LSR2
2.2.2.2             192.168.1.22        OSPF      To-LER2
127.0.0.1           127.0.0.1           -         lo0
192.168.1.0/30      192.168.1.13        OSPF      To-LSR1
192.168.1.4/30      192.168.1.25        OSPF      To-LSR2
192.168.1.8/30      192.168.1.13        OSPF      To-LSR1
                    192.168.1.25        OSPF      To-LSR2
192.168.1.12/30     directly connected  -         To-LSR1
192.168.1.16/30     192.168.1.13        OSPF      To-LSR1
                    192.168.1.22        OSPF      To-LER2
192.168.1.20/30     directly connected  -         To-LER2
192.168.1.24/30     directly connected  -         To-LSR2
```

_
_

LER2 Configuration and Forwarding Table

```
vlan create LAN ip id 100
vlan add ports et.3.(1-4) to LAN
interface create ip To-LSR1 address-netmask 192.168.1.18/30 port gi.2.1
interface create ip To-LSR3 address-netmask 192.168.1.22/30 port gi.2.2
interface create ip LAN address-netmask 172.17.1.1/24 vlan LAN
interface add ip lo0 address-netmask 2.2.2.2/32
ip-router global set router-id 2.2.2.2
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 2.2.2.2 to-area backbone cost 10
ospf start
system set name LER2
```

LER2# ip show routes

```
Destination         Gateway             Owner     Netif
-----------         -------             -----     -----
1.1.1.1             192.168.1.17        OSPF      To-LSR1
1.1.1.2             192.168.1.17        OSPF      To-LSR1
                    192.168.1.21        OSPF      To-LSR3
1.1.1.3             192.168.1.21        OSPF      To-LSR3
2.2.2.1             192.168.1.17        OSPF      To-LSR1
2.2.2.2             2.2.2.2             -         lo0
127.0.0.1           127.0.0.1           -         lo0
172.17.1.0/24       directly connected  -         LAN
192.168.1.0/30      192.168.1.17        OSPF      To-LSR1
192.168.1.4/30      192.168.1.17        OSPF      To-LSR1
                    192.168.1.21        OSPF      To-LSR3
192.168.1.8/30      192.168.1.17        OSPF      To-LSR1
192.168.1.12/30     192.168.1.17        OSPF      To-LSR1
                    192.168.1.21        OSPF      To-LSR3
192.168.1.16/30     directly connected  -         To-LSR1
192.168.1.20/30     directly connected  -         To-LSR3
192.168.1.24/30     192.168.1.21        OSPF      To-LSR3
```

Traffic Engineering Database - TED

Traffic Engineering Database - TED

Traffic Engineering Database - TED

Traffic Engineering Database - TED

The extension to the IGP allow associate link state database to include the link specific resource information. The information about resource availability and usage is contained in the *Traffic Engineering Database*, or *ted*. The traditional link state database and the new traffic engineering database are used to determine the optimally suited end-to-end path through the network over which to signal and establish the LSP. As is true with the link state database, the traffic engineering database has a consistent view from all the routers in the same area.

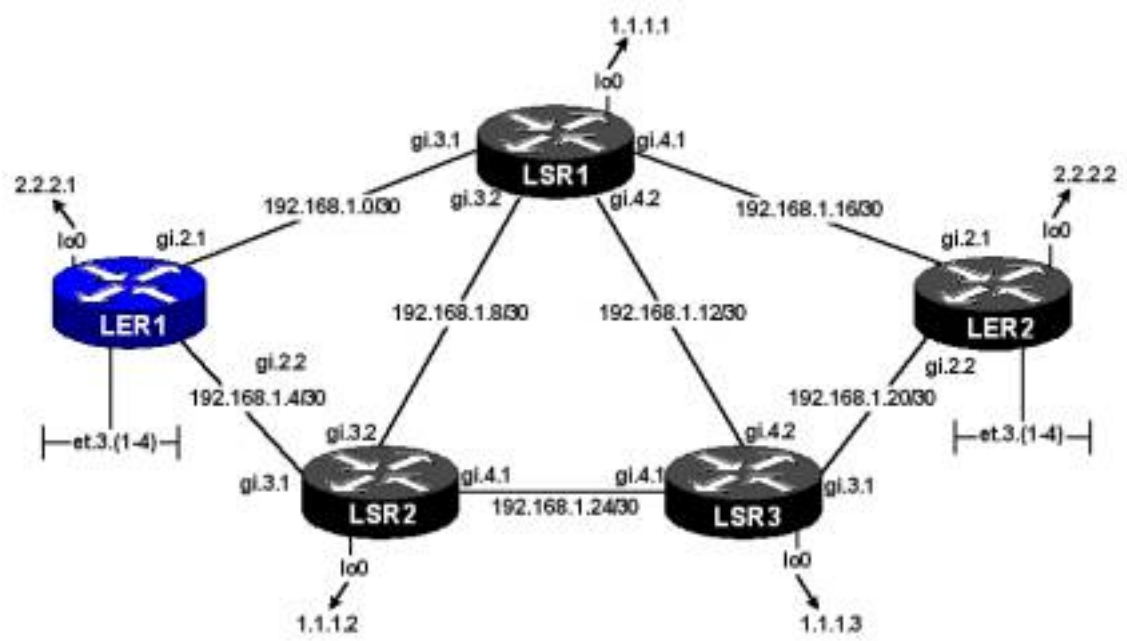
To display the traffic engineering database for OSPF

```
RS(config)# OSPF show ted
```

To display the traffic engineering database for IS-IS

```
RS(config)# ISIS show ted
```

When the traffic engineering capable network is in the base state, defined as no configured or established label switched paths, or any configurations that causes the links to deviate from their defaults, the ted reports the native link resources. All links record the " *Max BW*" as the link speed, "*Max Reservable BW*" equal to the "*Max unreserved BW*" for all label switch paths per priority classes.



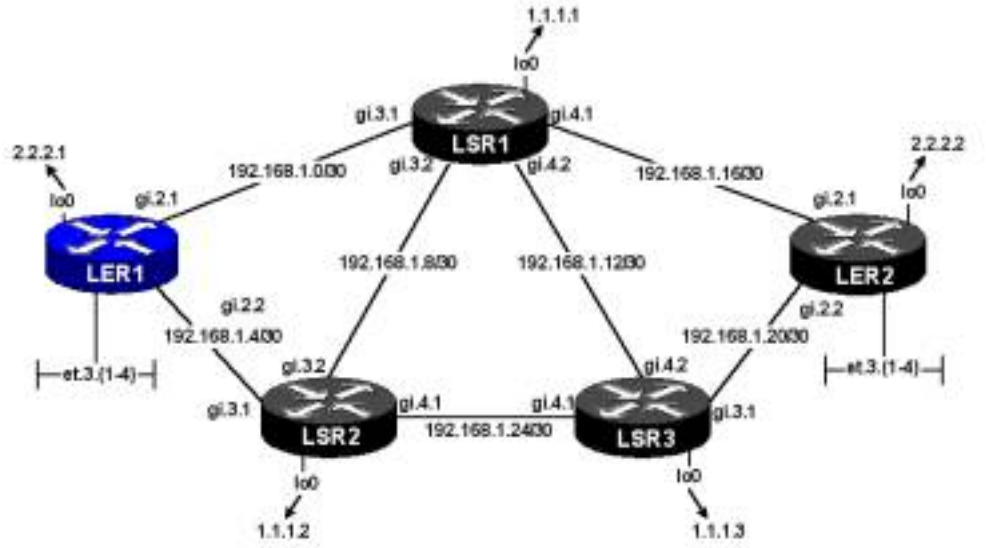
A subset of the traffic engineering database is presented from the perspective of LER1. A brief look at the database reveals which advertising router has flooded the link information and the complete resource information for the advertised link. The entire ted database display from the perspective of LER1 is available here, although the ted is the same on all routers. This means, the traffic engineering database information on one router will match the information contained in the ted of all other routers in the area.

```
LER1# ospf show ted
      OSPF Router with ID (2.2.2.1)

VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.2
                SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
                Remote Addr c0a80102
                Metric 2
                Max BW            :             1 Gbps
```

```
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.18
          (Link Data) Router Interface address: 192.168.1.17
                  SPF data LSID c0a80112 Data c0a80111 Cost 2 Link
type 2
                  Link Type  2
                  Link ID  c0a80112
                  Local Addr  c0a80111
                  Remote Addr c0a80111
                  Metric 2
                  Max BW            :             1 Gbps
                  Max reservable BW :             1 Gbps
                  Max unreserved BW :
                    [0]             1 Gbps  [1]             1
Gbps
                    [2]             1 Gbps  [3]             1
Gbps
                    [4]             1 Gbps  [5]             1
Gbps
                    [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
                Number of TOS metrics: 0
                TOS 0 Metrics: 2
```

The traffic engineering database reflects changes to the "*max unreserved BW*" any time a new LSP is established that includes a bandwidth reservation constraint. Any reservations that cause a deviation of 5% or more of the available bandwidth will result in a triggered link state advertisement to alter the other routers to the resource reduction. The LSA is used to update the ted and Dijkstra's algorithm is not run.

In the example below, a signaled RSVP-TE map is instantiated from ingress LER1 with a destination of egress LER2, with a bandwidth requirement of 750Mbps. Since no path is explicitly specified, the IGP chooses the shortest path to the destination and establishes the LSP across the northerly most path. This simple example is only meant to demonstrate the effects of bandwidth reservations on the ted.



LER1 Configuration

```
vlan create LAN ip id 100
vlan add ports et.3.(1-4) to LAN
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface create ip LAN address-netmask 172.16.1.1/24 vlan LAN
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
```

```
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create label-switched-path To-LER2-1 from 2.2.2.1 to 2.2.2.2
bps 750000000
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
system set name LER1
```

To display detailed information about an LSP on a router…….

RS# mpls show label-switched-paths <name> verbose

```
RS# mpls show label-switched-paths To-LER2-1 verbose
     Label-Switched-Path: "To-LER2-1"
        state: Up                lsp-id: 0x7
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: none
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR BPS>

        Path-Signalling-Parameters:
            attributes: <>
            inherited-attributes: <>
            retry-limit: 5000    retry-int: 15 sec.
            retry-count: 5000    next_retry_int: 0.000000 sec.
            bps: 750000000       preference: 7
            hop-limit: 255       opt-int: 600 sec.
            mtu: 1500
            ott-index: 1         ref-count: 1
            cspf-path:  num-hops: 3
                192.168.1.2      - strict  (LER1 local interface)
                192.168.1.1      - strict  (LSR1 interface)
```

```
                192.168.1.18    - strict  (LER2 interface)
            record-route:
                192.168.1.1
                192.168.1.18
```

The traffic-engineering database reflects the change in resources usage throughout the network. It is important to remember an LSP is unidirectional and the Gigabit Ethernet connections between the routers are full duplex. This means that links advertised in the direction of the LSP will now indicate the new resource information. However, the links same links advertised in the reverse direction have not reduced the *Max unreserved* values. Simply there is currently no complimentary LSP in the reverse direction to enable bi-directional conversations, so the original bandwidth remains available.

The complete updated ted database is available here

Using the same partial list that was presented earlier, the resource information now reflects the links connecting the ingress and egress LER across the northerly most route, in the direction from ingress to egress, has 250Mbps of bandwidth remaining. The new resources information is used by any future CSPF calculations. The resource view stored in as part of the ted is consistent throughout the area.

```
LER1# ospf show ted
          OSPF Router with ID (2.2.2.1)

VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
       Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.2
                SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
                Remote Addr c0a80102
```

```
                Metric 2
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]             250 Mbps  [1]             250
Mbps
                  [2]             250 Mbps  [3]             250
Mbps
                  [4]             250 Mbps  [5]             250
Mbps
                  [6]             250 Mbps  [7]             250
Mbps
            Resource class 0
            Number of TOS metrics: 0
            TOS 0 Metrics: 2

VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
 Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.18
        (Link Data) Router Interface address: 192.168.1.17
                SPF data LSID c0a80112 Data c0a80111 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80112
                Local Addr  c0a80111
                Remote Addr c0a80111
                Metric 2
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]             250 Mbps  [1]             250
Mbps
                  [2]             250 Mbps  [3]             250
Mbps
                  [4]             250 Mbps  [5]             250
Mbps
                  [6]             250 Mbps  [7]             250
Mbps
            Resource class 0
            Number of TOS metrics: 0
```

TOS 0 Metrics: 2

```
LER1# ospf show ted
        OSPF Router with ID (2.2.2.1)

VTX ID RTR:1.1.1.2 [1.1.1.2] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.6
        (Link Data) Router Interface address: 192.168.1.5
                SPF data LSID c0a80106 Data c0a80105 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80106
                Local Addr  c0a80105
                Remote Addr c0a80105
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.26
        (Link Data) Router Interface address: 192.168.1.25
                SPF data LSID c0a8011a Data c0a80119 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8011a
                Local Addr  c0a80119
                Remote Addr c0a80119
                Metric 2
                Max BW             :             1 Gbps
```

```
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.10
          (Link Data) Router Interface address: 192.168.1.10
                SPF data LSID c0a8010a Data c0a8010a Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8010a
                Local Addr  c0a8010a
                Remote Addr c0a8010a
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

VTX ID RTR:2.2.2.2 [2.2.2.2] Flags RtrTLV
```

```
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.22
        (Link Data) Router Interface address: 192.168.1.22
                SPF data LSID c0a80116 Data c0a80116 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80116
                Local Addr  c0a80116
                Remote Addr c0a80116
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.18
        (Link Data) Router Interface address: 192.168.1.18
                SPF data LSID c0a80112 Data c0a80112 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80112
                Local Addr  c0a80112
                Remote Addr c0a80112
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
```

```
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.6
          (Link Data) Router Interface address: 192.168.1.6
                  SPF data LSID c0a80106 Data c0a80106 Cost 2 Link
type 2
                  Link Type  2
                  Link ID  c0a80106
                  Local Addr  c0a80106
                  Remote Addr c0a80106
                  Metric 2
                  Max BW            :             1 Gbps
                  Max reservable BW :             1 Gbps
                  Max unreserved BW :
                    [0]             1 Gbps  [1]             1
Gbps
                    [2]             1 Gbps  [3]             1
Gbps
                    [4]             1 Gbps  [5]             1
Gbps
                    [6]             1 Gbps  [7]             1
Gbps
                  Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.2
          (Link Data) Router Interface address: 192.168.1.2
                  SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
```

```
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
                Remote Addr c0a80102
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

VTX ID RTR:1.1.1.3 [1.1.1.3] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.26
        (Link Data) Router Interface address: 192.168.1.26
                SPF data LSID c0a8011a Data c0a8011a Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8011a
                Local Addr  c0a8011a
                Remote Addr c0a8011a
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
```

```
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2

            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.14
            (Link Data) Router Interface address: 192.168.1.14
                    SPF data LSID c0a8010e Data c0a8010e Cost 2 Link
type 2
                    Link Type  2
                    Link ID  c0a8010e
                    Local Addr  c0a8010e
                    Remote Addr c0a8010e
                    Metric 2
                    Max BW             :             1 Gbps
                    Max reservable BW :              1 Gbps
                    Max unreserved BW :
                      [0]              1 Gbps  [1]              1
Gbps
                      [2]              1 Gbps  [3]              1
Gbps
                      [4]              1 Gbps  [5]              1
Gbps
                      [6]              1 Gbps  [7]              1
Gbps
                    Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2


            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.22
            (Link Data) Router Interface address: 192.168.1.21
                    SPF data LSID c0a80116 Data c0a80115 Cost 2 Link
type 2
                    Link Type  2
                    Link ID  c0a80116
```

```
                Local Addr  c0a80115
                Remote Addr c0a80115
                Metric 2
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]               1 Gbps  [1]               1
Gbps
                  [2]               1 Gbps  [3]               1
Gbps
                  [4]               1 Gbps  [5]               1
Gbps
                  [6]               1 Gbps  [7]               1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
        Link connected to: TRANS Flags

        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.1
                SPF data LSID c0a80102 Data c0a80101 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80101
                Remote Addr c0a80101
                Metric 2
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]               1 Gbps  [1]               1
Gbps
                  [2]               1 Gbps  [3]               1
Gbps
                  [4]               1 Gbps  [5]               1
Gbps
                  [6]               1 Gbps  [7]               1
```

```
Gbps
                Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2

            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.10
            (Link Data) Router Interface address: 192.168.1.9
                SPF data LSID c0a8010a Data c0a80109 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8010a
                Local Addr  c0a80109
                Remote Addr c0a80109
                Metric 2
                Max BW             :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2

            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.14
            (Link Data) Router Interface address: 192.168.1.13
                SPF data LSID c0a8010e Data c0a8010d Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8010e
                Local Addr  c0a8010d
                Remote Addr c0a8010d
                Metric 2
```

```
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.18
        (Link Data) Router Interface address: 192.168.1.17
                SPF data LSID c0a80112 Data c0a80111 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80112
                Local Addr  c0a80111
                Remote Addr c0a80111
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2
```

```
LER1# ospf show ted
        OSPF Router with ID (2.2.2.1)

VTX ID RTR:1.1.1.2 [1.1.1.2] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.6
        (Link Data) Router Interface address: 192.168.1.5
                SPF data LSID c0a80106 Data c0a80105 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80106
                Local Addr  c0a80105
                Remote Addr c0a80105
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.26
        (Link Data) Router Interface address: 192.168.1.25
                SPF data LSID c0a8011a Data c0a80119 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8011a
                Local Addr  c0a80119
                Remote Addr c0a80119
                Metric 2
                Max BW            :             1 Gbps
```

```
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.10
          (Link Data) Router Interface address: 192.168.1.10
                SPF data LSID c0a8010a Data c0a8010a Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8010a
                Local Addr  c0a8010a
                Remote Addr c0a8010a
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

VTX ID RTR:2.2.2.2 [2.2.2.2] Flags RtrTLV
```

```
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.22
        (Link Data) Router Interface address: 192.168.1.22
                SPF data LSID c0a80116 Data c0a80116 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80116
                Local Addr  c0a80116
                Remote Addr c0a80116
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.18
        (Link Data) Router Interface address: 192.168.1.18
                SPF data LSID c0a80112 Data c0a80112 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80112
                Local Addr  c0a80112
                Remote Addr c0a80112
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
```

```
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.6
          (Link Data) Router Interface address: 192.168.1.6
                  SPF data LSID c0a80106 Data c0a80106 Cost 2 Link
type 2
                  Link Type  2
                  Link ID  c0a80106
                  Local Addr  c0a80106
                  Remote Addr c0a80106
                  Metric 2
                  Max BW            :             1 Gbps
                  Max reservable BW :             1 Gbps
                  Max unreserved BW :
                    [0]             1 Gbps  [1]             1
Gbps
                    [2]             1 Gbps  [3]             1
Gbps
                    [4]             1 Gbps  [5]             1
Gbps
                    [6]             1 Gbps  [7]             1
Gbps
                  Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

          Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.2
          (Link Data) Router Interface address: 192.168.1.2
                  SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
```

```
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
                Remote Addr c0a80102
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]            250 Mbps  [1]            250
Mbps
                  [2]            250 Mbps  [3]            250
Mbps
                  [4]            250 Mbps  [5]            250
Mbps
                  [6]            250 Mbps  [7]            250
Mbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

VTX ID RTR:1.1.1.3 [1.1.1.3] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.26
        (Link Data) Router Interface address: 192.168.1.26
                SPF data LSID c0a8011a Data c0a8011a Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8011a
                Local Addr  c0a8011a
                Remote Addr c0a8011a
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]              1
Gbps
                  [2]             1 Gbps  [3]              1
Gbps
                  [4]             1 Gbps  [5]              1
```

```
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2

            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.14
            (Link Data) Router Interface address: 192.168.1.14
                    SPF data LSID c0a8010e Data c0a8010e Cost 2 Link
type 2
                    Link Type  2
                    Link ID  c0a8010e
                    Local Addr  c0a8010e
                    Remote Addr c0a8010e
                    Metric 2
                    Max BW             :             1 Gbps
                    Max reservable BW :              1 Gbps
                    Max unreserved BW :
                      [0]              1 Gbps  [1]              1
Gbps
                      [2]              1 Gbps  [3]              1
Gbps
                      [4]              1 Gbps  [5]              1
Gbps
                      [6]              1 Gbps  [7]              1
Gbps
                    Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2


            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.22
            (Link Data) Router Interface address: 192.168.1.21
                    SPF data LSID c0a80116 Data c0a80115 Cost 2 Link
type 2
                    Link Type  2
                    Link ID  c0a80116
```

```
                Local Addr  c0a80115
                Remote Addr c0a80115
                Metric 2
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]               1 Gbps  [1]               1
Gbps
                  [2]               1 Gbps  [3]               1
Gbps
                  [4]               1 Gbps  [5]               1
Gbps
                  [6]               1 Gbps  [7]               1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
        Link connected to: TRANS Flags

        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.1
                SPF data LSID c0a80102 Data c0a80101 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80101
                Remote Addr c0a80101
                Metric 2
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]               1 Gbps  [1]               1
Gbps
                  [2]               1 Gbps  [3]               1
Gbps
                  [4]               1 Gbps  [5]               1
Gbps
                  [6]               1 Gbps  [7]               1
```

```
Gbps
                Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2

            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.10
            (Link Data) Router Interface address: 192.168.1.9
                SPF data LSID c0a8010a Data c0a80109 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8010a
                Local Addr  c0a80109
                Remote Addr c0a80109
                Metric 2
                Max BW             :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
            Number of TOS metrics: 0
                TOS 0 Metrics: 2

            Link connected to: TRANS Flags
            (Link ID) Designated Router address: 192.168.1.14
            (Link Data) Router Interface address: 192.168.1.13
                SPF data LSID c0a8010e Data c0a8010d Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a8010e
                Local Addr  c0a8010d
                Remote Addr c0a8010d
                Metric 2
```

```
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]             1 Gbps  [1]             1
Gbps
                  [2]             1 Gbps  [3]             1
Gbps
                  [4]             1 Gbps  [5]             1
Gbps
                  [6]             1 Gbps  [7]             1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.18
        (Link Data) Router Interface address: 192.168.1.17
                SPF data LSID c0a80112 Data c0a80111 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80112
                Local Addr  c0a80111
                Remote Addr c0a80111
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             1 Gbps
                Max unreserved BW :
                  [0]           250 Mbps  [1]           250
Mbps
                  [2]           250 Mbps  [3]           250
Mbps
                  [4]           250 Mbps  [5]           250
Mbps
                  [6]           250 Mbps  [7]           250
Mbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2
```

Manipulating Link Resource Information

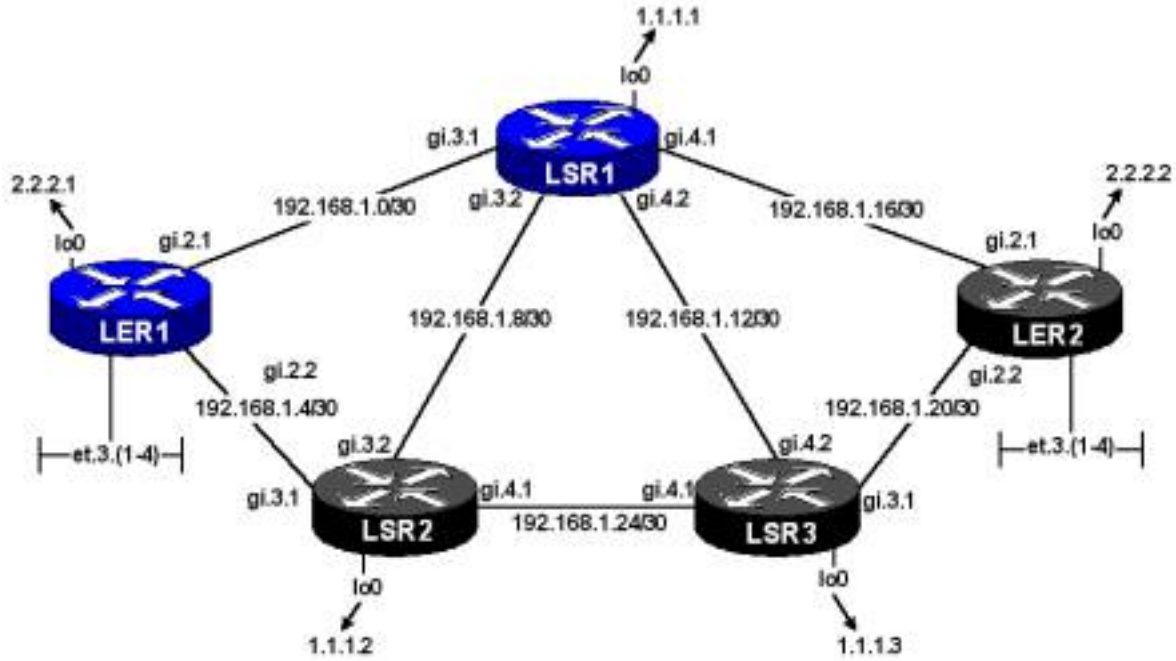
Manipulating Link Resource Information

Over Subscription Using Ratios

By default link resources are advertised for what they are. Though, in many cases allowing, yet controlling, over subscription of link capacity is quite acceptable. Today's traffic patterns tend to be bursty in nature, with the bursts across different segments of the network occurring at different times. This nature may allow providers to over subscribe certain links, all links or no links. These configurations are based on the understanding of traffic patterns, traffic types, service level agreements and risk tolerance..

To set the *Max Reservable Bandwidth* to a value that is not equal to the maximum advertised link bandwidth, or *Max Bandwidth*, the subscription option can be used as part of the mpls set interface command. The ratio is a percentage in the range of <0-64000>, where 100 advertises the maximum reservable bandwidth equal to the link bandwidth. Remember advertisements are unidirectional. This means, if the same physical link has differing subscription rates, the maximum reservable bandwidth across the physical link will not be the same. Changes in the reservable bandwidth in excess of 5% will trigger a link state advertisement, but not the execution of Dijkstra's algorithm.

```
RS(config)# mpls set interface <Name|ALL> subscription <ratio>
```



LER1 Configuration

```
vlan create LAN ip id 100
vlan add ports et.3.(1-4) to LAN
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface create ip LAN address-netmask 172.16.1.1/24 vlan LAN
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls set interface To-LSR1 subscription 200
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
system set name LER1
ospf set traffic-engineering on
```

LSR1 configuration

```
interface create ip To-LER1 address-netmask 192.168.1.1/30 port
gi.3.1
interface create ip To-LER2 address-netmask 192.168.1.17/30 port
gi.4.1
interface create ip To-LSR2 address-netmask 192.168.1.9/30 port
gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.13/30 port
gi.4.2
interface add ip lo0 address-netmask 1.1.1.1/32
ip-router global set router-id 1.1.1.1
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.1 to-area backbone cost 10
ospf start
mpls add interface all
mpls set interface To-LER1 subscription 200
mpls start
rsvp add interface all
rsvp start
system set name LSR1
ospf set traffic-engineering on
```

The ted output shows that both routers are advertising the physical link with a 2 time reservations capacity to maximum bandwidth.

```
LER1# ospf show ted
          OSPF Router with ID (2.2.2.1)
VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.2
                SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
```

```
                Remote Addr c0a80102
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             2 Gbps

                Max unreserved BW :
                  [0]             2 Gbps  [1]             2
Gbps
                  [2]             2 Gbps  [3]             2
Gbps
                  [4]             2 Gbps  [5]             2
Gbps
                  [6]             2 Gbps  [7]             2
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2

VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
                Link connected to: TRANS Flags
          (Link ID) Designated Router address: 192.168.1.2
          (Link Data) Router Interface address: 192.168.1.1
                SPF data LSID c0a80102 Data c0a80101 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80101
                Remote Addr c0a80101
                Metric 2
                Max BW            :             1 Gbps
                Max reservable BW :             2 Gbps
                Max unreserved BW :
                  [0]             2 Gbps  [1]             2
Gbps
                  [2]             2 Gbps  [3]             2
Gbps
                  [4]             2 Gbps  [5]             2
Gbps
                  [6]             2 Gbps  [7]             2
Gbps
```

```
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2
```

Using the mpls show interface command it is possible to look at the incoming and outgoing resources information specific to a single interface, instead of the ted resource dump of the entire database. What you loose is the detailed link state information presented in the ted.

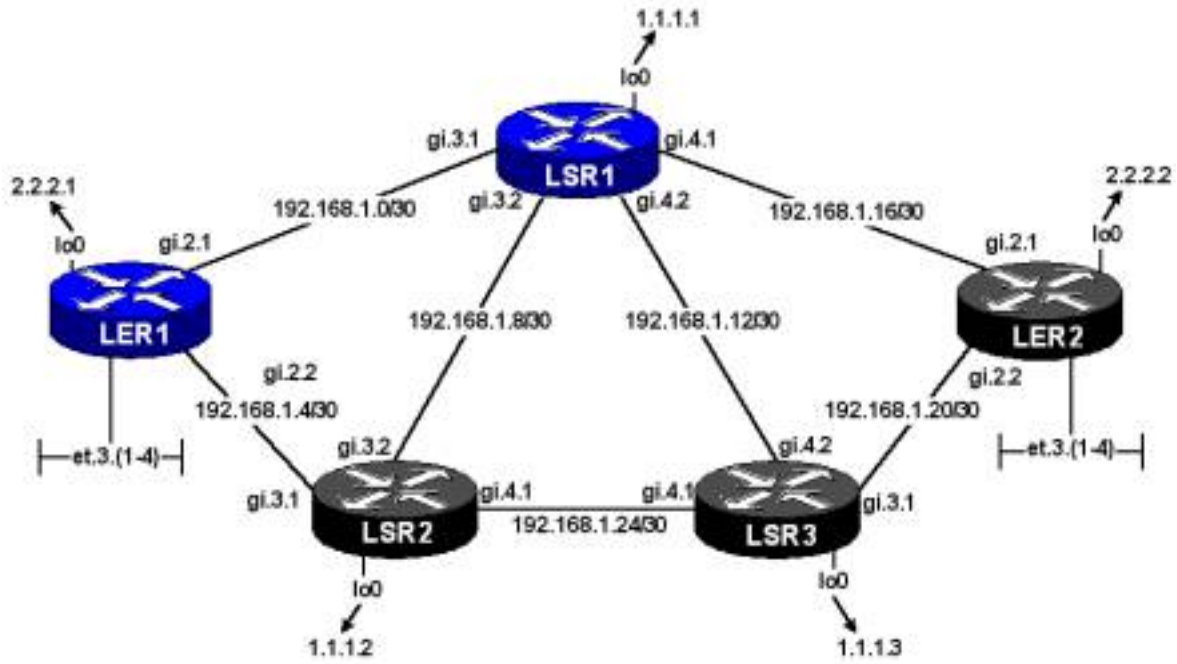
```
LER1# mpls show interface To-LSR1 verbose
Interface: <To-LSR1>
    Index: 3 Vlan: 2 Address 802.2 0:0:1d:a3:4e:97  Change: <>
    State: <Up Broadcast Multicast Simplex>
    Refcount: 2     Up-down transitions: 1
     192.168.1.2
            Metric: 32      MTU: 1436
            Refcount: 1     Preference: 0   Down: 120
            Broadcast Address:   192.168.1.3
            Subnet Number: 192.168.1        Subnet Mask:
255.255.255.252

    MPLS active configuration:
    --->proto: <static rsvp>        end-of-tunnel-label: 16
        flags: <>
    MPLS saved configuration:
    --->proto: <static rsvp>        end-of-tunnel-label: 0
        flags: <>

    Label-map [Config]:
        in-label:
    Label-map [Active]:
        in-label:

Subscription 200
Incoming Direction:
StaticBW 1000000000bps, AvailableBW 2000000000bps
ReservedBW [0]        0bps [1]        0bps [2]        0bps
[3]        0bps
           [4]        0bps [5]        0bps [6]        0bps
```

```
[7]        0bps
Outgoing Direction:
StaticBW 10000000000bps, AvailableBW 2000000000bps
ReservedBW [0]        0bps [1]        0bps [2]        0bps
[3]        0bps
           [4]        0bps [5]        0bps [6]        0bps
[7]        0bps
```

Under Subscription Using Ratios

There may instances where not all the link bandwidth, "*Max BW*", should be made available to the CSPF process. To reduce the available bandwidths used by CPSF specify a ratio of less than 100.

The following example presents both the under subscription capability and unidirectional nature of the command. The two different routers that share a common link advertise different *Max Reservable* vales. Note, the subscription command used in the configuration of LSR1 is meant only as a point of reference. As stated earlier, the default action is to advertise the *Max Reservable* equal to the *Max BW*.



LER1 Configuration

```
vlan create LAN ip id 100
vlan add ports et.3.(1-4) to LAN
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface create ip LAN address-netmask 172.16.1.1/24 vlan LAN
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls set interface To-LSR1 subscription 50
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
system set name LER1
ospf set traffic-engineering on
```

LSR1 configuration

```
interface create ip To-LER1 address-netmask 192.168.1.1/30 port
gi.3.1
interface create ip To-LER2 address-netmask 192.168.1.17/30 port
gi.4.1
interface create ip To-LSR2 address-netmask 192.168.1.9/30 port
gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.13/30 port
gi.4.2
interface add ip lo0 address-netmask 1.1.1.1/32
ip-router global set router-id 1.1.1.1
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR2 to-area backbone
```

```
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.1 to-area backbone cost 10
ospf start
mpls add interface all
mpls set interface To-LER1 subscription 100
mpls start
rsvp add interface all
rsvp start
system set name LSR1
ospf set traffic-engineering on


LER1# ospf show ted
          OSPF Router with ID (2.2.2.1)

VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.2
                SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
                Remote Addr c0a80102
                Metric 2
                Max BW             :             1 Gbps
                Max reservable BW :           500 Mbps
                Max unreserved BW :
                  [0]            500 Mbps  [1]            500
Mbps
                  [2]            500 Mbps  [3]            500
Mbps
                  [4]            500 Mbps  [5]            500
Mbps
                  [6]            500 Mbps  [7]            500
Mbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2
```

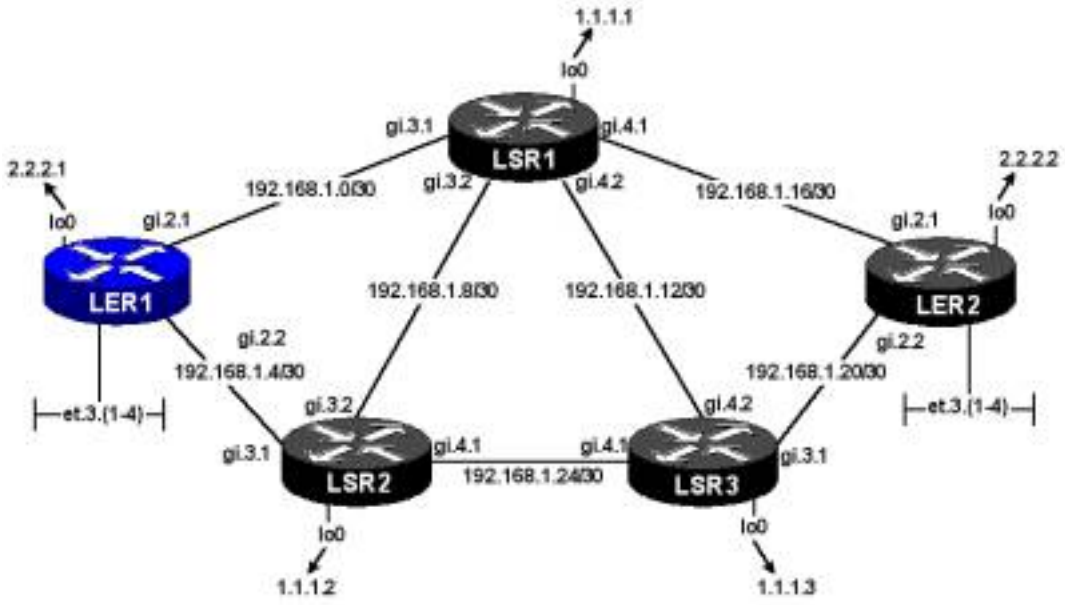
```
VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.1
                SPF data LSID c0a80102 Data c0a80101 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80101
                Remote Addr c0a80101
                Metric 2
                Max BW             :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]              1 Gbps  [1]              1
Gbps
                  [2]              1 Gbps  [3]              1
Gbps
                  [4]              1 Gbps  [5]              1
Gbps
                  [6]              1 Gbps  [7]              1
Gbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2
```

Manipulating Link Bandwidth Values

Riverstone also provides the ability to set the "*Max* BW" information conveyed to the traffic-engineering database less than the actual link speed.

RS(config)# mpls set interface <Name|All> bandwidth <bps>

Altering a links bandwidth in this manner will automatically cause the area or level flooding of the new link state information. There is no restriction for combining this command and the "*Subscription*" command.



_

LER1 configuration

```
vlan create LAN ip id 100
vlan add ports et.3.(1-4) to LAN
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface create ip LAN address-netmask 172.16.1.1/24 vlan LAN
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls set interface To-LSR1 bandwidth 250000000
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
system set name LER1
ospf set traffic-engineering on
```

```
LER1# ospf show ted
        OSPF Router with ID (2.2.2.1)

VTX ID RTR:2.2.2.1 [2.2.2.1] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.2
                SPF data LSID c0a80102 Data c0a80102 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80102
                Remote Addr c0a80102
                Metric 2
                Max BW             :           250 Mbps
                Max reservable BW :           250 Mbps
                Max unreserved BW :
                  [0]            250 Mbps  [1]            250
Mbps
                  [2]            250 Mbps  [3]            250
Mbps
                  [4]            250 Mbps  [5]            250
Mbps
                  [6]            250 Mbps  [7]            250
Mbps
                Resource class 0
        Number of TOS metrics: 0
            TOS 0 Metrics: 2

VTX ID RTR:1.1.1.1 [1.1.1.1] Flags RtrTLV
        Link connected to: TRANS Flags
        (Link ID) Designated Router address: 192.168.1.2
        (Link Data) Router Interface address: 192.168.1.1
                SPF data LSID c0a80102 Data c0a80101 Cost 2 Link
type 2
                Link Type  2
                Link ID  c0a80102
                Local Addr  c0a80101
                Remote Addr c0a80101
                Metric 2
```

```
                Max BW              :              1 Gbps
                Max reservable BW :              1 Gbps
                Max unreserved BW :
                  [0]               1 Gbps  [1]               1
Gbps
                  [2]               1 Gbps  [3]               1
Gbps
                  [4]               1 Gbps  [5]               1
Gbps
                  [6]               1 Gbps  [7]               1
Gbps
                Resource class 0
          Number of TOS metrics: 0
              TOS 0 Metrics: 2
```

Approaches To Signaling in MPLS

Approaches to Signaling in MPLS

The Role of Signaling in MPLS

Signaling is a means by which routers exchange relevant information across the control plane. In an MPLS network, the type of information exchanged between routers depends on the signaling protocol being used. At a base level, labels must be distributed to all MPLS enabled routers that are expected to forward data for a specific FEC. The *Label Distribution Protocol, or LDP*, deals only with this facet of signaling. However, other dynamic signaling protocols, like RSVP-TE and CR-LDP, are capable of signaling and establishing paths through a network to include different types of constraints or reservation of network resources.

Benefits of Signaling Protocols

It is possible to create MPLS enable networks without the aid of signaling protocols, just as it is possible to create IP forwarding networks without a routing protocol. The ability to statically configure each MPLS node with the necessary information for packet forwarding, including in and out labels, actions and next hops are available. This approach is roughly equivalent to using static routes in an IP network. As the networks change, get larger and more complex with increased dynamics, the static approach becomes increasing complex and resource intensive to operate, if it can be maintained at all.

Dynamic signaling protocols have been designed to allow single routers to request the establishment and label binding to FEC for an end-to-end path. The instantiating router simply determines the best path through the network conforming to the local constraints and requests the routers in the in the path to establish a path and distribute the label binding to FEC. Configuring a new LSP, over a core that is MPLS and signaling enabled, does not require anything beyond the configuration in the instantiating router.

Dynamic Signaling Protocols for MPLS

Certainly, MPLS is enhanced by a protocols ability to create a path using a dynamic signaling mechanism. Three signaling protocols are available for use in MPLS networks today.

Label Distribution Protocol – LDP: RFC3036 defines the protocol specifically designed for the distribution of information required to properly interpret label binding to FEC. It does not represent an end-to-end path; rather it is a hop by approach. Riverstone Networks supports LDP, documented under Label Distribution Protocol (LDP).

Resource Reservation Protocol Traffic Extensions – RSVP-TE: The roots of the original RSVP protocol, as defined by RFC2205, have it following the Intserv QoS model. Traditional RSVP is strictly a control plane protocol that is responsible for signaling reservation requests at the micro-flow level between two end stations. The extensions to the original specification result in the MPLS capable signaling protocol, RSVP-TE. This newly extended protocol introduced new ways to allow RSVP-TE to scale in large and complex, MPLS label dependant networks underpinned by IP. At a high level the micro-flow host-to-host reservations were replaced by the aggregation of packets requiring the same treatment into a single FEC, perform label distribution functions and alleviate the soft state concerns of a protocol that does not use a reliable transport. RSVP-TE runs directly over IP using protocol number 46. Riverstone Networks supports RSVP-TE, documented in Resource Reservation Protocol - Traffic Engineering (RSVP-TE).

Constraint-based Routed Label Distribution Protocol – CR-LDP: Extensions to the LDP protocol to incorporate traffic engineering capabilities.

Both RSVP-TE and CR-LDP share a strikingly similar functional set.

Comparing RSVP-TE & CR-LDP

A basic comparison for the two end-to-end TE based signaling methods.

	RSVP-TE	**CR-LDP**[2]
History	Extensions to RFC2205 RSVPv1	Extensions to RFC3036 LDP
Transport	IP Protocol 46	UDP 646 (hello & discovery) TCP 646 (Adjacency)
Session State	Soft – Refresh Reduction	Hard – TCP
Recovery from failure	No TCP connection to maintain across nodes	Loss of TCP connection causes loss of LSP
Reservation Direction	Reverse Direction	Forward Direction
Call admission Control	Yes – Reverse Direction	Yes – Forward Direction
Signaling Traffic Parameters	Bandwidth, link affinity	Bandwidth, link affinity
Explicit Route	Loose & Strict	Loose & Strict
Route Record Object	Yes	Yes
Route Pinning	Yes[2]	Yes
Policing	Yes[1]	Yes
Label Distribution Methods	Downstream-on-demand	Downstream-on-demand
Fast Reroute	Yes[1]	Yes
Pre-emption	Setup & Hold Priorities	Setup & Hold Priorities
Route Re-optimization	Yes[1]	Yes
Security	Yes – MD5	TCP connection & MD5

1 – RS future support
2 – Under consideration for future

Label Distribution Protocol (LDP)

Introduction To LDP

- History
- Intent of LDP

LDP Peering

- Label Distribution Peers
- Establishing Peering Sessions

Exchanging Label Binding to FEC

- Label Management
- Creating & Distributing Label Binding to FEC
- Peer Exchange of Labels
- Making Sense of the LDP Database
- Mapping the Active Next-Hop to a Label

Timers & Session Maintenance

- Hellos & Keepalives
- Default Timer Values
- Configuring Timer Values

Loop Detection

- **Loop Detection**

Authenticating LDP Sessions

- **MD5 Signature Authentication**

LDP Show Commands

- **LDP Show Commands**

Resource ReSerVation Protocol – Traffic Engineering (RSVP-TE)

Signaling Different Types of RSVP-TE Paths

- Extending the RSVP for MPLS Networks
- Hop-by-Hop
- Explicit Route Object - ERO

RSVP-TE Record Route Object - RRO

- Record Route Object

How an RSVP-TE LSP is Signaled

- Path and RESV Messages
- Message Flow During Path Creation
- Resource Reservation Styles

RSVP-TE Refresh Overhead Reduction

- Without Refresh Reduction Techniques
- RSVP-TE Refresh Overhead Reduction
- RSVP-TE Bundle Message
- Summary Refresh
- Hello Messages

- Deploying Refresh Reduction Techniques

RSVP-TE Phases & States

- RSVP-TE Phase & State Flags
- Path Determination Phase
- CSPF Failure Messages
- Signaling Phase

Path Protection Using RSVP-TE

- Designating Backup Paths
- Fast Re-route and Detour LSP

Path Pre-emption and Priorities

- Path Pre-emption & Priorities

Authenticating RSVP-TE Sessions

- Authentication
- Enabling MD5 Authentication

Introduction to Constraint Based Routing

Configuring Link Affinity (Admin Groups)
Configuring Bandwidth Constraints
Configuring Hop Count Constraint

RSVP-TE Show Commands

- **RSVP-TE Show Commands**

Introduction to LDP

Introduction to LDP

History

Standards track protocol defined by RFC3036

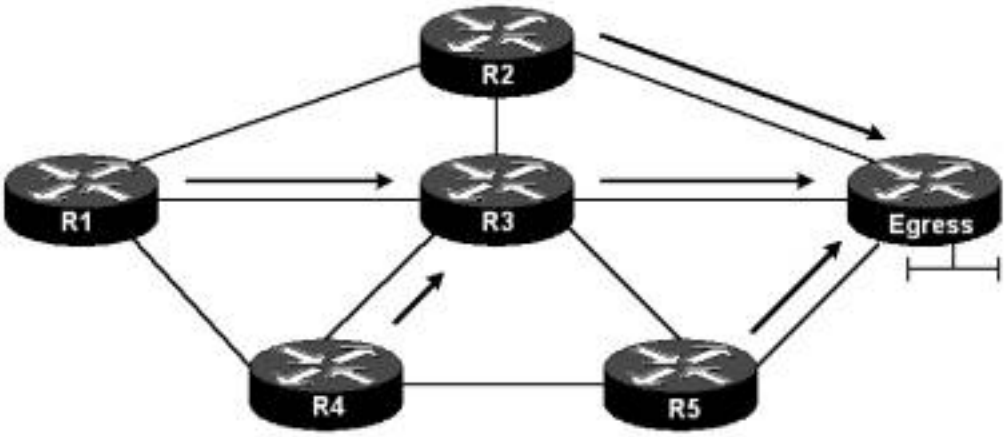
LDP applicability statement RFC3037

Intent of LDP

The *Label Distribution Protocol, or LDP,* was written specifically to perform label distribution in MPLS environments. The prefix to label bindings is stored in the *Label Information Base, or LIB.* Utilizing liberal label retention a FEC may map to many possible labels, one per peer. It relies on the underlying routing information to select the active next hop and associates that gateway to an appropriate outbound label. Established LDP peers are free to exchange prefix and label binding information with each other. Unlike traffic engineered paths, which use constraints and explicit routes to establish end-to-end LSPs, LDP has no such concept. The *Forwarding Information Base, or FIB,* is responsible for determining the hop-by-hop path through the network. Since LDP issues labels from the global label space, it issues the same label binding to FEC out all interfaces that will use the same outbound label.

The example below shows router sending MPLS labeled traffic toward a destination. R3 has been selected as the active next hop for both R1 and R4. It

merges the label it receives on the two different interfaces to a single outbound interface.



This rather dynamic protocol automatically discovers local peers forming relationships automatically. These peers exchange label binding to FEC using the *LDP Identifier* (router-id). In fact all loopback addresses on an RS platform will have a label binding to FEC passed to the LDP peers. Remember, LDP relies on the underlying routing information in order to forward label packets. If a loopback address and LDP identifier are configured on an LSR but no routing information exists for this prefix the local LDP peering session will be established but the exchange of label binding to FEC will not proceed normally, due to the lack of routing knowledge to reach the loopback address.

A natural application for LDP is IBGP core offload. The requirement to run an IBGP full mesh or route reflectors in the core of the network when EBGP peers are separated by an IGP is removed if LDP is deployed in the core of the network. Simply configuring LDP inside the core of the network and distributing the external routes to other routers that require the entire Internet routing table over IBGP, the core routers only have to understand which label to apply to reach the peering routers. There is no need for the internal routers to maintain the entire Internet route table. For complete detailed configuration information refer to the MPLS section under “Configuring L3 Label Switched Paths – BGP Traffic over an LSP Configuration Example” in the RS Switch Router User Guide v 8.0.

LDP is also a key enabler of L2 Virtual Private Networks and Transparent LAN Services.

LDP Peering

LDP Peering

Label Distribution Peers

There are two types of peering relationships in a LDP environment, *local peer* and *remote peer*.

Local Peer: LDP nodes that form this relationship are directly connected, reachable across a common link. Direct adjacencies are automatically discovered using hello multicasts to the “all routers on this subnet” group using UDP port 646. Routers on that receive the hello message may form a local peering relationship using TCP port 646. The hello generation of the hello process is automatic on all LDP enabled interfaces no configuration is necessary. This is the *Basic Discovery* mechanism.

Remote Peer: LDP nodes that do not share link level connectivity must be specifically configured to for a peering relationship. This method uses the same hello message format as the local peers except it is a directed hello, unicast message specifically address to the configured remote peer address. Remote peers play a key role in an LSP hierarchy, where many virtual channels exist within shared tunnels. The ability to create these LDP remote peering sessions are key to scaling L2 Virtual Private Networks and Transparent LAN Services across and MPLS core.

Establishing Peering Sessions

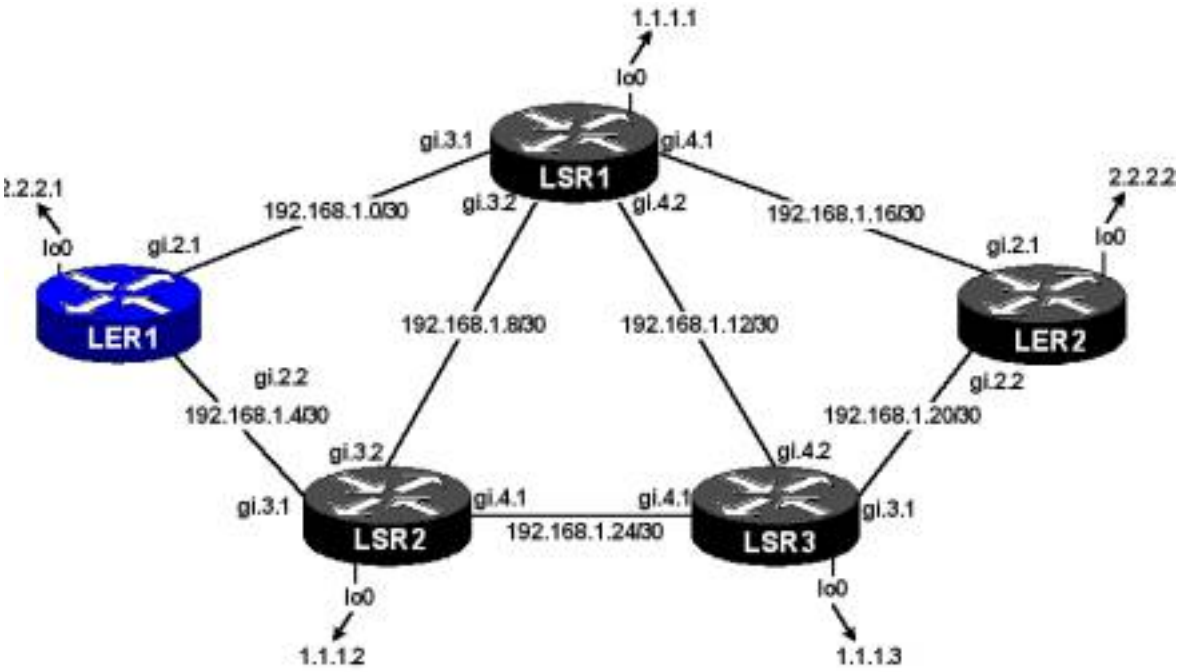
Receipt of a hello message is either accepted by default, unless explicit local policy forbids it. Assuming the hello is accepted, an acknowledgement is returned to the originator using either the source IP address of the original hello or the embedded *Transport Object TLV*. The RS platform does not set the transport object TLV but will accept it and respond according to the object contents if it is included in the hello message.

Once each router accepts the other as a possible peer, the router with the higher IP address will take the active role in establishing the TCP session over which all negotiations and LDP session information will flow. If either potential peers is unable to negotiate parameters that are suitable for their specific needs the LDP session will not complete, with the peer that terminates the session issuing a *Session Reject* message. The potential peers negotiate label distribution methods, protocol version, timers and the like.

An *LDP Identifier* is used to uniquely identify LDP nodes in the network. The identifier is derived from the *router-id* and the label space the label was issued from (two bytes). The label space used for LDP sessions in the Riverstone Networks implementation is global and therefore the value of both the last two bytes is always "0:0". The global space is represented as a single "0" in most display commands. When using the LDP protocol the router-id must also exist as an IP interface associated with loopback interface. The sample configuration demonstrates how to add an IP address to the loopback interface, set the router-id and inject the loopback address into an OSPF area. Obviously the example below is incomplete and is meant only to demonstrate the requirements mentioned above.

```
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf add stub-host 2.2.2.1 to-area backbone cost 10
```

To demonstrate messaging from basic discovery through and session establishment a new router, *LER1* is configured and connected to the network as depicted below.




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls start
ldp add interface To-LSR1
ldp add interface To-LSR2
ldp start
```

The new router will establish local LDP peering relationships with the two

directly connected label switched routers. Examination of packets exchange between peers locally connected over network 192.168.1.0/30, 2.2.2.1:0-1.1.1.1:0, demonstrates the basic discovery and session establishment phase.

Send hello discovery

```
LDP Send Link Hello msg on Interface To-LSR1
Our LDP Id: 2.2.2.1:0, Hold time: 15 seconds
```

Receive hello discovery

```
LDP Receive Link Hello msg on Interface To-LSR1
Peer LDP Id: 1.1.1.1:0, Hold time: 15 seconds
```

Hello adjacency formed with local peer using global label space

```
LDP1(if-3): New hello adjacency 1.1.1.1.0.0 lblSpace(0)
```

Initialize TCP connection and session parameters (LER1 active - higher IP)

```
LDP Send Initialization msg on Interface To-LSR1, Link Neighbor
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
LDP version: 1, Keepalive Time : 30 seconds, Downstream-
Unsolicited
Max PDU Length: 4096, Receiver LDP identifier: 1.1.1.1:0
```

Opened TCP connection and session parameters negotiated with peer

```
LDP Receive Initialization msg on Interface To-LSR1, Link
Neighbor
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
LDP version: 1, Keepalive Time : 30 seconds, Downstream-
Unsolicited
Max PDU Length: 4096, Receiver LDP identifier: 2.2.2.1:0
```

Keepalives ensure protocol status

```
LDP Send Keepalive msg on Interface To-LSR1, Link Neighbor
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0

LDP Receive Keepalive msg on Interface To-LSR1, Link Neighbor
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
```

Hello messages confirm local reachbility

```
LDP Send Link Hello msg on Interface To-LSR1
```

```
Our LDP Id: 2.2.2.1:0, Hold time: 15 seconds

LDP Receive Link Hello msg on Interface To-LSR1
Peer LDP Id: 1.1.1.1:0, Hold time: 15 seconds
```

The session establishment process is the same regardless of peer type, local or remote. The only difference is which type of discovery mechanism that will be used, basic or extended, and the configuration sets requirements to complete generate the extended discovery process for remote peering sessions.

The hello multicast used to automatically discover local peers is issued using a source IP of the physical interface local to the subnet. The LDP identifier in the hello is the router-id. In the case of local peering, the loopback address need not be added to the list of LDP capable interfaces. The LDP session is established between the physical interfaces. However, when for a remote peering session the directed unicast hello uses the local lo0 address as the source IP address and the target hello is destined for the remote loopback. This means the lo0 interfaces must be added to the list of LDP capable interfaces or the session will not establish. The LDP identifier remains the router-id. Remember, there could be many addresses coded on the loopback interface to meet various operational requirements.

To add the loopback interface to the list of LDP capable interfaces…

```
RS(config)# ldp add interface lo0
```

To create the remote peering session…

```
RS(config)# ldp add remote-peer <IP of peer>
```

To create a remote peering session between the two label edge routers…

```
ldp add interface lo0
ldp add remote-peer 2.2.2.2
```

The only changes to the messaging are in the hello packets…

Send target hello

```
LDP Send Target Hello msg on Interface lo to remote peer 2.2.2.22
Our LDP Id: 2.2.2.1:0, Hold time: 15 seconds
```

Receive target hello

```
LDP Receive Target Hello msg on Interface lo from remote peer
2.2.2.22
Peer LDP Id: 2.2.2.2:0, Hold time: 15 seconds
```

Exchanging Label Binding to FEC

Exchanging Label Binding to FEC

Label Management

LDP uses ordered control with downstream-unsolicited label distribution to exchange label binding to FEC between peers. The RS platform implements liberal label retention to improve recovery times to failure conditions.

Creating & Distributing Label Binding to FEC

By default, the RS platform only distributes label binding to FEC for the loopback interfaces, all other interfaces would have to be manual distributed.

To distribute a label binding to FEC for a prefix other than the loopback interface the local router must be configured with a router a route-map to identify the prefix and an LDP egress-policy that reference that policy. Also the must be information in the FIB for the prefix. If there is no information in the FIB the label binding to FEC is not advertised, or if the prefix disappears from the FIB the

label binding to FEC is withdrawn.

Assuming the terminating interface is on the local router the interface must be active for an entry to exist in the FIB. To create and distribute a label binding for prefix 9.9.9.0/24 and distribute it to local peers…

```
interface create ip Net9 address-netmask 9.9.9.1/24 port et.3.8
route-map Net9 permit 10 network 9.9.9.0/24 match-prefix
ldp set egress-policy route-map Net9 sequence 10
```

Without an active next hop in the local peers FIB, they will accept the label mapping for prefix 9.9.9.0/24 but will not distribute a label binding for it. There are many ways to inject a route into a network, added it to the routing protocol, use a route-map or simply redistribute it.

Using a simple redistribution command…

```
ip-router policy redistribute from-proto direct network 9.9.9.0/24
exact to-proto ospf
```

Peer Exchange of Labels

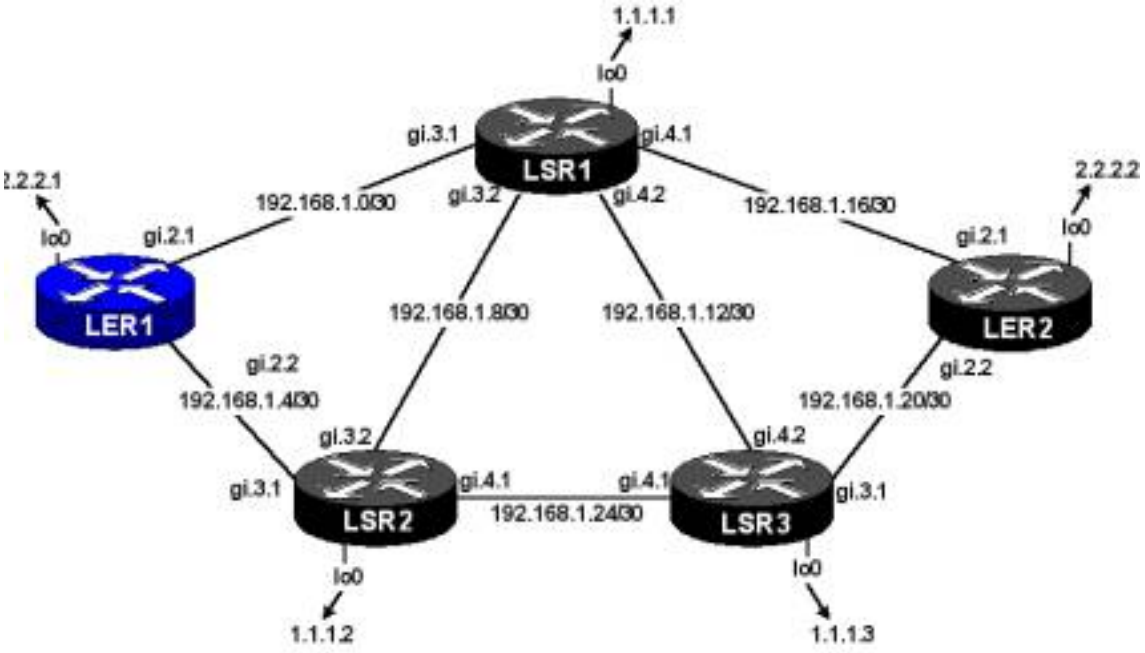
As routers receive new label binding to FEC as part of the *Label Mapping* message they are stored in the *Label Information Base, or LIB.* For each label binding to FEC received inbound one is created and distributed outbound to all LDP peers, including the advertisement of a label binding to FEC back toward the originator. The label mapping message consists of a *FEC TLV* and a *Label TLV*, with a *Message ID* acting as a uniquely identifies for the message.

FEC TLV: This objects object includes the prefix information that defines the forwarding equivalence class.

Label TLV: Contains the 20 bit label value that to be used when forwarding

packets belonging to the FEC to a particular router.

Label binging to FEC information can be distributed as soon as the session has bee established. In the network diagram below assume the new router has completed the session establishment with the to local peers. The information exchange is monitored from the perspective of the new router. It is necessary to show the process for each local peer in order to get a complete understanding of how the label exchange is accomplished. One important thing to notice below is that no label to label binding to FEC (prefix 1.1.1.2) until it is sent from LER1-LSR1 until the new router receives the label binding to FEC from the active next hop for that prefix, LSR2.




```
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
```

```
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls start
ldp add interface To-LSR2
ldp add interface To-LSR1
ldp start
system set name LER1
ospf set traffic-engineering on
```

The new router will exchange LDP capable interface information followed by the label exchange. They gray italicized notations on the right indicate the direction of the exchange. Two sessions have been established one to each local peer, where the LDP ID is the router-id.

LER1-LSR1 (2.2.2.1:0-1.1.1.1:0) : Exchange LDP Interface Addresses

```
LDP Send Address msg on Interface To-LSR1, Link Neighbor (LER1-LSR1)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Addresses: 192.168.1.6

LDP Send Address msg on Interface To-LSR1, Link Neighbor (LER1-LSR2)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Addresses: 192.168.1.2

LDP Receive Address msg on Interface To-LSR1, Link Neighbor (LSR1-LER2)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Addresses: 192.168.1.17

LDP Receive Address msg on Interface To-LSR1, Link Neighbor (LSR1-LSR3)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Addresses: 192.168.1.13

LDP Receive Address msg on Interface To-LSR1, Link Neighbor (LSR1-LSR2)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
```

Addresses: 192.168.1.9

LDP Receive Address msg on Interface To-LSR1, Link Neighbor *(LSR1-LER1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Addresses: 192.168.1.1

LER1-LSR1 (2.2.2.1:0-1.1.1.1:0) : Exchange Label→FEC Binding
LDP Receive Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LER2)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2051
 FEC: IP 2.2.2.2/32

LDP Send Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LER2)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2048
 FEC: IP 2.2.2.2/32

LDP Receive Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LSR3)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2050
 FEC: IP 1.1.1.3/32

LDP Send Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LSR3)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2049
 FEC: IP 1.1.1.3/32

LDP Receive Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LSR2)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2049
 FEC: IP 1.1.1.2/32

LDP Receive Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LSR1)*

Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 3
 FEC: IP 1.1.1.1/32

2001-08-31 22:15:08 LDP Send Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LSR1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2050
 FEC: IP 1.1.1.1/32

2001-08-31 22:15:08 LDP Send Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LER1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 3
 FEC: IP 2.2.2.1/32

2001-08-31 22:15:08 LDP Receive Label Mapping msg on Interface To-LSR1, Link Neighbor *(lo0 of LER1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2048
 FEC: IP 2.2.2.1/32

LER1-LSR2 (2.2.2.1:0-1.1.1.2:0) : Exchange LDP Interface Addresses
LDP Send Address msg on Interface To-LSR2, Link Neighbor *(LER1-LSR2)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Addresses: 192.168.1.6

LDP Send Address msg on Interface To-LSR2, Link Neighbor *(LER1-LSR1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Addresses: 192.168.1.2

LDP Receive Address msg on Interface To-LSR2, Link Neighbor *(LSR2-LSR3)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Addresses: 192.168.1.25

LDP Receive Address msg on Interface To-LSR2, Link Neighbor *(LSR2-*

LSR1)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Addresses: 192.168.1.10

LDP Receive Address msg on Interface To-LSR2, Link Neighbor *(LSR2-LER1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Addresses: 192.168.1.5

LER1-LSR2 (2.2.2.1:0-1.1.1.1:0) : Exchange Label→FEC Binding
LDP Send Label Mapping msg on Interface To-LSR2, Link Neighbor *(lo0 of LSR1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2050
 FEC: IP 1.1.1.1/32

LDP Receive Label Mapping msg on Interface To-LSR2, Link Neighbor *(lo0 of LSR1)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2048
 FEC: IP 1.1.1.1/32

LDP Send Label Mapping msg on Interface To-LSR2, Link Neighbor *(lo0 of LSR3)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2049
 FEC: IP 1.1.1.3/32

LDP Receive Label Mapping msg on Interface To-LSR2, Link Neighbor *(lo0 of LSR3)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2049
 FEC: IP 1.1.1.3/32

LDP Send Label Mapping msg on Interface To-LSR2, Link Neighbor *(lo0 of LER2)*
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2048
 FEC: IP 2.2.2.2/32

```
LDP Receive Label Mapping msg on Interface To-LSR2, Link Neighbor (lo0 of LER2)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2051
 FEC: IP 2.2.2.2/32

LDP Receive Label Mapping msg on Interface To-LSR2, Link Neighbor (lo0 of LSR2)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 3
 FEC: IP 1.1.1.2/32

LDP Send Label Mapping msg on Interface To-LSR1, Link Neighbor (lo0 of LSR2 advert LER1 to LSR1)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.1:0
Label: 2051
 FEC: IP 1.1.1.2/32

LDP Send Label Mapping msg on Interface To-LSR2, Link Neighbor (lo0 of LSR2)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2051
 FEC: IP 1.1.1.2/32

LDP Send Label Mapping msg on Interface To-LSR2, Link Neighbor (lo0 of LER1)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 3
 FEC: IP 2.2.2.1/32

LDP Receive Label Mapping msg on Interface To-LSR2, Link Neighbor (lo0 of LER1)
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 1.1.1.2:0
Label: 2050
 FEC: IP 2.2.2.1/32
```

Making Sense of the LDP Database

Presenting the LDP database for the peering relationships between these three

routers summarizes the events that occurred above. The color-coding indicates the table relationships between the peers, with common fonts relating one tables Input to the other tables output. When viewing the signaling protocol label information base, it is important to realize the *Input Label Database* represents labels received into the local router from a peer. Where as, the *Output Label Database* are labels the local routers has sent to its peer. View it from the perspective of the signaling protocol.

LER1# ldp show database

Input label database, 2.2.2.1:0-1.1.1.1:0

Label	Prefix
2048	2.2.2.1/32
2049	1.1.1.2/32
2050	1.1.1.3/32
3	1.1.1.1/32
2051	2.2.2.2/32

Output label database, 2.2.2.1:0-1.1.1.1:0

Label	Prefix
2048	2.2.2.2/32
2049	1.1.1.3/32
2050	1.1.1.1/32
3	1.1.1.2/32
2051	2.2.2.1/32

LSR1# ldp show database

Input label database, 1.1.1.1:0-2.2.2.1:0

Label	Prefix
2048	2.2.2.2/32
2049	1.1.1.3/32
2050	1.1.1.1/32
2051	1.1.1.2/32
3	2.2.2.1/32

Output label database, 1.1.1.1:0-2.2.2.1:0

Label	Prefix
2048	2.2.2.1/32
2049	1.1.1.2/32
2050	1.1.1.3/32
3	1.1.1.1/32
2051	2.2.2.2/32

LSR2# ldp show database

Input label database, 1.1.1.2:0-2.2.2.1:0

Label	Prefix
2048	2.2.2.2/32
2049	1.1.1.3/32
2050	1.1.1.1/32
3	2.2.2.1/32
2051	1.1.1.2/32

Output label database, 1.1.1.2:0-2.2.2.1:0

Label	Prefix
2048	1.1.1.1/32
2049	1.1.1.3/32
2050	2.2.2.1/32
3	1.1.1.2/32
2051	2.2.2.2/32

Input label database, 2.2.2.1:0-1.1.1.2:0

```
  Label     Prefix
   2048     1.1.1.1/32
   2049     1.1.1.3/32
   2050     2.2.2.1/32
      3     1.1.1.2/32
   2051     2.2.2.2/32
```

Input label database, 1.1.1.1:0-1.1.1.2:0

```
  Label     Prefix
   2048     1.1.1.1/32
   2049     1.1.1.3/32
   2050     2.2.2.1/32
   2051     2.2.2.2/32
      3     1.1.1.2/32
```

Input label database, 1.1.1.2:0-1.1.1.1:0

```
  Label     Prefix
   2048     2.2.2.1/32
   2049     1.1.1.2/32
   2050     1.1.1.3/32
   2051     2.2.2.2/32
      3     1.1.1.1/32
```

Output label database, 2.2.2.1:0-1.1.1.2:0

```
  Label     Prefix
   2048     2.2.2.2/32
   2049     1.1.1.3/32
   2050     1.1.1.1/32
      3     2.2.2.1/32
   2051     1.1.1.2/32
```

Output label database, 1.1.1.1:0-1.1.1.2:0

```
  Label     Prefix
   2048     2.2.2.1/32
   2049     1.1.1.2/32
   2050     1.1.1.3/32
   2051     2.2.2.2/32
      3     1.1.1.1/32
```

Output label database, 1.1.1.2:0-1.1.1.1:0

```
  Label     Prefix
   2048     1.1.1.1/32
   2049     1.1.1.3/32
   2050     2.2.2.1/32
   2051     2.2.2.2/32
      3     1.1.1.2/32
```

Mapping The Active Next Hop to A Label

The routing table shows IP address of the next hop in the path. The router uses this information along with the list of LDP capable interfaces exchange just prior to label distribution to map the *Gateway* to an MLPS label.

```
LER1# ip show routes
Destination          Gateway              Owner      Netif
-----------          -------              -----      -----
1.1.1.1              192.168.1.1          OSPF       To-LSR1
1.1.1.2              192.168.1.5          OSPF       To-LSR2
```

```
1.1.1.3              192.168.1.1           OSPF      To-LSR1
                     192.168.1.5           OSPF      To-LSR2
2.2.2.1              2.2.2.1               -         lo0
2.2.2.2              192.168.1.1           OSPF      To-LSR1
127.0.0.1            127.0.0.1             -         lo0
172.16.1.0/24        directly connected    -         LAN
192.168.1.0/30       directly connected    -         To-LSR1
192.168.1.4/30       directly connected    -         To-LSR2
192.168.1.8/30       192.168.1.1           OSPF      To-LSR1
                     192.168.1.5           OSPF      To-LSR2
192.168.1.12/30      192.168.1.1           OSPF      To-LSR1
192.168.1.16/30      192.168.1.1           OSPF      To-LSR1
192.168.1.20/30      192.168.1.1           OSPF      To-LSR1
                     192.168.1.5           OSPF      To-LSR2
192.168.1.24/30      192.168.1.5           OSPF      To-LSR2
```

To display the inbound and outbound label information including the labels that are in use…

RS(config)# mpls show ip-bindings

View this display information from the perspective of MPLS, a label switching protocol. The *in label* is the label that action is to be taken on, inbound toward the switch. The *out label* is the label that is pushed or swapped onto the outgoing packet. The gray italicized notations beside the FEC at the start of each table entry indicate the name of the router that owns the loopback address. The notations on the far right show indicate the name of the active next hop, indicated by the *inuse* field.

Examples using the three MPLS router types:

Ingress Router: A native IP packet is received; a longest prefix match is performed and the active next hop is determined, noted as the gateway in the forwarding table. The gateway is matched to the previously received LDP capable interface and LDP ID owner. The packet is encapsulated and sent on its way. If two active next hops exist both are used in a round robin fashion.

Transit Router: Matches label of inbound packet to outbound label and forwards accordingly. There is no load distribution across multiple active outbound paths.

Egress Router: Acts on native packet.

```
LER1# mpls show ip-bindings
  2.2.2.1/32 (LER1)
Best Path
        in label:     3             Active, Egress          (local
LER1)
        out label:    2051          lsr: 1.1.1.2:0
        out label:    2051          lsr: 1.1.1.1:0
  1.1.1.2/32 (LSR2)
        in label:     2048          Active
        out label:    imp-null      lsr: 1.1.1.2:0   inuse  (via
LSR2 - Direct)
        out label:    2049          lsr: 1.1.1.1:0
  1.1.1.3/32 (LSR3)
        in label:     2050          Active
        out label:    2049          lsr: 1.1.1.2:0   inuse  (Via of
LSR2)
        out label:    2050          lsr: 1.1.1.1:0   inuse  (Via of
LSR1)
  1.1.1.1/32 (LSR1)
        in label:     2051          Active
        out label:    2050          lsr: 1.1.1.2:0
        out label:    imp-null      lsr: 1.1.1.1:0   inuse  (Via of
LSR1)
  2.2.2.2/32 (LER2)
        in label:     2049          Active
        out label:    2048          lsr: 1.1.1.2:0
        out label:    2048          lsr: 1.1.1.1:0   inuse  (Via of
LSR1)
```

Timers & Session Maintenance

Timers & Session Maintenance

Hellos & Keepalives

TCP provides the guaranteed transport for control information. Periodic hello messages are used to reset hello hold timers and provide a quick and simple means to validate the reachability of the peer. Once the LDP session between the peers is established keepalives are sent across the TCP connection to monitor the status of the session. Each individual session is identified by the use of the *Message_ID* object in each LDP packet. It is important to note that the hello and keepalive timers are negotiated at session establishment. Each peer advertises the locally significant timers, hellos and keepalives. If the values are different on the two hosts the lowest value of each, hello and keepalive, will be used.

Default Timer Values

Default Hello and Keepalive timers on the RS platform.

Session Monitoring Parameters	Direct Connect LDP Peer	Remote LDP Peer
Hello Hold Time	15 Seconds	15 Seconds

Hello Send Interval (1/3 of hold time) – Not configurable	5 Seconds	5 Seconds
Keepalive Timeout	30 Seconds	30 Seconds
Keepalive Send Interval (1/3 of hold time) – Not Configurable	10 Seconds	10 Seconds

Configuring Timer Values

Both the hello hold time and the keepalive timeout are configurable. However, the send interval is automatically adjusted to one third of these settings.

To change the Hello Hold Time

```
RS(config)# ldp set interface <name|all> link-hello-hold-time
<seconds>
```

To change the Keepalive Timeout

```
RS(config)# ldp set interface <name|all> Keepalive-timeout
<seconds>
```

Both send intervals will automatically be adjusted to 1/3 of the new settings.

LDP Loop Detection

LDP Loop Detection

Loop Distribution

Loop Distribution

Riverstone Networks supports both loop detection mechanisms defined by the LDP specification, Path Vector TLV and Hop Count. The Path Vector TLV instructs each intermediate LDP node to check the TLV and ensure that its router-id, or LDP identifier, does not appear in the recorded intermediate hops. If it does the router must discard the label request and inform the instantiating router there is a loop in the network. If however, the local router does not find its IP address in the list of intermediate hops it appends its router-id and forwards the packet to the next hop along the path. If the hop count approach is taken, each router increments the hop count field in the LDP packet by one before passing the label request to the downstream node. If the hop count exceeds the acceptable limit defined by the instantiating router, a loop is assumed and the label request is dropped.

By default, loop detection mechanisms are disabled on the RS platform. Enabling loop detection is a global command. Hop Count and Path Vector can be enabled separately, depending on the required loop detection mechanisms preferences. By default the hop count value is 10, if no other path-vector-limit is specified.

To enable both loop detection schemes with a maximum hop count of 20.

```
ldp set global hop-count-loop-detection-enable
ldp set global path-vector-loop-detection-enable
```

ldp set global path-vector-limit 20

To display what type of loop detection is enabled

RS# ldp show global

```
LER1# ldp show global
Ordered control mode
Path vector loop detection enabled, Path vector limit 20
Hop count loop detection enabled, Hop count limit 20
```

Authenticating LDP Sessions

Authenticating LDP Sessions

Using MD5 Signatures for Security

Using MD5 Signatures for Security

Riverstone Networks provides session authentication using the IETF standard MD5 signature option. When the key information is stored in the configuration file it is encrypted. If the peer or interface options are not specified authentication is applied to all interfaces with LDP enabled. Remember, the password is case sensitive. Warning: if applying the MD5 signature option to an LDP interface that already has a session established it will drop the TCP connection and all associate label binding to FEC entries for that session. The session will reestablish the database information for that session once both interfaces agree on a common security method and password.

To enable MD5 authentication between peers…

```
RS(config)# ldp set md5-password <password> peer|interface <name>
```

An example may look something like this…

```
Ler1(config)# ldp set md5-password ldp-peer-key interface To-LSR1
Ler1(config)# ldp set md5-password ldp-peer-key interface To-LSR2
```

The md5 passwords would be encrypted in the configuration file for security purposes…

```
ldp set md5-password <encrypted>58a95d7c2e3c0329 interface To-
LSR1
ldp set md5-password <encrypted>58a95d7c2e3c0329 interface To-
LSR2
```

The LDP hello adjacencies will form even with inconsistent security on peering interfaces. However, the TCP session will fail to authenticate, as noted by the continuous *connection* phase of *opening*. The fail state for the peers will oscillate between the one indicated below for 192.168.1.5 and a similar state that identifies the "*State: Nonexistent*" with no "*connection*" information.

To check the status of the LDP peering sessions…

```
RS# ldp show sessions
```

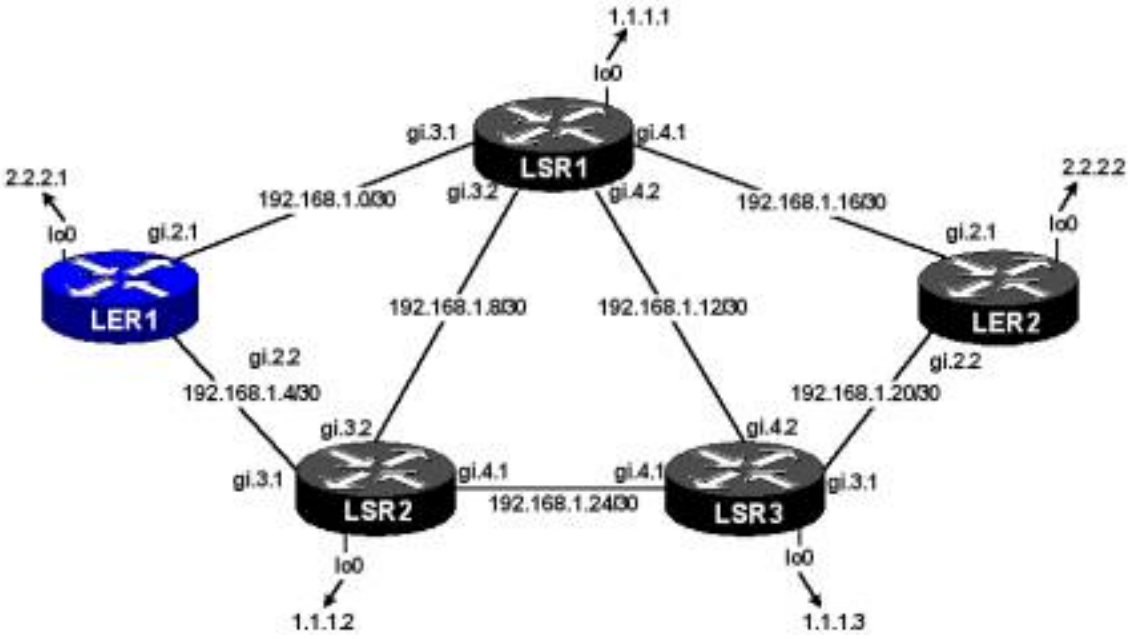
```
LER1#  ldp show sessions
LER1# ldp show session
Codes: Tx - Sent, Rx tot - Received Total, Rx fltd - Received
Filtered
Address          State         Connection       Hold Time(sec)
Tx/Rx tot/Rx fltd
192.168.1.5      Nonexistent   Opening          11
0/0/0
192.168.1.1      Operational   Open             12
4/5/0
```

LDP Show Commands

LDP Show Commands

Sample Network

The IGP, MPLS and LDP are only enabled on the core facing interfaces. This network and the associated configuration will form the basis for show commands that follow.




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls start
ldp add interface To-LSR1
ldp add interface To-LSR2
ldp add interface lo0
ldp start
```

LDP Session Information

High-level session information for each peer:

```
RS# ldp show sessions all
```

- Address of peer
- State of the session
- TCP Connection state (closed, opening, open)
- Time before session expires without a keepalive
- Number of labels sent, received and received labels filtered

Possible Session States	Description
Nonexistent	No session exists
Connecting	TCP connection is in progress
Initialized	TCP connection established
OpenSent	Initialization or keepalive messages being transmitted
OpenRec	Keepalive message being transmitted
Operational	Session established
Closing	Closing session

```
LER1# ldp show session all
Codes: Tx - Sent, Rx tot - Received Total, Rx fltd - Received
Filtered

Address          State         Connection       Hold Time(sec)
Tx/Rx tot/Rx fltd
192.168.1.1      Operational   Open             7
5/5/0
192.168.1.5      Operational   Open             12
5/5/0
```

A more detailed view of the session information is available by coding the *verbose* option. The following additional information is available with this option…

- Session ID (comprised of both LDP Identifiers)

- Timer values
- Local and remote physical interfaces
- A list of LDP enabled interfaces on the remote peer

```
LER1# ldp show session all verbose
Address: 192.168.1.1, State: Operational, Connection: Open, Hold
time: 12
  Labels Sent 5, Received total 5, Received filtered 0
  Session ID: 2.2.2.1:0--1.1.1.1:0
  Next keepalive in 1 seconds
  Active, Maximum PDU: 4096, Hold time: 15 seconds
  Keepalive interval: 10 seconds, Connect retry interval: 14
seconds
  Local address: 192.168.1.2, Remote address: 192.168.1.1
  Next-hop addresses received:
    1.1.1.1
    192.168.1.17
    192.168.1.9
    192.168.1.13
    192.168.1.1

Address: 192.168.1.5, State: Operational, Connection: Open, Hold
time: 12
  Labels Sent 5, Received total 5, Received filtered 0
  Session ID: 2.2.2.1:0--1.1.1.2:0
  Next keepalive in 0 seconds
  Active, Maximum PDU: 4096, Hold time: 15 seconds
  Keepalive interval: 10 seconds, Connect retry interval: 14
seconds
  Local address: 192.168.1.6, Remote address: 192.168.1.5
  Next-hop addresses received:
    1.1.1.2
    192.168.1.10
    192.168.1.25
    192.168.1.5
```

LDP Neighbor Information

Detailed neighbor information for each peer

RS# ldp show neighbor verbose

- Address where neighbor was discovered and interface used to reach neighbor
- Label Space ID indicating the LDP identifier and label space the label was issued from - ":0" being from the global space
- Time before session expires without a keepalive
- Transport address used for establishing the session

```
LER1# ldp show neighbor verbose
Address              Interface            Label space ID      Hold
Time(seconds)
192.168.1.5          To-LSR2              1.1.1.2:0             12
  Transport address: 192.168.1.5
192.168.1.1          To-LSR1              1.1.1.1:0              7
  Transport address: 192.168.1.1
```

LDP Statistics Information

The statistical information about the LDP protocol is broken into two horizontal planes each with a cumulative and a five second representation. The tables are self-explanatory. One thing to note, if the statistics are cleared all the cumulative information is lost, obviously. So when reviewing the statistics following a clear, the "*Event Type - Sessions Opened*" may be zero even though there are open sessions. Don't let this field mislead you into thinking no sessions are formed. The session display command is the authority on session related information.

RS# ldp show statistics

```
LER1# ldp show statistics
Message type                 Total                        Last 5
seconds
============                 =====
=============
```

	Sent	Received	Sent	Received
Hello	378	378	2	2
Initialization	2	2	0	0
Keepalive	190	190	2	2
Notifcation	0	0	0	0
Address	4	7	0	0
Address withdraw	0	0	0	0
Label mapping	12	24	0	0
Label request	0	0	0	0
Label withdraw	2	4	0	0
Label release	2	4	0	0
Label abort	0	0	0	0
All UDP	378	378	2	2
All TCP	212	208	2	2

Event type	Total	Last 5 seconds
Sessions opened	2	0
Sessions closed	0	0
Shutdown received	0	0
Shutdown sent	0	0
Keep alive expired	0	0
Malformed TLV	0	0
Bad TLV length	0	0
Bad message length	0	0
Bad PDU length	0	0
Bad LDP identifiers	0	0
Hello errors	0	0
Advertisement errors	0	0
Max PDU errors	0	0
Label range errors	0	0

LDP Interface Information

A detailed view of the LDP interfaces indicates the following for each LDP enabled interface…

RS# ldp show interface all verbose

- Label Space indicating the LDP identifier and label space the label was issued from - ":0" being from the global space
- The number of neighbor sessions that exist on this interface
- Timer information
- Label management – retention and distribution

```
LER1# ldp show interface all verbose

Interface             Label space              Nbr count  Next
hello(seconds)
lo                    2.2.2.1:0                    0          0
  Hold time: 15, Liberal label retention, Downstream unsolicited
To-LSR2               192.168.1.6:0                1          1
  Hold time: 15, Liberal label retention, Downstream unsolicited
To-LSR1               192.168.1.2:0                1          2
  Hold time: 15, Liberal label retention, Downstream unsolicited
```

L2-FEC Information

This is more appropriately discussed as part of L2 Virtual Private Networks.

RS# ldp show l2-fec <options>

LDP Global Settings

Used to display the LSP control mode for label distribution and loop detection settings.

RS# ldp show gloabl

```
LER1# ldp show global
Ordered control mode
Path vector loop detection disabled
Hop count loop detection disabled
```

LDP Summary Command

To review all configuration information, distribution of information and settings with a single command…

RS# ldp show all

```
LER1 ldp show all

Global parameters
------ ----------
Ordered control mode
Path vector loop detection disabled
Hop count loop detection disabled

Interface parameters
--------- ----------
Interface            Label space              Nbr count  Next
hello(seconds)
lo                   2.2.2.1:0                    0          0
To-LSR1              192.168.1.2:0                1          0
To-LSR2              192.168.1.6:0                1          0

Neighbor parameters
-------- ----------
Address              Interface           Label space ID      Hold
Time(seconds)
192.168.1.1          To-LSR1             1.1.1.1:0             10
192.168.1.5          To-LSR2             1.1.1.2:0             10

Session parameters
------- ----------
```

Codes: Tx - Sent, Rx tot - Received Total, Rx fltd - Received Filtered

Address	State	Connection	Hold Time(sec)	Tx/Rx tot/Rx fltd
192.168.1.1	Operational	Open	10	5/5/0
192.168.1.5	Operational	Open	10	5/5/0

LDP Statistics

Message type	Total		Last 5 seconds	
	Sent	Received	Sent	Received
Hello	1176	1176	2	2
Initialization	0	0	0	0
Keepalive	588	588	0	0
Notifcation	0	0	0	0
Address	0	0	0	0
Address withdraw	0	0	0	0
Label mapping	0	0	0	0
Label request	0	0	0	0
Label withdraw	0	0	0	0
Label release	0	0	0	0
Label abort	0	0	0	0
All UDP	1176	1176	2	2
All TCP	588	588	0	0

Event type	Total	Last 5 seconds
Sessions opened	0	0
Sessions closed	0	0

```
Shutdown received                0                        0
Shutdown sent                    0                        0

Keep alive expired               0                        0
Malformed TLV                    0                        0
Bad TLV length                   0                        0
Bad message length               0                        0
Bad PDU length                   0                        0
Bad LDP identifiers              0                        0
Hello errors                     0                        0
Advertisement errors             0                        0
Max PDU errors                   0                        0
Label range errors               0                        0

Label Database
----- --------
Input label database, 2.2.2.1:0-1.1.1.1:0
  Label     Prefix
   2048     2.2.2.2/32
   2049     1.1.1.2/32
   2050     1.1.1.3/32
   2051     2.2.2.1/32
      3     1.1.1.1/32

Output label database, 2.2.2.1:0-1.1.1.1:0
  Label     Prefix
   2048     1.1.1.2/32
   2049     2.2.2.2/32
   2050     1.1.1.3/32
   2051     1.1.1.1/32
      3     2.2.2.1/32

Input label database, 2.2.2.1:0-1.1.1.2:0
  Label     Prefix
   2048     2.2.2.2/32
   2049     1.1.1.3/32
   2050     1.1.1.1/32
   2051     2.2.2.1/32
      3     1.1.1.2/32
```

```
Output label database, 2.2.2.1:0-1.1.1.2:0
  Label     Prefix
   2048     1.1.1.2/32
   2049     2.2.2.2/32
   2050     1.1.1.3/32
   2051     1.1.1.1/32
      3     2.2.2.1/32
```

Signaling Different Types of RSVP-TE Paths

Signaling Different Types of RSVP-TE Paths

Extending RSVP for MPLS Networks

Standards track protocol defined by RFC3209, “RSVP-TE: Extensions to RSVP for LSP Tunnels”

The Applicability statement for RSVP-TE is described by RFC3210.

Signaling a Path Using RSVP-TE

RSVP-TE can be used to signal hop-by-hop and explicit paths through an MPLS network. Once the network is MPLS ready and the link state routing protocol has been deployed, with or without traffic engineering extensions, a dynamically signal LSP can be established by simply configuring the instantiating router. Traffic engineering can be applied to either of these signaling approaches. Creating an RSVP path through a network is a rather simple process.

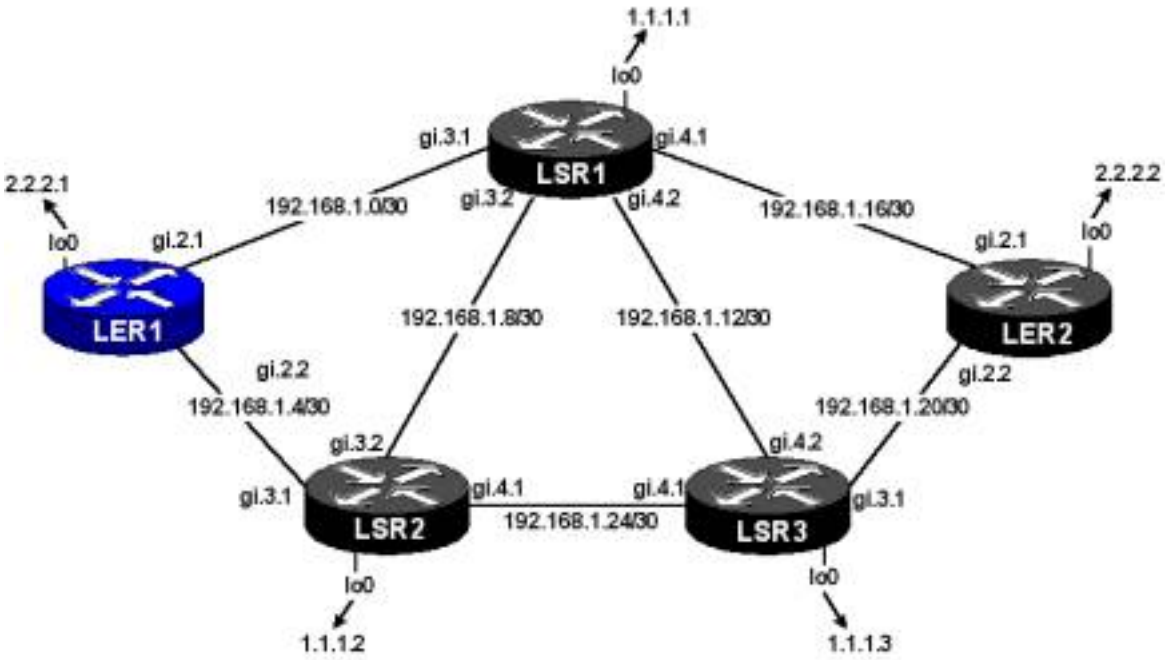
Hop-by-Hop

This method determines a path through the network based on the interior gateway protocols view of the network. If no constraints are applied to the LSP then the instantiating router simply sends the request for a path to the active next hop for that destination, no explicit routing. The IGP at each router is free to select active next hops based on the link state database. In the event of path failure, like a link failure somewhere in the network, the hop-by-hop method will eventually establish a path around the failure based on updated link state database information. Re-optimization is under development on the RS platform.

To create a simple hop-by-hop path…

RS(config)# mpls create label-switched-path <name> from <SIP-to-use> to <destination IP> <options>

A sample network shows how an instantiating router requests a hop-by-hop end-to-end RSVP path through the MPLS network to a destination without any constraints or resource requirements.



interface create ip To-LSR1 address-netmask 192.168.1.2/30 port

```
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create label-switched-path LSP from 2.2.2.1 to 2.2.2.2 no-
cspf
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

The northerly most router represented the active next hop for the destination and the instantiating router followed the information in the forwarding information base and sent the RSVP request to the active next hop indicated in the FIB. The result, the IGP used the shortest path between the edge routers over which to signal and establish the path.

To display high level LSP information for an LSP including start and end points, state and labels used…

```
RS# mpls show label-switched-paths <name|all> <options>
```

```
LER1# mpls show label-switched-paths all
Ingress LSP:
LSPname                    To                 From
State LabelIn LabelOut
LSP                        2.2.2.2            2.2.2.1
Up    -       17
```

```
Transit LSP:
LSPname                    To                  From
State LabelIn LabelOut

Egress LSP:
LSPname                    To                  From
State LabelIn LabelOut
```

A more detailed view of the LSP information can be found using the "*verbose*" option. This includes the various session attributes for the LSP and the associated path information. The path information includes path attributes, labels associated with the LSP, timers, resources constraints and the confirmation the path the LSP has taken through the MPLS network (record-route).

RS# mpls show label-switched-paths <name> verbose

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP"
        state: Up                lsp-id: 0x9
        status: Success
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: none
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR>

        Path-Signalling-Parameters:
            attributes: <NO-CSPF>
            inherited-attributes: <>
            label in:            label out: 17
            retry-limit: 5000    retry-int: 15 sec.
            retry-count: 5000    next_retry_int: 0.000000 sec.
            preference: 7        metric: 1
            ott-index: 1         ref-count: 1
            bps: 0               mtu: 1500
            hop-limit: 255       opt-int: 600 sec.
            record-route:
                192.168.1.1
```

```
                192.168.1.18

Transit LSP:

Egress LSP:
```

The same display commands can be used on the transit router for this LSP. Remember, an outbound label “3” indicates penultimate hop pop is performed on the router preceding the last router in the LSP. When this is done, the router makes a forwarding decision based on the inbound label sending it to the next hop without applying a new upper level label on the outbound.

```
LSR1# mpls show label-switched-paths all
Ingress LSP:
LSPname                   To                From
State LabelIn LabelOut

Transit LSP:
LSPname                   To                From
State LabelIn LabelOut
LSP                       2.2.2.2           2.2.2.1           -
     17      3

Egress LSP:
LSPname                   To                From
State LabelIn LabelOut


LSR1# mpls show label-switched-paths all verbose
Ingress LSP:

Transit LSP:

    Label-Switched-Path: "LSP"
        state: Up               lsp-id: 9
        to: 2.2.2.2             from: 2.2.2.1
        Path-Signalling-Parameters:
            setup-pri: 7                holding-pri: 0
            label in: 17                label out: 3
            path recvfrom: 192.168.1.2  path sendto: 192.168.1.18
```

```
            explicit-path:
            record-route:
                192.168.1.18

Egress LSP:
```

To complete the picture, the display commands can be used on the egress router. The inbound label 3 indicates this router has told its upstream to pop the label and send only the encapsulated packet to the egress. Also notice, the output label. Okay so there isn't one. The "*label out:*" field is blank and this omission indicates this router is to act on the native IP packet it receives from the upstream router.

```
LER2# mpls show label-switched-paths all
Ingress LSP:
LSPname                   To                From
State LabelIn LabelOut

Transit LSP:
LSPname                   To                From
State LabelIn LabelOut

Egress LSP:
LSPname                   To                From
State LabelIn LabelOut
LSP                       2.2.2.2           2.2.2.1           -
     3       -
Egress LSP:

LER2# mpls show label-switched-paths all verbose
Ingress LSP:

Transit LSP:

Egress LSP:
    Label-Switched-Path: "LSP"
        state: Up               lsp-id: 9
        to: 2.2.2.2             from: 2.2.2.1
```

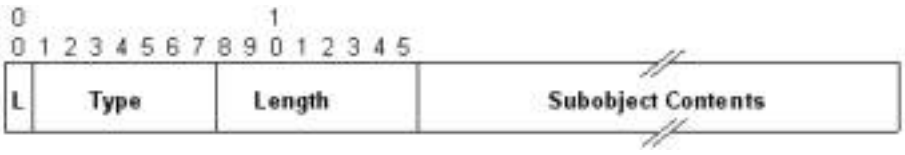
```
Path-Signalling-Parameters:
    setup-pri: 7                holding-pri: 0
    label in: 3                 label out:
    path recvfrom: 192.168.1.17 path sendto: 2.2.2.2
    explicit-path:
    record-route:
```

Explicit Route Object - ERO

The hop-by-hop method allows the interior gateway protocol to select the path through the network. However, there are many benefits that can be realized by having the instantiating router dictate the hops an LSP will traverse. The ERO is the creation and inclusion of the list of routers that comprise the most suitable path through the MPLS network. This is analogous to the source routing, where the instantiating router dictates, either in whole or in part, the path through the network.

The ERO object may contain two kinds of explicit routes, *strict* or *loose* hops. A strict hop indicates that the two nodes must be adjacent to one another with no intermediate hops separating them. A loose hop indicates the nodes do not have to be adjacent to each other and the IGP can be used to determine the best path to the loose hop. This allows the router building the ERO to apply some abstract level of configuration, indicating that the path needs to traverse a particular router without dictating how to reach that hop. By default, any hop specified as part of the ERO is strict unless otherwise configured as loose. Information contained in the ERO is stored in the path state block for each router. Currently implementations on the RS platform support loose and strict routing in the form of IP address.

The Internet draft defines the fields of the ERO subobject as follows:



L: The disposition of the particular hop. A value of 0 indicates the subobject is strict. This is the default if the configuration omits the type field for this hop. A value of 1 indicates the type of this hop is loose.

Type: A seven bit field indicating the value of the subobjects contents. The draft currently defines four reserved values. Of these, Riverstone supports IP addressing.

0	Reserved
1	IPv4 Prefix
2	IPv6 prefix
32	Autonomous system

Length: An 8 bit field that represents the number of bytes for the entire subobject, inclusive of all fields.

Subobject Contents: The addressing information specific to the type. A minimum of two bytes represents the smallest possible type field, AS Number.

Configuring an explicit route on the RS platform is done by creating a path with a specified number of hops, defining those hops with their disposition, strict or loose and associating that path to an LSP as primary or secondary. Note: If the path is created without specifying any number of hops the interior gateway protocol determines the active next hop for the destination and sends the request to that node. It is equivalent to creating a hop-by-hop path, no explicit route.

To create the path with an explicit route…

```
RS(config)# mpls create path <name> num-hops <number>
```

To define the hops for a created path…

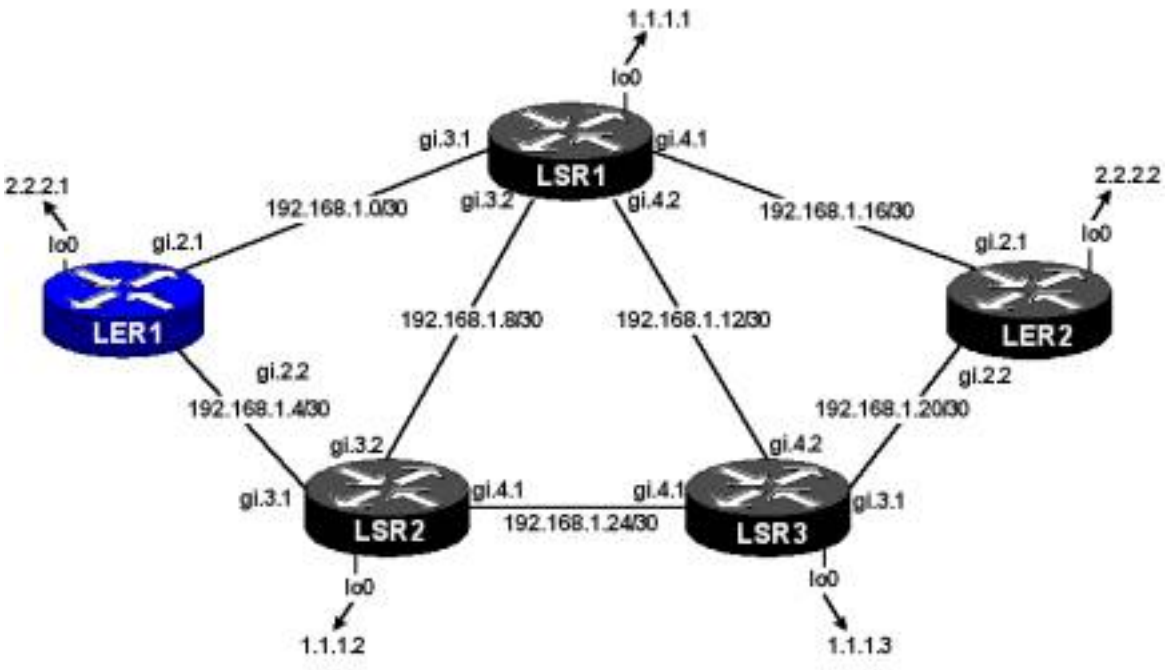
```
RS(config)# mpls set path <name> hop <number> ip-addr <ip-
```

address> type <strict|loose>

Associating a path to an LSP…

RS(config)# mpls set label-switched-path <name> primary|secondary <path name>

Example configuration of a completely strict route from ingress (LER1) to egress(LER2)..




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create path ERO-Path1 num-hops 4
```

```
mpls set path ERO-Path1 hop 1 ip-addr 192.168.1.6 type strict
mpls set path ERO-Path1 hop 2 ip-addr 192.168.1.5 type strict
mpls set path ERO-Path1 hop 3 ip-addr 192.168.1.26 type strict
mpls set path ERO-Path1 hop 4 ip-addr 192.168.1.22 type strict
mpls create label-switched-path LSP from 2.2.2.1 to 2.2.2.2 no-cspf
mpls set label-switched-path LSP primary ERO-Path1
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

The resulting subobjects of the ERO would look like this…

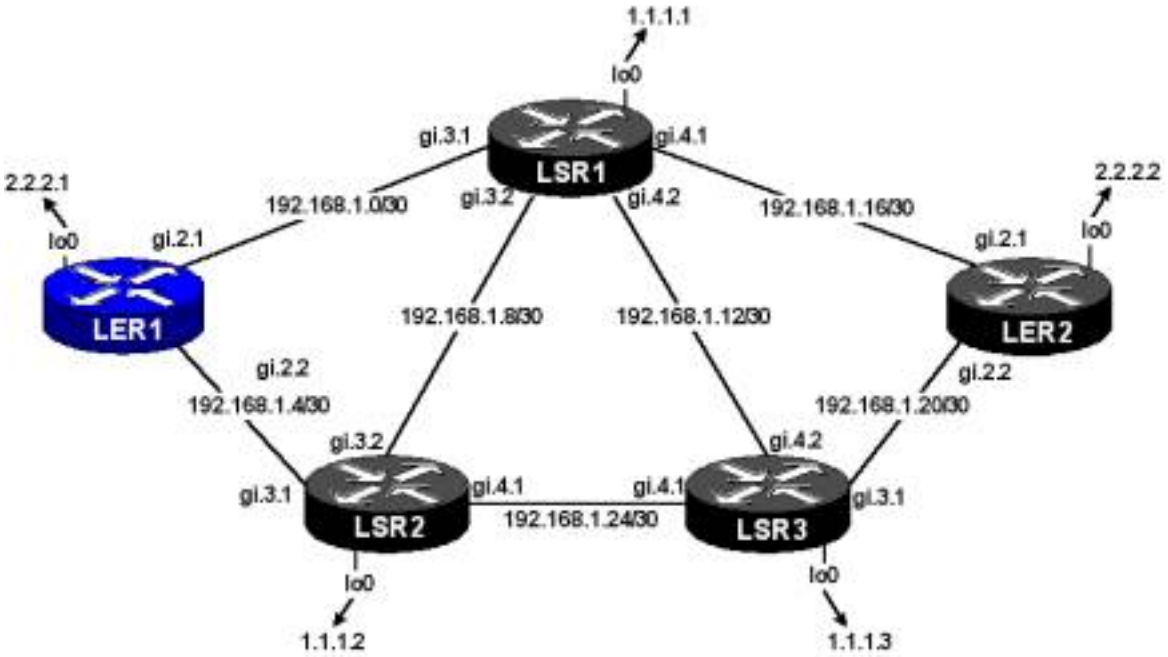
```
L=0;Type=4;Length=64;Contents=192.168.1.6 (instantiating router interface)
L=0;Type=4;Length=64;Contents=192.168.1.5
L=0;Type=4;Length=64;Contents=192.168.1.26
L=0;Type=4;Length=64;Contents=192.168.1.22
```

Looking at the LSP information, the explicit route information is logged in the “explicit-path” field.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP"
        state: Up                lsp-id: 0x9
        status: Success
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: primary
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR PRI>

        Protection-Path "ERO-Path1": <Active, Primary>
            state: Up            lsp-id: 0x4002
            status: Success
            attributes: <>
```

```
        inherited-attributes: <>
    Path-Signalling-Parameters:
        attributes: <>
        inherited-attributes: <NO-CSPF>
        label in:           label out: 17
        retry-limit: 5000   retry-int: 15 sec.
        retry-count: 5000   next_retry_int: 0.000000 sec.
        preference: 7       metric: 1
        ott-index: 1        ref-count: 1
        bps: 0              mtu: 1500
        hop-limit: 255      opt-int: 600 sec.
        explicit-path: "ERO-Path1"  num-hops: 4
            192.168.1.6     - strict
            192.168.1.5     - strict
            192.168.1.26    - strict
            192.168.1.22    - strict
        record-route:
            192.168.1.5
            192.168.1.26
            192.168.1.22

Transit LSP:

Egress LSP:
```

This example demonstrates a slightly different approach, using loose hops and the loopback interfaces on the routers instead of physical interface addressing. In this example all traffic is forced through a single router, LSR3, by coding one of the loose hops to be the IP address of that routers loopback interface. Also notice the first hop and last hop are actually the tunnel end points. In essence the path can be established across any links as long as one of the transit nodes is LSR3. This provides a certain level of link failure protection but still leaves the single point of failure should LSR3 become unusable. Loose and strict routes are may be combined in a path message, allowing for the description of a path that must pass through some hops in an specific point to point fashion (strict) while other parts of the path may be derived by the IGP (loose).




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create path To-LER2-Prime num-hops 3
mpls set path To-LER2-Prime hop 1 ip-addr 2.2.2.1 type loose
mpls set path To-LER2-Prime hop 2 ip-addr 1.1.1.3 type loose
mpls set path To-LER2-Prime hop 3 ip-addr 2.2.2.2 type loose
mpls create label-switched-path To-LER2-1 from 2.2.2.1 to 2.2.2.2
no-cspf
mpls set label-switched-path To-LER2-1 primary To-LER2-Prime
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

The resulting subobjects of the ERO would look like this…

```
L=0;Type=4;Length=64;Contents=2.2.2.1 (instantiating router
interface)
L=1;Type=4;Length=64;Contents=1.1.1.1
L=1;Type=4;Length=64;Contents=2.2.2.2
```

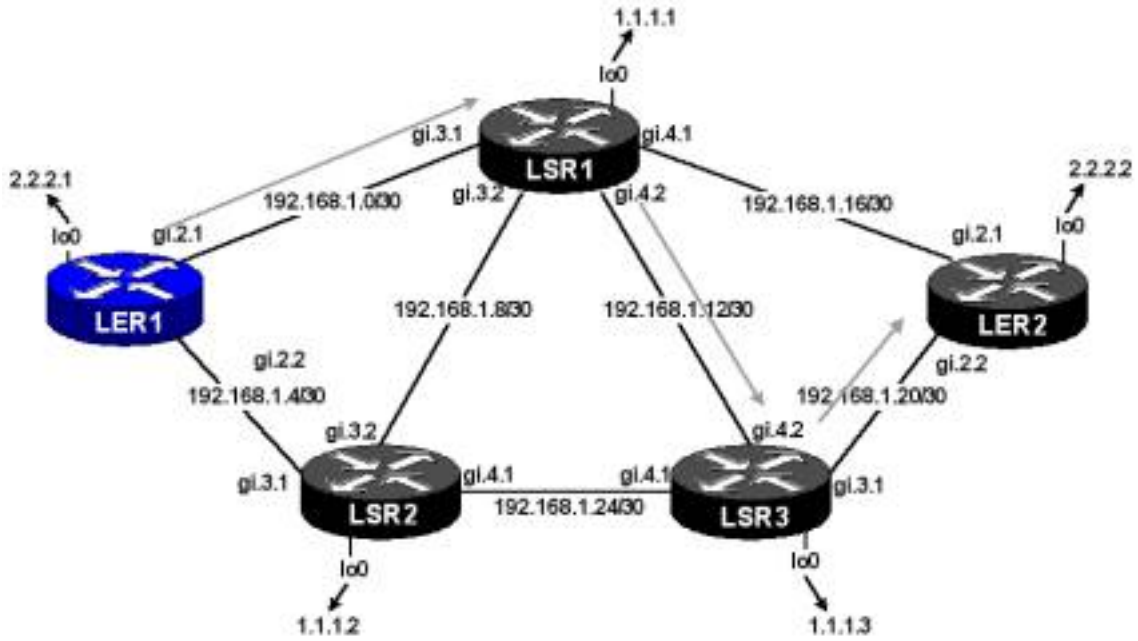
Looking at the LSP information reveals the differences between the two approaches. The completely strict explicit route was kind of this way or no way approach to signaling and establishing the LSP. If any node or link specified in the explicit path failed that path statement would fail the LSP. The loose approach provides slightly more resilience at the expense of complete control. Should the path be established and a node or link that the LSP was using fail, the instantiating router would determine a new path through the network, signal and establish it.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP"
        state: Up               lsp-id: 0x9
        status: Success
        to: 2.2.2.2             from: 2.2.2.1
        proto: <rsvp>           protection: primary
        setup-pri: 7            hold-pri: 0
        attributes: <FROM_ADDR PRI>

        Protection-Path "ERO-Path1": <Active, Primary>
            state: Up           lsp-id: 0x4003
            status: Success
            attributes: <>
            inherited-attributes: <>
        Path-Signalling-Parameters:
            attributes: <>
            inherited-attributes: <NO-CSPF>
            label in:           label out: 17
            retry-limit: 5000   retry-int: 15 sec.
```

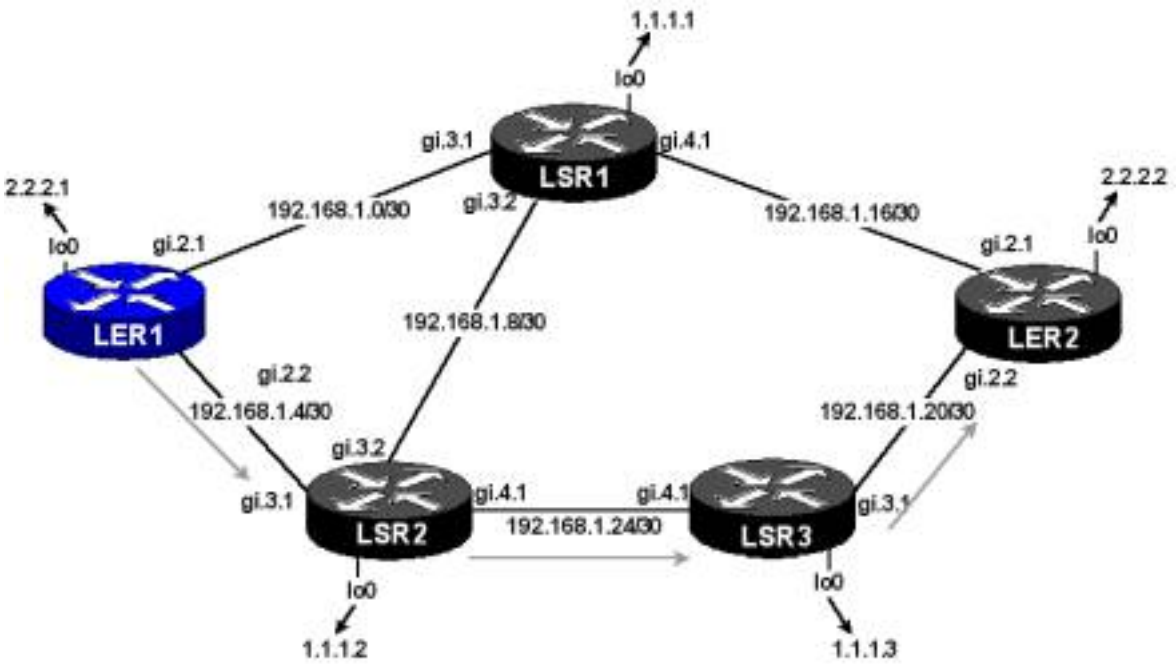
```
retry-count: 5000   next_retry_int: 0.000000 sec.
preference: 7       metric: 1
ott-index: 1        ref-count: 1
bps: 0              mtu: 1500
hop-limit: 255      opt-int: 600 sec.
explicit-path: "ERO-Path1"  num-hops: 3
    2.2.2.1         - strict
    1.1.1.3         - loose
    2.2.2.2         - loose
record-route:
    192.168.1.1
    192.168.1.14
    192.168.1.22
```

```
Transit LSP:

Egress LSP:
```

The actual explicit route object is built using a combination of the IGP forwarding table and the manually configured hops. For example, the first hop in the path was derived from the forwarding table based on the next best hop for the loose route of 1.1.1.3. Similarly the next best hop from 1.1.1.3 to 2.2.2.2 was also determined form the routing table.



Assuming there was a failure, as long as the loopback interfaces specified as part of the ERO are reachable the IGP will converge and the path will be established across an alternate set of transit routers. An example is presented below.



-

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP"
        state: Up                lsp-id: 0x9
        status: Success
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: primary
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR PRI>

        Protection-Path "ERO-Path1": <Active, Primary>
            state: Up            lsp-id: 0x4003
            status: Success
            attributes: <>
            inherited-attributes: <>
        Path-Signalling-Parameters:
            attributes: <>
            inherited-attributes: <NO-CSPF>
```

```
        label in:           label out: 17
        retry-limit: 5000   retry-int: 15 sec.
        retry-count: 5000   next_retry_int: 0.000000 sec.
        preference: 7       metric: 1
        ott-index: 1        ref-count: 1
        bps: 0              mtu: 1500
        hop-limit: 255      opt-int: 600 sec.
        explicit-path: "ERO-Path1"  num-hops: 3
            2.2.2.1         - strict
            1.1.1.3         - loose
            2.2.2.2         - loose
        record-route:
            192.168.1.5
            192.168.1.26
            192.168.1.22


Transit LSP:

Egress LSP:
```

RSVP-TE Record Route Object - RRO

RSVP-TE Record Route Object - RRO

Record Route Object - RRO

Record Route Object - RRO

A *Record Route Object, or RRO,* the inclusion of this object indicates each node along the label switch path to include its IP address as a new top level subobject in the RRO field of the PATH or RESV message. Optionally, each node may be requested to also record its label information for the LSP. If *label recording* is included in the session attribute, the nodes along the path insert the label value information and then their IP addressing information above it. The RRO is organized in a last-in-first-out format, where the most recent router to write their route information as a subobject is the top level entry. Label recording is not currently supported on the RS platform.

Loop Detection: When a router receives a message that contains the RRO object it checks to ensure it has not entered an IP address into one of the subobject fields. If a router's IP is already contained in the RRO a loop exists and an error message is returned.

Management: The instantiating router provides the complete path information for an LSP at a single location. For routes that use strict explicit paths it may seem redundant. However, it is extremely useful for paths that use the IGP to either resolve loose hops in an ERO or use the IGP to determine the entire path.

To display the RRO information…

LER1# mpls show label-switched-path <name|all> verbose

```
LER1# mpls show label-switched-paths LSP verbose
Ingress LSP:

    Label-Switched-Path: "LSP"
        state: Up                lsp-id: 0x6
        status: Success
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: secondary
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR PRI SEC>

        Protection-Path "ERO-Path1": <Primary>
            state: Up            lsp-id: 0x4001
            status: Success
            attributes: <>
            inherited-attributes: <>
        Path-Signalling-Parameters:
            attributes: <>
            inherited-attributes: <NO-CSPF>
            label in:            label out: 17
            retry-limit: 5000    retry-int: 15 sec.
            retry-count: 5000    next_retry_int: 0.000000 sec.
            preference: 7        metric: 1
            ott-index: 2         ref-count: 1
            bps: 0               mtu: 1500
            hop-limit: 255       opt-int: 600 sec.
            explicit-path: "ERO-Path1"  num-hops: 4
                192.168.1.6      - strict
                192.168.1.5      - strict
                192.168.1.26     - strict
                192.168.1.22     - strict
            record-route:
                192.168.1.5
                192.168.1.26
                192.168.1.22
```

Route Pinning: The RRO may be used as input into the ERO function. When an LSP is established and the RRO is available to the instantiating router, it is

possible to re-establish the LSP using a strict explicit route that follows the RRO. This requires the generation of new PATH and RESV messages, tearing down the original. Riverstone does not support route pinning at this time.

The default action for the RS platform is to record the route information. If so desired, possibly for security or visibility purposes, this default action can be overridden on all or on a per LSP basis. Note: when changing any configurations of an established LSP the path and reservation will be torn down.

RS(config)# mpls <create|set> label-switched-path <name|all> no-record-route

```
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create path ERO-Path1 num-hops 4
mpls create path ERO-Path2 num-hops 3
mpls create path Path-Back1
mpls set path ERO-Path1 hop 1 ip-addr 192.168.1.6 type strict
mpls set path ERO-Path1 hop 4 ip-addr 192.168.1.22 type strict
mpls set path ERO-Path2 hop 1 ip-addr 192.168.1.2 type strict
mpls set path ERO-Path2 hop 2 ip-addr 192.168.1.1 type strict
mpls set path ERO-Path2 hop 3 ip-addr 192.168.1.18 type strict
mpls set path ERO-Path1 hop 2 ip-addr 192.168.1.5 type strict
mpls set path ERO-Path1 hop 3 ip-addr 192.168.1.26 type strict
mpls create label-switched-path LSP from 2.2.2.1 to 2.2.2.2 no-
cspf
mpls set label-switched-path LSP primary ERO-Path1
mpls set label-switched-path LSP secondary Path-Back1 preference
10
mpls set label-switched-path LSP secondary ERO-Path2 preference
100 standby
mpls set label-switched-path LSP no-record-route
mpls start

LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP"
```

```
state: Up                lsp-id: 0x6
status: Success
to: 2.2.2.2              from: 2.2.2.1
proto: <rsvp>            protection: primary
setup-pri: 7             hold-pri: 0
attributes: <FROM_ADDR PRI SEC>

Protection-Path "ERO-Path1": <Active, Primary>
    state: Up            lsp-id: 0x4001
    status: Success
    attributes: <>
    inherited-attributes: <>
Path-Signalling-Parameters:
    attributes: <>
    inherited-attributes: <NO-RRO NO-CSPF>
    label in:            label out: 17
    retry-limit: 5000    retry-int: 15 sec.
    retry-count: 5000    next_retry_int: 0.000000 sec.
    preference: 7        metric: 1
    ott-index: 1         ref-count: 1
    bps: 0               mtu: 1500
    hop-limit: 255       opt-int: 600 sec.
    explicit-path: "ERO-Path1"  num-hops: 4
        192.168.1.6      - strict
        192.168.1.5      - strict
        192.168.1.26     - strict
        192.168.1.22     - strict
    record-route:
```

How an RSVP-TE LSP is Signaled

How an RSVP-TE LSP is Signaled

PATH and RESV Messages

RSVP-TE makes use of PATH and RESV messages and some newly defined objects to signal, establish and maintain label switched paths.

Path Message: This is used to signal and request information required to establish the LSP from end-to-end, from ingress to egress. The RSVP PATH message includes all the necessary session attributes and the *label request* object. RSVP-TE environments exhibit is ordered downstream-on-demand label manageability, thus if a downstream router does not have a label binding to FEC to satisfy the request, it further propagates the request downstream until the egress router receives the request. Each router that receives the PATH message stores the information in the *Path State Block, or psb.* The LSP-ID is used as a unique identifier for each LSP.

Information contained in a PATH message…

```
RSVP_Path <rsvp_1>: Send Path
        session-attr: name: LSP2 flags: 0x0 setup-pri: 7 holding-
pri: 0
        session: end-point: 2.2.2.2 tunnel-id: 7 ext-tunnel-id:
```

```
0x2020201
        send-templ: sender: 2.2.2.1 lsp-id: 12
        next-hop: 192.168.1.1 <To-LSR1>
```

To display the path state block…

RS# mpls show psb <name|all> <options>

```
LER1# rsvp show psb
Path State Blocks:
--------------------
RSVP_PSB <rsvp_1>: (psb = 0x82a45690)
        session-attr: name: LSP2 flags: 0x0 setup-pri: 7 holding-
pri: 0
        session: end-point: 2.2.2.2 tunnel-id: 7 ext-tunnel-id:
0x2020201
        send-templ: sender: 2.2.2.1 lsp-id: 12
        prev-hop: 0.0.0.0 lih: 0
        in-if: <Local-API> out-if: <To-LSR1>
        explicit-route:
        sender-tspec: qos: CL cdr: 0 pbs: 0 pdr: 0 mpu: 20 mtu:
1436
        block-tspec:
        psb refresh timer:  time-to-expire: 6.810000 sec.
        psb cleanup timer:  time-to-expire: 136.040000 sec.
        ref-count: 1
        LSP-handle: 0x82a42b88
        Session: 0x82a45510
```

RESV Message: The egress router responds to the PATH message with a reservation or RESV message. The purpose of this response is to have all router along the path perform the *Call Admission Control, or CAC*, make the necessary bandwidth reservations and distribute the label binding to FEC to the upstream router. The label is distributed using the *Label Object*. The labels sent up stream become the output labels for the routers receiving the label object. The labels that a router issues to an upstream become the inbound label used as the lookup into the hardware output tag table. The reservation specific information is stored in

the *reservation state block, or rsb.*

Note: The mapping of inbound label to outbound label, associated values and actions is established in hardware, even before the first data packet arrives. This means, that every packet that arrives at an LSR with an MPLS label that is known is hardware switched.

The RESV message includes the unique identifier (lsp-id), the style of the reservation, and label information received from the downstream router.

```
2001-10-02  9:19:06 RSVP_Resv <rsvp_1>: Recv Resv
        session: end-point: 2.2.2.2 tunnel-id: 7 ext-tunnel-id:
0x2020201
        style: FF
        flow-descr: filter-spec: sender: 2.2.2.1 lsp-id: 12
        remote-labels: [19]
        in-if: <To-LSR1>
```

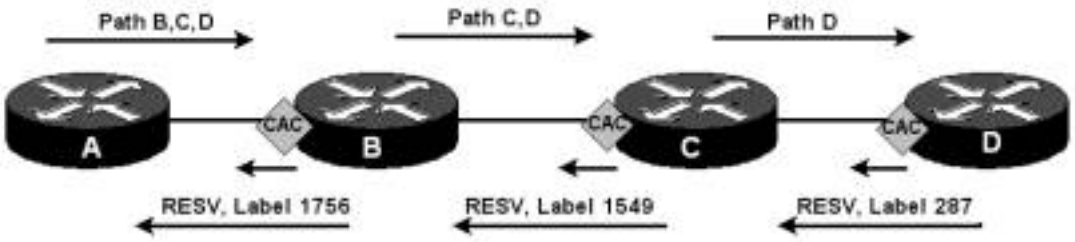
To display the reservation state block…

RS# mpls show rsb <name|all> <options>

```
LER1# rsvp show rsb
RSVP_RSB <rsvp_1>: (rsb = 0x82a45b78)
        session: end-point: 2.2.2.2 tunnel-id: 7 ext-tunnel-id:
0x2020201
        style: FF
        in-if: <To-LSR1>
        rsb refresh timer:  time-to-expire: 0.000000 sec.
        filter-spec: sender: 2.2.2.1 lsp-id: 12
        remote-labels: [19]
        local-labels: []
        filt-spec cleanup timer:  time-to-expire: 150.300000 sec.
        Session: 0x82a45510
```

Messaging Flow During Path Creation

Below, a representation that demonstrates how the PATH and RESV messaging work in conjunction with ordered downstream-on-demand label management.



The table is meant to provide a high level definition for each of the PATH and RESV type messages.

Requestor →	← **Recipient**
Path → Establishment request for new path, includes label request object	**Resv** ← Response to establishment request, performing reservations, label distribution (via the label object) across the same nodes the PATH traversed.
PathErr → Problem establishing path or refreshing state	**ResvErr** ← Problems establishing reservation or refreshing state
PathTear → Signal paths removal from service	**ResvTear** ← Release all resources for an LSP
ResvConfirm → Success confirmation of the Resv message, path established	

Resource Reservation Styles

Resource reservation styles are important functions to consider when configuring the LSP. The reservation style determines how bandwidth is accounted for across related label switch paths. There are two types of reservation styles applicable to MPLS networks, *Shared Explicit* & *Fixed Filter*.

Fixed Filter –FF: Relationships between LSP are not considered and bandwidth

is accounted for on an individual basis. This is the default reservation style on the RS platform.

Shared Explicit – SE: Reserves and accounts for bandwidth based on the uniqueness of the label switch path. A path for two related LSPs that crosses common physical links only makes one bandwidth reservation. It does not make a reservation for each path, since they cannot both be in use at the same time. This is beneficial for the recovery from a detour path, entered into service due to some failure causing the primary path to incur a service interruption. It is very likely the LSP that will assume the primary role from the temporary local repair will share some common links. In this case the reservation process for the new LSP should be able to count on the bandwidth reserved by the existing path to be available for its needs. This type of scenario also holds true for re-optimization and LSP alterations, like increased reservations. There is a definite requirement to support make-before-break.

To configure an LSP to use the shared explicit reservation style…

RS(config)# mpls <create|set> label-switched-path <name|all> from <Sip> to <DIP> adaptive

The reservation state block shows the style of each LSP.

```
RSVP_RSB <rsvp_1>: (rsb = 0x82a4ac68)
        session: end-point: 2.2.2.2 tunnel-id: 8 ext-tunnel-id:
0x2020201
        style: SE
        in-if: <To-LSR1>
        rsb refresh timer:  time-to-expire: 0.000000 sec.
        filter-spec: sender: 2.2.2.1 lsp-id: 13
        remote-labels: [19]
        local-labels: []
        filt-spec cleanup timer:  time-to-expire: 129.040000 sec.
        Session: 0x82a4c500
```

The RS platform is implementing support for make-before-break in the ros90

code series.

RSVP-TE Refresh Overhead Reduction

RSVP-TE Refresh Overhead Reduction

Without Refresh Reduction Techniques

Due to the soft state nature of the RSVP-TE protocol, constant refreshing must occur to ensure the integrity of the path state blocks and reservation state blocks. These refresh messages are sent using regular PATH and RESV messages on a per LSP basis. This may cause protocol-scaling issues, not to mention wasting network resources.

The refresh messages contain all the PATH and RESV messaging. Sample refresh messages are included below.

```
RSVP_Path <rsvp_1>: Send Path
        session-attr: name: LSP2 flags: 0x4 setup-pri: 7 holding-
pri: 0
        session: end-point: 2.2.2.2 tunnel-id: 13 ext-tunnel-id:
0x2020201
        send-templ: sender: 2.2.2.1 lsp-id: 20
```

```
        next-hop: 192.168.1.1 <To-LSR1>

RSVP_Resv <rsvp_1>: Recv Resv
        session: end-point: 2.2.2.2 tunnel-id: 13 ext-tunnel-id:
0x2020201
        style: SE
        flow-descr: filter-spec: sender: 2.2.2.1 lsp-id: 20
        remote-labels: [17]
        in-if: <To-LSR1>
```

RSVP Refresh Overhead Reduction

"RSVP Refresh Overhead Reduction Extensions" are defined in RFC2961.

Until recently, the soft state nature of RSVP-TE was a main concern in large provider and backbone networks. However, these concerns have been largely addressed RSVP-TE extension that the protocol to move from a per LSP refresh model to almost a per LSR approach. Each RSVP capable router is given the ability to advertise whether or not it is capable of supporting the refresh reductions specified, accomplished by setting the "flag" bit to "0x01" in the common header of each of the RSVP messages it sends. When two RSVP-TE peers both exhibit this support they make use of "RSVP Bundle Message" and "Summary Refresh Extension" defined in the RFC. It is important to note, all RSVP-TE messages set the router alert flag in the header, so the CPU processes all messaging.

RSVP-TE Bundle Messages

This allows RSVP capable routers to send many standard PATH and RESV update or refresh messages as *sub-messages* in a single packet, thus reducing the number of packets required to convey state. The standard message in the sub-message uses the *Message_ID* to allow the receiving router to map the appropriate message in the bundle to the applicable path or reservations state block. The savings with bundling standard messages is minimizing the required

number of packets and subsequent overhead required to convey state.

By default the RS platform does not enable bundled messages. To override the default and enable bundled messages…

```
RS(config)# rsvp set interface <name|all> aggregate-enabled
```

The default timer for the bundle refresh is set to refresh every 5 seconds. To change the default timer value use…

```
RS(config)# rsvp set global bundle-interval <1-65535>
```

Summary Refresh

Taking this to the next level, the summary refresh messages send only a list of *Message_IDs* without including the standard PATH or RESV messaging. Individual entries are checked against the path and reservation state blocks and if they exist they are refreshed. The use of acknowledgments, positive and negative, allows the receiving router to convey success and failure to the requesting router. The summary refresh is returned with ACK indicators for successful updates and NAK indicators if the path or reservation state block did not exist. This approach significantly reduces the amount of network resources consumed by refresh requirements.

By default the RS platform does not enable summary refresh techniques. To override the default and enable summary refresh…

```
RS(config)# rsvp set interface <name|all> msgid-extensions-enabled
```

The default timer for the summary refresh is set to refresh every 3 seconds, with an associated 1 second default on the receipt of the acknowledgement.. To change the default timer value use…

```
RS(config)# rsvp set global msgid-list-interval <1-65535>
```

```
RS(config)# rsvp set global msgack-Interval <1-65535>
```

Hello Messages

A simple adjacency test between RSVP-TE neighbors that quickly allows the status of the relationship to be determined using physical and process reachability.

By default the RS platform does not enable adjacency reachability testing. To override the default and enable hellos…

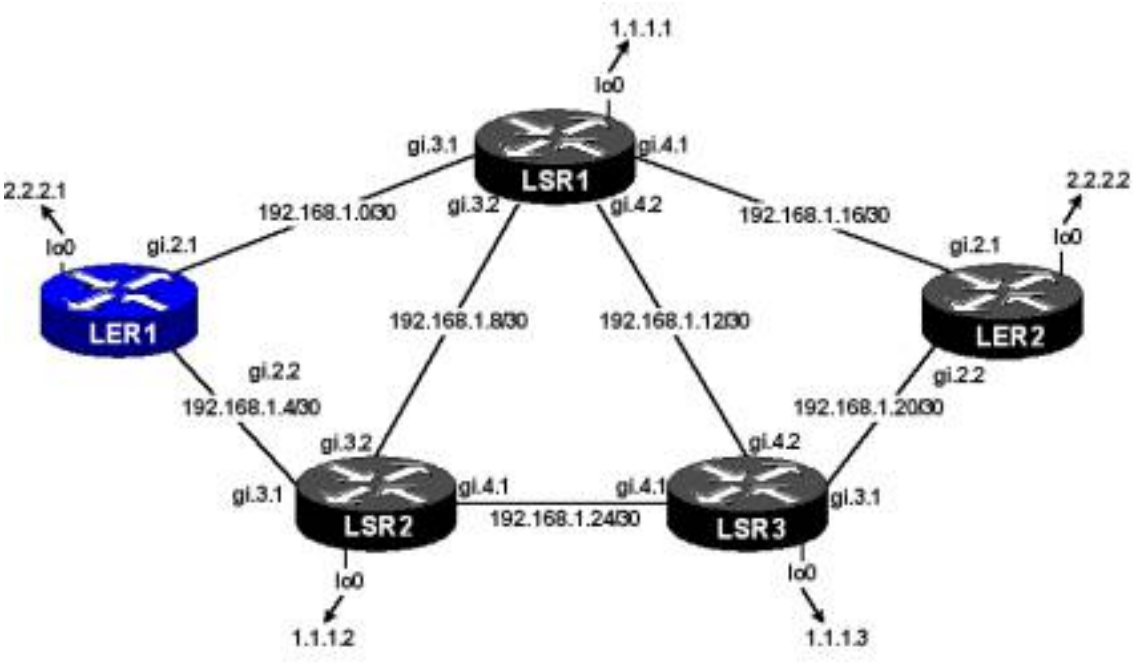
```
RS(config)# rsvp set interface <name|all> hello-enabled
```

The default hello timer is set to 3 with an associate 3 times multiplier. Should timer expire in three times the hello timer, a path tear message will be sent to terminate the LSP. To change the default timer value use…

```
RS(config)# rsvp set global hello-interval <1-65535>
RS(config)# rsvp set global hello-multiplier <1-255>
```

Deploying Refresh Reduction Techniques

Deploying these techniques reduces the resources required to maintain state for the RSVP-TE PATH and RESV messages. Two LSPs have been configured and pass through common nodes. Across these common nodes the refresh reduction techniques have been configured.




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create label-switched-path LSP2 from 2.2.2.1 to 2.2.2.2 no-
cspf
mpls create label-switched-path LSP1 from 2.2.2.1 to 2.2.2.2 no-
cspf
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp set interface To-LSR1 aggregate-enable hello-enable msgid-
extensions-enable
rsvp set interface To-LSR2 aggregate-enable hello-enable msgid-
extensions-enable
rsvp start
```

ospf set traffic-engineering on

Adjacency and message bundling and summarization are now reduced to this type of messaging…

```
2001-10-02 12:20:12 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:12 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:12 RSVP_SM <rsvp_1>: Send Srefresh, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:15 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:15 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:17 RSVP_SM <rsvp_1>: Send Bundle, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:18 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:18 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:21 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:21 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:24 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:24 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:27 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:27 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:30 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:30 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:32 RSVP_SM <rsvp_1>: Recv Bundle, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:32 RSVP_SM <rsvp_1>: Recv Srefresh, SrcAddr:
```

192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:33 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:33 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:36 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:36 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:39 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:39 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:42 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:42 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:42 RSVP_SM <rsvp_1>: Send Srefresh, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:45 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:45 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:47 RSVP_SM <rsvp_1>: Send Bundle, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:48 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:48 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:51 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:51 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:54 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:54 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:54 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:20:54 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:

```
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:57 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:20:57 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:21:00 RSVP_SM <rsvp_1>: Send Hello, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:21:00 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:21:00 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1>
2001-10-02 12:21:00 RSVP_SM <rsvp_1>: Recv HelloAck, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:21:03 RSVP_SM <rsvp_1>: Recv Bundle, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:21:03 RSVP_SM <rsvp_1>: Recv Srefresh, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:21:03 RSVP_SM <rsvp_1>: Recv Hello, SrcAddr:
192.168.1.1 in-if: <To-LSR1>
2001-10-02 12:21:03 RSVP_SM <rsvp_1>: Send HelloAck, DstAddr:
192.168.1.1 out-if: <To-LSR1
```

Checking the interface attributes reveals the refresh reduction techniques have been enabled on both core facing interfaces. To display the attribute information…

RS# rsvp show interface

```
LER1# rsvp show interface
all
Interface          Type       Attributes                    Path-
MTU
-------------------------------------------------------------------
To-LSR1            Enet/POS   <HELLO REF-RED MSG-ID>        1436
To-LSR2            Enet/POS   <HELLO REF-RED MSG-ID>         1436
```

To view the timer settings…

RS# rsvp show gloabl

```
LER1# rsvp show global
RSVP Global Configuration:
---------------------------
        RSVP Instance: <rsvp_1>
        ------------------------
        path refresh interval:  30 sec. path multiplier:        3
        resv refresh interval:  30 sec. resv multiplier:        3
        hello interval:         3 sec.  hello multiplier:       3
        bundle interval:        5 sec.
        msgid-list interval:    3 sec.
        msgack interval:        1 sec.
        blockade aging interval:60 sec.

        trace flags:            0c00    trace level:    4
```

RSVP-TE Phases & States

RSVP-TE Phases & States

RSVP-TE Phase & State Flags

On the RS platform, two phases exist to establish an LSP, *Path Determination Phase* and *Signaling Phase.* There are two fields that help in determining which phase is currently being processed, *State* and *Status.*

To view the state and status information…

RS# mpls show label-switched-paths <name> verbose

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP2"
        state: Null                 lsp-id: 0x9
        status: CspfInProgress
        to: 2.2.2.2                 from: 2.2.2.1
        proto: <rsvp>               protection: none
        setup-pri: 7                hold-pri: 0
        attributes: <FROM_ADDR>
```

```
Path-Signalling-Parameters:
    attributes: <RETRYING ADAPTIVE>
    inherited-attributes: <>
    label in:           label out:
    retry-limit: 5000   retry-int: 15 sec.
    retry-count: 4999   next_retry_int: 12.080000 sec.
    preference: 7       metric: 1
    ott-index: 0        ref-count: 0
    bps: 0              mtu: 1500
    hop-limit: 255      opt-int: 600 sec.
```

Path Determination Phase

The time spent by the instantiating router to execute the CSPF process or validate the ERO. During the path computation the status of the LSP is indicated with a status of *CspfInProgress* and an LSP state of *Null*. The null state indicates that no attempt can be made until the CSPF process has completed. Upon the successful completion of this phase the status field changes to *success.* Should the CSPF function fail the status information will indicate *CspfFail* and a description of the problem.

CSPF Failure Messages

Failure Message	Description
Bandwidth, Include, Exclude	Depends on the request of CSPF
Neighbor not found	Neighbor not found with TE capabilities
System not found	Either of Intermediate/Source/Dest hop systems not known to IGP
IGP is disabled	IGP is currently disabled
CSPF not ready	CSPF is not ready as SPF timer is fired
Hop count exceeded	hop count exceeded the requested number
Maximum hop count exceeded	CSPF maximum hop count exceeded (200)

Strict hop failed	One of the strict hops failed in request
Area not found	PATH might be out of one area
Parameter error	Request does not have valid parameters
Source not known	Intermediate/Source/Dest hop systems not found
PDU parse error	There was a parse error while parsing the LSP/LSA

Signaling Phase

Signaling Phase: After the successful completion of the CSPF process the LSP state indicator changes to *Init.* The initialization phase is the start of the signaling process in which PATH and RESV messages are exchanged.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP2"
        state: Init             lsp-id: 0xd
        status: Success
        to: 2.2.2.2             from: 2.2.2.1
        proto: <rsvp>           protection: none
        setup-pri: 7            hold-pri: 0
        attributes: <FROM_ADDR>

        Path-Signalling-Parameters:
            attributes: <RETRYING ADAPTIVE>
            inherited-attributes: <>
            label in:           label out: 0
            retry-limit: 5000   retry-int: 15 sec.
            retry-count: 4998   next_retry_int: 12.140000 sec.
            preference: 7       metric: 1
            ott-index: 0        ref-count: 0
            bps: 0              mtu: 1500
            hop-limit: 255      opt-int: 600 sec.
            cspf-path:  num-hops: 3
                192.168.1.2     - strict
                192.168.1.1     - strict
```

```
                192.168.1.18    - strict

Transit LSP:

Egress LSP:
```

Upon the success completion of the negotiations, the LSP state changes to *Up*.

A fully functional LSP will look something like this.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP2"
        state: Up               lsp-id: 0xd
        status: Success
        to: 2.2.2.2             from: 2.2.2.1
        proto: <rsvp>           protection: none
        setup-pri: 7            hold-pri: 0
        attributes: <FROM_ADDR>

        Path-Signalling-Parameters:
            attributes: <ADAPTIVE>
            inherited-attributes: <>
            label in:           label out: 17
            retry-limit: 5000   retry-int: 15 sec.
            retry-count: 5000   next_retry_int: 0.000000 sec.
            preference: 7       metric: 1
            ott-index: 1        ref-count: 1
            bps: 0              mtu: 1500
            hop-limit: 255      opt-int: 600 sec.
            cspf-path:  num-hops: 3
                192.168.1.2     - strict
                192.168.1.1     - strict
                192.168.1.18    - strict
            record-route:
                192.168.1.1
                192.168.1.18
```

```
Transit LSP:

Egress LSP:
```

Should a failure occur an error message will be displayed and the LSP will continue to attempt to restart based on the value of the next_retry_int. The next_retry_int is an increasing number, initially set to the retry interval, which grows with each subsequent failure.

Path Protection using RSVP-TE

Path Protecting using RSVP-TE

Designating Backup Paths

Creating a strict explicit path using only a single primary path provides pinpoint control over how the data associated to the label switch path will flow. However, in the even of failure service must be able to recover without manual intervention. Loose explicit routing allows RSVP-TE to signal a new path around a failure, at the expense of control.

There are two categories assigned to paths when assigned to an LSP, primary or secondary. There can only be a single primary path associated to an LSP. It is always preferred over any secondarily designated path. There can be numerous secondary paths associated to an LSP, with a configurable order of preference. The obvious benefit to this approach is the ability to define disparate paths across the backbone, should the network have such a physical configuration.

To associate an existing path to an LSP as a primary or secondary …

```
RS(config)# mpls set label-switched-path <name> primary|secondary
<path name>
```

The secondary paths are selected based on preference, with the higher numerical value preferred. To define the preference of secondary paths…

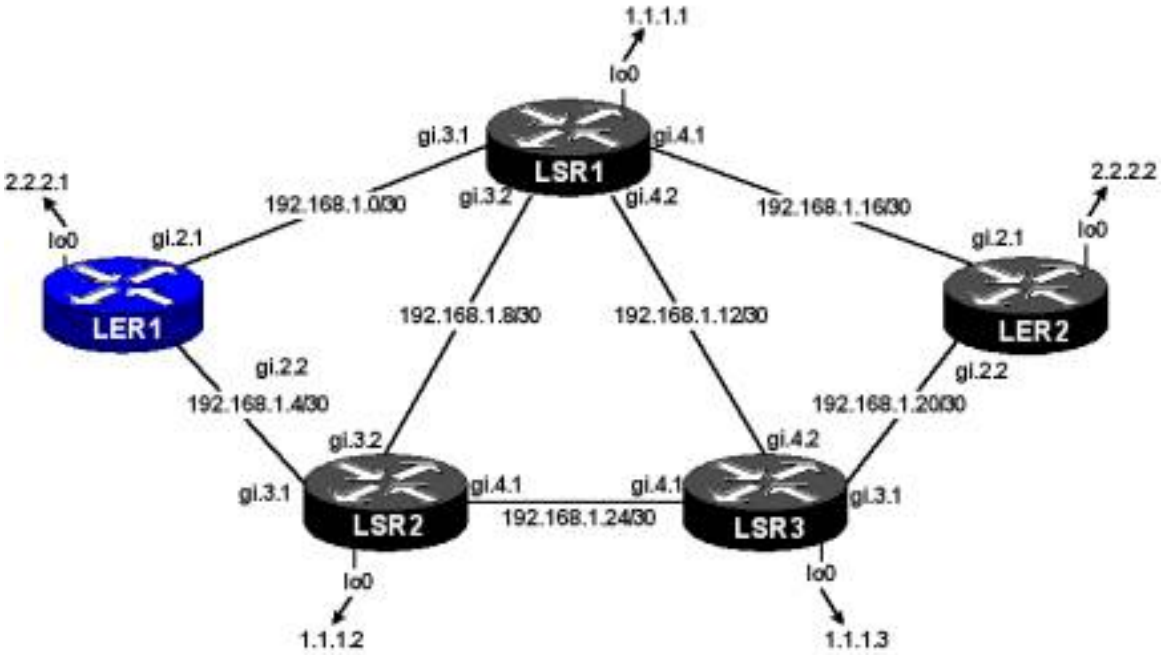
```
RS(config)# mpls set label-switched-path <name> secondary <path
```

name> preference <1-255>

Finally, the secondary path may be defined as a hot or cold standby. A cold standby is the default action and will not be established until the primary has failed. This means, once the instantiating router realizes the primary path is no longer valid the preferred secondary must be signaled and established before it can be used. A hot standby is the ability to have a pre-established backup that takes for a failed primary as soon as it is recognized. To configure a hot standby…

RS(config)# mpls set label-switched-path <name> secondary <path name> preference <1-255> standby

One possible solution may be to define a completely explicit primary route that is best suited for the traffic carried within the LSP. To protect the primary path a completely disparate path capable of servicing the LSP may exist and be explicitly configured as a preferred secondary. Since there are no common transit points it may make sense to configure this secondary path as a hot standby. Finally, a less preferred hop-by-hop backup path may be configured as a backup of last resort. It main role is to try to establish a path through a network that has already observed multiple failures that have impacted service on the primary and preferred secondary paths. It is pointless to pre-establish this path because it would be unforeseeable to predetermine what paths may remain from ingress to egress.




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create path ERO-Path1 num-hops 4
mpls create path ERO-Path2 num-hops 3
mpls create path Path-Back1
mpls set path ERO-Path1 hop 1 ip-addr 192.168.1.6 type strict
mpls set path ERO-Path1 hop 2 ip-addr 192.168.1.5 type strict
mpls set path ERO-Path1 hop 3 ip-addr 192.168.1.26 type strict
mpls set path ERO-Path1 hop 4 ip-addr 192.168.1.22 type strict
mpls set path ERO-Path2 hop 1 ip-addr 192.168.1.2 type strict
mpls set path ERO-Path2 hop 2 ip-addr 192.168.1.1 type strict
mpls set path ERO-Path2 hop 3 ip-addr 192.168.1.18 type strict
mpls create label-switched-path LSP from 2.2.2.1 to 2.2.2.2 no-
cspf
```

```
mpls set label-switched-path LSP primary ERO-Path1
mpls set label-switched-path LSP secondary ERO-Path2 preference
100 standby
mpls set label-switched-path LSP secondary Path-Back1 preference
10
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

A look at the high level LSP information only indicates the LSP based information, not the underlying path information.

```
LER1# mpls show label-switched-paths all

Ingress LSP:
LSPname                    To                 From
State LabelIn LabelOut
LSP                        2.2.2.2            2.2.2.1
Up    -       17

Transit LSP:
LSPname                    To                 From
State LabelIn LabelOut

Egress LSP:
LSPname                    To                 From
State LabelIn LabelOut
```

In order to check the path information the "*verbose*" option must be coded as part of the display command. When a specific path is being used to forward traffic for the LSP the status of the protection-path field state is <Active, disposition>. In steady state, the primary path will be filling the active role. However, during times of network issues one of the secondary paths will enter the active state, based on order of preference. Console messaging will show transition events.

```
LER1# mpls show label-switched-paths all verbose
```

```
Ingress LSP:

    Label-Switched-Path: "LSP"
        state: Up                lsp-id: 0xa
        status: Success
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: primary
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR PRI SEC>

        Protection-Path "ERO-Path1": <Active, Primary>
            state: Up            lsp-id: 0x4004
            status: Success
            attributes: <>
            inherited-attributes: <>
        Path-Signalling-Parameters:
            attributes: <>
            inherited-attributes: <NO-CSPF>
            label in:            label out: 17
            retry-limit: 5000    retry-int: 15 sec.
            retry-count: 5000    next_retry_int: 0.000000 sec.
            preference: 7        metric: 1
            ott-index: 3         ref-count: 1
            bps: 0               mtu: 1500
            hop-limit: 255       opt-int: 600 sec.
            explicit-path: "ERO-Path1"  num-hops: 4
                192.168.1.6      - strict
                192.168.1.5      - strict
                192.168.1.26     - strict
                192.168.1.22     - strict
            record-route:
                192.168.1.5
                192.168.1.26
                192.168.1.22

        Protection-Path "Path-Back1": <Secondary>
            state: Null          lsp-id: 0x4015
            status: Success
            attributes: <PREF>
            inherited-attributes: <>
```

```
        Path-Signalling-Parameters:
            attributes: <>
            inherited-attributes: <NO-CSPF>
            label in:           label out:
            retry-limit: 5000   retry-int: 15 sec.
            retry-count: 5000   next_retry_int: 0.000000 sec.
            preference: 10      metric: 1
            ott-index: 0        ref-count: 0
            bps: 0              mtu: 1500
            hop-limit: 255      opt-int: 600 sec.
            explicit-path: "Path-Back1" num-hops: 0

        Protection-Path "ERO-Path2": <Secondary>
            state: Up           lsp-id: 0x4012
            status: Success
            attributes: <PREF>
            inherited-attributes: <>
        Path-Signalling-Parameters:
            attributes: <STANDBY>
            inherited-attributes: <NO-CSPF>
            label in:           label out: 17
            retry-limit: 5000   retry-int: 15 sec.
            retry-count: 5000   next_retry_int: 0.000000 sec.
            preference: 100     metric: 1
            ott-index: 1        ref-count: 1
            bps: 0              mtu: 1500
            hop-limit: 255      opt-int: 600 sec.
            explicit-path: "ERO-Path2"  num-hops: 3
                192.168.1.2     - strict
                192.168.1.1     - strict
                192.168.1.18    - strict
            record-route:
                192.168.1.1
                192.168.1.18

Transit LSP:

Egress LSP:
```

When a primary fails and one of the secondary paths assumes the active role a

message is written to the console indicating the primary has failed and which secondary path is taking over.

```
%MPLS-I-LSPPATHSWITCH, LSP "LSP" switching to Secondary Path "ERO-Path2".
```

When the primary path is re-established, it automatically assumes the active role and the associated message is written to the console.

```
MPLS-I-LSPPATHSWITCH, LSP "LSP" switching to Primary Path "ERO-Path1".
```

The default switch back from secondary to primary can be overridden. If the overrides are in place to prevent the automatic switch back a failure along the active secondary path or manual intervention will cause the secondary to switch back to the primary. Manual intervention requires an operator to use the comment command to remove the active secondary path statement from the configuration. This will immediately fail the path back to the primary. The command line in the configuration can be commented back in immediately following the act of commenting it out.

To override the automatic switch back function…

RS(config)# mpls set label-switched-path <name> no-swicthback

The use of the comment command…

RS(config)# comment in|out <linenum>

Fast Re-route and Detour LSP

The Internet draft [draft-gan-fast-reroute-00.txt] defines a process that allows intermediate nodes along a main LSP to pre-establish detours around possible failure points. It introduces two new RSVP objects.

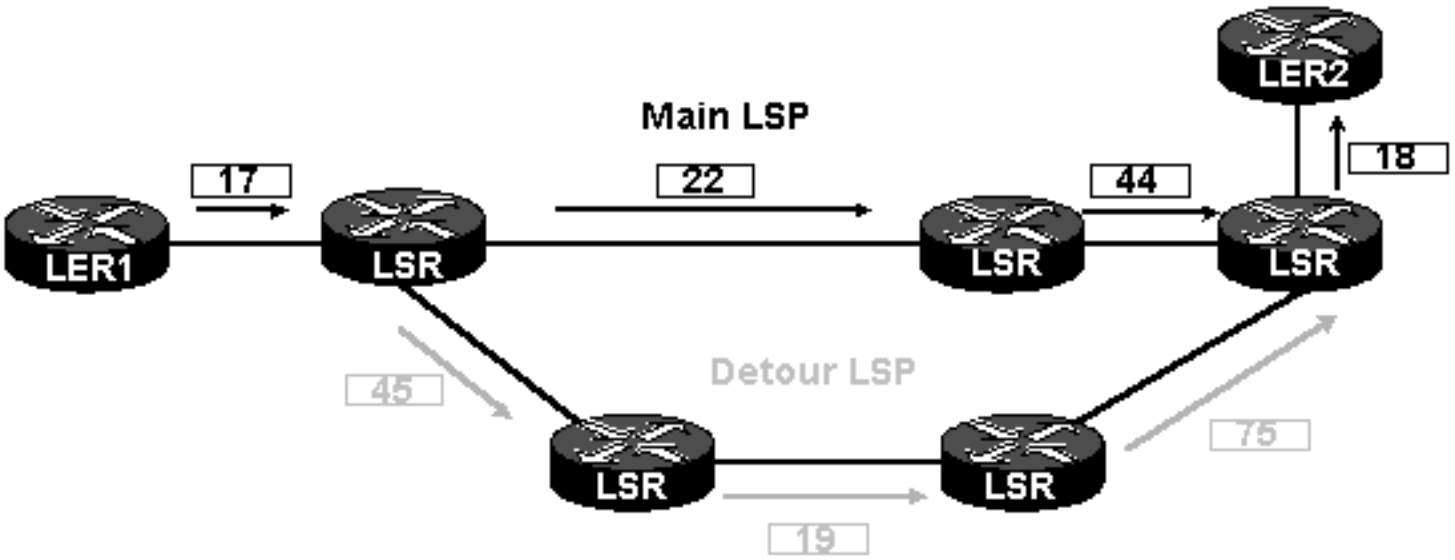
Fast Reroute: This is the trigger carried in the RSVP session information to indicate the main LSP requires a detour LSP to be pre-established across the components. This object includes all the necessary information to allow transit routers to execute the CPSF process and select detours that meet the criteria of the main LSP. The object information includes setup and hold priorities, bandwidth and link affinity (include and exclude). The constraint based information is not required to be the same on the detour path as it is on the main LSP. If these are not specifically configured when fast reroute is configured, the constraints are inherited.

Detour: Includes the IP address of the router requesting the detour LSP and the destination that should be bypassed in the event of failure. The goal is to create the detour around the link and the immediate next hop.

Merge Node: The node where the detour and the main path join. The merge node is responsible for mapping both the main LSP inbound label and the detour LSP inbound label to the same outbound action. On the RS platform this will mean, multiple related hw-ilm-tbl entries will share a common hw-ott-index, which determines the outbound action.

Branch Node: The node where a detour LSP is instantiated to protect the main path.

The simple diagram below represents a main LSP protected by a detour LSP. If the main LSP traverses a more complex longer path, detour LSPs will be established around all possible failure points that have an alternate available.



An older Internet draft (*draft-swallow-rsvp-bypass-label-01.txt – November 2000*) defines a different approach to the fast reroute concept. With the older approach, the detour LSP uses label stacking to create the "bypass" LSP. Encapsulating the main LSP label within the label for the bypass LSP the penultimate hop on the bypass LSP removes the top level label, the bypass tunnel label, delivering the main LSP label back to the merge node.

There is a possibility these drafts may merge into a single specification.

Support for the "Gan" draft is being developed for the RS code base.

RSVP-TE Path Pre-emption and Priorities

RSVP-TE Path Pre-emption and Priorities

Path Pre-emption & Priorities

Path Pre-emption & Priorities

RSVP-TE provides a means to ensure higher priority LSPs take precedence over lower priority ones when network resources are oversubscribed. Each LSP contains includes a setup and hold priority as part of the session information. LSPs are established based on setup priority and maintain their reservations base on the hold priority. This means if an LSP were unable to establish a path through the network due to a lack of resources, its setup priority would be compared to the hold priority of established paths. If the setup priority of the failed LSP were a higher priority, indicated by a lower numerical value, than the hold priority of an existing LSP pre-emption could occur.

Setup Priority: A numerical value in the range of 0-7, with 0 being the highest possible value.

Hold Priority: A numerical value in the range of 0-7, with 0 being the highest possible value.

By default pre-emption is not allowed because each LSP has a setup priority of 7, the lowest, and a hold priority of 0, the highest. Should a setup priority equal a hold priority pre-emption does not occur. The RS platform does not allow these values to be changed in the current release.

To display the values of the setup and hold priorities for an established LSP…

RS# rsvp show psb <name|all> <options>

```
LER1# rsvp show psb all verbose
Path State Blocks:
--------------------
RSVP_PSB <rsvp_1>: (psb = 0x82a6b5a8)
        session-attr: name: LSP1 flags: 0x0 setup-pri: 7 holding-
pri: 0
        session: end-point: 2.2.2.2 tunnel-id: 14 ext-tunnel-id:
0x2020201
        send-templ: sender: 2.2.2.1 lsp-id: 97
        prev-hop: 0.0.0.0 lih: 0
        in-if: <Local-API> out-if: <To-LSR1>
        explicit-route: 192.168.1.1=>192.168.1.18
        sender-tspec: qos: CL cdr: 0 pbs: 0 pdr: 0 mpu: 20 mtu:
1436
        block-tspec:
        psb refresh timer:  time-to-expire: 8.290000 sec.
        psb cleanup timer:  time-to-expire: 134.540000 sec.
        ref-count: 1
        LSP-handle: 0x82a57018
        Session: 0x82a6b790
```

Authenticating RSVP-TE Sessions

Authenticating RSVP-TE Sessions

Authentication

RSVP-TE sessions do not have the connection-oriented level of security that comes with using TCP as a reliable transport. This raw IP based protocol, must rely on other security mechanisms. The RS platform provides a level of protection allowing the RSVP sessions to be protected by MD5 authentication. This ensures, only nodes that are specifically code with the shared secrete and running the RSVP-TE protocol will be able to authenticate and participate in the RSVP-TE process. Of course the "*auth-key*" is case sensitive and both it and the "*auth-method*" must be the same on RSVP-TE neighbors that wish to communicate using the RSVP-TE protocol.

WARNING: If authentication is being applied on top of interfaces that have established LSPs, they will enter a failed state until the peers have been authenticated. The following error message is issued after the configuration is saved to alert you to the condition.

```
%MPLS-W-RSVPNEWAUTHKEY, Setting new authentication key. Existing
TE-RSVP LSPs on interface To-LSR3 may be torn down
```

The below demonstrates the issue where one side is using MD5 security and the peer on the other side is not.

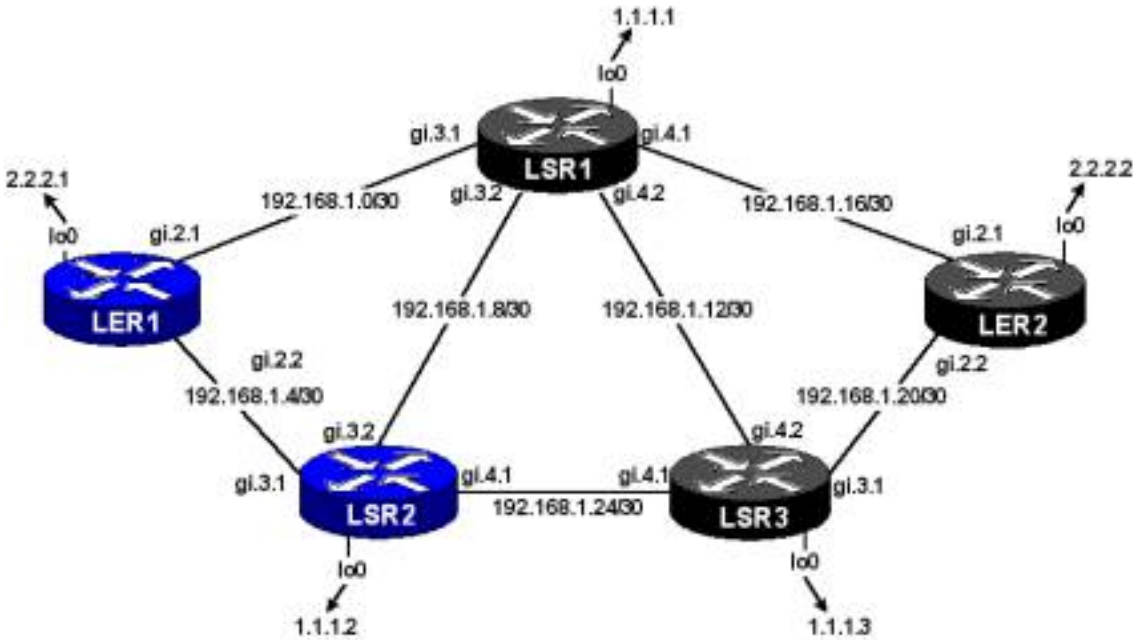
```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:
    Label-Switched-Path: "LSP"
        state: Down                lsp-id: 0x6
        status: RoutingProblem, NeighborFail
        to: 2.2.2.2                from: 2.2.2.1
        proto: <rsvp>              protection: none
        setup-pri: 7               hold-pri: 0
        attributes: <FROM_ADDR>
```

Enabling MD5 Authentication

Both peers sharing a link require the same security mechanism, whether that is MD5 or none. By default, authentication is disabled on the RS platform.

To enable MD5 authentication…

RS(config)# rsvp set interface <interface-name or IP |all> auth-key <key> auth-method <md5|none>



The RSVP-TE configurations to enable MD5 authentication is presented using the two hi-lighted routers above.

RSVP-TE configuration for LER1…

```
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp set interface To-LSR1 auth-key Unique2 auth-method md5
rsvp set interface To-LSR2 auth-key Unique3 auth-method md5
rsvp start
```

RSVP-TE configuration for LSR2…

```
rsvp add interface To-LER1
rsvp add interface To-LSR1
rsvp add interface To-LSR3
rsvp set interface To-LER1 auth-key Unique3 auth-method md5
rsvp set interface To-LSR1 auth-key Unique1 auth-method md5
rsvp set interface To-LSR3 auth-key Unique4 auth-method md5
rsvp start
```

Use the interface show command to display the RSVP attributes applied to an interface. The attributes list to two RSVP-TE enabled interfaces with authentication enabled.

```
RS# rsvp show interface <name|all> <options>
```

```
LER1#   rsvp show interface
Interface         Type       Attributes                    Path-
MTU
-------------------------------------------------------------------
To-LSR1           Enet/POS   <AUTH_EN>                     1436
To-LSR2           Enet/POS   <AUTH_EN>                     1436
```

For a more detailed view add the *verbose* option to the display command.

```
RS# rsvp show interface all verbose
```

```
LER1# rsvp show interface all verbose
RSVP Interface Configuration:
-----------------------------
```

To-LSR1

type:	Enet/POS	
attributes:	**<AUTH_EN>**	
path-mtu:	1436	
auth-key:	**Unique2**	
path-vector-limit:	8	
hop-count-limit:	255	
rapid-retransmit-interval:	1000	sec.
rapid-retransmit-delta:	2	sec.
rapid-retry-limit:	3	
current-msg-id:	0x98	
epoch:	0xb09	
seq-no:	0x0:0x43757265	

To-LSR2

type:	Enet/POS	
attributes:	**<AUTH_EN>**	
path-mtu:	1436	
auth-key:	**Unique3**	
path-vector-limit:	8	
hop-count-limit:	255	

Constraint Based Routing

Constraint Based Routing

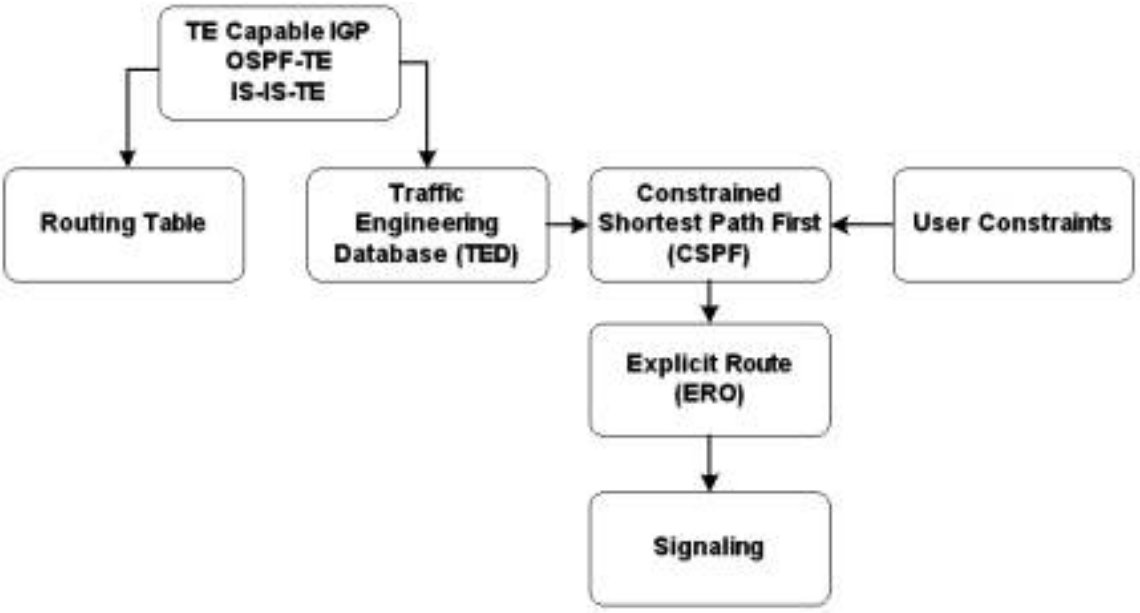
Introduction to CSPF

Introduction to CSPF

Traditional traffic engineering exercises require a complete, detailed and current understanding of the traffic patterns on the network. Constant tweaking and offline computations enable the proper mapping of customers to network resources, at a very high level. Using the traffic engineering found in the MPLS signaling protocols and link state routing protocols more tools are provided to help alleviate some of the burden. This by no means eliminates the need to understand the capacity of the overall network and the foundation on which it is built and its growth patterns. The new tools do allow an instantiating router, configured with some network resource allocations or constraints, to compute an acceptable path matching the requirements to available resources that exist end-to-end across the network.

The *Constrain-based Shortest Path First, or CSPF,* process is an extension to SPF process performed by link state databases. The CSPF calculation uses local configuration as input and computes the shortest path through the network that matches the configured requirements.

The decision process can be viewed as…



The RS platform supports two different types of constraints. The first relies on the IGP to carry information in link state updates used to populate the *traffic engineering database, or ted*. The second type of constraint compares a local constraint to the link state database to determine which paths should be accepts or rejected. Constrains are a means to prune the link state database of any links that do not meet the requirement specified for an LSP.

The RS platform supports the following constraints…

Admin Groups	list	Including and excluding link colors & resource classes	Carried in LSA
Bandwidth	value	Bandwidth requested	Carried n LSA
Hop Limit	number	Maximum number of hops from ingress to egress	Local

Configuring Link Affinity (Admin Groups)

Configuring Link Affinity (Admin Groups)

Link Affinity (Admin Groups)

Before a link can join a membership the administrative group must be defined. Up to 32 link memberships can be created. With the administrative groups in place, the interfaces are associated to the memberships to which they belong. An interface can have multiple memberships.

Names used to create the groups are symbolic, easy to remember placeholders that are not distributed in the link state advertisement. The actual 32 bit mask is announced using the LSA. To create the admin groups…

```
RS(config)# mpls create admin-group <name> group-value <0-31>
```

To join an interface to a membership…

```
RS(config)# mpls set interface <name|all> adming-group
<name|list>
```

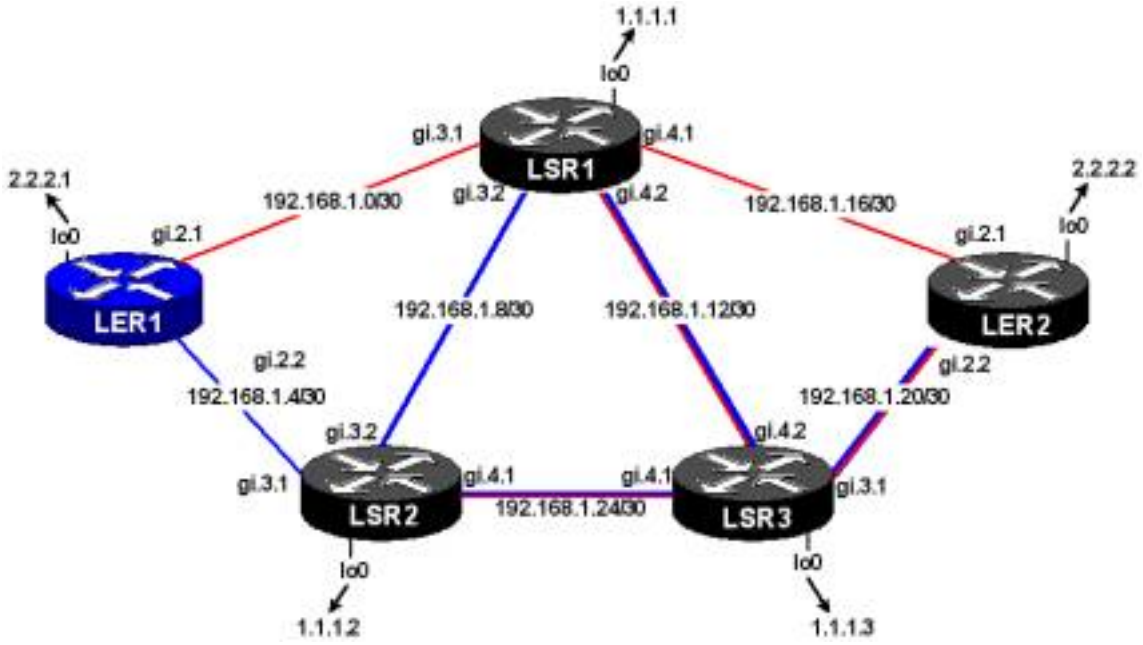
When an LSP is set to include certain memberships it will only consider interfaces with that membership when creating the LSP. Similarly, when certain memberships are excluded those links are pruned form the possible interfaces over which the LSP may be established. If an interfaces belongs to both the accepted membership, the include, but also belongs to a membership that is excluded the interface will not be considered for inclusion. The explicit exclude

has a higher precedence over an explicit include. When an LSP does not apply administrative group requirements, the CSPF process ignores the affinity information in the ted.

To apply link affinity to an LSP…

RS(config)# mpls set label-switched-path <name> admin-group <name|list> include|exclude

The following example creates two label switched paths that use differing administrative groups. Nothing is explicitly excluded. This means any interface that belongs to that membership will be considered for the LSP. All routers have defined interface memberships as depicted in the diagram below. A sample configuration of one of the routers is provided.




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
```

```
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls create admin-group Red group-value 0
mpls create admin-group Blue group-value 1
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls set interface To-LSR1 admin-group Red
mpls set interface To-LSR2 admin-group Blue
mpls create label-switched-path LSP1 from 2.2.2.1 to 2.2.2.2
mpls create label-switched-path LSP2 from 2.2.2.1 to 2.2.2.2
mpls set label-switched-path LSP1 include Red
mpls set label-switched-path LSP2 include Blue
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

A detailed look at the LSP shows the path chosen by the CSPF process, which was sent in the RSVP PATH message as an explicit route object.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:
    Label-Switched-Path: "LSP2"
        state: Up                lsp-id: 0xb
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: none
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR INCLUDE>

        Path-Signalling-Parameters:
            attributes: < >
            inherited-attributes: <>
            retry-limit: 5000    retry-int: 15 sec.
            retry-count: 5000    next_retry_int: 0.000000 sec.
            bps: 0               preference: 7
            hop-limit: 255       opt-int: 600 sec.
            include:             Blue
            mtu: 1500
```

```
            ott-index: 2        ref-count: 1
            cspf-path:  num-hops: 4
                192.168.1.6     - strict(blue)
                192.168.1.5     - strict(blue)
                192.168.1.26    - strict(red,
blue)
                192.168.1.22    - strict(red, blue)
            record-route:
                192.168.1.5
                192.168.1.26
                192.168.1.22


    Label-Switched-Path: "LSP1"
        state: Up               lsp-id: 0xa
        to: 2.2.2.2             from: 2.2.2.1
        proto: <rsvp>           protection: none
        setup-pri: 7            hold-pri: 0
        attributes: <FROM_ADDR INCLUDE>

        Path-Signalling-Parameters:
            attributes: < >
            inherited-attributes: <>
            retry-limit: 5000   retry-int: 15 sec.
            retry-count: 5000   next_retry_int: 0.000000 sec.
            bps: 0              preference: 7
            hop-limit: 255      opt-int: 600 sec.
            include:            Red
            mtu: 1500
            ott-index: 1        ref-count: 1
            cspf-path:  num-hops: 3
                192.168.1.2     - strict(red)
                192.168.1.1     - strict(red)
                192.168.1.18    - strict(red)
            record-route:
                192.168.1.1
                192.168.1.18
```

When displaying the information in the ted the *Resource Class* value is a hex representation of the 32 bit *Resource Class or Link Color* sub-TLV that was

received as part of the link state advertisement. To determine interface membership using the information in the ted, you must convert the hex value to a binary. Any position that has a value of one indicates that interface is a member of that admin group. The following details the interface membership based on the ted data.

Network	Admin-Group	Hex	32 Bit Binary
192.168.1.0/30	Red	1	00000000000000000000000000000001
192.168.1.4/30	Blue	2	00000000000000000000000000000010
192.168.1.8/30	Blue	2	00000000000000000000000000000010
192.168.1.12/30	Red, Blue	3	00000000000000000000000000000011
192.168.1.16/30	Red	1	00000000000000000000000000000001
192.168.1.20/30	Red, Blue	3	00000000000000000000000000000011
192.168.1.24/30	Red, Blue	3	00000000000000000000000000000011

Bandwidth Constraint

Bandwidth Constraint

Bandwidth

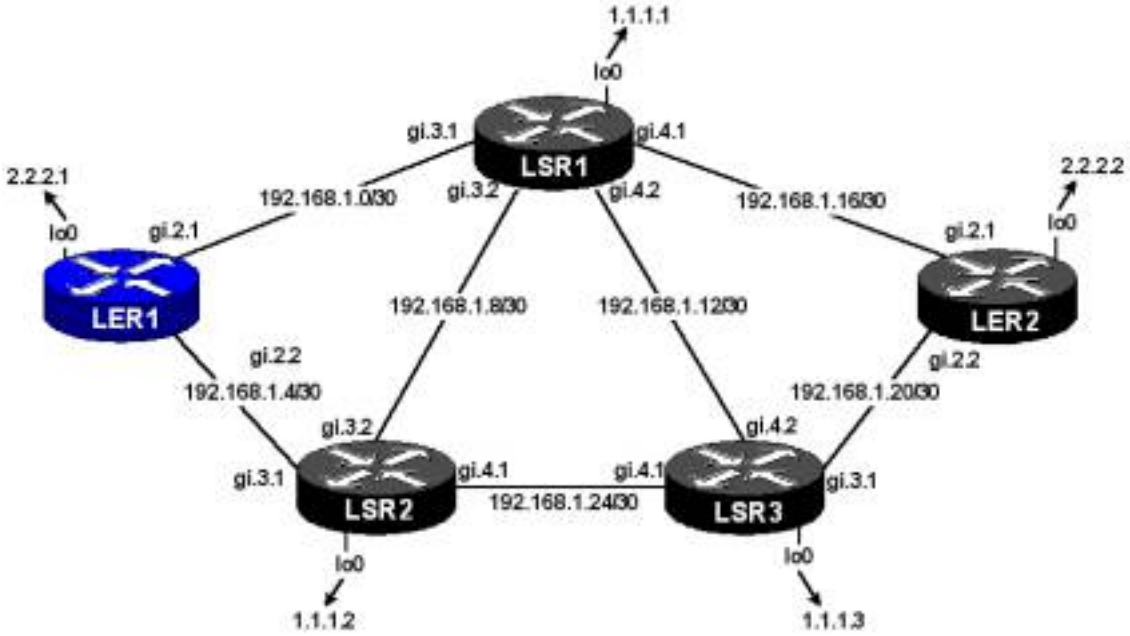
Bandwidth

This represents a bps resource request from end-to-end.

To configure an LSP with a bandwidth requirement…

```
RS(config)# mpls <create|set> label-switched-path <name|all> bps
<numer>
```

The sample router configuration below is deigned so the two LSPs will not be able to share any common network links, based on the size bandwidth requests. The CSPF process will establish the LSPs across completely separate links.



```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create label-switched-path LSP1 from 2.2.2.1 to 2.2.2.2 bps
750000000
mpls create label-switched-path LSP2 from 2.2.2.1 to 2.2.2.2 bps
500000000
mpls start
rsvp add interface To-LSR1

rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

Looking at the detailed LSP information, the cspf-path, also the explicit route object, shows that both LSPs been established across disparate network elements, each receiving the required bandwidth.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:

    Label-Switched-Path: "LSP2"
        state: Up                lsp-id: 0xd
        to: 2.2.2.2              from: 2.2.2.1
        proto: <rsvp>            protection: none
        setup-pri: 7             hold-pri: 0
        attributes: <FROM_ADDR BPS>

        Path-Signalling-Parameters:
            attributes: < >
```

```
        inherited-attributes: <>
        retry-limit: 5000   retry-int: 15 sec.
        retry-count: 5000   next_retry_int: 0.000000 sec.
        bps: 500000000      preference: 7
        hop-limit: 255      opt-int: 600 sec.
        mtu: 1500
        ott-index: 2        ref-count: 1
        cspf-path:  num-hops: 4
            192.168.1.6     - strict
            192.168.1.5     - strict
            192.168.1.26    - strict
            192.168.1.22    - strict
        record-route:
            192.168.1.5
            192.168.1.26
            192.168.1.22

Label-Switched-Path: "LSP1"
    state: Up               lsp-id: 0xc
    to: 2.2.2.2             from: 2.2.2.1
    proto: <rsvp>           protection: none
    setup-pri: 7            hold-pri: 0
    attributes: <FROM_ADDR BPS>

    Path-Signalling-Parameters:
        attributes: < >
        inherited-attributes: <>
        retry-limit: 5000   retry-int: 15 sec.
        retry-count: 5000   next_retry_int: 0.000000 sec.
        bps: 750000000      preference: 7
        hop-limit: 255      opt-int: 600 sec.
        mtu: 1500
        ott-index: 1        ref-count: 1
        cspf-path:  num-hops: 3

            192.168.1.2     - strict
            192.168.1.1     - strict
            192.168.1.18    - strict
        record-route:
            192.168.1.1
```

192.168.1.18

Hop Count Constraint

Hop Count Constraint

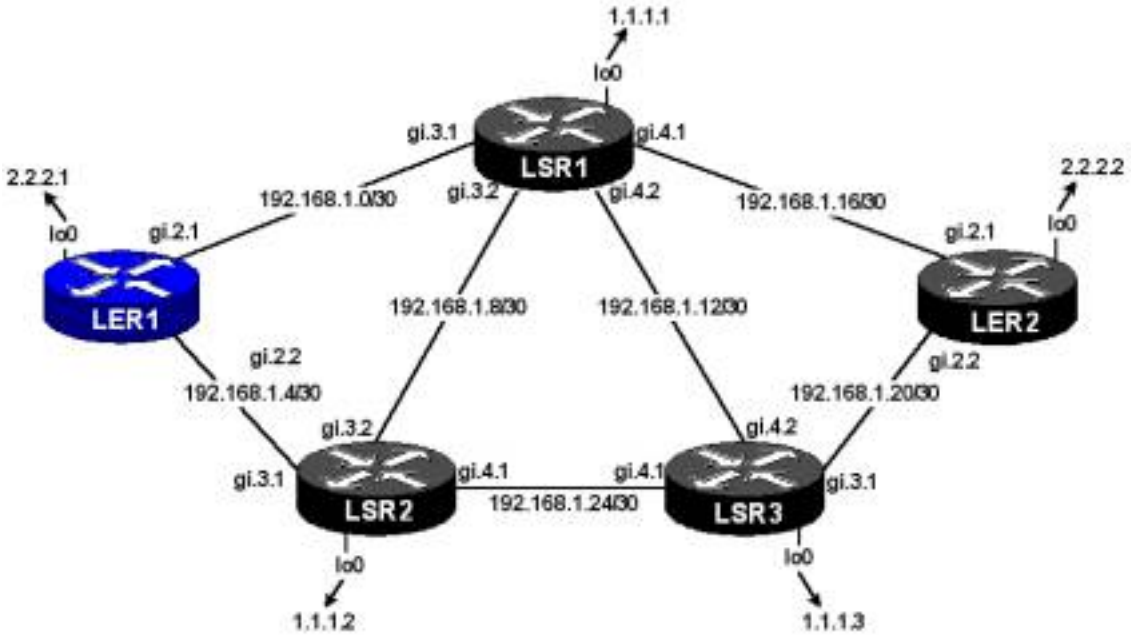
Hop Limit

Hop Limit

Defining a hop limit confines the CSFP process to consider only paths that do not exceed the specified number of hops. This information is not included in any link state advertisement. Rather it is applied locally at the ingress router against routes in the link state database. If a hop count limit is not coded, then the maximum number of hops, 255, is used as the input to the CSFP process. It is important to remember, the number of hops starts with the local interface of the instantiating router.

To configure an LSP with a hop count constraint…

```
RS(config)# mpls <create|set> label-switched-path <name|all> hop-limit <1-255>
```




```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create label-switched-path To-LER2-1 from 2.2.2.1 to 2.2.2.2
hop-limit 4
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ospf set traffic-engineering on
```

A detailed look at the LSP shows the resulting cspf-path, also the explicit route.

```
LER1# mpls show label-switched-paths all verbose
Ingress LSP:
```

```
Label-Switched-Path: "To-LER2-1"
    state: Up                lsp-id: 0x6
    to: 2.2.2.2              from: 2.2.2.1
    proto: <rsvp>            protection: none
    setup-pri: 7             hold-pri: 0
    attributes: <FROM_ADDR HOP_LIMIT>

    Path-Signalling-Parameters:
        attributes: < >
        inherited-attributes: <>
        retry-limit: 5000    retry-int: 15 sec.
        retry-count: 5000    next_retry_int: 0.000000 sec.
        bps: 0               preference: 7
        hop-limit: 4         opt-int: 600 sec.
        mtu: 1500
        ott-index: 1         ref-count: 1
        cspf-path:  num-hops: 3
            192.168.1.2      - strict
            192.168.1.1      - strict
            192.168.1.18     - strict
        record-route:
            192.168.1.1
            192.168.1.18
```

Layer Two Virtual Private Networks Introduction

Layer Two Virtual Private Networks Introduction

Introduction

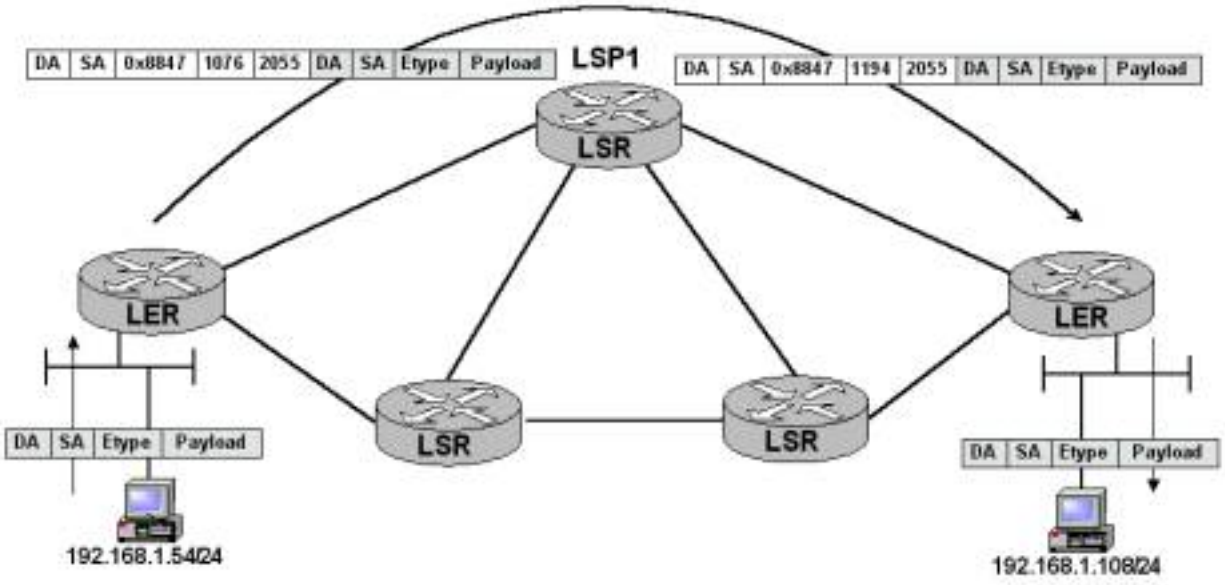
Introduction

One of the keys building blocks of MPLS is the separation of the forwarding mechanism from the layer three network protocols. The label implanted between the layer two and layer three OSI information, or as part of the native layer two header, is used to switch the packet across the MPLS network. Core transit routers switch packets based on the label without relying on an upper layer network protocol. The network layer protocol information is hidden, encapsulated inside the MPLS envelop. *Layer Two Virtual Private Networks, L2 VPN*, takes full advantage of this. The entire inbound packet is wrapped in an MPLS header and forwarded through the MPLS core. Reaching the other side, the labels are removed and the packet that arrives at the ultimate destination exactly the same as the packet that entered the MPLS network at the ingress. This represents the layer two portion of the L2 VPN service.

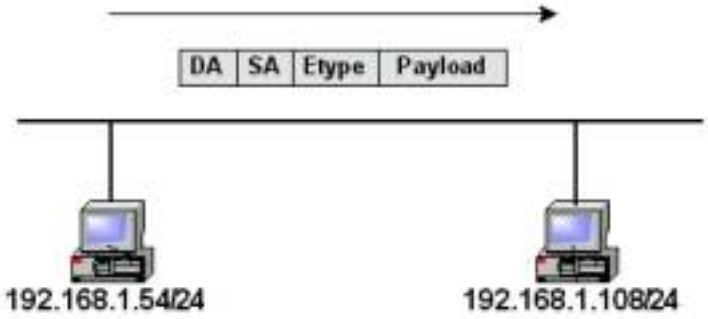
Applying different labels to each individual customer that uses the shared network provides the privacy element, analogous to ATM or Frame Relay. Consider a simple definition of a VPN in the context of MPLS simply as customer separation using labels. It does not encompass the security encryption mechanisms found in protocols like IPSec. However, there is nothing to preclude the clients from using encryption to communicate with each other across an L2

VPN. The complexities of the core MPLS network are hidden from the end nodes.

The end nodes don’t see this.



They see this.



Virtual Lease Lines - Martini

Virtual Leased Lines - Martini

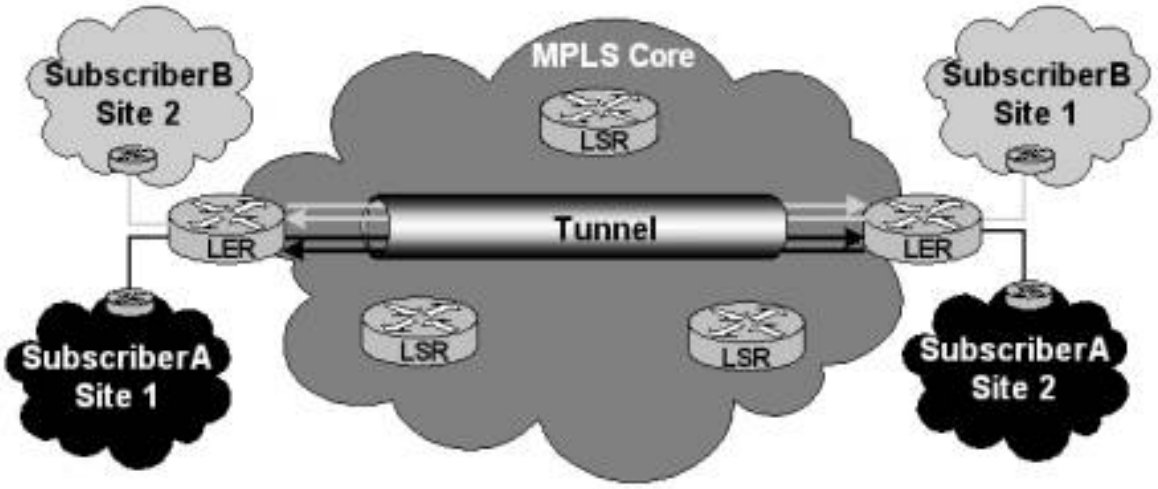
“Martini” Drafts

draft-martini-l2circuit-encap-mpls-04.txt defines the handling and encapsulation of layer two packets.

draft-martini-l2circuit-trans-mpls-08.txt defines the signaling and forwarding of traffic across the MPLS network.

In order to provide L2 VPN services across an MPLS network a methodology needed to be defined. The two martini drafts define the different encapsulation

techniques accompanied by signaling and transport functions. They combine to form the basis for point-to-point layer two services across an MPLS network, equivalent in concept to ATM or Frame Relay. The point-to-point service is facilitated through a pair of LSPs in opposite directions, which form a single virtual pipe. Label stacking is used to create hierarchies separating the common tunnel LSP and the virtual channels that exist inside. The *VC Label* represents each virtual channel. This label is used by the egress in order to map the channel to the individual *Group ID and Virtual Circuit ID, or VCID*. If present the Group ID is a collection of VC IDs, which identifies the different services within a single VC Label. If the Group ID is not present the VC ID represents the service.



Tunnel LSP Creation

The *Tunnel LSP* is the point-to-point connection within which individual customer virtual circuits will exist. The tunnel LSP scales the network core by aggregating many virtual circuits into a single common tunnel LSP. It is not feasible to have a tunnel per customer, or even worse, a tunnel per customer per service. A model that followed that direction is doomed to run into scalability issues early on in the deployment. The protocol with which to create the tunnel LSP is left to the discretion of the implementer and is largely based on the question, which protocol services the needs better. For example, if traffic engineering, including the signaling explicit paths is required, RSVP-TE provides the solution. However, if

it is acceptable to allow the IGP to make the hop-by-hop decisions and end-to-end path significance is not required LDP may be acceptable. The RS platform supports LDP over LDP and LDP over RSVP-TE, so either LDP or RSVP-TE can be used for the tunnel LSP.

Signaling the VC Label

A peering session must exist for LDP information to be exchanged between LDP capable nodes. Since the edges of the network represent remote LDP peers the configuration must explicitly instruct nodes to form a peering relationship using extended discovery. Once the peering session has been established Virtual Circuit Labels, *VC Labels,* can be exchanged to identify the components of the individual L2-FEC. The signaling of the VC Label is performed using LDP extensions to the *Label Mapping* message. When an edge router is configured with a new L2-FEC, a local LDP label is selected form the database and using the extensions to the Label Mapping Message, all the information relating to the L2-FEC (Group ID, VC ID, Interface and other) is forwarded to the remote peer.

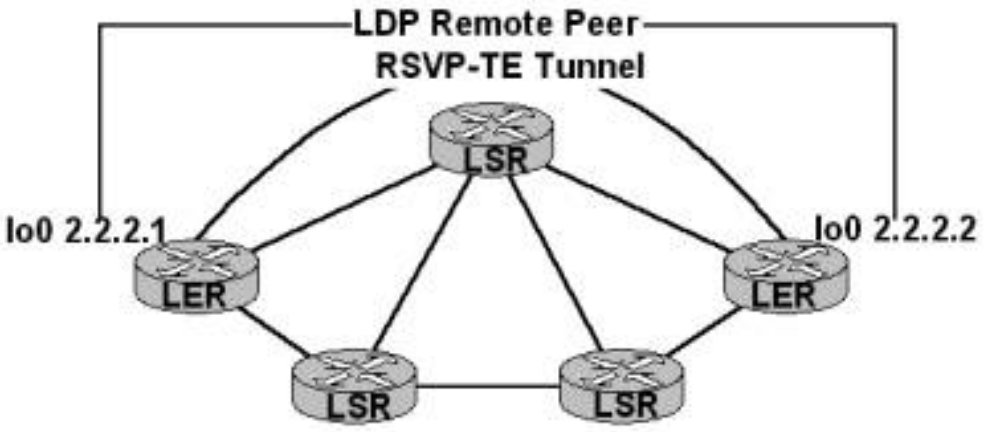
The RS supports the following types of layer two FECs.

RS L2-FEC	Description	Group ID	VC ID
VLAN	802.1Q VLAN		Value of VLAN ID
Customer-Id	Physical Port		Value of Customer-id configured
Customer-Id, VLAN	Physical Port, 802.1Q VLAN	Value of Customer-id configured	Value of VLAN ID

Monitoring the Signaling Process

Starting with the base network having only the tunnel LSP and the remote LDP peering session established, we examine the effects on the LDP database as we start to add L2-FECs to the router configurations. No L2-Fecs have been created

at this time.



With just the base network configured, the LDP databases should receive (Input label database) the remote peers loopback address and send (Output label database) the remote peer the local loopback address.

lo0 2.2.2.1	**lo0 2.2.2.2**

`LER1# ldp show database`
 	
`Input label database, 2.2.2.1:0-2.2.2.2:0`
`  Label     Prefix`
`      3     2.2.2.2/32`
 	
`Output label database, 2.2.2.1:0-2.2.2.2:0`
`  Label     Prefix`
`      3     2.2.2.1/32`
	`LER2# ldp show database`
	
	`Input label database, 2.2.2.2:0-2.2.2.1:0`
	`  Label     Prefix`
	`      3     2.2.2.1/32`
	
	`Output label database, 2.2.2.2:0-2.2.2.1:0`
	`  Label     Prefix`
	`      3     2.2.2.2/32`

If the following L2-FEC is added to the configuration on router 2.2.2.1…

```
ldp map ports et.3.(5-6) customer-id 2001
ldp add l2-fec customer-id 2001 to-peer 2.2.2.2
```

A *Label Mapping* message will be sent to peer 2.2.2.2 binding label to FEC, in this case the VC ID, “customer-id 2001”.

```
LDP Send Label Mapping msg on Interface lo, Remote Neighbor
Our LDP Id: 2.2.2.1:0 Peer LDP Id: 2.2.2.2:0
Label: 2048
 FEC:
Customer-id 2001
```

The LDP database will reflect the addition of the FEC. The L2-FEC must be configured on both routers to allow for traffic to reach the ultimate destination. If the L2-FEC is only created on one of the LDP peers the packets will reach the destination with the appropriate VC Label however there are no associated output ports possible for the received L2-FEC. The packet would be dropped at the egress node.

lo0 2.2.2.1	**lo0 2.2.2.2**
LER1# ldp show database Input label database, 2.2.2.1:0-2.2.2.2:0 Label Prefix 3 2.2.2.2/32 Output label database, 2.2.2.1:0-2.2.2.2:0 Label Prefix **2048 Customer ID 2001** 3 2.2.2.1/32	LER2# ldp show database Input label database, 2.2.2.2:0-2.2.2.1:0 Label Prefix **2048 Customer ID 2001** 3 2.2.2.1/32 Output label database, 2.2.2.2:0-2.2.2.1:0 Label Prefix 3 2.2.2.2/32

Adding the complimentary configuration on router 2.2.2.2 will allow it to properly forward traffic out a specific set of ports to the ultimate destination.

```
ldp map ports et.5.1 customer-id 2001
ldp add l2-fec customer-id 2001 to-peer 2.2.2.1
```

lo0 2.2.2.1	**lo0 2.2.2.2**

```
LER1# ldp show database

Input label database, 2.2.2.1:0-
2.2.2.2:0
  Label     Prefix
   2048     Customer ID 2001
      3     2.2.2.2/32

Output label database, 2.2.2.1:0-
2.2.2.2:0
  Label     Prefix
   2048     Customer ID 2001
      3     2.2.2.1/32
```

```
LER2# ldp show database

Input label database, 2.2.2.2:0-
2.2.2.1:0
  Label     Prefix
   2048     Customer ID 2001
      3     2.2.2.1/32

Output label database, 2.2.2.2:0-
2.2.2.1:0
  Label     Prefix
   2048     Customer ID 2001
      3     2.2.2.2/32
```

Forwarding Packets

Ingress Node: The ingress node is responsible for classifying inbound traffic based on the configured L2-FEC information. Once the traffic has been classified the ingress router strips the preamble and FCS and pushes the two labels found in the output tag table. The bottom of stack label represents the VC Label and the top level label represents the tunnel LSP.

Transit Routers: The transit routers are oblivious to any tunneling that is occurring beneath the top level label. They perform their label operation based on the tunnel label and never see the VC Label inside.

Egress Router: The egress router may receive a packet that has already had the tunnel label popped by the penultimate router or receive the complete stack if penultimate hop popping is disabled. Regardless, the egress router uses the VC Label to point to the Group Id and VC ID and determines which local ports are included as possible output ports. Once that is determined, regular L2 functions apply, learning and bridging etc.

Creating and Signaling Group ID & VC ID

The RS platform creates and signals the Group ID & VC ID when an L2-FEC is created locally and the then added to a remote peer. Basically the process can be broken down into three steps.

1) Create the L2-FEC
2) Signal the L2-FEC
3) Optionally set any transport level preferences

The three different approaches will be detailed below. Before starting the discussion on configuring the three L2-FEC types it is important to realize all the MPLS ports connecting the edge LER to the core MPLS network should be configured as 802.1Q trunks. This allows the MPLS ports to be part of multiple customer VLANs. There are two options when defining these ports as 802.1Q trunk-ports. The packets sent out into the MPLS core can either carry a 802.1Q tag as part of the layer two information that precedes the MPLS label if layer two switches are deployed between the LER and LSR, or the "untagged" option which does not insert an 802.1Q tag between the layer two information and the MPLS label. The untagged option is deployed where the next hop is an MPLS aware device, switching on MPLS label information.

To set the MPLS ports to 802.1Q trunk ports…

```
RS(config)# vlan make trunk-port <port>
```

To send the packet out to the MPLS cloud without the outer VLAN header information…

```
RS(config)# vlan make trunk-port <port> untagged
```

1. L2-FEC Physical Port: The ability to map a port or group of ports to unique identifier, isolating those ports from the rest the rest of the ports.

Using the example below, packets that arrive on port et.3.5, not destined to a local host, will be encapsulated, less the FCS and preamble, inside another layer two header followed by a stack of two MPLS labels. When using port based solutions it is important to note that port may receive 802.1Q tagged or untagged packets. By default the RS does not accept both 802.1Q tagged and untagged packets on the same port. In order to allow a port to act as a hybrid *Native VLAN* must be configured.

Case I - Traditional Access Port (Non-802.1Q tagged packets accepted, tagged packets dropped)

```
vlan make trunk-port gi.2.(1-2) untagged                (set MPLS
ports on LER to 802.1q capable but do not send 802.1Q tag)
ldp map ports et.3.(5-6) customer-id 2001               (map the
ports to an identifier)
ldp add l2-fec customer-id 2001 to-peer 2.2.2.2         (signal VC
Label {vcid=customer-id 2001} to the remote peer)
```

Case II - Traditional Trunk-Port (802.1Q tagged packets accepted, untagged packets dropped)

```
vlan make trunk-port gi.2.(1-2) untagged                (set MPLS
ports on LER to 802.1q capable but do not send 802.1Q tag)
vlan make trunk-port et.3.(5-6)                         (accept
802.1Q tag on the customer facing ports)
ldp map ports et.3.(5-6) customer-id 2001               (map the
ports to an identifier)
ldp add l2-fec customer-id 2001 to-peer 2.2.2.1         (signal VC
Label and {vcid customer-id 2001}to the remote peer)
```

Case III – Hybrid Ports (Both 802.1Q tagged and untagged accepted)

This case requires *Native VLAN* support. When the Native VLAN configuration is deployed it means the provider does not have to know whether the customer traffic arriving on the port is or is not 802.1Q tagged, nor does the provider have

to be aware of the customer VLAN IDs. This will be completely transparent to both the provider and the customer. Traffic that arrives will ultimately arrive at the destination with the exact same disposition it entered with. The Native VLAN acts as a catch all for those packets that arrive without an 802.1Q tags, inferring a VLAN for it.

To configure the native VLAN…

```
RS(config)# vlan set native-vlan <ports> <protocol|all> <VLAN name>
```

The following maps all packets that arrive on ports et.3.(1-8) without a tag to the DEFAULT VLAN (1). Notice, for the ports that are to be configured for hybrid, they must be configured as trunk ports first. WARING: VLAN names are case sensitive! When coding a VLAN name, DEFAULT is not the same as default.

```
vlan make trunk-port et.3.(1-8)
vlan set native-vlan et.3.(1-8) all DEFAUL
```

It is not required to use the DEFAULT VLAN as the native VLAN. Any VLAN can be specified as the native VLAN. However, a conflict will arise if an 802.1Q labeled packet arrives on a native VLAN port with the a VLAN tag in the packet that is the same as the configured Native VLAN.

```
vlan make trunk-port et.3.(1-8)
vlan create Native port-based id 4093
vlan add ports et.3.(1-8) to Native
vlan set native-vlan et.3.(1-8) all Native
```

Applying the hybrid approach to the L2 VPN service, the following allows customers to send 802.1Q tagged packets or un-tagged packets.

```
vlan make trunk-port gi.2.(1-2) untagged          (set MPLS ports on
LER to 802.1q capable sending packets untagged)
vlan make trunk-port et.3.(1-8)                   (accept 802.1Q
```

```
tag)
vlan set native-vlan et.3.(1-8) all DEFAULT     (set the ports to
apply native DEFAULT VLAN when 802.1Q tag not present)
ldp map ports et.3.(5-6) customer-id 2001       (create the L2-
FEC)
ldp add l2-fec customer-id 2001 to-peer 2.2.2.1 (signal VC Label
and {vcid customer-id 2001}to peer)
ldp map ports et.3.8 customer-id 4000           (create the L2-
FEC)
ldp ad l2-fec customer-d  4000 to-peer 2.2.2.1  (signal VC Label
and {vcid customer-id 4000} to peer)
```

2. L2-FEC VLAN ID: When this is used as the classification mechanism the VLAN ID value represents the VC ID. Typically, this approach is used when the provider assigns a VLAN-ID on a per customer basis. For this case, a VLAN must be created for the customer and the trunk port that connects to this customer and the MPLS enabled ports facing the core must be added to the customer defined VLAN. The connection can either be direct to the customer or a connection to a metro network that supports many subscribers of that provider.

In this example the trunk-port connects to a metro network using the et.3.3 interface. The VLAN ID 1001 and 1002 are extended to the remote peer 2.2.2.2.

```
vlan make trunk-port gi.2.(1-2) untagged              (set MPLS
ports on LER to 802.1q capable sending packets untagged)
vlan make trunk-port et.3.3                           (customer
facing - accept 802.1Q tag)
vlan create Cust1001 port-based id 1001               (create
customer VLAN)
vlan add port et.3.3 to Cust1001                      (add the
customer facing ports to customer VLAN)
vlan add ports gi.2.(1-2)                             (add the
MPLS ports to customer VLAN)
vlan create Cust1002 port-based id 1002               (create
customer VLAN)
vlan add port et.3.3 to Cust1002                      (add the
customer facing ports to customer VLAN)
```

```
vlan add ports gi.2.(1-2)                                    (add the
MPLS ports to customer VLAN)
lpd add l2-fec vlan 1001 to-peer 2.2.2.2                     (signal VC
Label and {vcid customer-id 1001}to peer)
ldp add l2-fec vlan 1002 to-peer 2.2.2.2                     (signal VC
Label and {vcid cusomter-id 1002}to peer)
```

3. L2-FEC Physical Port and VLAN: Finally, there is the a physical port and a VLAN ID to al L2-FEC. In this case the VC ID is the VLAN ID and the Group-ID is the customer-id. This approach allows a single customer to use multiple VLAN IDs without depleting the provider VLAN space. The Group-id (customer-id) engulfs all the VC ID (VLAN) information within it. The provider can offer site-specific VLAN significance within a customer-id. The port that connects to an individual customer or shared metro is configured as a trunk, with the port assigned an encapsulating group-ID. This example distributes the traffic to specific sites based n VLAN information within the customer-id.

```
vlan make trunk-port et.3.8                                   (customer
facing - accept 802.1Q tag)
vlan create 100 port-based id 100                            (create the
vlans that are to be received
vlan create 300 port-based id 300                            on customer
ports. Overlapping customer
vlan add ports et.3.8 to 100                                 ports in the VLAN are
protected from cross
vlan add ports et.3.8 to 300                                 communications
through backend filters)
ldp map ports et.3.8 customer-id 3000                        (create the
L2-FEC)
ldp add l2-fec vlan 100 customer-id 3000 to-peer 2.2.2.2(signal VC
Label and {GID=customer-id 3000:VCID=Vlan 100} to peer 2.2.2.2)
ldp add l2-fec vlan 200 customer-id 3000 to peer 2.2.2.3
(signal VC Label and {GID=customer-id 3000:VCID=Vlan 200}
                                                             to peer
                                                             2.2.2.3)
```

L2-Fec Transport LSP

By default, adding an L2-fec to a remote peer, will automatically select an LSP over which to signal the VC Label and Group-ID and VC-ID. The selection process depends on the number of label switched paths that exist between the remote LDP peers. If only a single LSP exists, the selection process is simple it uses that one. If more than one exists LSP selection is based on preference. The lowest numerical preference the most preferred LSP. Should the preference be equal, the process checks to determine if a corresponding VC ID has been received from the remote LDP peer. If the VC Label mapping to the same Group-ID and VC ID has been received from a remote peer the same LSP will be selected.

In most cases it is beneficial, when more than one LSP exists, to map the traffic to a preferred LSP. This allows the provider to explicitly engineer which are the preferred paths in the network. This does not mean if the LSP fails it represents a single point of failure; alternates can be accepted should a failure occur.

To set specific l2-fec specific transport parameters…

RS(config)# ldp set l2-fec ?

```
ler1(config)# ldp set l2-fec
?
 alternate-acceptable     - An alternate LSP (RSVP or LDP) is
acceptable in case
                            the transport LSP is not active. If
the transport
                            LSP comes up later, it will override
the alternate
                            LSP selected.
 customer-id              - Sets the transport LSP to be used for
this
                            customer-id FEC. The vlan option if
specified
                            additionally will select the transport
LSP for this
```

```
                                customer-id, vlan combination FEC.
 no-switchback                - Do not switch back from non-preferred
LSP to
                                preferred LSP if preferred LSP comes
up
 to-peer                      - Sets the transport LSP to this peer
 transport-lsp                - Name of the RSVP LSP which should be
used as a
                                transport LSP
 vlan                         - Sets the transport LSP to be used for
this vlan FEC.
                                The customer-id option if specified
additionally
                                will select the transport LSP for this
customer-id,
                                vlan combination FEC.
```

Specific to selecting the preferred LSP…

Option	Example	Action
Only one LSP specified in configuration	***ldp set l2-fec customer-id 2001 to-peer 2.2.2.2 transport-lsp TunLSP***	Use specified LSP. In event of LSP failure service interruption will occur.
LSP is specified but alternates are specified in the configuration	***ldp set l2-fec customer-id 2001 to-peer 2.2.2.2 transport-lsp TunLSP alternate-acceptable***	Use specified LSP. In the event of failure select any other LSP that terminates on the required remote peer.

Consider the following, no VC Labels have been exchanged between remote peers and the local router has decided the most preferred label switched path is a local decision. After the label exchange has been completed between two remote LDP peers with an established session the data starts to flow. By looking at the L2-FEC and the associated output tag table it is noticed all traffic for these L2-FEC are using a specific Transport LSP (TunLSP2) out a specific interface (To-LSR2).

```
LER1# ldp show l2-fec
FEC: Forward Equivalence class, in-lbl: Label received, out-lbl:
Label sent
Remote neighbor 2.2.2.2:0
FEC                                 in-lbl  out-lbl Transport LSP
name/label
VLAN ID 1002                        2048    2049    TunLSP2/17
VLAN ID 1001                        2049    2048    TunLSP2/17
Customer ID 2001                    -       2050

LER1# mpls show ott-table
Interface       OTT RefCount  HW-OTT RefCount   NextHop
Vlan Labels
-------------------------------------------------------------------
--------
lo              1    1          0       0       192.168.1.5
3    [17]
                2    1          0       0       0.0.0.0
1002 [2048]
                3    1          1       1       192.168.1.5
1002 [17|2048]
                4    1          0       0       0.0.0.0
1001 [2049]
                5    1          2       1       192.168.1.5
1001 [17|2049]
                6    1          0       0       0.0.0.0
0    [3]
                7    1          0       0       192.168.1.1
2    [17]

To-LSR1

To-LSR2         3    1          1       1       192.168.1.5
1002 [17|2048]
                5    1          2       1       192.168.1.5
1001 [17|2049]
```

Simply specifying which Transport LSP that should be used and whether or not alternates are acceptable will change the Transport LSP.

```
ldp set l2-fec vlan 1002 to-peer 2.2.2.2 transport-lsp TunLSP
alternate-acceptable
ldp set l2-fec vlan 1001 to-peer 2.2.2.2 transport-lsp TunLSP
alternate-acceptable
```

Tracing the path of the L2-fec now reveals the change to Transport LSP TunLSP out interface To-LSR1. Should this LSP fail an alternate interconnecting the two remote LDP peers will be selected to forward packets.

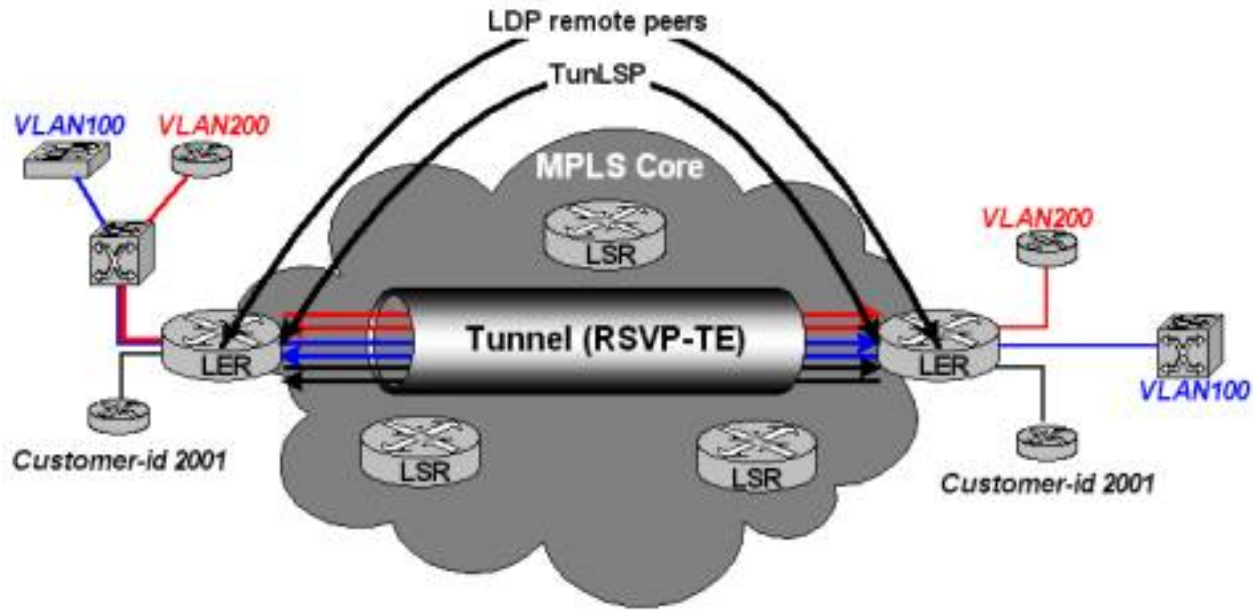
```
LER1# ldp show l2-fec
FEC: Forward Equivalence class, in-lbl: Label received, out-lbl:
Label sent
Remote neighbor 2.2.2.2:0
FEC                                 in-lbl  out-lbl Transport LSP
name/label
VLAN ID 1002                        2048    2049    TunLSP/17
VLAN ID 1001                        2049    2048    TunLSP/17
Customer ID 2001                    -       2050

LER1# mpls show ott-table
Interface       OTT RefCount  HW-OTT RefCount   NextHop
Vlan Labels
-------------------------------------------------------------------
--------
lo              1    1          0       0       192.168.1.5
3    [17]
                2    1          0       0       0.0.0.0
1002 [2048]
                3    1          1       1       192.168.1.1
1002 [17|2048]
                4    1          0       0       0.0.0.0
1001 [2049]
                5    1          2       1       192.168.1.1
1001 [17|2049]
                6    1          0       0       0.0.0.0
0    [3]
                7    1          0       0       192.168.1.1
2    [17]
```

```
To-LSR1          3    1          1       1       192.168.1.1
1002 [17|2048]
                 5    1          2       1       192.168.1.1
1001 [17|2049]
```

To-LSR2

Steps to Delivering VLL Services

The following network will demonstrate how to create different L2-FEC, mapping them to specific label switch paths, and ultimately exchanging the L2-FEC information between peers. The sample network will use RSVP to establish the Pop-Pop Tunnel LSP, MPLS for forwarding, and the required LDP extensions to exchange L2-FEC information between peers.



VLL Services delivery can be summarized into five high-level steps.

1) Create the MPLS core network
    a. IGP required to distributed reachability information with traffic engineering optional but recommended
    b. MPLS and Tunnel LSP signaling protocol (RSVP-TE or LDP) is required on all core facing interfaces

2) Edge routers instantiate the Tunnel LSP (RSVP-TE for explicit route support or LDP for hop-by-hop paths)

3) Enable LDP and establish remote LDP peering sessions, only required on label edge routers acting as the gateway between the MPLS cloud and the customers

a. Add the loopback interface of each LER to LDP
b. Start the LDP protocol
c. Establish the remote peering sessions using targeted hello messages

4) Define L2-FEC and distribute the VC Label mapping to Group ID and VC ID (required on both sides)

a. Create or set the L2-Fec
b. Add L2-Fec to remote peer

5) Transfer data

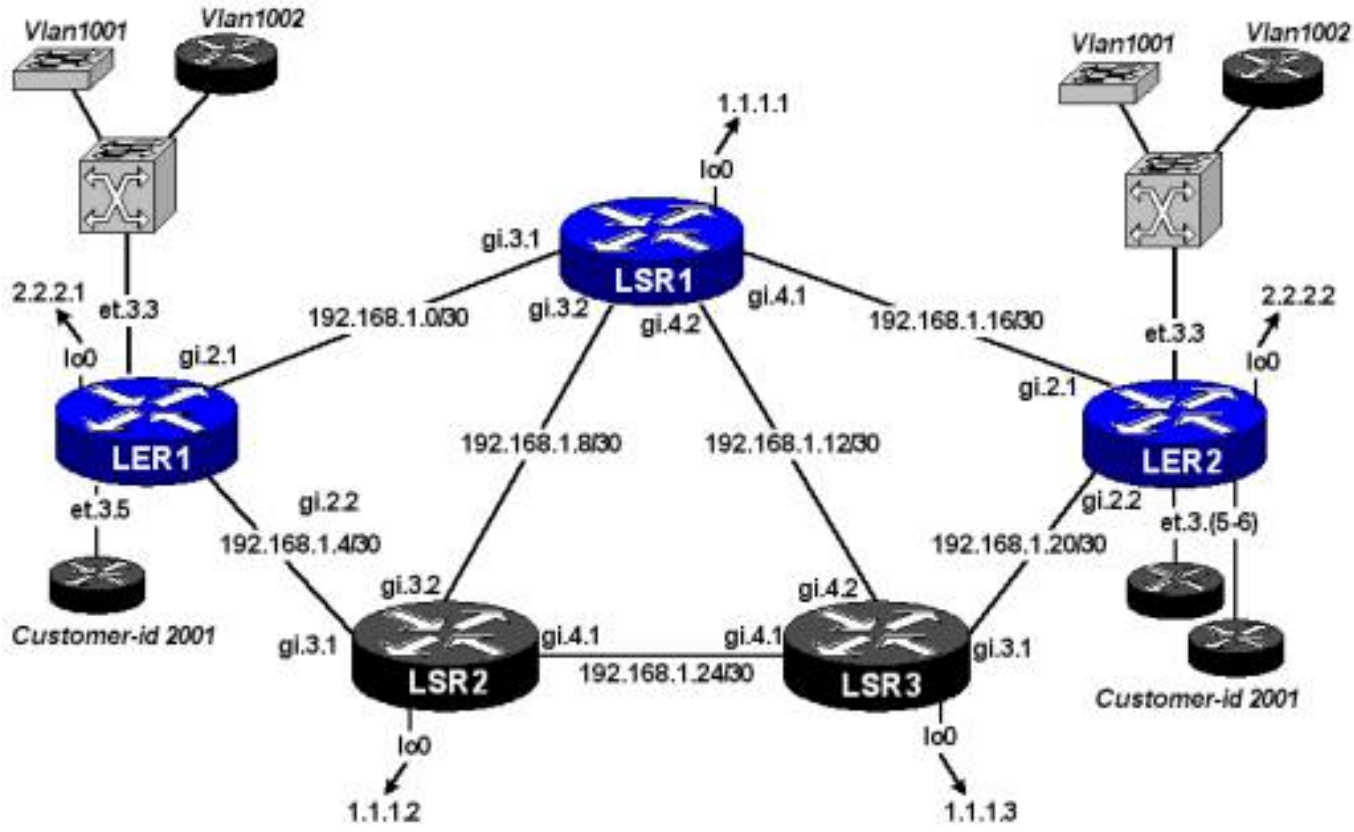
At the end of these steps, VLL services are available. Adding new subscribers means configuration of end nodes, allowing the dynamic signaling process to perform the necessary information exchange.

VLL Examples

Expanding on the above sample network, the following walks through the configuration of the VLL related services. RVSP-TE has been chosen as the signaling protocol for the Tunnels LSP and OSPF-TE has been chosen the IGP. Here, three subscribers have requested service. Two of the subscribers who are already connected to the traditional L2 Metro network would like to extend their layer two service to new facilities across the provider's backbone. A third subscriber would like to establish a layer two connection between two of their sites, which will have direct connections to the MPLS edge routers. When the

service is deployed, the subscribers will see a traditional layer two service between their sites. The MPLS cloud will be transparent to them.

For the two existing customers, an L2-FEC will be based on their provider issued VLAN-ID, which is part of the existing layer two Metro. The customer issued VLAN-ID will become the customer specific VC ID in the MPLS network. The new customer will be configured with a port based L2-FEC and the Customer–Id will become the VC ID.



1) Router configurations for both label edge routers and one label switch router have been provided to demonstrate the core configuration requirements discussed in the "Steps to Delivering VLL Services" section. After this step, there are no more core changes required to deliver the VLL service. After the core configuration is completed, the transit routers need not be touched. The entire configuration to establish new L2 VPN connections is done on and signaled from the edge routers.

LER1 Configuration

```
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
system set name LER1
ospf set traffic-engineering on
```

LSR1 Configuration

```
interface create ip To-LER1 address-netmask 192.168.1.1/30 port
gi.3.1
interface create ip To-LER2 address-netmask 192.168.1.17/30 port
gi.4.1
interface create ip To-LSR2 address-netmask 192.168.1.9/30 port
gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.13/30 port
gi.4.2
interface add ip lo0 address-netmask 1.1.1.1/32
ip-router global set router-id 1.1.1.1
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.1 to-area backbone cost 10
ospf start
mpls add interface all
mpls start
```

```
rsvp add interface all
rsvp start
system set name LSR1
ospf set traffic-engineering on
```

LER2 Configuration

```
interface create ip To-LSR1 address-netmask 192.168.1.18/30 port
gi.2.1
interface create ip To-LSR3 address-netmask 192.168.1.22/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.2/32
ip-router global set router-id 2.2.2.2
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 2.2.2.2 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR3
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR3
rsvp start
system set name LER2
ospf set traffic-engineering on
```

2) Once the core MPLS network is in place the Tunnel LSP should be created to interconnect the remote provider points of presences. RSVP or LDP can be used to signal the network path. Choose the signaling protocol based on your requirements. Of course, if one chose not to use a signaling protocol, static hop-by-hop mappings could be nailed up throughout the network, if required.

LSP Example one: Two label switched paths defined across distinct routes over the MPLS network both with different preferences. If no "transport-lsp" is specified for an L2-FEC the lowest numerical preference is selected.

LER1

```
mpls add interface To-LSR1
```

```
mpls add interface To-LSR2
mpls create path T-Prime num-hops 3
mpls create path T-Second num-hops 4
mpls set path T-Prime hop 1 ip-addr 192.168.1.2 type strict
mpls set path T-Prime hop 2 ip-addr 192.168.1.1 type strict
mpls set path T-Prime hop 3 ip-addr 192.168.1.18 type strict
mpls set path T-Second hop 1 ip-addr 192.168.1.6
mpls set path T-Second hop 2 ip-addr 192.168.1.5
mpls set path T-Second hop 3 ip-addr 192.168.1.26
mpls set path T-Second hop 4 ip-addr 192.168.1.22
mpls create label-switched-path TunLSP from 2.2.2.1 to 2.2.2.2
mpls create label-switched-path TunLSP2 from 2.2.2.1 to 2.2.2.2
mpls set label-switched-path TunLSP primary T-Prime preference 100
mpls set label-switched-path TunLSP2 primary T-Second preference
200
```

LER2

```
mpls add interface To-LSR1
mpls add interface To-LSR3
mpls create path T-Prime num-hops 3
mpls create path T-Second num-hops 4
mpls set path T-Prime hop 1 ip-addr 192.168.1.18 type strict
mpls set path T-Prime hop 2 ip-addr 192.168.1.17 type strict
mpls set path T-Prime hop 3 ip-addr 192.168.1.1 type strict
mpls set path T-Second hop 1 ip-addr 192.168.1.22
mpls set path T-Second hop 2 ip-addr 192.168.1.21
mpls set path T-Second hop 3 ip-addr 192.168.1.25
mpls set path T-Second hop 4 ip-addr 192.168.1.5
mpls create label-switched-path TunLSP from 2.2.2.2 to 2.2.2.1
mpls create label-switched-path TunLSP2 from 2.2.2.2 to 2.2.2.1
mpls set label-switched-path TunLSP primary T-Prime preference 100
mpls set label-switched-path TunLSP2 primary T-Second preference
200
```

LSP Example two: A slightly different configuration may see a single LSP with designated primary and secondary paths used for path protection. The strategy below allows the primary path to be backed up with a pre-established backup path, or hot standby.

LER1

```
mpls create path T-Prime num-hops 3
mpls create path T-Second num-hops 4
mpls set path T-Prime hop 1 ip-addr 192.168.1.2 type strict
mpls set path T-Prime hop 2 ip-addr 192.168.1.1 type strict
mpls set path T-Prime hop 3 ip-addr 192.168.1.18 type strict
mpls set path T-Second hop 1 ip-addr 192.168.1.6
mpls set path T-Second hop 2 ip-addr 192.168.1.5
mpls set path T-Second hop 3 ip-addr 192.168.1.26
mpls set path T-Second hop 4 ip-addr 192.168.1.22
mpls create label-switched-path TunLSP from 2.2.2.1 to 2.2.2.2
mpls set label-switched-path TunLSP primary T-Prime preference
mpls set label-switched-path TunLSP secondary T-Second preference
standby
```

LER2

```
mpls create path T-Prime num-hops 3
mpls create path T-Second num-hops 4
mpls set path T-Prime hop 1 ip-addr 192.168.1.18 type strict
mpls set path T-Prime hop 2 ip-addr 192.168.1.17 type strict
mpls set path T-Prime hop 3 ip-addr 192.168.1.1 type strict
mpls set path T-Second hop 1 ip-addr 192.168.1.22
mpls set path T-Second hop 2 ip-addr 192.168.1.21
mpls set path T-Second hop 3 ip-addr 192.168.1.25
mpls set path T-Second hop 4 ip-addr 192.168.1.5
mpls create label-switched-path TunLSP from 2.2.2.2 to 2.2.2.1
mpls set label-switched-path TunLSP primary T-Prime preference
mpls set label-switched-path TunLSP secondary T-Second preference
standby
```

3) Once the Tunnel LSP is signaled and established the LDP relationship between remote peers must be created using directed hello packets. Remember, all communication between the LDP remote peers is performed using the LDP Identifier, the router-id. This means the loopback interface must be added to the LDP protocol. Then the remote peer must be defined and the LDP protocol started.

LER1

```
ldp add interface lo0
ldp add remote-peer 2.2.2.2
ldp start
```

LER2

```
ldp add interface lo0
ldp add remote-peer 2.2.2.1
ldp start
```

4) By this stage the network is “Service Ready”. The core network is MPLS and RSVP enabled and the edge routers have established their remote LDP peering relationships. Now the task of mapping the customer traffic is at hand. The options for mapping the customer traffic are by VLAN-ID, Physical Port or combination of Physical Port and VLAN-ID. The first step is to ensure the MPLS enabled core facing ports are defined as trunk ports to allow the MPLS ports to be added to multiple customers.

Remember, as is the entire MPLS concept, these mappings are unidirectional. This means customer information needs to be mapped on both remote LDP peers if bi-directional traffic is a requirement.

LER1

```
vlan make trunk-port gi.2.(1-2) untagged
vlan make trunk-port et.3.3
vlan make trunk-port et.3.8
vlan make trunk-port et.3.5

vlan create Cust1001 port-based id 1001
vlan add ports et.3.3 to Cust1001
vlan add ports gi.2.(1-2) to Cust1001
ldp add l2-fec vlan 1001 to-peer 2.2.2.2

vlan create Cust1002 port-based id 1002
vlan add ports et.3.3 to Cust1002
vlan add ports gi.2.(1-2) to Cust1002
ldp add l2-fec vlan 1002 to-peer 2.2.2.2

vlan set native-vlan et.3.5 all DEFAULT
```

```
ldp map ports et.3.5 customer-id 2001
ldp add l2-fec customer-id 2001 to-peer 2.2.2.2
```

LER2

```
vlan make trunk-port gi.2.(1-2) untagged
vlan make trunk-port et.3.3
vlan make trunk-port et.3.8
vlan make trunk-port et.3.(5-6)

vlan create Cust1001 port-based id 1001
vlan add ports gi.2.(1-2) to Cust1001
vlan add ports et.3.3 to Cust1001
ldp add l2-fec vlan 1001 to-peer 2.2.2.1

vlan create Cust1002 port-based id 1002
vlan add ports et.3.3 to Cust1002
vlan add ports gi.2.(1-2) to Cust1002
ldp add l2-fec vlan 1002 to-peer 2.2.2.1

vlan set native-vlan et.3.(5-6) all DEFAULT
ldp map ports et.3.(5-6) customer-id 2001
ldp add l2-fec customer-id 2001 to-peer 2.2.2.1
```

Related Show Commands

Some useful show commands are presented in this section to help work around the new network. Complete network configurations are available here.

Starting from the ingress LER and working through all the way to the egress LER, here are some interesting things found along the way.

The LDP database stores all the labels that have been distributed between LDP peers, input (received from) and output (sent to). Notice that the remote LDP peers that are traveling within the RSVP tunnel advertise themselves as directly connected, penultimate hop label of 3 for their loopback prefix. This is because; as far as the remote LDP peers are concerned the LDP sessions are directly

connected. There are no other LDP enabled nodes between them, as is the case. When the label information is pushed on the outgoing packet, the LDP label within the Tunnel LSP is actually the identifier on how to handle the L2-FEC on the egress, issued from the LDP global space.

RS# ldp show database

```
LER1# ldp show database

Input label database, 2.2.2.1:0-2.2.2.2:0
  Label     Prefix
   2048     Customer ID 2001
   2049     VLAN ID 1002
   2050     VLAN ID 1001
      3     2.2.2.2/32

Output label database, 2.2.2.1:0-2.2.2.2:0
  Label     Prefix
   2048     VLAN ID 1001
   2049     VLAN ID 1002
   2050     Customer ID 2001
      3     2.2.2.1/32
```

The virtual circuit identifier is associated with an in and out VC Label and the Transport LSP and its label. From this information the starting point of the L2 VPN can be associated to the label stack and the tunnel LSP.

RS# ldp show l2-fec

```
LER1# ldp show l2-fec
FEC: Forward Equivalence class, in-lbl: Label received, out-lbl:
Label sent
Remote neighbor 2.2.2.2:0
FEC                                 in-lbl  out-lbl Transport LSP
name/label
Customer ID 2001                    2048    2050    TunLSP/17
VLAN ID 1002                        2049    2049    TunLSP/17
VLAN ID 1001                        2050    2048    TunLSP/17
```

The “output tag table” provides the information necessary to determine which L2-FEC is using which outbound interface on the router, as well as providing the necessary hardware ott index value “HW-OTT” which is required to determine various outbound action information. The important information revealed in this display is the interfaces information for the label stack.

RS# mpls show ott-table

```
LER1# mpls show ott-table
Interface       OTT RefCount  HW-OTT RefCount   NextHop
Vlan Labels
-------------------------------------------------------------------
--------
lo              1    1          0       0       192.168.1.1
2    [17]
                2    1          0       0       0.0.0.0
2    [18]
                3    1          0       0       0.0.0.0
0    [2048]
                4    1          1       1       192.168.1.1
2    [17|2048]
                5    1          0       0       0.0.0.0
1002 [2049]
                6    1          2       1       192.168.1.1
1002 [17|2049]
                7    1          0       0       0.0.0.0
1001 [2050]
                8    1          3       1       192.168.1.1
1001 [17|2050]
                9    1          0       0       0.0.0.0
0    [3]

To-LSR1         4    1          1       1       192.168.1.1
2    [17|2048]
                6    1          2       1       192.168.1.1
1002 [17|2049]
                8    1          3       1       192.168.1.1
1001 [17|2050]
```

To-LSR2

In order to get find the proper index in the hardware output tag table, the actual MPLS port being used by the LSP is also required. That information can be obtained by displaying the information for the interface in use for the L2-FEC.

RS# interface show ip To-LSR1

```
LER1# interface show ip To-LSR1
Interface To-LSR1:
    Admin State:           up
    Operational State:     up
    Capabilities:          <BROADCAST,SIMPLEX,MULTICAST>
    Configuration:
       VLAN:               SYS_L3_To-LSR1
       Ports:              gi.2.1
       MTU:                1500
       MAC Encapsulation:  ETHERNET_II
       MAC Address:        00:00:1D:A3:4E:97
       IP Address:         192.168.1.2/30  (broadcast: 192.168.1.3)
```

The action information when an L2-FEC match occurs can be found in the hardware output tag table using the index represented by the “hw-ott” indicator and the port in use by the label switched path. The following displays all the action information for L2-FEC “customer ID 2001”. Remember, the output tag table is an MPLS table and represents the label that must be pushed on the packet toward the destination. This corresponds to the labels stored in the LDP database as the input labels. A close look at the table detailed exactly what output action will occur including the number and labels that will be pushed, the byte count information for the L2-FEC and the next_hop_mac address of the LSR.

The label is represented using this form {LABEL:EXP:S:TTL}. In the example below, two labels are pushed, “push_n”. The top level label represents the Tunnel LSP {17:0:0:255}. The LDP VC Label tunned within the tunnel LSP indicates the bottom of the stack {2048:0:1:255}.

RS# mpls show hw-ott-tbl index 1 port gi.2.1

```
LER1# mpls show hw-ott-tbl index 1 port gi.2.1
Port: gi.2.1
-------------
Entry 1, Total: 7680
 dot1p source         : 0           output_vlan id : 2
 dot1p enabled        : 0     output vlan priority : 0
 dot1q enabled        : 0                      mtu : 1568
 overwrite_ttl        : 1            next_hop_mac : 00e0:630f:265a
 rate limit violated  : 0               php_etype : 34887
 rate limit enable    : 0                  label0 : 17:0:0:255
 no exp support       : 0                  label1 : 2048:0:1:255
 start of l2 tunnel   : 1                  label2 : 0:0:0:0
 trunk lsp            : 1                   ip_da : 0
 ip tunnel            : 0              byte count : 91154
 lsr hop hide         : 0            packet count : 893
 llc snap encaps      : 0   malformed_label_drops : 0
 send to cpu          : 0                   pop_n : 0
 state                : 1                  push_n : 2
```

Checking the transit LSR: Using the information gathered in previous steps the transit LSR can be check for pertinent LSP information, including label information.

RS# mpls show label-switched-paths transit verbose

```
LSR1# mpls show label-switched-paths transit verbose
Transit LSP:
    Label-Switched-Path: "TunLSP_T-Prime"
        state: Up                lsp-id: 16386
        to: 2.2.2.1              from: 2.2.2.2
        Path-Signalling-Parameters:
            setup-pri: 7                 holding-pri: 0
            label in: 17                 label out: 3
            path recvfrom: 192.168.1.18 path sendto: 192.168.1.2
            explicit-path: 192.168.1.2
            record-route:
```

```
                192.168.1.2

    Label-Switched-Path: "TunLSP_T-Prime"
        state: Up                lsp-id: 16385
        to: 2.2.2.2              from: 2.2.2.1
        Path-Signalling-Parameters:
            setup-pri: 7                holding-pri: 0
            label in: 17                label out: 3
            path recvfrom: 192.168.1.2  path sendto: 192.168.1.18
            explicit-path: 192.168.1.18
            record-r       192.168.1.18
```

At the egress: The L2-fec information on the egress label edge router reveals mirror image of the label information from the ingress. It is important to note the transport LSP and transport label are outbound form the local router. It does not represent the inbound label switched path. That information is gathered from the ingress label edge router.

```
LER2# ldp show l2-fec
FEC: Forward Equivalence class, in-lbl: Label received, out-lbl:
Label sent
Remote neighbor 2.2.2.1:0
FEC                                  in-lbl  out-lbl Transport LSP
name/label
VLAN ID 1001                         2048    2050    TunLSP/17
VLAN ID 1002                         2049    2049    TunLSP/17
Customer ID 2001                     2050    2048    TunLSP/17
```

Looking at the dynamically created static filters, which occurs when the L2-FEC is created and the VC Label is mapped, shows the ports that are part of the customer specific virtual private network. A filter is created in each direction to support bi-directional traffic. This is a subset of the display, as it pertains to “customer-id 2001”.

The, first block represents the direction toward the customer facing ports received from the MPLS network. Any mpls packet arriving with a label of 2048 is associated with customer-id 2001 and can egress on ports et.3.(5-6). The second

block represents the reverse direction. Any packet that arrives on ports et.3.(5-6) can use the MPLS port the active label switched path is currently on or the local customer facing ports, for local switching. These tables also give you the information you need to look into the input label map, with the index (label value: 2048) and the port (active MPLS LSP port: gi.2.1). If the active MPLS LSP switch to a different port, the Out-List ports in the second block would automatically update to reflect that.

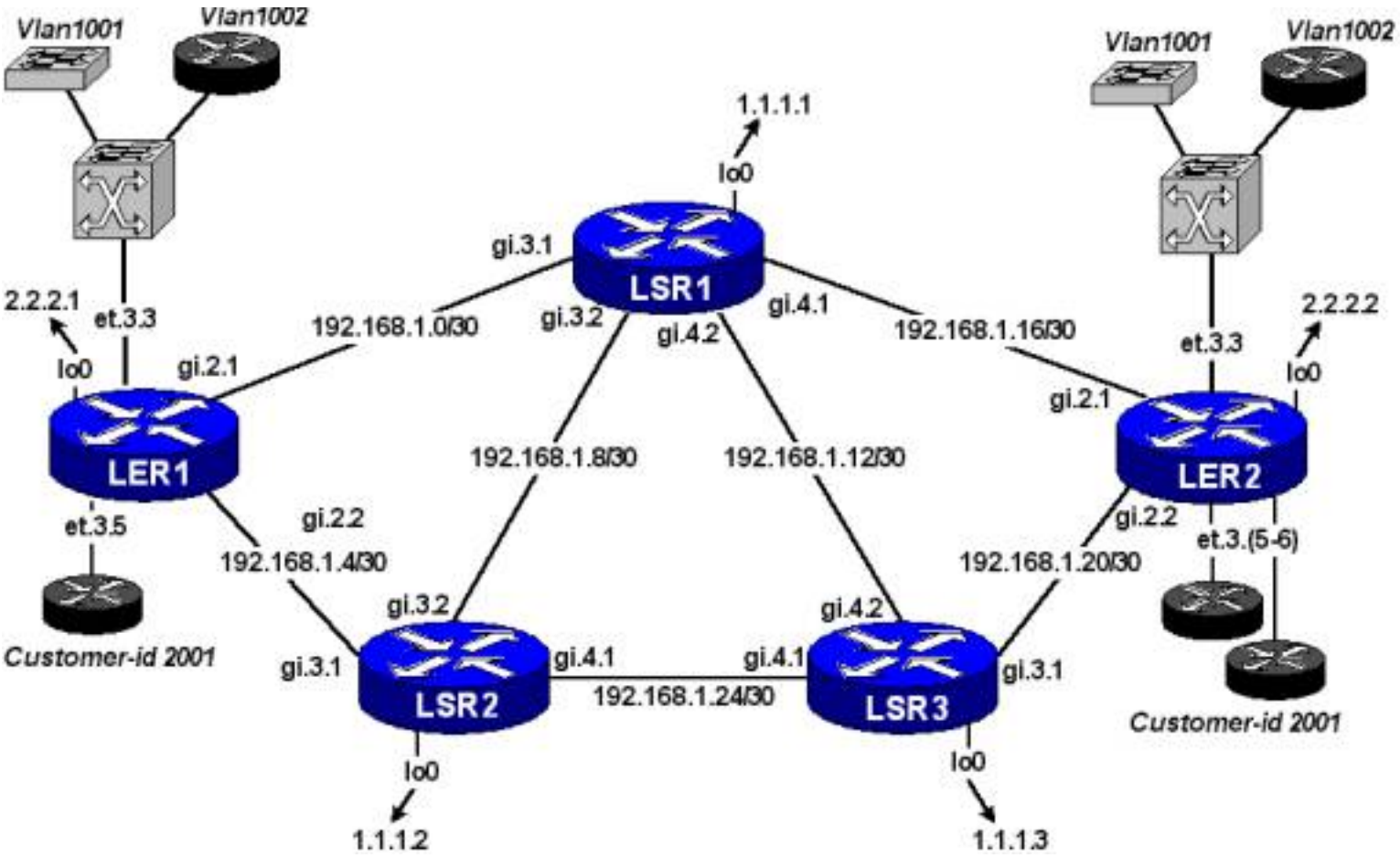
RS# filters show static-entry

```
LER2# filters show static-entry
Name:            Port-To-Port  MPLS: 2
----
Direction:       destination
Restriction:     allow-to-go
VLAN:            any VLAN
Label:           2048
Customer:        2001
Source MAC:      any
Source MAC Mask:000000:000000
Dest MAC:        any
Dest MAC Mask:   000000:000000
In-List ports:   gi.2.(1-2)
Out-List ports: et.3.(5-6)

Name:            Port-To-Port  MPLS: 7
----
Direction:       destination
Restriction:     allow-to-go
VLAN:            any VLAN
Customer:        2001
Source MAC:      any
Source MAC Mask:000000:000000
Dest MAC:        any
Dest MAC Mask:   000000:000000
In-List ports:   et.3.(5-6)
Out-List ports: gi.2.1,et.3.(5-6)
```

To display the input lable map…

RS# mpls show hw-ilm-tbl port <active MPLS port> index <Inbound LDP label>

```
LER2# mpls show hw-ilm-tbl port gi.2.1 index 2048
Port: gi.2.1
-------------
Entry 2048, UNICAST, Total: 15360
 check vlan          : 0              egress tos : 0
 end of mpls tunnel : 0           egress vlan id : 4095
 end of l2 tunnel   : 1     egress vlan priority : 0
 send to cpu         : 0             egress etype : 2048
 exp rewrite         : 0                      exp : 0
 vlan overwrite      : 0          output channel : 16
 prio overwrite      : 0             output port : 15
 ip tunnel           : 0               ott index : 0
 lsr hop hide        : 0              exp21qprio : 0
 overwrite tos mode : 0               byte count : 347925
 overwritetos        : 0            packet count : 3524
 state               : 1             packet drop : 0
```



LER1

```
vlan make trunk-port gi.2.(1-2) untagged
vlan make trunk-port et.3.3
vlan make trunk-port et.3.5
vlan create Cust1001 port-based id 1001
vlan create Cust1002 port-based id 1002
vlan add ports et.3.3 to Cust1001
vlan add ports et.3.3 to Cust1002
vlan add ports gi.2.(1-2) to Cust1001
vlan add ports gi.2.(1-2) to Cust1002
interface create ip To-LSR1 address-netmask 192.168.1.2/30 port
gi.2.1
interface create ip To-LSR2 address-netmask 192.168.1.6/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.1/32
vlan set native-vlan et.3.5 all DEFAULT
ip-router global set router-id 2.2.2.1
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
```

```
ospf add stub-host 2.2.2.1 to-area backbone cost 10
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR2
mpls create path T-Prime num-hops 3
mpls create path T-Second num-hops 4
mpls set path T-Prime hop 1 ip-addr 192.168.1.2 type strict
mpls set path T-Prime hop 2 ip-addr 192.168.1.1 type strict
mpls set path T-Prime hop 3 ip-addr 192.168.1.18 type strict
mpls set path T-Second hop 1 ip-addr 192.168.1.6
mpls set path T-Second hop 2 ip-addr 192.168.1.5
mpls set path T-Second hop 3 ip-addr 192.168.1.26
mpls set path T-Second hop 4 ip-addr 192.168.1.22
mpls create label-switched-path TunLSP from 2.2.2.1 to 2.2.2.2
mpls set label-switched-path TunLSP primary T-Prime
mpls set label-switched-path TunLSP secondary T-Second standby
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR2
rsvp start
ldp add interface lo0
ldp map ports et.3.5 customer-id 2001
ldp add remote-peer 2.2.2.2
ldp add l2-fec customer-id 2001 to-peer 2.2.2.2
ldp add l2-fec vlan 1002 to-peer 2.2.2.2
ldp add l2-fec vlan 1001 to-peer 2.2.2.2
ldp start
system set name LER1
ospf set traffic-engineering on
```

LSR1

```
interface create ip To-LER1 address-netmask 192.168.1.1/30 port
gi.3.1
interface create ip To-LER2 address-netmask 192.168.1.17/30 port
gi.4.1
interface create ip To-LSR2 address-netmask 192.168.1.9/30 port
gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.13/30 port
gi.4.2
interface add ip lo0 address-netmask 1.1.1.1/32
```

```
ip-router global set router-id 1.1.1.1
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.1 to-area backbone cost 10
ospf start
mpls add interface all
mpls start
rsvp add interface all
rsvp start
system set name LSR1
ospf set traffic-engineering on
```

LSR2

```
interface create ip To-LER1 address-netmask 192.168.1.5/30 port
gi.3.1
interface create ip To-LSR1 address-netmask 192.168.1.10/30 port
gi.3.2
interface create ip To-LSR3 address-netmask 192.168.1.25/30 port
gi.4.1
interface add ip lo0 address-netmask 1.1.1.2/32
ip-router global set router-id 1.1.1.2
ospf create area backbone
ospf add interface To-LER1 to-area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 1.1.1.2 to-area backbone cost 10
ospf start
mpls add interface all
mpls start
rsvp add interface all
rsvp start
system set name LSR2
ospf set traffic-engineering on
```

LSR3

```
interface create ip To-LER2 address-netmask 192.168.1.21/30 port
gi.3.1
```

```
interface create ip To-LSR1 address-netmask 192.168.1.14/30 port
gi.4.2
interface create ip To-LSR2 address-netmask 192.168.1.26/30 port
gi.4.1
interface add ip lo0 address-netmask 1.1.1.3/32
ip-router global set router-id 1.1.1.3
ospf create area backbone
ospf add interface To-LER2 to-area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR2 to-area backbone
ospf add stub-host 1.1.1.3 to-area backbone cost 10
ospf start
mpls add interface all
mpls start
rsvp add interface all
rsvp start
system set name LSR3
ospf set traffic-engineering on
```

LER2

```
vlan make trunk-port gi.2.(1-2) untagged
vlan make trunk-port et.3.3
vlan make trunk-port et.3.(5-6)
vlan create Cust1001 port-based id 1001
vlan create Cust1002 port-based id 1002
vlan add ports et.3.3 to Cust1002
vlan add ports gi.2.(1-2) to Cust1001
vlan add ports gi.2.(1-2) to Cust1002
vlan add ports et.3.3 to Cust1001
interface create ip To-LSR1 address-netmask 192.168.1.18/30 port
gi.2.1
interface create ip To-LSR3 address-netmask 192.168.1.22/30 port
gi.2.2
interface add ip lo0 address-netmask 2.2.2.2/32
vlan set native-vlan et.3.(5-6) all DEFAULT
ip-router global set router-id 2.2.2.2
ospf create area backbone
ospf add interface To-LSR1 to-area backbone
ospf add interface To-LSR3 to-area backbone
ospf add stub-host 2.2.2.2 to-area backbone cost 10
```

```
ospf start
mpls add interface To-LSR1
mpls add interface To-LSR3
mpls create path T-Prime num-hops 3
mpls create path T-Second num-hops 4
mpls set path T-Prime hop 1 ip-addr 192.168.1.18 type strict
mpls set path T-Prime hop 2 ip-addr 192.168.1.17 type strict
mpls set path T-Prime hop 3 ip-addr 192.168.1.2
mpls set path T-Second hop 1 ip-addr 192.168.1.22
mpls set path T-Second hop 2 ip-addr 192.168.1.21
mpls set path T-Second hop 3 ip-addr 192.168.1.25
mpls set path T-Second hop 4 ip-addr 192.168.1.5
mpls create label-switched-path TunLSP from 2.2.2.2 to 2.2.2.1
mpls set label-switched-path TunLSP secondary T-Second standby
mpls set label-switched-path TunLSP primary T-Prime
mpls start
rsvp add interface To-LSR1
rsvp add interface To-LSR3
rsvp start
ldp add interface lo0
ldp map ports et.3.(5-6) customer-id 2001
ldp add remote-peer 2.2.2.1
ldp add l2-fec vlan 1001 to-peer 2.2.2.1
ldp add l2-fec vlan 1002 to-peer 2.2.2.1
ldp add l2-fec customer-id 2001 to-peer 2.2.2.1
ldp start
system set name LER2
ospf set traffic-engineering on
```

Transparent LAN Services - TLS

Transparent LAN Services - TLS

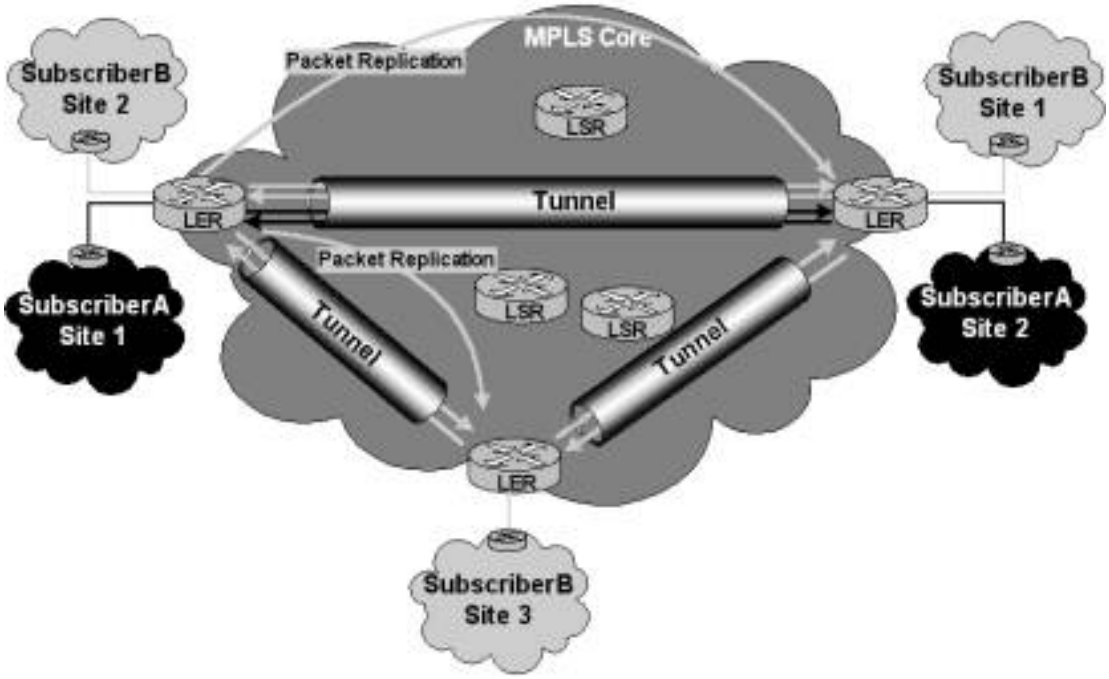
Transparent LAN Services

Transparent LAN Services

Transparent LAN Services, TLS, is an extension to the VLL model that allows for point-to-multipoint connectivity. From the perspective of the end nodes attached to TLS enabled MPLS network, the MPLS backbone takes on the appearance of a learning layer two switch capable of learning and aging. However, the TLS network must maintain a loop free environment. A loop free topology removes the scaling issue that one might find by deploying a spanning tree type topology in the core. This, of course would be very undesirable in a backbone network that required to deal with many hundreds if not thousands of discrete customers and intersecting label switched paths. A logical full meshed of point-to-point label switched paths are required to ensure full customer connectivity and receipt of packets on one label switched path are never forwarded to any other label switch path. This implementation detail is similar to the split horizon approach used by some routing protocols.

These extensions to the VLL model augment the point-to-point methodology with the ability to replicate packets and perform address learning and aging. The actual MPLS core does not change, the intelligence for deploying either the VLL or TLS model is at the edge. There are two Internet drafts that have been tabled to address the TLS services model of the learning bridge using the Ethernet circuits defined in the previously mention "*Martini*" drafts. The good news is the authors will merge their works into a single draft, internet-drafts/draft-lasserre-tls-mpls-00.txt and draft-vkompella-ppvpn-vpsn-mpls-00.txt.

Broadcast packets and unknown unicast packets are flooded to all label switched paths on which that customer particular customer id has been received. Currently, multicast packets are treated as broadcast based traffic. In the example below, packet replication for SubscriberA will only be replicated across the top path. Where as, replication of packets for SubscriberB will be replicated on both Tunnels, neither of the tunnels used by SubscriberB will in turn forward packets that it did not generate. Once layer two addressing information has been learned, unicast packets will only be forwarded to the LSP on which the MAC address is found. Should MAC addresses move or age out, the learning bridge methodology will remove the applicable MAC addresses from the LSP to which they once belonged.



Subscribers who contract this service from their provider must make ample provisions to ensure the addition of a layer two connections between sites does not have negative implications in their specific environment. Simply put, this is a layer two connection between two sites and subscriber networks must be able to resolve any loops this introduces for in their environment.