The LSP Protection/Restoration Mechanism in GMPLS

by

Ziying Chen

The LSP Protection/Restoration Mechanism in GMPLS

by

Ziying Chen

A graduation project submitted to

the Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of

Master of Computer Science

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario, Canada, K1N 6N5

October 1, 2002

Abstract

This report introduces the new switching technology Generalized Multiprotocol Label Switching (GMPLS) and traffic engineering. It outlines the components of the GMPLS path protection/restoration mechanism, and it specifies how different protocols contribute to path protection/restoration in GMPLS. This report specifies different path protection/restoration mechanisms. It illustrates how they work and how the signaling protocol supports them. Also, some case studies are provided to illustrate how the recovery mechanism is constructed in practice. At the end, the report compares these path protection/restoration mechanisms and introduces the current trend of protection/restoration in the industry.

Acknowledgements

I would like to thank my supervisor, Professor Gregor Bochmann, for his support and care during my study under his supervision. His guidance is very appreciated!

I would like to thank my uncle, Chi Kan Leung. His long-term support makes my study dream in Canada come true.

I would like to thank my family for their encouragement and care.

I would like to thank my grandmother and all the relatives in our big family for their moral support and help.

I would also like to thank my friends that study with me throughout the years in the school.

And, I would also like to thank all the people at the University of Ottawa I have had the pleasure to meet.

Table of Contents

1. Introduction

Multiprotocol Label Switching (MPLS) [1] is a recent switching technology that has been proposed for IP networks with two main objectives: (a) providing a more efficient mechanism for packet forwarding than traditional routing, and (b) providing tools for quality of service and traffic engineering. It is based on a switching principle very similar to ATM cell switching (VPI/VCI correspond to labels) and Time-Division Multiplexing (time slots correspond to labels).

With the increased traffic within the Internet, there is a tendency to have backbone connections with very high bandwidth capability, including optical fibers possibly with Dense Wavelength Division Multiplexing (DWDM). The principle of DWDM is again very similar to time-division multiplexing (wavelengths correspond to labels).

It can be foreseen that the future data networks will include various switching techniques at various levels of the capacity hierarchy, from optical transmission up to the packet level. Since the switching techniques expected to be used at these different levels, that is, MPLS, optical space switching, DWDM, and time division multiplexing, all require that a logical connection between the source and the destination must be established before the data can be sent, it has been proposed that it would be good if the same signaling protocol could be used for controlling the establishment of such logical connections at all these different levels. While the signaling protocols at these different levels may not be completely identical because they may require certain level-dependent parameters, nevertheless, the logical structure and most of the message content could be identical for the signaling at these different levels. General MPLS (GMPLS) [2] is intended as such a signaling protocol that could be used at these different switching levels.

With the development of networks, new technologies provide high bandwidth capacity, which makes a significant data loss if a failure cannot be recovered timely. It is imperative for GMPLS networks to provide protection/restoration of traffic.

This report gives an introduction to the general area and provides an overview of the GMPLS protocol and related standards. The main emphasis of this report is on the path protection/restoration mechanisms that can be used with GMPLS. It specifies how different protocols contribute to path protection/restoration in GMPLS, including signaling and routing protocols. This report specifies different path protection/restoration mechanisms. It illustrates how they work and how the signaling protocol supports them. It also addresses some problems remaining to be solved, and provides some answers. Some case studies are provided to illustrate how the recovery mechanism can be used in practice. At the end, this report compares these path protection/restoration mechanisms and introduces the current status of path protection/restoration mechanisms in the industry.

2. Overview of GMPLS

MPLS evolved from several similar technologies that were invented in the middle of the 1990s, for example, *IP switching* by Ipsilon [3] [4] [5], *Tag Switching* by Cisco [6], *Aggregated Route-based IP Switching* by IBM [7], and *Cell Switching Router* by Toshiba [8]. They all use label swapping to forward data, and they all use IP addressing and IP-based routing protocols like OSPF. At the end of the 1990s, the Internet Engineering Task Force (IETF) standardized the technology and named it MPLS [1].

A label is a short, fixed-length entity and it does not encode any information from the network layer header. A node that supports MPLS is called Label Switching Router (LSR). A label is inserted in front of each data packet on the entry in the network. At each LSR, the packet is forwarded based on the value of the label, and forwarded to an outgoing interface with a new label. In some situations, the incoming interface is also a factor to determine the outgoing interface. This operation is called Label Swapping. When the data packet arrives at the destination node, the label is stripped off and the packet is handed to the upper layer to process. The path that data is forwarded by label swapping across a network is called Label Switched Path (LSP). In the illustration in Figure 1.0, the LSP is (Node1, Node2, Node3). The head node of the LSP is called ingress node, e.g., Node 1, and the ending node of the LSP is called egress node, e.g., Node 3.

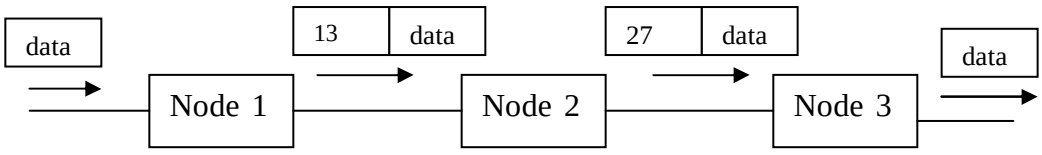


Figure 1.0: data is forwarded along the LSP

The function of forwarding can be partitioned into two components: control component and forwarding component. The forwarding component is responsible for the forwarding of data from the input port to the output port in a router according to the forwarding table. The control component is responsible for the construction and maintenance of the forwarding table. These two components are also named forwarding plane and control plane.

MPLS [1] provides routers with the label switching technology to forward data. The router can make a forwarding decision based on two sources of information: the label forwarding table and the label carried in the data. Based on the incoming label (and maybe also the incoming interface), the forwarding table provides enough information to forward the data, e.g., outgoing label, outgoing interface, and so on (see Figure 1.1 for a forwarding entry).

Incoming information	Outgoing information
Incoming label	Outgoing label
…	Outgoing interface
	…

Figure 1.1 the logical view of an entry in the forwarding table.

MPLS supports data forwarding based on a label. The original MPLS architecture [1] assumes that a Label Switching Router (LSR) has a forwarding plane which can (a) recognize packet (or cell) boundaries, and (b) process packet (or cell) headers. However, there are routers that cannot recognize packet boundaries or process packet headers, e.g., TDM switches, optical cross-connects (OXCs), etc. But different label modeling techniques can allow these routers (switches) to forward data using the same principle of label switching. For example, the time slot of TDM, the lambda (or wavelength) of a WDM switch, the port of an OXC, etc, can be modeled as a label. That means the forwarding plane is different, but the control plane can be same. Such a technology is called Generalized MPLS (GMPLS) [2]. GMPLS extends MPLS. With GMPLS, a switch whose forwarding plane recognizes neither packet nor cell boundaries can also forward data using this extended label switching technology. GMPLS supports multiple types of switching: packet (cell), TDM, lambda, and space (port) switching. This means that GMPLS can forward data based on time slots, wavelengths, physical ports and labels.

GMPLS models wavelength, TDM channels or time slots as labels [9], and the name *generalized label* refers to all these different "labels" [10].

2.1 LSP Hierarchy

So far, GMPLS supports five types of interfaces (see [2]).

(1) Packet Switch Capable (PSC) interfaces

They are interfaces that can recognize packet boundaries and can forward data based on the content of the packet header. An example is an Ethernet interface of an IP router, which can recognize the header boundary of an IP packet.

(2) Layer-2 Switch Capable (L2SC) interfaces

They are interfaces that recognize frame/cell boundaries and can forward data based on the content of the frame/cell header. An example is an interface of an ATM switch that forwards cells based on the label encoded by ATM VCI/VPI.

(3) Time-Division Multiplex Capable (TDM) interfaces
They are interfaces that forward data based on the data's time slot in a repeating cycle. An example is an interface of a SONET switch.

(4) Lambda Switch Capable (LSC) interfaces
They are interfaces that forward data based on the wavelength on which the data is received. An example includes the interface of an Optical Cross-Connect (OXC), which can distinguish lambdas.

(5) Fiber-Switch Capable (FSC) interfaces
They are interfaces that forward data based on a position of the data in the real world physical spaces. An example is an interface of a Photonic Cross-Connect (PXC), which can operate on a per-fiber basis.

We can see that interfaces (3), (4) and (5) are unable to check the content of the user data, while (1) and (2) can process the packet (cell) headers.

A circuit can be established only between, or through, interfaces of the same type. Depending on the particular technology being used for each interface, different circuit names can be used, e.g. SONET/SDH circuit, light-path, etc. In the context of GMPLS, all these are referred to a common name: Label Switched Path (LSP).

In MPLS, LSPs can be nested, e.g., several LSPs of the same level can be multiplexed into a single LSP of another level. The nested LSP concept in MPLS has been extended to GMPLS [11]. A new LSP is multiplexed inside an existing higher-order LSP so that the preexisting LSP serves as a link along the path of the new LSP [12]. This is referred to as LSP hierarchy. The ordering of LSPs is based on the link multiplexing capabilities of the nodes. A hierarchical LSP can be established using the same type of interface, or between different types of interface.

A hierarchical LSP can be established if an interface is capable of multiplexing several LSPs from the same technology (layer). For example, 4 OC-48 links can be multiplexed into an OC-192 link. A lower order SDH/SONET LSP (OC-48) can be nested in a higher order SDH/SONET LSP (OC-192) (see Figure 1.2).

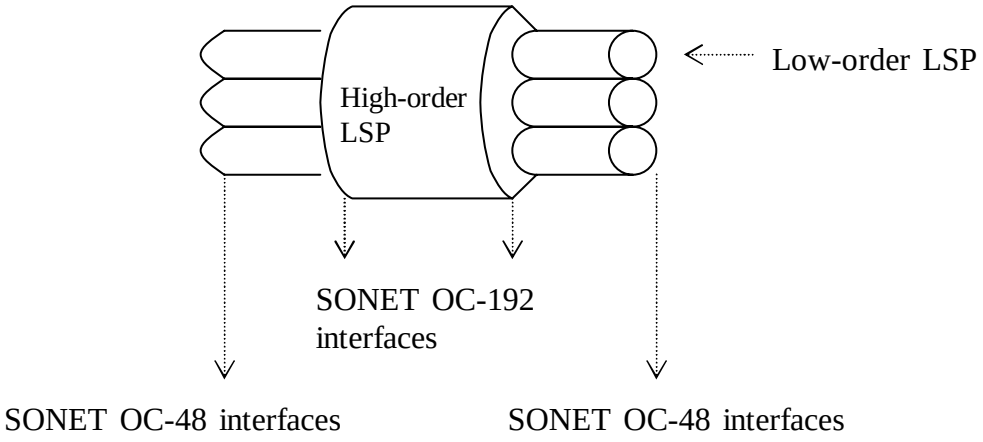


Figure 1.2: a hierarchical LSP is established on the same type of interfaces.

A hierarchical LSP can also be established between different types of interface. Let us discuss the following example. An LSP which starts and ends on Packet Switch Capable (PSC) interfaces can be nested (together with other LSPs) into an LSP which starts and ends on SONET (TDM) interfaces – assuming that the SONET interfaces have bigger capacity. That LSP which starts and ends on SONET interfaces can again be nested into an LSP which starts and ends on Lambda Switch Capable (LSC) interfaces.

Figure 1.3 shows an example where nested LSPs occur between different types of interfaces.

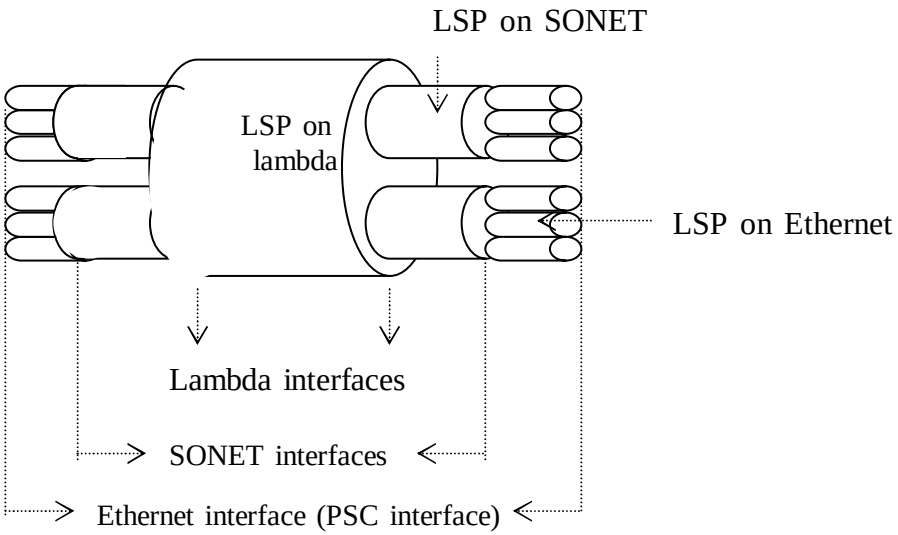


Figure 1.3: LSP hierarchy between different interfaces

At the top of this LSP hierarchy is the LSP with FSC interfaces, followed by LSC, then by TDM, L2SC and PSC interfaces (the reversed order of the above 5 interfaces). So, an

LSP which starts and ends on PSC interfaces can be nested into an LSP which starts and ends on L2SC interfaces. This LSP, further, can be nested into an LSP that starts and ends on TDM interfaces, which further can be nested into an LSP that starts and ends on LSC interfaces. Again, the LSP starts and ends on LSC interfaces can further be nested into an LSP that starts and ends on FSC interfaces. The example in Figure 1.3 shows a three-level hierarchical LSP. For each level of a given hierarchy, there is a separate control instance. The LSP is independently computed based on that level of routing information, and independently signaled. Examples follow in the subsequent sections.

2.2 The Mesh Network

The trend of the Internet transport infrastructure is to have an optical network core interconnecting high-speed routers (and switches) (see [13]).

A lightpath is a point-to-point optical layer connection between two access points in an optical network (see [14] for the definition). An example is shown in Figure 1.4. A wavelength connects two edge OXCs through two ports of the OXCs. Note that the two edge OXCs may be bridged by a number of OXCs and the wavelength may be switched by these transit OXCs. The lightpath is referred to as an LSP in the context of GMPLS if the lightpath is set up by GMPLS signaling.

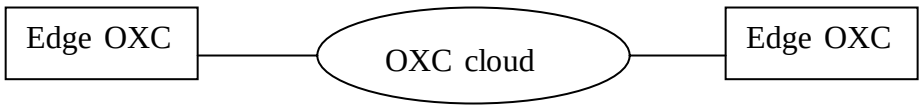


Figure 1.4: a lightpath

This report only considers the LSP recovery mechanism in a mesh network. An example of a mesh network is shown in Figure 1.5. In the example, LSRs which are packet-switch capable (called PSC LSRs) are connected to SONET switches. And the SONET switches are connected to an optical core network. One PSC LSR is connected to its peer over dynamically established LSPs across the optical core. The optical core is assumed to be incapable of processing packet headers. It is also assumed that a path must be established across the optical core network before the PSC LSRs can communicate.

The optical core network consists of OXCs that are connected by point-to-point optical links. The OXC can operate at the level of individual wavelength. The OXCs are mesh-connected (to form a general topology). Each node has the GMPLS-implemented control plane. What does it mean? (a) The nodes can forward data using label switching. For example, OXCs can forward data by label switching - based on the input wavelength, which is modeled as a label, to make a forwarding decision. (b) Each node uses GMPLS signaling (e.g., RSVP-TE with extensions) and GMPLS routing protocols (e.g. OSPF-TE with extensions).

It is recommended that the optical network control plane should utilize IP-based protocols (e.g., signaling and routing) for dynamic provisioning and restoration of light-paths within

and across optical networks. This is because signaling and routing mechanisms developed for IP traffic engineering applications can be reused in optical networks [15].

The OXC provides lambda-switch capable interfaces, and the multiplexing capacity of the interface is usually much bigger than that of the packet-switch capable interface. Furthermore, wavelength (or lambda) cannot multiplex packets directly. Therefore, SONET switches, e.g., OC-192/OC-48 switches, provide the optical core network access to the PSC LSRs. In this example and rest of this report, it is assumed that the edge OXC has interfaces that provide WDM capabilities for lambda-switch capable interfaces, also it has interfaces that provide SONET section level signals (e.g., OC-192 including all overheads). The SONET switch can multiplex a number of same-level LSPs that deliver packets into a single SONET path. The SONET switch also has a GMPLS-implemented control plane – it uses label switching to forward data and GMPLS signaling and routing protocols.

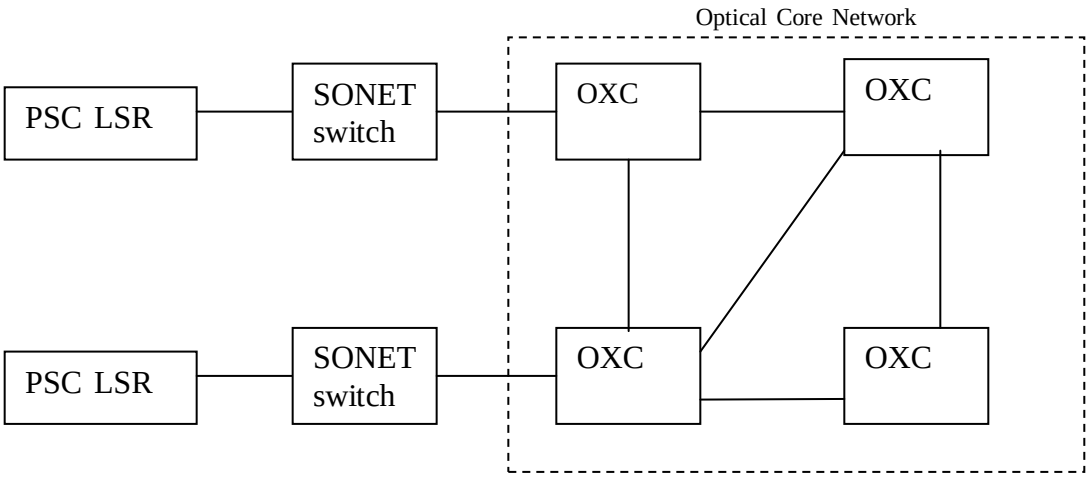


Figure 1.5: a GMPLS mesh network example

2.3 Traffic Engineering

The task of mapping traffic flows onto an existing physical network topology to optimize the network resource utilization and facilitate the network operations is called Traffic Engineering (see [16] for detailed definition). Traffic Engineering (TE) provides the ability to move traffic flow away from congestions and onto a potentially less congested physical path across a network.

TE properties are information used to support traffic engineering. For example, TE properties for a link include: available bandwidth, maximum bandwidth, etc.

Traditional routing protocols (e.g., OSPF) do not consider Traffic Engineering and they have been extended to advertise TE properties in a network by IETF, e.g., *TE LSAs to extend OSPF for Traffic Engineering* [17], *OSPF Extensions to Support Multi-Area Traffic Engineering* [18]. For example, assuming that two routers are connected by a link, with the TE information advertised by the extended OSPF, both routers understand

the available bandwidth of the link, the maximum bandwidth of the link, etc. Each router stores the TE properties in a database, which are learnt from the advertisement provided by the routing protocol. With the TE properties in the database, a node understands the TE properties of the network. And the database of the routers will be synchronized within the entire routing area. The information in the database can be used for a path computation algorithm to compute a path across the network to meet the Traffic Engineering requirements.

The Traffic Engineering (TE) link concept is introduced with the current development of traffic engineering and optical networks. A TE link is a logical link that has TE properties [19]. The Internet draft [19] explains the meaning of "logical": it is a way to group/map the information about certain physical resources (and their properties) into the information that is used by CSPF for the purpose of path computation, and by GMPLS signaling. Both ends of the link must do the mapping/grouping consistently. By "consistent", it means the information advertised by one end of the link does not conflict with that advertised by the other end of the link. Examples of a TE link are: a physical link, an LSP, or a bundle of physical links. The TE properties of a TE link are exchanged like traditional link information by routing protocols, e.g., carried by OSPF advertisement messages.

As we said, an LSP can be regarded as a TE link. Because of the benefits introduced by optical networks, e.g., high bandwidth, the capacity of an LSP constructed by lambdas likely cannot be utilized completely by one user. The routing protocol can advertise this LSP as a TE link into the routing domain, which can be used for the path computation algorithm to calculate paths, path aggregation (e.g., shared by other LSPs that require a portion of the LSP capacity), etc. We say that there is a "forwarding adjacency" (FA) between the end-nodes of the advertised LSP [20]. And such an LSP is named FA-LSP [20]. As a TE link, the TE properties are also associated with the FA-LSP.

In a hierarchical LSP, the high-order LSPs tunnel low-order LSPs. The high-order LSP should be advertised by the routing protocol as a TE link (and they become a FA-LSP), so that the unreserved bandwidth is utilized.

We will see examples of the TE link and FA-LSP in the subsequent sections.

2.4 The GMPLS Control Plane

There are five major functions in the control plane of GMPLS: resource discovery, routing, path computation, link management and signaling. We briefly introduce these functions here and we will specify the portions of these functions that are related to this report in the subsequent sections.

Resource discovery is the procedure through which nodes within a network find out the resource in the network. It provides the information for signaling and path computation. Path computation uses an algorithm to calculate an explicit-routed LSP (ER-LSP).

The routing function uses the IP-based routing protocols to distribute and maintain the information about the topology and resources of the network. The routing protocol is the means by which non-local resources are discovered. The topology and resources of the network will be taken into account as parameters for the path algorithm to calculate an ER-LSP.

Signaling is the procedure through which service provisioning is done. The service provisioning includes LSP establishment, LSP deletion and LSP modification.

Link management is used to manage TE links, e.g., maintain control channel connectivity, localize link failure, and so on.

Control information, e.g., signaling messages, routing messages, link management messages, is exchanged through the control channel. The control channel should be separated from the data channel as IETF recommended [10]. One of the good reasons for separation is that the control channel should not share the fate with the data channel. And it does not have to be the same physical medium as the data channel. For example, an OXC uses lambda to transport data, but uses an Ethernet link to transport control signals.

2.4.1 Resource Discovery

Local resource discovery is the procedure that a router takes to find out what resource it has for service provisioning.

When a node starts up, it goes through the neighbor and link discovery procedure, for example, by manual configuration or an automatic procedure. By combining the results, each node has a database about the local resource, for example, link capacity, wavelength, etc.

After the local resource discovery, each node uses the routing protocol to distribute its local resource. When a node receives other nodes' resources, it stores them in a database. Then, any changes to the resource will also be advertised by the routing protocol. Thus each node knows about the resource of the entire network.

2.4.2 Enhancements in the Routing Protocol to Support GMPLS

Conventional routing protocols are reused and enhanced with extensions to support GMPLS, e.g., OSPF with extensions [21], IS-IS with extensions [22]. They are used to discover network topology, distribute Traffic Engineering properties and GMPLS-specific features.

Here we introduce the extensions of conventional routing protocols to support unnumbered links, different interfaces, link protection type and Shared Risk Link Group distribution in GMPLS.

Extensions to support unnumbered links

One of the fundamental issues in routing is addressing. Because of WDM, an optical fiber may have a number of channels. The IETF draft [14] suggests an addressing scheme: an IP address is used to identify a node (e.g., a router ID), and a "selector" is used to identify further fine-grain information within each node.

A numbered link means its interfaces are IP addressed. An unnumbered link means its interfaces are not IP addressed. In the optical network, optical fibers connect OXCs as point-to-point links. Point-to-point links need not to be numbered. In this case, the router (or an OXC) that connects an unnumbered link can assign a 32-bit identifier to the link. The identifier uniquely identifies the link within that router. So the identifier is locally significant. This local identifier is called the remote identifier from the point of view of the other OXC that is connected by the same unnumbered link. For example, OXC A and B are connected by unnumbered link L. OXC A assigns identifier L1 to L, which is a local identifier to A; OXC B assigns L2 to L, which is a local identifier to B. When the routing protocol exchanges the information between two routers, L1 is a remote identifier to B, and L2 is a remote identifier to A. The link can be uniquely identified globally by <router ID, (local) unnumbered link identifier> (see the example in Figure 1.6). Note that the router ID is always a 32-bit IP address.

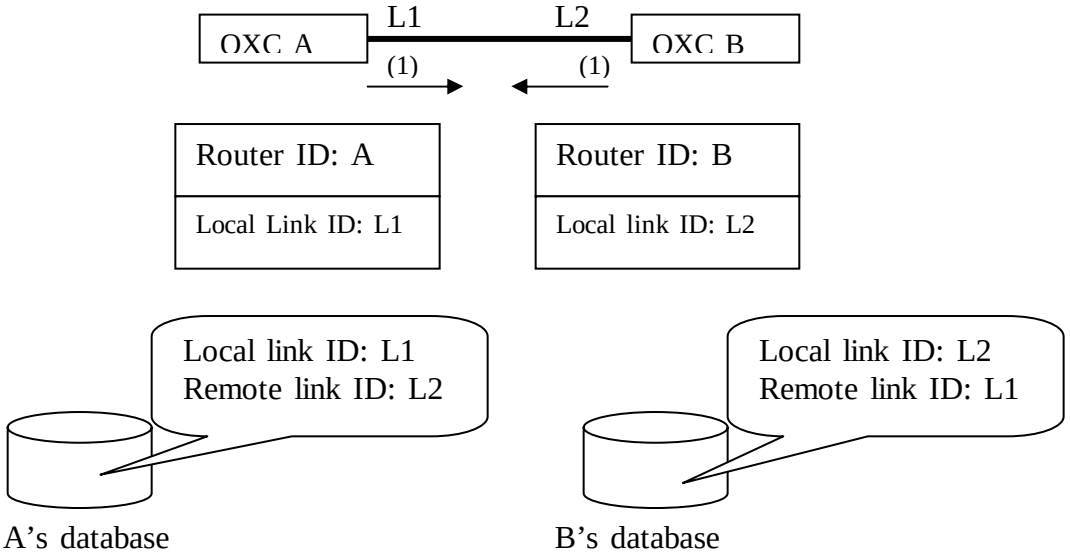


Figure 1.6: naming unnumbered link

It is assumed that an edge router that has physical connectivity to an OXC is able to provide optical-electrical data conversion. An edge router between the optical network and the IP network has interfaces that connect to OXCs and interfaces that connect to regular IP routers (see Figure 1.7). Assuming that the link F between the OXC and edge router is an optical fiber, and the link between the IP router and the edge router is a regular link (e.g., an Ethernet link). At the start-up, the edge router knows that the optical fiber F connects itself through interface I_1 to an OXC by neighbor discovery (e.g., by manual configuration, and see [23] for more about how a router discovers its

neighbors). And it knows that an Ethernet link connects itself through interface I_2 to an IP router by neighbor discovery. When the edge router creates its routing adjacency relationship with its neighbors, it understands what parameters, options and protocol extensions it is going to use. Thus the routing protocol will send out advertisement messages carrying unnumbered link identifiers to identify link F, and it will send out advertisement messages carrying IP addresses to identify link L.

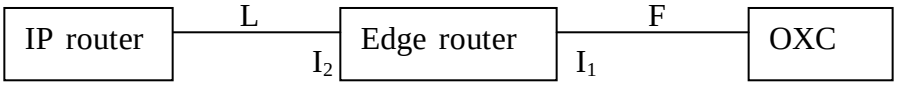


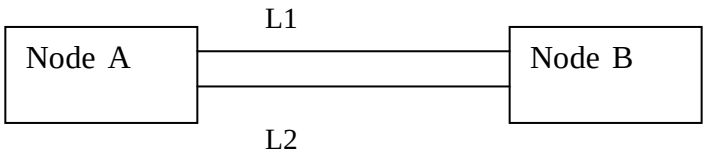
Figure 1.7: edge router knows about the links

Extensions to support link protection type

If a link has a protection capability provided by the link layer, then such a link capability should be considered by the path computation component when calculating/selecting the path. The link protection type (e.g., 1+1 protection) is one of the traffic engineering properties of a link and it is distributed by the routing protocol. The link protection type does not have the same meaning when it is carried by signaling protocols as when it is carried by routing protocols, because it is from a different point of view. When the routing protocol distributes the link protection type for a given link, it means the link has the protection capability indicated by the link protection type. Let see what these link protection types are.

Extra Traffic

A link with type Extra Traffic means it is protecting another link or other links.



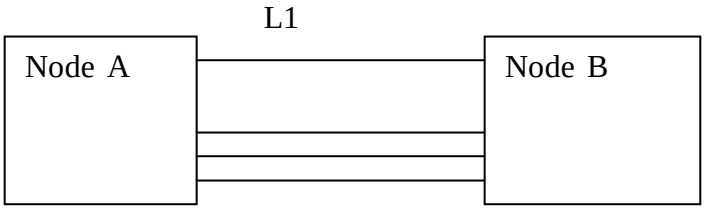
For instance, Link 1 and Link 2 connect Node A and Node B. Traffic is going through L2. If Link 1 is of type “Extra Traffic”, it is protecting L2, but there is no traffic going through L1 yet, or the traffic going through L1 is different from that going through L2.

In Internet draft *Routing Extensions in Support of Generalized MPLS* [19], the sentence “The LSPs on a link of this type will be lost if any of the links it is protecting fail” means a link of this type will be activated when a link it is protecting fails. So any LSP that is on such a link will be preempted.

Unprotected

No link is protecting the link that is of type unprotected. If it fails, then the LSP is lost and so is the traffic.

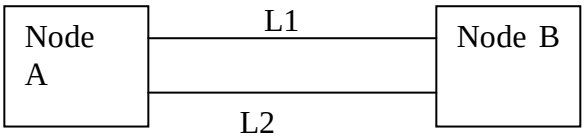
Shared
If the link is of type Shared, it means that there are one or more disjoint links of type Extra Traffic that are protecting this link.



For instance, Link 1 is protecting one or more links, which is of type Extra Traffic. Other links that are protected by L1 are of type Shared – they share the protection relationship.

Dedicated 1:1
If the link is of type Dedicated 1:1, it means that there is one dedicated disjoint link of type Extra Traffic that is protecting this link. For instance, in example of Type *Extra Traffic* (see above), Link 2 is typed *Dedicated 1:1*.

Dedicated 1+1
If the link is of type *Dedicated 1+1*, it means that a dedicated disjoint link is protecting this link. However, the protecting link is not advertised in the link state database. So if the switchover occurs for a failure, the LSP is still there.



For instance, traffic is sent between two links: L1 and L2. The receiver takes the healthy one to accept user traffic. Link 2 and Link 1 both are of type *Dedicated 1+1*.

Enhanced
A link of type Enhanced means it has a protection capability that is more reliable than *Dedicated 1+1*.

If the link information distributed by the routing protocol does not have the link protection type, it means it is unknown.

Extensions to support Shared Risk Link Group
With the development of optical network, e.g., WDM, a number of links can have the same fate. Because they share the same physical resource, and if the resource is not available, then all these links are broken. For example, an optical fiber can contain a

number of links. Such a set of links constitutes a Shared Risk Link Group (SRLG) [19]. Based on different physical resource, a link may belong to multiple SRLGs.

For path protection/restoration, the links of the backup path must belong to different SRLG(s) from the ones of the working path. Therefore, the SRLG information is useful for the path computation component to compute the path.

Extensions to support different interfaces

A link is connected to a node by an interface. GMPLS supports different types of interface, e.g., interface which is capable of packet switching, interface which is capable of lambda switching, etc. Different types of interface have different switching capabilities, and even same type of interface have different switching capabilities. The switching capability of the interface introduces a new constraint for path computation and signaling. In GMPLS LSP set up, a LSP must start and end at the same type of interface. So this information needs to be distributed onto the network.

The Interface Switching Capability Descriptor [24] describes the switching capability of an interface. The IETF draft *Routing Extensions in Support of Generalized MPLS* [19] defines the following interface switching types:

Packet-Switch Capable-1 (PSC-1)
Packet-Switch Capable-2 (PSC-2)
Packet-Switch Capable-3 (PSC-3)
Packet-Switch Capable-4 (PSC-4)
Layer-2 Switch Capable (L2SC)
Time-Division-Multiplex Capable (TDM)
Lambda-Switch Capable (LSC)
Fiber-Switch Capable (FSC)

If an interface is of type PSC, it means that the node receiving data over this interface can switch the received data on a packet-by-packet basis. An example is the Ethernet interface. Types PSC-1 through PSC-4 stand for different levels of capability. It means potentially an LSP starts and ends on PSC interface can also be nested into another LSP that also starts and ends on PSC interface assuming that the LSP interfaces have different switching capabilities. However the PSC types 1-4 has not been detailed in the draft yet.

If an interface is of type L2SC, it means that the node receiving data over this interface can switch the received frames based on the layer 2 address. An example is the ATM interface – based on ATM VCI/VPI to switch data.

If an interface is of type TDM, it means that the node receiving data over this interface can switch the received data based on the time slot. An example is the SONET interface.

If an interface is of type LSC, it means that the node receiving data over this interface can recognize and switch individual lambdas within the interface. An example is the interface of an OXC (or PXC) that can operate on an individual lambda.

If an interface is of type FSC, it means that the node receiving data over this interface can switch the entire contents to another interface. An example is the interface of an OXC (or PXC) that can operate on an individual fiber.

Besides the switching type, the Interface Switching Capability Descriptor also contains the maximum bandwidth for each priority (range from 0 to 7) that may be reserved on this link.

A link can be used to transport different data encoded in a different way, e.g., SONET, Lambda, Packet, etc. The data encoding method specifies this information in the Interface Switching Capability Descriptor.

So the Interface Switching Capability Descriptor contains three necessary pieces of information: (1) interface switching type, (2) max (reservable) bandwidth and (3) data encoding type. Optional information may be attached in the descriptor for some specific interface types, for example, if the interface is PSC, the Maximum Transport Unit should be specified.An example of an Interface Switching Capability Descriptor is like:

Interface Switching Capability = PSC-1
Encoding = Ethernet 802.3
Max Bandwidth[0] = 1.0 Gbps, for priority 0

When a node advertises its link information carrying the descriptor, the descriptor only describes the interface that connects the node originating the message. In the example in Figure 1.8, interface I and interface K connect the router A to other nodes. The Interface Switching Capability Descriptor (ISCD) originated by A only describes interface I and K, not the interface of another end of the link.

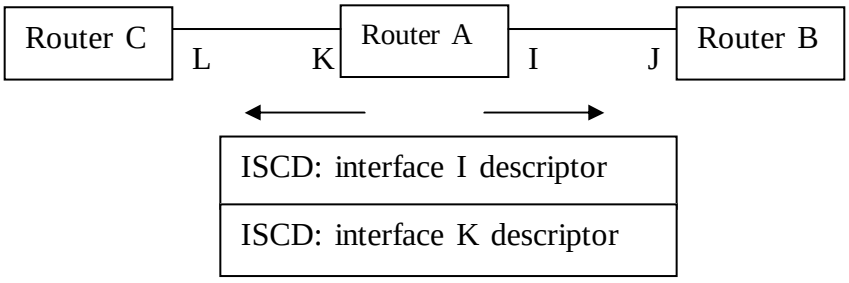


Figure 1.8: a router advertises the interface descriptor

Traffic Engineering properties
Besides the above information, there are other TE properties that are distributed by routing protocols, e.g., maximum bandwidth, available bandwidth, etc. Because these TE properties are not specific for GMPLS, they will be introduced in the subsequent sections.

We are going to see how these extensions are implemented in OSPF as an example in the subsequent sections.

2.4.3 Enhancements in MPLS Signaling to Support GMPLS

Signaling refers to exchange of information between involved components in the network required to provide and maintain service. GMPLS signaling provides LSP control (e.g., LSP set-up/release, LSP modification), and it may be used to reserve resources at the same time when LSP is being established. GMPLS signaling uses enhanced protocols CR-LDP [25] or RSVP-TE [26].

Generalized Label Request and Generalized Label
In the context of GMPLS, an LSP can be a mix of different types of link. For example, an LSP may have links that connects ATM switches, SONET switches, OXCs and others. And the label should take a different form. These forms of "label" are referred to as a *generalized label*.

In the GMPLS signaling, a node explicitly requests a label from its downstream peer when it needs one. The signaling message carries a label request, which should tell the downstream node enough information about the application environment of the desired label. The downstream node responds with a generalized label. It should contain enough information to allow nodes of the LSP to program their label forwarding tables.

Therefore, the signaling message should be extended to support the widening scope of GMPLS signaling. The label request message should include the following information:

(1) LSP encoding type;
(2) Switching type;
(3) Generalized Payload ID (G-PID).

An LSP can be used to transport different data encoded in a different way, e.g., SONET, Lambda, Packet, etc. The LSP encoding types are defined in [27].

An interface connects a link to a node. The interfaces supported by GMPLS may have different switching capabilities, for example, packet-switch capable, lambda-switch capable, TDM capable, etc. These are named switching types in GMPLS signaling. A list of the switching types is defined in [24].

The Generalized Payload ID is an identifier of the payload carried by an LSP. Examples include lambda (using fiber), Ethernet (using fiber or lambda), etc. G-PID is defined in [27].

A generalized label has a variable length, which can model different types of "label", e.g., wavelength, port, etc, in the context of GMPLS.

Bi-directional LSP setup

There are a number of reasons [28] for using one signaling session to build a bi-directional LSP, instead of building two unidirectional LSP to do the same job. The advantages are obvious, e.g., the signaling overhead is less. From the restoration point of view, the delay to establish a bi-directional LSP to restore the service for a failed bi-directional LSP is less than the restoration delay for a unidirectional LSP. So the GMPLS signaling should be able to support bi-directional LSP set-up.

Label Set

There are cases in GMPLS that result in label allocation trouble. For example, OXC A and OXC B are signaling neighbors for the set-up of a new LSP. OXC B (a downstream node) assigns label 10 to OXC A (an upstream node), which works as the outgoing label in A for forwarding data to B. But that label is not available in A (e.g., it does not have wavelength 10 at the interface to B). So the label set is defined in GMPLS signaling, which restricts the label range. For example, assuming that OXC A and OXC B both support GMPLS-RSVP-TE signaling, OXC A puts all the labels that are acceptable to A itself into the label set. The Path message carries the label set from A to B (from upstream to downstream). B can pick one of the labels in the set. However, if none of the labels in the label set is acceptable to B, B will generate an error and the path set-up will not continue.

Signaling Link Protection for LSP establishment

During LSP signaling in GMPLS, label distribution protocols (RSVP-TE, or CR-LDP) may carry the link protection type. If the link protection type is carried, it means the LSP to be established requires link layer protection. The link protection type indicates what link protection capability is desired for the links constructing the LSP to be set up. The link protection type is one of the TE requirements (or a constraint) for an LSP, so the signaling for the LSP will not continue if the desired link protection cannot be provided. There are six link protection types defined by [27]. They have been specified in the previous section of this report. For example, the signaling protocol carries link protection type *Dedicated 1+1*, and it means the LSP to be established requires the link that has *Dedicated 1+1* protection.

Indication of the LSP role

There are two LSP roles: primary or secondary (backup). The GMPLS signaling protocol carries a flag that indicates if the LSP being set up is primary or secondary. The resources allocated for a backup LSP are not used until the primary LSP fails. Because the resource allocation has priorities (carried by the signaling protocol), the resource allocated for a backup LSP may be used by an LSP that has lower priority until the primary LSP fails and the traffic is switched over to the backup. At that time, all the LSPs using the resource allocated for the backup LSP must be preempted.

2.4.4 Path Computation

Traditional IP routing algorithms aim to find a path that optimizes a certain scalar metric (e.g. minimizes the number of hops), and such a method causes a number of network problems, e.g., network congestions, violation of network administration, etc.

Constraint-based routing algorithms set out to find a path that optimizes a certain scalar metric and at the same time does not violate a set of constraints. Such a path is called constraint-based path. It is the ability to find a path that does not violate a set of constraints that distinguishes constraint-based routing from conventional IP routing.

The constraints include QoS requirements, administrative policies, etc. Because we are studying the LSP protection/restoration mechanism, the constraint of interest is that the backup path must not share a link/node with the primary path except the initiator node and the terminator node. In particular, the information of Shared Risk Link Group and Link Protection Type are of interest to us. Note that the LSP role is for resource allocation and usage.

We need to compute a path to implement constraint-based routing. The path computation component in GMPLS control plane is used to do such a job. Path computation is used to select an appropriate route between two clients through the optical network for explicit routing.

In each node of the network, there is a database TE-LSDB that stores the information of all the links in the network, e.g., TE properties. This is the prerequisite for path computation. After all, we must know about the network before we calculate anything. Also, it means that the constraints we considered in the path computation are within the scope defined by the information in the TE-LSDB.

For a hop-by-hop routed LSP, there is no need to have path computation. When the signaling is done, it carries the desired Link Protection Type. Every node receiving the signaling message must honor the desired link protection for the LSP being established; otherwise, the signaling will not go through (see the subsequent section for more). Note that a hop-by-hop routed LSP cannot be the backup LSP, because there is no guarantee that the links/nodes traveled by such an LSP are not part of the primary LSP. The transit node is not supposed to keep track of the information about primary/backup LSP pairs, because there could be thousands of LSPs that go through a node.

Path computation is used to provide end-to-end LSP protection using the explicit-routed LSP (ER-LSP). If the primary LSP is an ER-LSP, then the backup LSP can be calculated following the primary LSP computation. If the primary LSP is a hop-by-hop routed LSP, and we know the nodes traveled by a hop-by-hop routed LSP, then we can also compute a path and use ER-LSP to create its backup. Otherwise, end-to-end LSP protection is not applicable.

Usually, constraint-based routing requires path computation at the LSP initiator node. This is because different LSP initiator node may have different constraints for a path to the same destination, and the constraints associated with a particular LSP initiator node are only known to that node. The reason is similar to source routing – the source determines the path.

The Shortest Path First (SPF) algorithm computes a path that is optimal with respect to some scalar metric. Many people (see [29]) propose that it is possible to modify the SPF algorithm in such a way that it can take into account the constraints. The algorithm is referred to as Constraint-based Shortest Path First (CSPF). There have been a number of proposals for CSPF, like [29]. The study of CSPF is out of the scope of this report, but a simple algorithm for CSPF is introduced to illustrate what CSPF is. It consists of three major steps:

(1) Among all the links, exclude the ones that violate the constraints we defined.
(2) According to the administration policy, map one (or more) link TE property as the scalar metric (cost) of the link.
(3) Use the SPF algorithm to calculate the path.

Based on (1), we know that all the links we consider will not violate the constraints, and so will be the path. For example, the link color stands for an administrative constraint. If we want a path that is only within the "red" domain, then only the links with color "red" are considered. The user's requirement is also a constraint – in fact, it is the most important one from the service point of view. If a user wants a path in which each link must have bandwidth 5Mb/s, then we do not consider all the links whose available bandwidth (the difference between the maximum bandwidth that may be reserved on this link and the bandwidth that has bee allocated) is less than that.

With regard to path computation for LSP protection/restoration, the constraint is that the links traveled by the backup LSP must not belong to the same Shared Risk Link Group (SRLG) as the primary LSP. Therefore, after the computation for the primary LSP, all links belonging to the SRLG to which the links of the primary LSP belong are excluded (not considered).

In order to avoid the protection contention between LSP layer and link layer (see Section 5.1.2), [30] proposes that the Link Protection Type of the links traveled by the LSPs that construct the protection mechanism should be "unprotected". Such a proposal is the second constraint that should be the considered if we follow that proposal.

With regard to (2), we can take any of the TE properties or administrative distance.

Let us have an example. We will establish an LSP that requires T1 bandwidth (1.544 Mb/s), which travels from Node 1 to Node 5. In Figure 1.9, the link directly from Node 1 to Node 5 has only 1 Mb/s available; others have enough or more. So the link from Node 1 to Node 5 is excluded. Then we consider the available bandwidth as the metric.

The cost of a link is calculated by (10^8 / available bandwidth). The link from Node 1 to Node 2 has available bandwidth 10 Mb/s, so the cost is 10. In such a way, the metric of every link is calculated (see Figure 1.10). Then, using the SPF algorithm, the shortest path from Node 1 to Node 5 is (Node1, Node4, Node3, Node5).

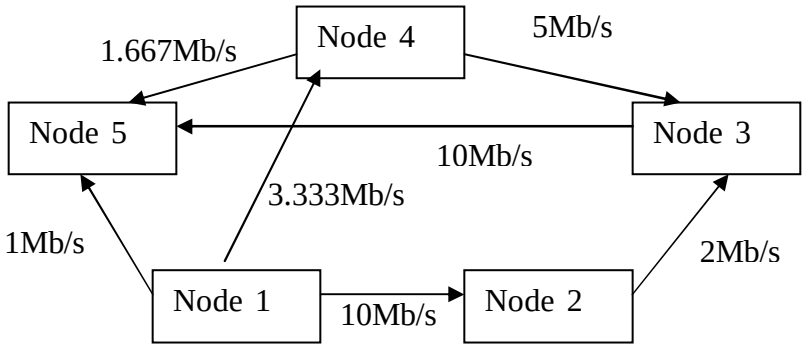


Figure 1.9: available bandwidth in the network

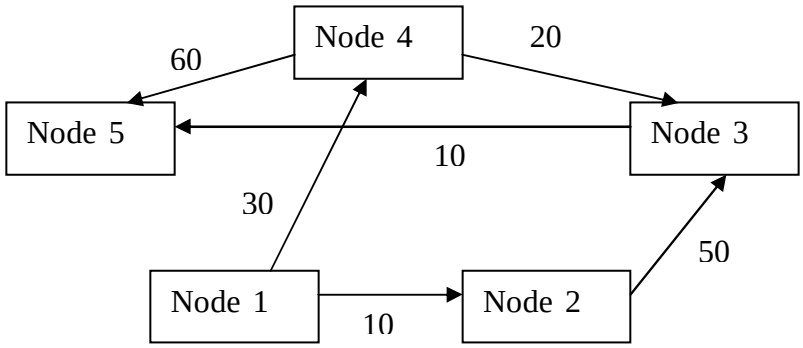


Figure 1.10: the metric of the links to be used by SPF algorithm

In general, path computation can be control-driven or data-driven. If the path computation is triggered by administrative control, e.g., the network administer configures a path and requires the path computation for an ER-LSP, then the path computation is called control-driven. The data-driven path computation does not require administration. User data arrives at a node. In order to deliver the data, the node computes a path before signaling the LSP. Path computation is triggered by the data's arrival, and it is called data-driven. Using the control-driven mode, the path can be pre-calculated and even pre-established (before user data arrives), so it is faster in response to data delivery.

3. Overview of Path Protection/Restoration

With the development of networks, new technologies provide high bandwidth capacity. The ever-increasing bandwidth leads to a significant data loss if a failure cannot be recovered timely. Users and network service providers require network survivability. For example, real-time applications require very fast network recovery. No network service provider wants unprotected networks. On the other hand, transmission systems deployment gives chances to network failure, for example, telecommunication fiber cables share the same ducts of other utility transport media. Cable cuts are difficult to avoid.

Network survivability has been a hot research topic in the industry. Today, multiple layer protection/restoration is possible. The protection/restoration mechanism can be implemented in the link layer or in the IP/GMPLS layer. For example, the architecture of an IP-over-WDM node can be viewed logically as:

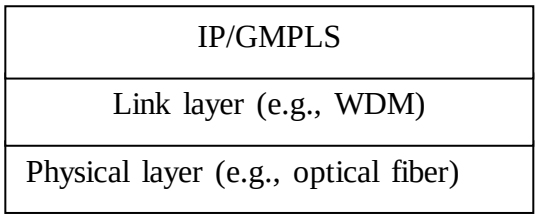


Figure 2.1: a logical view of the architecture of a GMPLS node

Protection/restoration mechanisms at the IP/GMPLS require relatively more time to recover, and using higher levels of recovery mechanisms may require more resources [31]. But there are limitations and disadvantages in the link layer protection, particularly in the optical network, e.g., complicated implementation, cost, instability due to duplication of functions, etc. It is still a challenge to implement recovery mechanisms at the WDM layer for the time being. Today a number of proposals have been studied in the industry to search for recovery mechanisms at the WDM layer, such as [32] and [33]. Furthermore, link layer protection cannot easily provide node protection [34]. The study of link layer recovery mechanisms is out of the scope of this report.

The motivation for using multiple layer protection is to provide the desired level of service in the most cost-effective manner [35]. With multiple layer protections, we need to prioritize them. The recovery mechanism that has higher priority is triggered first to recover failures. Usually, it is expected that lower layer recovery mechanism is closer to the failure, so it has higher priority. Also we need a coordination mechanism to avoid contention between different layer recovery schemes. One of the most popular coordination mechanisms is the hold-off timer. The hold-off time is the waiting time between the detection of a failure and taking MPLS-based recovery action. It allows time for lower layer protection to take effect [36]. If MPLS-based recovery is the only recovery mechanism desired, then the hold-off time may be zero. Assuming that we have SONET Automatic Protection Switch (APS) link protection, for example, within the hold-off time, GMPLS LSP path protection waits for the APS protection to switch. If the SONET APS succeeds protection within the hold-off time, then the hold-off timer is reset

and no further protection is needed. The original LSP can remain there. From this point of view, the link layer protection provides a means for LSP protection. Section 2.4.3 specifies how LSP signaling requires link layer protection when the LSP is being established. If the hold-off time expires, the LSP protection/restoration is triggered. The coordination mechanism introduces a tradeoff between rapid recovery and creation of a race condition where several layer protection mechanisms respond to the same fault.

GMPLS widens the application scope of MPLS, and people propose using GMPLS to build a unified control plane to manage all kinds of network nodes [14]. The GMPLS LSP protection/restoration has been an important recovery mechanism for network survivability.

Differently from traditional IP networks, MPLS networks establish label switched paths (LSPs) before data forwarding occurs. This potentially allows MPLS networks to pre-establish protection (backup) LSPs for working LSPs, and achieve better survivability than traditional IP networks.

Here we introduce what we need for the LSP protection/restoration mechanism in GMPLS networks.

(1) A method for computing the working and protection paths;
(2) A method for working and protection path signaling;
(3) A fault detection mechanism;
(4) A fault localization and notification mechanism to localize the fault and convey the information;
(5) A recovery mechanism to move the traffic over from the working path to the protection path or to reroute the fault;
(6) A repair detection mechanism to detect the original working path is fixed;
(7) An optional switchback or restoration mechanism to restore the traffic to the original working path.

Item (7) is optional and it is not time-sensitive. In some cases, it may not be desirable. For example, switching the traffic back to the original working path can disrupt the traffic (even for a very short time). It may not be desired under the user requirements. Item (6) may not be necessary in some cases. For example, if (7) is not wanted, then (6) is no needed.

Item (1) is implemented by the path computation component. For example, it uses CSPF to compute a path and selects the working and protection path. Usually it is proprietary. The path computation considers the Traffic Engineering properties of the network, administrative constraints and user requirements to calculate the backup and working path. For example, if both Link L and K share the same physical resource (e.g., they exist in the same optical cable), then either L or K should be considered in a particular working path computation and its backup.

As we introduced in the last sub-section, the GMPLS signaling protocols carries the link protection information, which can allow the nodes on the network to identify the working and backup path.

Traditional methods to monitor the health of data links may not be useful any more. For example, pure optical switches may not allow these methods to check the bit-rate, format or wavelength. Fault detection should work at the layer closest to the failure in order to achieve quick response. In optical network, this should be located in the physical layer (e.g., optical layer). For example, one method of fault detection at the optical layer is detecting the loss of light (LOL). Using software can also detect a faulty link/node, and it will be introduced in the subsequent section. However, fault detection at the physical layer provides fast and reliable solution, and it is preferred if it is applicable.

The optical network has its own character in failure. When one link is broken, e.g., a fiber cut, all the downstream nodes (in terms of data flow) can detect loss of light. Therefore, we also need a method to localize the failure. The Link Management Protocol provides a method, which will be introduced in the subsequent section.

Both GMPLS signaling protocols [26] and [25] are being extended to provide methods to support LSP protection/restoration. For simplicity, we use the term RSVP-TE to refer to [26] and CR-LDP to [25] from now on.

There are a number of objectives for the LSP protection/restoration mechanism. The LSP protection/restoration mechanism should
(1) optimize the use of resources;
(2) provide fast recovery and minimize the disruption to data traffic of any failure;
(3) minimize degrading the traffic and preserve the constraints on the traffic after switchover;
(4) minimize the recovery overhead (be simple);
(5) be cost-efficient.

At the end of our discussion, we will see that some of the above objectives are conflicting. There is a trade-off between them. It is impossible to achieve all of these objectives at the same time, and the choice depends on what the user wants and what is the network administration goal.

4. Multiple Protocols Contribute to GMPLS LSP Protection/Restoration

4.1 OSPF Extensions

The current routing protocols OSPF and IS-IS are extended to support Traffic Engineering and GMPLS. Here we take the popular OSPF as an example to see how it works.

The OSPF protocol is re-used to distribute information to support Traffic Engineering and GMPLS features. Two types of extensions have been added to the OSPF: TE extensions and extensions for GMPLS. The former is named OSPF-TE, which distributes TE properties over the network. The latter is referred to as GMPLS-OSPF, which distributes extensions dedicated to support GMPLS.

In the OSPF protocol, the message describing the local link information that is flooded throughout the network is named Link State Advertisement (LSA). A new LSA - TE LSA is defined to support Traffic Engineering and GMPLS (see [37] for more information).

The Type-Length-Value (TLV) structure (see Figure 3.1) is used as the payload in the TE LSA. The Type specifies the type of the data; the length specifies the length of the whole TLV structure, and the Value describes the information regarding to Traffic Engineering and GMPLS support.

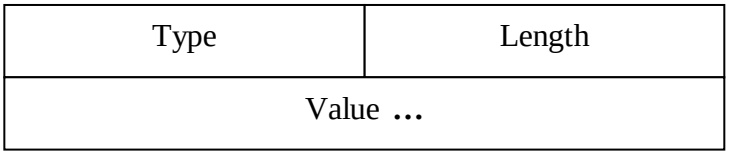


Figure 3.1: the TLV structure

The TLV structure can be nested, for example, sub-TLVs are carried as the value in the higher-level TLV. So it is extendable, which is good for future development. There are two TLVs: router address TLV and link TLV.

Router address TLV
The router address is the router ID of the node that advertises the LSA. The TE LSA must carry a router address TLV. It is type 1, the length is 4, and the value is the 4-octet IP address.

Link TLV
The link TLV contains information about the link. And it consists of a set of sub-TLVs, each of which describes a piece of particular information about Traffic Engineering or GMPLS features. The information of these sub-TLVs are introduced in the subsequent sections. The Link TLV is type 2 and the length varies.

OSPF does not process the contents of the TE LSA.

4.1.1 Introduction to Traffic Engineering Extensions to OSPF (OSPF-TE)

When a router starts, it discovers the information about its own links (interfaces) – the links connecting the router to networks (or other routers). Then the routing protocol is used to advertise the information to other routers. The information is passed around from router to router. Ultimately, every router has identical information about the network and

the information is stored in a database named Link State Database (LSDB). Each router will independently calculate the best path to other nodes in the network using a path computation algorithm. For example, the popular OSPF protocol uses Dijkstra's Shortest Path First (SPF) algorithm to come up with a SPF tree, which serves as a map for data routing (see Figure 3.2). Then according to routing policies, an appropriate route is selected and put into the routing table.

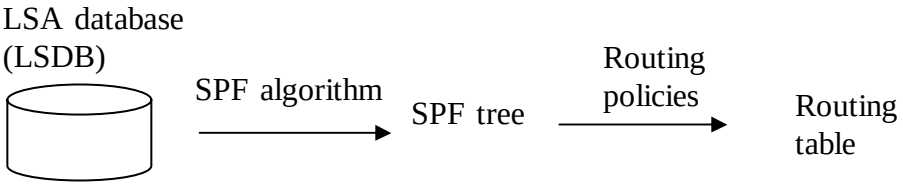


Figure 3.2: from LSDB to an appropriate route

Conventionally, the information of the links includes the status of the links (e.g., up/down), metric (cost), etc. The information does not support Traffic Engineering. For example, the metric is assigned to routes as a means of ranking them from the most preferred to the least preferred. The calculation of the metric is static. The bandwidth metric used in Cisco routers is calculated as: metric = 10^8/(link bandwidth). Thus a higher-bandwidth path is always preferred over a lower-bandwidth path. But what if a T1 link of the preferred path is heavily loaded with traffic and a 64k link is lightly loaded?

Because the TE properties (e.g., bandwidth availability, administrative constraints) are not provided or considered in conventional routing protocols, the routing decision does not support Traffic Engineering.

Relying on the current routing protocols, TE properties are added into the messages that are flooded throughout the network. In IETF, the draft OSPF-TE [37] proposes the following TE properties that should be considered to support Traffic Engineering, and they rely on the OSPF opaque LSA advertising mechanism to distribute the TE properties. Each of the following 9 items constructs a sub-TLV in the link TLV of the TE LSA. Note that they are optional except the first two sub-TLVs: Link type and Link ID.

1 - Link type
2 - Link ID
3 - Local interface IP address
4 - Remote interface IP address
5 - Traffic engineering metric
6 - Maximum bandwidth
7 - Maximum reservable bandwidth
8 - Unreserved bandwidth
9 - Resource class/color

Link type
It specifies if the link is (1) point-to-point or (2) multi-access link. For the time being, only point-to-point link is completely supported.

Link ID
The Link ID identifies the remote end of the link. For point-to-point links, this is the Router ID of the neighbor.

Remote Interface IP Address
It specifies the IP address of the neighbor's interface corresponding to this link. For unnumbered links, this is the link remote identifier (see Section 2).

Local Interface IP address
It specifies the IP address(es) of the interface corresponding to this link. If there are multiple local addresses on the link, they are all listed in the appropriate structure of a routing message. For unnumbered links, this is the link local identifier (see Section 2).

The local and remote interface IP addresses identify the parallel links between two nodes.

Traffic Engineering Metric
A metric is a variable assigned to routes as a means of ranking them from best to worst or from most preferred to least preferred. The Traffic Engineering metric specifies the link metric for traffic engineering purposes. This metric may be different than the standard OSPF link metric.

Maximum Bandwidth
It specifies the maximum bandwidth that can be used on this link from the LSA-originating router to its neighbor. For example, a T1 link has maximum bandwidth 1.544 Mb/s, an OC-48 link has around 2.5 Gb/s.

Maximum Reservable Bandwidth
It specifies the maximum bandwidth that may be reserved on this link in the direction from the LSA-originating router to its neighbor. Note that this may be greater than the maximum bandwidth (the link may be oversubscribed). For example, an OC-48 link may be configured to have maximum reservable bandwidth 2.75 Gb/s (10% oversubscribed).

Unreserved Bandwidth
It is the difference between the Maximum Reservable Bandwidth and the bandwidth that has been reserved. There are eight priority levels (from 0 to 7) of unreserved bandwidth. This information specifies the unreserved bandwidth of each priority level. Priority 0 is the highest.

Resource Class/Color
It specifies administrative group membership for this link, in terms of a bit mask. A link may belong to multiple groups - if so it has multiple bit masks.

A node advertises the TE-LSA whenever one of its own links gets the TE properties updated. The routers that receive these TE-LSAs store them in a database that is named TE Link State Database (TE-LSDB). The TE LSDB is synchronized across all nodes supporting OSPF-TE within an area. So each node in that area has an identical view of the TE properties of the network. The path computation component of the control plane can use the information provided by TE LSDB to compute a path that meets a user's requirements and the traffic engineering goals (see Figure 3.3).

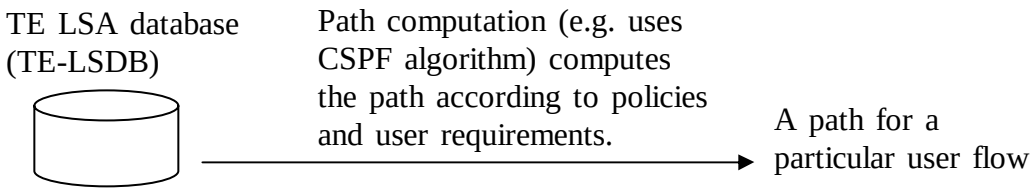


Figure 3.3: from OSPF-TE to a path

Like the regular links, FA-LSPs (an LSP is advertised as a link in the network - see the Section *Traffic Engineering* before) also have the TE properties we just introduced. They are also stored in the database TE-LSDB. This information is also used by the path component to compute a path. As examples, here we list some of the TE properties of a FA-LSP (see [20] for more).

(1) Link type: an FA-LSP must be a "point-to-point" link;
(2) Local and Remote interface address: if the FA-LSP is to be numbered, then the local interface IP address is the head-end address of the FA-LSP link. And the remote interface IP address is the address of the ending node of the FA-LSP;
(3) Maximum Bandwidth (also named Maximum LSP Bandwidth): It specifies the maximum bandwidth that may be reserved on this LSP. Therefore, it is like the Maximum Reservable Bandwidth of a link.
(4) Interface Switching Capability: it is the Interface Switching Capability of the first link of the FA-LSP.

As it is introduced, the above TE properties are carried by the TLV structure within the TE LSA and distributed by OSPF.

4.1.2 Extensions to OSPF for supporting GMPLS

The following information is needed to support GMPLS: (1) unnumbered link identifier; (2) Link Protection Information; (3) Shared Risk Link Group (SRLG) Information; (4) Interface Switching Capability Descriptor. They also rely on the TE LSA of OSPF to be distributed into the network.

4.1.2.1 Unnumbered link support in OSPF

How unnumbered link is supported has been introduced in Section 2. In OSPF, the 32-bit unnumbered link identifier (local and remote) is simply put into the value field of the TLV

structure. The type is 11. If the remote identifier is unknown (e.g., at the router start-up), then it is 0. Carried by the TE LSA, the unnumbered link identifier is advertised.

4.1.2.2 Shared Risk Link Group (SRLG)

The SRLG is also a link property and it is advertised by the link sub-TLV. The SRLG is specified by a 32-bit word, contained in the Value field of the sub-TLV structure. The sub-TLV type is 16. If a link can belong to multiple SRLG, then all of them are listed in the sub-TLV structure and the order is irrelevant. An example is shown in Figure 3.4.

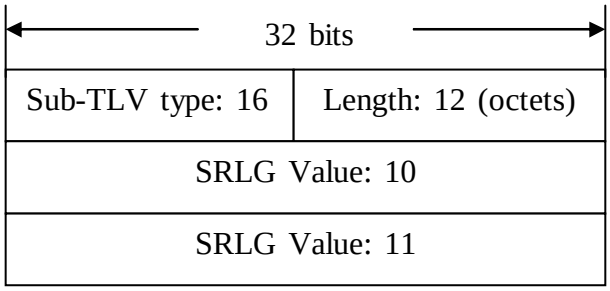


Figure 3.4: SRLG sub-TLV

4.1.2.3 Link Protection Type

The link protection type can be considered by the path computation component to compute a path and it is distributed throughout the network. There are six link protection types (See Section 2.4.2 of this report for what they are.):

(1) Extra Traffic;
(2) Unprotected;
(3) Shared;
(4) Dedicated 1:1;
(5) Dedicated 1+1;
(6) Enhanced.

If the routing protocol does not distribute the link protection type for a link, then the protection attribute of that link is unknown.

The link protection type is encoded in a sub-TLV of the link TLV. The sub-TLV type is 14 and it has 4 octets (see Figure 3.5). But only the first octet is used. The first octet is used for indicating protection types and the other octets are reserved. The first octet may contain the following value to indicate the link protection type:

0x01 Extra Traffic
0x02 Unprotected
0x04 Shared
0x08 Dedicated 1:1
0x10 Dedicated 1+1
0x20 Enhanced
0x40 Reserved
0x80 Reserved

Protection type	reserved

Figure 3.5: the Value field of the sub-TLV for link protection type

4.1.2.4 Interface Switching Capability Descriptor

The interface switching capability is encoded by a sub-TLV (type 15) of a link TLV. The field contains one of the following codes. And each code signals the correspondent type.

Code	Type
1	Packet-Switch Capable-1 (PSC-1)
2	Packet-Switch Capable-2 (PSC-2)
3	Packet-Switch Capable-3 (PSC-3)
4	Packet-Switch Capable-4 (PSC-4)
51	Layer-2 Switch Capable (L2SC)
100	Time-Division-Multiplex Capable (TDM)
150	Lambda-Switch Capable (LSC)
200	Fiber-Switch Capable (FSC)

The code is not consecutive, as it allows for future extension.

4.2 Link Management Protocol (LMP)

Neighboring nodes may run the Link Management Protocol (LMP) [38] for link management. With the development of optical networks, nodes include photonic switches (PXCs), optical crossconnects (OXCs), routers, switches, add-drop multiplexors, WDM systems and so on. LMP support any type of nodes. And LMP supports TE links.

The link multiplexing capability has an effect on how to do the link management, e.g., resource allocation. To allow interworking between links with different multiplexing capability, sub-channels of a component link should be able to be configured as a data link. For example, several Ethernet links are multiplexed into an OC-12 link, which is connected to a node. The node should allow each Ethernet link to be configured as a data link. So that link management on each Ethernet link is possible if required.

To run LMP, a control channel must be established between the node pair. The control channel should be separated from the data channel [10]. And, the node pair can communicate bi-directionally at least through one of the control channels. If so, then an LMP adjacency can be formed between the two nodes. Multiple active control channels are possible in an LMP adjacency, and the control channel ID (CCID) is used to identify each one.

LMP messages are encoded as data in IP packets, and it runs directly over IP except for the LMP Test message. The LMP Test message is sent over the data links (in-band) for

link connectivity verification. So optionally it is limited by the transport media, e.g., not necessarily encoded as data in IP packets.

LMP functions are: control channel management, link property correlation, link connectivity verification, and fault management.

(1) **Control channel management** is used to establish and maintain control channels between LMP adjacent nodes. The control channel can be used to exchange routing, signaling, and other control messages.

To establish the control channel, the IP address for the far-end of the control channel must be known (e.g., by configuration). A node sends a LMP Config message to its neighbor, which contains parameters, e.g., the LMP keep-alive interval. The receiver of the Config message must reply an acknowledgement. If both sides agree on the parameters, the control channel is established. After that, the LMP keep-alive message is sent periodically to maintain the control channel.

After two neighboring nodes successfully establish the control channel, control messages can be exchanged through the control channel. Examples of these control messages may be label distribution information implemented by RSVP-TE, network topology and state distribution information implemented by OSPF-TE, fault management implemented by LMP, and so on.

(2) **Link property correlation** is used to synchronize the properties of the TE link and verify the configuration. An example of TE link is shown in following figure. LSP is taken as a TE link by Node1 and Node3, which is constructed by link (A, B) and link (C, D). Link (A, B) or link (C, D) is called a data link.

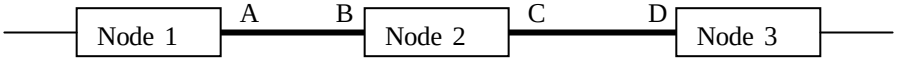


Figure 3.6: An LSP as a TE link starting from Node1 to Node3

After the LSP is established by the signaling protocol, LMP may be used to synchronize the properties of the TE link. So, Node1 may send a LMP LinkSummary message to Node3, which is constructed by LMP objects as:

<LinkSummary Message> ::= <LMP message header><Message ID>
<TE Link><Data Link (A, B)><Data Link (C, D)>

Within each Data Link object, sub-objects may contain information about link reservable bandwidth, wavelength if there is any, interface switching capability such as interface A for data link (A, B).

The receiver of LinkSummary message must verify that the information obtained from the message makes sense and matches the information that is stored in the routing database or configuration inventory. For instance, the interfaces A and D must be of the same interface switching capability type in the example shown in Figure 3.6. The receiver of a LinkSummary message must reply an acknowledgement, which reports the correctness of the TE link properties.

(3) **Link connectivity verification** is used to verify the physical connectivity of the data links between the nodes.

In the example shown in Figure 3.6, Node1 and Node3 may exchange LMP Test messages between interface A and D through link (A, B) and (C, D) to verify the physical connectivity of the TE link on a periodic basis. The verification messaging must be transported by the data-bearing channel, not the control channel.

(4) **Fault management** provides a fault localization procedure. Because the LMP fault management is within the scope of this report, let us discuss it in detail.

The Link Management Protocol introduces a fault localization procedure to localize failures. It can localize the path failure by quickly reporting the status of one or more data link. It is designed to work for both unidirectional and bi-directional LSPs.

During the Link Property Correlation, both LMP-capable nodes can signal whether they support LMP fault management. If they do, then LMP fault management messaging becomes one of the control signals between these two nodes.

In optical networks, e.g., nodes are PXCs in the network, if one of the data links fails, then all the downstream nodes (in terms of data flow) may detect the failure due to the nature of light, e.g., loss of light. The LMP fault management requires each node that has detected the failure to send a LMP ChannelStatus message to the upstream node. This ChannelStatus message can report all the broken channel/links together. The upstream node must acknowledge the message by a LMP ChannelStatusAck message. Then the upstream node checks if there is any local data link failure, for example, it checks if the input side has any signal. If the input side is working fine, the failure is localized; otherwise, the node will continue sending LMP ChannelStatus messaging upstream. After the local checking, the upstream node must send a ChannelStatus message to the downstream node to report the status.

On the other hand, after the downstream node receives the ChannelStatusAck, it expects a ChannelStatus from the upstream node. If it receives no ChannelStatus, it should send a ChannelStatusRequest to solicit the message.

The time-sequence diagram in Figure 3.7 outlines how it works. Let us suppose that Node 2 is the downstream node relatively to Node 1 (in terms of data flow). Node 2 detects a failure.

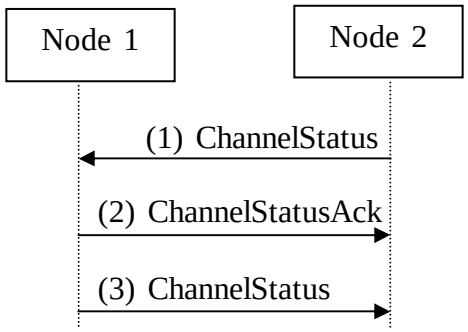


Figure 3.7: channelStatus messaging

When the fault is localized, the upstream node which connects the failed link should trigger the signaling to get protection/restoration. And it does not perform LMP ChannelStatus messaging to upstream nodes any more.

Let us have an example to see how it works in a pure optical network. There are three PXCs in the example shown in Figure 3.8. An LSP travels the data links of these three nodes. The control channel is out-of-band. Assuming that the data link through which the LSP with the flow direction from PXC 1 to PXC3 is failed. Both PXC 2 and 3 can detect the failure. For simplicity, only one direction of the LSP is shown.

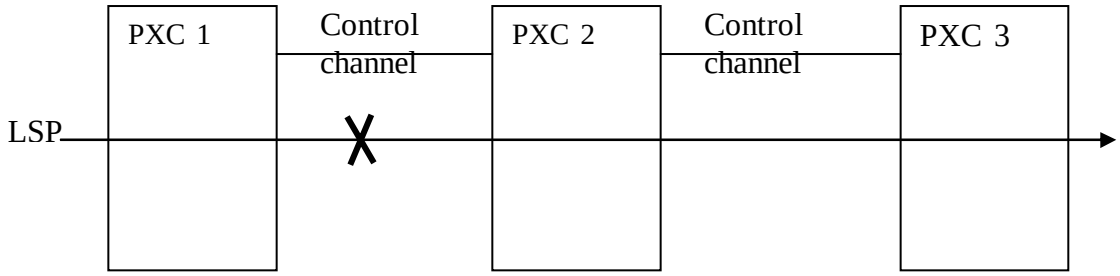


Figure 3.8: LMP fault management localizes the fault

PXC 3 sends the LMP ChannelStatus message to PXC 2, which acknowledges with a ChannelStatusAck. PXC 2 locally finds out that there is no input signal and the failure is propagated from upstream. So it tells PXC 3 also by a ChannelStatus message. Meanwhile, PXC 2 sends another ChannelStatus message to PXC 1, which tells PXC 1 that no signal comes in. PXC 1 replies with a ChannelStatusAck. PXC 1 locally finds out that the input is fine. So it sends PXC 2 a ChannelStatus message, which tells PXC 2 that it is clear. Thus PXC 1 has localized the failure. After that, the recovery will be triggered, for example, signaling starts to establish a reroute. Section 5 will specify the recovery mechanisms in detail.

If the failure affects both directions of the LSP, e.g., a fiber cut, then the same procedure is performed on each direction.

4.3 GMPLS Signaling

There are two major label distribution protocols to perform GMPLS signaling: RSVP-TE with extensions and LDP with extensions.

4.3.1 GMPLS signaling: RSVP-TE with extensions

Traditional RSVP (RFC2205) provides a means for an application to communicate its QoS requirements to an Integrated Services Internet infrastructure. RSVP is a control protocol that signals QoS requirements on behalf of a data flow. Before data delivery occurs, RSVP establishes a resource reservation for a simplex (one way) flow along its path. A simplex flow is a unidirectional flow traveling from its source to its destination. To allow duplex (two-way) communication, we need RSVP to reserve resource twice – one for each direction. RSVP consults a routing table in a router for the next hop. RSVP relies on IP or UDP for message transport.

RSVP must carry the following information:

- Information for flow identification, so that the flows with particular QoS requirements can be recognized within the network. This may include sender IP address, receiver IP address, port numbers and so on.
- Traffic specification and QoS requirements.

RSVP carries the information from the source host to the destination host along the router/switch on the path. There are two basic messages in RSVP: PATH and RESV messages. A PATH message travels from the sender to the receiver and include traffic specification and classification information provided by the sender. The PATH message identifies the path from the sender to the receiver and it collects status about the resource along the path. When the PATH arrives at the receiver, the receiver sends back a RESV message back toward the sender along the reverse of the path. The RESV message communicates with every router to make a resource reservation. See the following figure for PATH/RESV messaging.

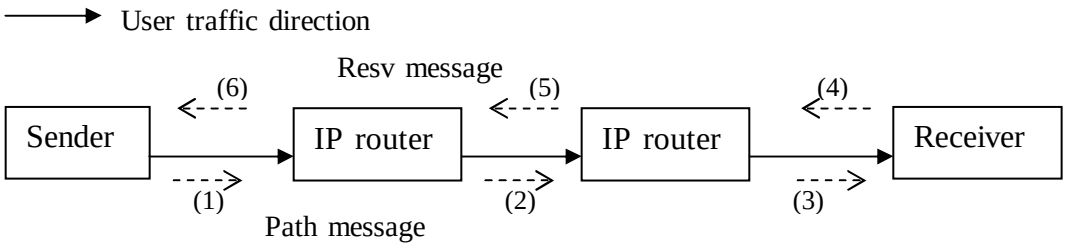


Figure 3.9: RSVP signaling to reserve resource

Each router along the path creates a software record (software state) for the particular flow, which keeps the flow classifier, QoS requirements, next hops, previous hops and other related information. These records have a timer, which means these software states will be removed after some time-out. So after some time period, the PATH message is

transmitted and the RESV travels the reverse path – the process repeats on a regular time interval basis. This is called refresh messaging, which keeps the software states and the router can continue providing QoS to the flow.

Extensions have been added to RSVP (RSVP-TE) to support label distribution for LSP signaling in MPLS. To establish an LSP, the sender node, with respect to the path, creates an RSVP Path message which contains a LABEL_REQUEST object. The LABEL_REQUEST object indicates that a label binding for this path is requested. A SESSION_ATTRIBUTE object is introduced to provide additional control information such as setup and hold priorities, local protection and so on. The RSVP-TE Path message carries this object during LSP signaling. When the Path message arrives at the destination node of a LSP, the node responds to the LABEL_REQUEST object with a LABEL object in its RSVP-TE Resv message. If the node is not the sender node, it allocates a free label and puts it into the LABEL object. And the Resv message is sent to the upstream node. The node that receives a Resv message with a LABEL object will use this label as the outgoing label in the forwarding entry of its forwarding table. It also allocates a free label for the upstream node, and puts it into the LABEL object attached to the Resv message. The Resv message is sent upstream again. Such a label distribution procedure repeats until the Resv message arrives at the sender node. The LSP establishment is done. The sender node has some criteria to classify different traffics and puts the predefined traffic into the appropriate LSP. In the example shown in Figure 3.10, the sender node will attach label 3 to all the packets before it forwards the packets out. When the packet arrives at the transit IP router, the label is replaced by 6 and then forwarded again. Such a label swapping procedure repeats on each node and the packet finally reaches the destination.

For label distribution, the LABEL_REQEUST and LABEL objects are mandatory, but other objects defined in RSVP-TE are optional, e.g., the SESSION_ATTRIBUTE mentioned above.

Note that because the label distribution is done with RSVP, each router can associate the resources with the LSP during LSP signaling. Therefore, resource reservation can be done in the meanwhile.

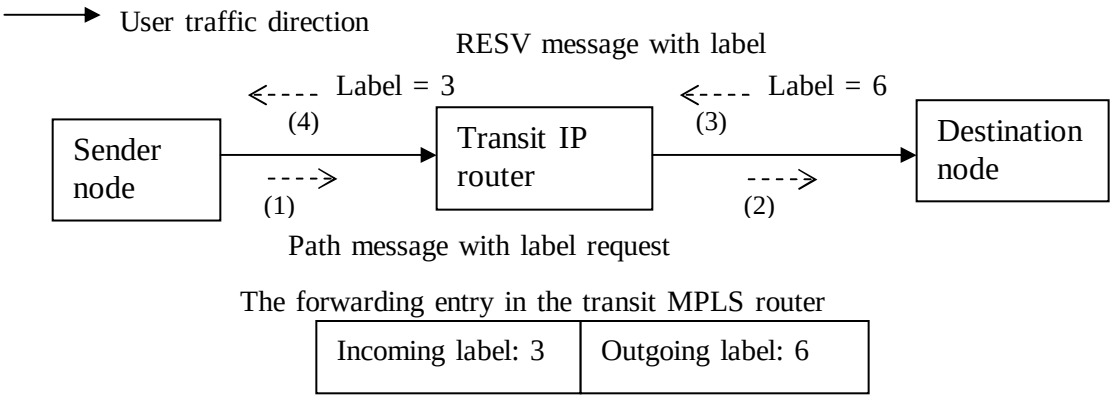


Figure 3.10: RSVP-TE signaling to distribute labels for establishing LSP

In each intermediate node, RSVP-TE consults the local routing table for the next hop. The LSP established in this way by RSVP-TE is named hop-by-hop routed LSP. Signaling in this way does not meet the requirements of many applications, for example, traffic engineering. So, the Explicit Route Object (ERO) is added to RSVP-TE to support the explicitly routed LSP (ER-LSP), which is similar to source routing. This object allows the path taken by RSVP-TE messaging to be pre-determined by the source. With ERO, the ingress node of the LSP can define which transit node the LSP will travel to reach the destination (egress node of the LSP). And the ER-LSP can be routed away from network failures, bottlenecks, or congestion.

The ERO is carried by the RSVP Path message. It contains a sequence of IP prefixes or a sequence of Autonomous Systems. The ERO tells the routing mechanism where to forward the Path message. We consider the following an example shown in Figure 3.11.

R1 is going to establish an explicitly routed LSP (R2, R3, R4, R5). R1 constructs the object ERO to have the sequence of nodes - R2, R3, R4 and R5. And each node can be represented by an IP address prefix. Then R1 creates the RSVP PATH message carrying the ERO as well as the LABEL_REQUEST object. Before the message is sent out, R1 checks the top of the ERO, and ERO tells R1 the next hop is R2. R1 sends it to R2. R2 looks at the top of the ERO and finds itself is on the top. R2 looks at the next one, which is the IP address prefix for R3, and takes it as the next hop for the message. R2 removes the top IP address prefix that is one of its interfaces through which the message comes in, before it forwards the message. R3, R4 and R5 follow the same algorithm as R2 does. When R5 receives the PATH message, ERO only has one prefix, which is one of the interfaces of R5. Note that a RSVP state has been created on every router along the path.

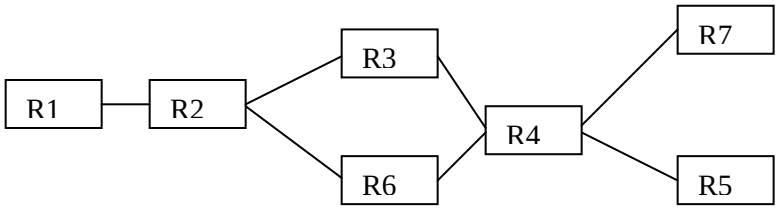


Figure 3.11: ER-LSP from R1 to R5

In order to respond to the LABEL_REQUEST object, the R5 constructs a RSVP RESV message along with the LABEL object. The message can be forwarded to R4 by R5. R4 updates the LABEL object and further forwards the message to R3. The message follows the RSVP state that the PATH message has created along the routers R4, R3, R2 and finally reaches R1. Thus a LSP is created. Note that an intermediate router may not be able to tell the difference between a label for an established, explicitly routed LSP and

one for a hop-by-hop routed LSP, as it does not need to make this distinction during programming the data forwarding plane.

If the ERO specifies every node of the LSP or every autonomous system traveled by the LSP, then the LSP is called “strictly” explicitly routed. If the ERO specifies some nodes or some autonomous systems traveled by the LSP, then the LSP is called “loosely” explicitly routed.

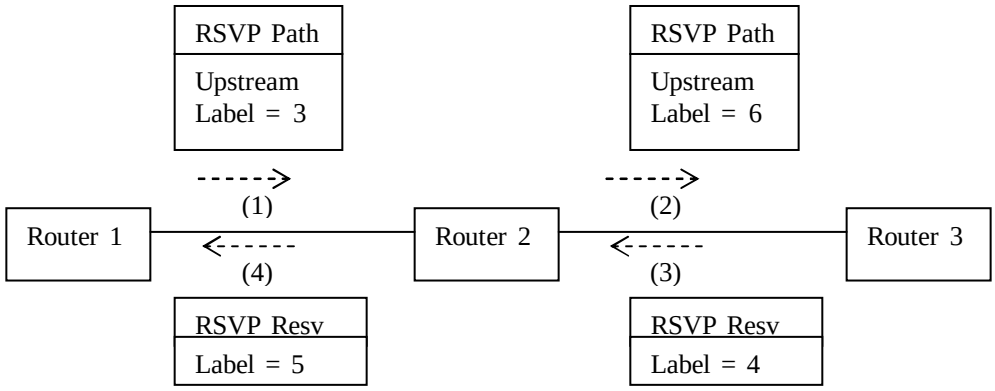
GMPLS extends MPLS to support multiple different interfaces. RSVP-TE is also extended to support GMPLS signaling. The label request object and label object must be generalized (see Section 2.4.2 of this report). In [39], the Generalized Label Request Object (carried by the Path message) and Generalized Label Object (carried by the Resv message) are defined. The Generalized Label Request Object allows different transit nodes with different data links to allocate labels.

When the Path message carrying the Generalized Label Request Object arrives at a node, the node makes sure the label request information (including the switching type, LSP encoding type and generalized payload ID) must be satisfied by the interface through which the traffic comes (incoming interface), the node itself and the interface through which traffic gets forwarded (outgoing interface). The node itself and the interfaces through which the traffic is transmitted should be able to support the LSP encoding type. The incoming interface should be able to support the switching type. Note that the label switched path (LSP) can be established only between (or through) interfaces of the same switching type. Usually only the egress will check the generalized payload ID (because the payload is transparent to transit nodes). If the egress does not support the payload, the LSP cannot be established. In all of these cases, a RSVP-TE PathErr message is generated.

There is no internal structure within a label. If we want nested LSPs (an LSP within another LSP), each LSP must be established separately.

The RSVP-TE Resv message carries the generalized label upstream along the reverse path set up by the Path message. The node that receives the Resv message must verify that the label is acceptable. In some situations, the label assigned by the downstream node could not be available, for example, an optical cross-connect does not have the wavelength to model the label. If the label is not acceptable, the node will generate a RSVP-TE ResvErr message.

In GMPLS-RSVP-TE, a procedure for bi-directional LSP set-up is introduced. The procedure is added to the establishment of a unidirectional LSP. The Upstream_Label object is defined in [39] and it is carried by the RSVP-TE Path message. This object is similar to the Generalized Label object. It contains a generalized label that is allocated by the upstream node and used by the downstream node for label swapping. An example is shown in Figure 3.12. The node that receives the Upstream_Label must verify the label is acceptable.



The forwarding entry in Router 2 for direction from R1 to R3

Incoming label 5	Outgoing label 4

The forwarding entry in Router 2 for direction from R3 to R1

Incoming label 6	Outgoing label 3

Figure 3.12: bi-directional LSP set-up using RSVP-TE

To support explicitly routed LSP in the context of GMPLS, just the IP address or the identifier of an autonomous system may not be adequate. For example, the LSP set-up needs to concatenate two LSPs to form an LSP at the edge of two different networks (e.g., an optical network and an IP network). There may be a number of wavelengths in a fiber (a link), and a particular wavelength (a label) is needed. The ingress of the LSP needs to specify the particular label (wavelength). So to support GMPLS signaling, a Label subobject is defined, which follows the IP address or the identifier of an autonomous system in the ERO. The Label subobject allows the ingress of the LSP to specify a particular label of a data link.

To improve network survivability, the protection information is considered in GMPLS signaling. It includes
(1) link protection type;
(2) indication of whether the path is primary or backup.

The link protection type indicates what link protection capability is desired for the links constructing the LSP to be set up (see Section 4.1.3.3 for the link protection types). During LSP signaling in GMPLS, label distribution protocols (RSVP-TE, or LDP) may carry the protection information. The link protection type in the protection information is one of the TE requirements (or a constraint) for a LSP to be set up. So the LSP set-up will not continue if the desired link protection cannot be provided.

Signaling a hierarchical LSP

GMPLS supports interfaces that have different switching capabilities. The Interface Switching Capability Descriptor describing the capability is distributed by the routing protocol throughout the network (see the section *Enhancements in the Routing Protocol to Support GMPLS*), and each node stores this information in the TE link state database (TE-LSDB).

An edge node is the one that connects two different networks constructed by different nodes, for example, an optical switch that has interfaces providing SONET signals and interfaces providing WDM capability for photonic cross-connects. When an edge node signals an LSP, relying on the Interface Switching Capability Descriptor provided by the TE-LSDB, it can find out whether the interface the signaling comes in has different switching capability from the outgoing interface. If so, it knows it may be at the boundary of two levels of LSP. For example, an edge node may have the Interface Switching Capability Descriptor of its interfaces like:

Descriptor for Interface 1:
Interface Switching Capability = TDM
Encoding = SONET
Max Bandwidth[0] = 10 Gbps, for priority 0

Descriptor for Interface 2:
Interface Switching Capability = FSC (Fiber Switch Capable)
Encoding = Ethernet 802.3
Max Bandwidth[0] = 100 Gbps, for priority 0

When the signaling message comes in from interface 1 and the outgoing interface for it will be interface 2, the edge node understands that a hierarchical LSP will be established (see the example in Figure 3.13). The low-order LSP is tunneled through the high-order LSP, and multiple low-order LSPs can be aggregated into the high-order LSP.

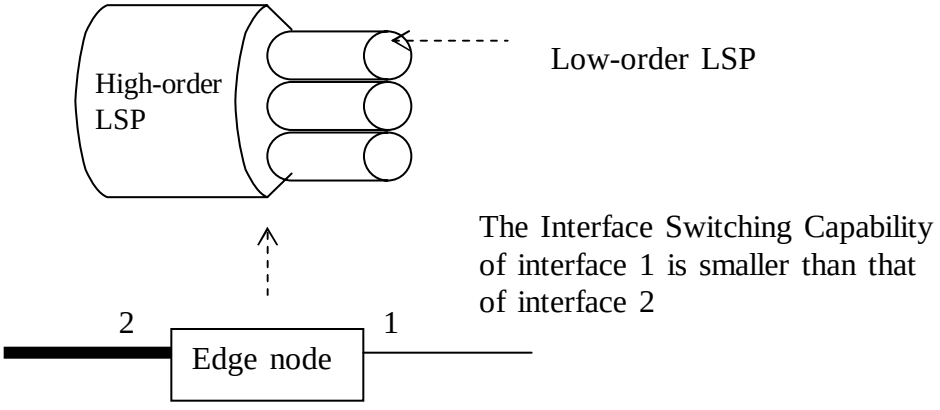


Figure 3.13: the edge node knows if a hierarchical LSP will be established

Here we illustrate how the hierarchical LSP set up is done using RSVP-TE with extensions to support GMPLS. Lower-order LSPs trigger the set-up of a higher-order LSP. Nodes at the border of two different networks in terms of multiplexing capabilities are responsible for establishing higher-order LSPs and aggregate lower-order LSPs. Figure 3.14 shows an example. Packet-Switch Capable (e.g., IP packets) LSR 1 and 2 are connected by a 500 Mb/s Ethernet link, so are LSR 7 and 8. SONET switches and LSRs are connected by OC-12 links; SONET switches and PXCs are connected by OC-192 links; PXCs are connected by optical fibers. Note that PXC 4 and 5 may not be connected directly, e.g., there are other PXCs between the two. Let us assume that the edge PXC has the capability to convert electrical signals to optical signals. They have interfaces that can provide SONET signals and interfaces that can provide WDM capability. An LSP (LSP 1) is going to be established from LSR 1 to LSR 8, which requires 500 Mb/s bandwidth.

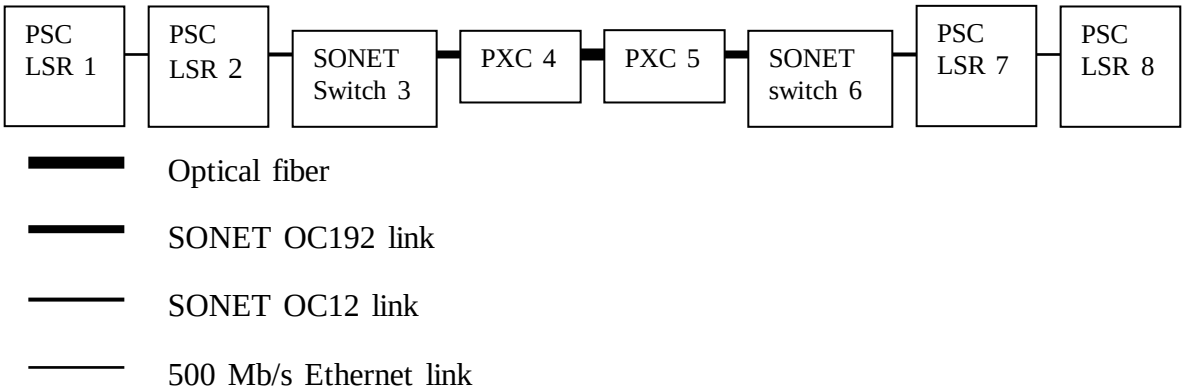


Figure 3.14: a hierarchical LSP is established between LSR1 and LSR8

We assume that all links have enough bandwidth for the LSPs to be established, and that there is no existing LSP between the different nodes. The GMPLS signaling using RSVP-TE starts from LSR1 (see the following figure).

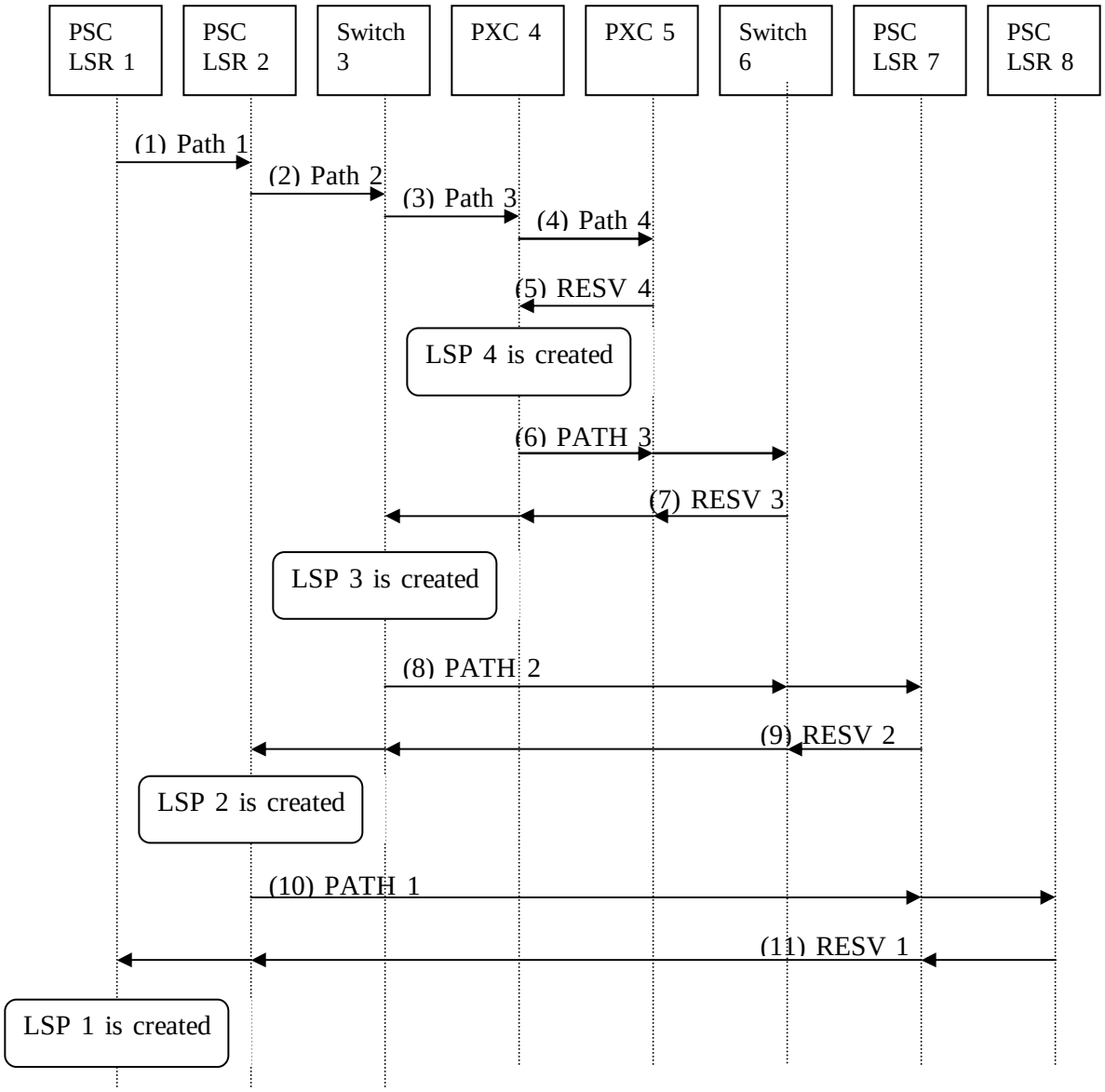


Figure 3.15: the time-sequence of establishing a hierarchical LSP

(1) The RSVP-TE Path message (Path 1) generated by Router 1 arrives at Router 2. This is the Path message for LSP 1, so let us call it Path 1. Based on the link information from the TE Link State Database, Router 2 knows that the path must cross links that are different (e.g., different types of interface, bigger multiplexing capacities). So Router 2 is triggered to establish a new path LSP2 that will be terminated on Router 7. This represents the next-higher LSP through which the LSP from Router 1 to Router 8 will be multiplexed. Router 2 generates another RSVP-TE Path message (Path 2 for LSP 2) destined to Router 7.

(2) Path 2 arrives at SONET Switch 3. Again, Switch 3 finds out that the LSP must cross different links. Switch 3 is going to establish LSP 3, and it generates RSVP-TE Path message destined to Switch 6 (Path 3).

(3) Path 3 arrives at PXC 4, which triggers PXC 4 to establish LSP 4. So PXC 4 generates a Path message destined to PXC 5 (Path 4).

(4) Path 4 arrives at PXC 5.

(5) PXC 5 responds with a RSVP-TE RESV message. Let us call this RESV message Resv 4. Resv 4 arrives at PXC 4, and the LSP 4 is created. LSP 4 is a TE link. It will be advertised by the routing protocol that runs at this level (e.g., the network constructed by the PXCs) as a lambda-switch-capable link. This LSP is a FA-LSP. The capacity of this TE link in the advertisement is the difference between its maximum capacity (e.g., a number of lambdas) and the share (e.g., one lambda) that has been allocated for the OC-192 bandwidth.

(6) Then PXC 4 continues signaling for LSP 3. The PATH message Path 3 goes on.

(7) Path 3 arrives at Switch 6, and Switch 6 responds with a RSVP-TE RESV message. Let us call it RESV 3.

(8) RESV 3 arrives at Switch 3. LSP 3 is created. LSP 3 is a TE link. It will be advertised by the routing protocol that runs at this level (e.g., the network constructed by OC-192 switches) as a TDM-switch-capable link. This LSP is a FA-LSP. The capacity of this TE link in the advertisement is the difference between its maximum capacity (e.g., OC-192 bandwidth) and the share (e.g., OC-12 bandwidth) that has been allocated for the LSP 2 being established. Then the LSP 2 set up procedure continues, and Path 2 goes on to Router 7.

(9) Router 7 responds with a RSVP-TE RESV message (RESV 2).

(10) RESV 2 arrives at Router 2 and LSP 2 is created. LSP 2 is a TE link. It will be advertised by the router protocol that runs at this level (e.g., the network constructed by OC-12 switches) as a TDM-switch-capable link. This LSP is a FA-LSP. The capacity of this link in the advertisement is the difference between its maximum capacity (e.g., OC-12 bandwidth) and the share (e.g., 500 Mb/s) that has been allocated for the LSP 1 being established. Then the LSP 1 set up procedures continues and Path 1 goes on to Router 9.

The hierarchical LSP established in the above example is illustrated in Figure 3.16.

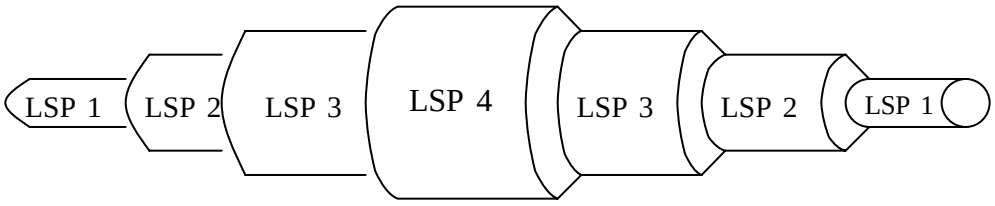


Figure 3.16: the hierarchical LSP in the example

If there is an existing FA-LSP that can satisfy the LSP being established, e.g., its unreserved bandwidth is bigger than what the LSP being established needs, then the edge node is responsible for tunneling the low-order LSP onto the existing high-order FA-LSP. In the above example, if LSP 4 has already been established (between PXC 4 and 5) when the Path message for LSP 3 (Path 3) arrives, then LSP 4 is treated as a single link and the Path 3 message will take PXC 5 as its destination, which is the ending node of LSP 4.

If the LSP being established is an explicit-routed LSP (ER-LSP), the RSVP-TE Path message carries an Explicit-Routed Object (ERO). A node receiving this message determines if it is the edge node at the boundary of two LSPs. If it is not, the conventional signaling goes on. If it is, it must determine which node is the ending node of the high-order LSP (the other edge). Then it must extract from the ERO the subsequence of hops from itself to the edge of the network. Let us call this subsequence of hops S1.

An example is shown in Figure 3.17. Node 1, 2 and 3 are part of an optical network. The RSVP-TE Path message carrying the ERO arrives at Node 1. Node 1 checks the nodes in the ERO one by one. From the routing database, it finds out the first 3 nodes starting from the beginning of the ERO are in the same network.

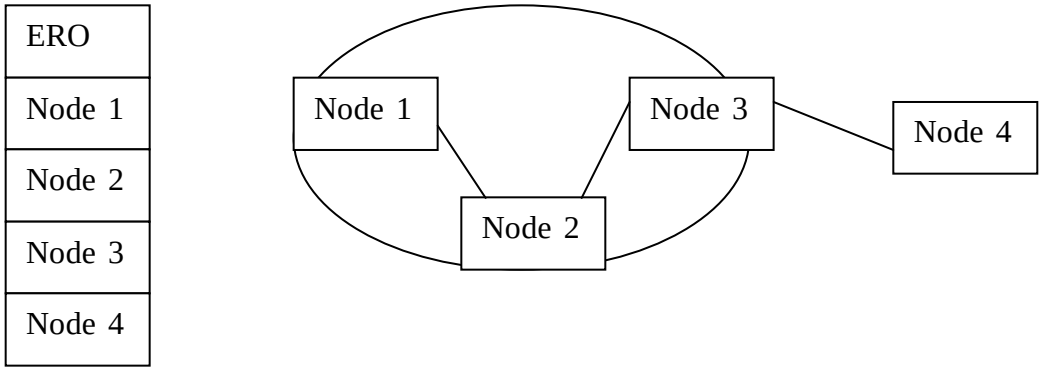


Figure 3.17: the ERO drives RSVP-TE to establish a hierarchical ER-LSP

Then the edge node checks the TE LSDB to see if there is an existing FA-LSP whose hops exactly match S1. If there is, it further checks if the properties of that FA-LSP can meet the requirements of the LSP being established, e.g., the unreserved bandwidth can satisfy the LSP being established. If so, the node replaces the hops S1 in the ERO with the end node of the FA-LSP. In the above example, Let us assume that there is a FA-LSP constructed by Node 1, 2 and 3. Node 1 replaces Node 1, 2 and 3 with Node 3 in the ERO. Then the destination address of the Path message is set to Node 3, and sent out by Node 1. We can see that the FA-LSP is treated as one link. After that, the TE properties of the FA-LSP are adjusted, e.g., the unreserved bandwidth is the difference between the previous unreserved bandwidth and the requirement of the LSP being established. They are advertised by the routing protocol in the current routing domain, e.g., by OSPF TE-LSA.

If there is no existing FA-LSP or the existing FA-LSPs do not satisfy the requirement of the LSP being established, then the edge node will signal a new high-order LSP, which will tunnel the low-order LSP. And it would be advertised as a FA-LSP.

The unreserved bandwidth of the FA-LSP is the difference between the maximum reservable bandwidth and what all the multiplexed low-order LSPs request.

The FA-LSP should be torn down if no tunneled LSP is there. There are a number of ways to trigger the FA-LSP tear-down. For example, if the maximum reservable bandwidth is as same as the unreserved bandwidth, then it means the FA-LSP is not being used, and it should be torn down.

4.3.1.1 Signaling Support for Fault Notification

The Notification mechanism in the signaling protocol RSVP-TE [26] is dedicated to support the fault notification in GMPLS recovery.

Fault notification is to notify the nodes of the failure in the path that are responsible for recovery. RSVP-TE defines a rapid fault notification mechanism to convey the information. The Notification mechanism includes the Notify Request object and the Notify message.

The Notify Request object contains the IP address of the node that should be notified of the failure, which is named *Notify Node Address*. This address can be configured, or automatically determined by the protection mechanism. For example, in the 1+1 protection mechanism, the LSP initiator node is responsible for switching the traffic to the backup LSP when the failure occurs, so the *Notify Node Address* should be that node. The LSP initiator node may be responsible for attaching this request object in the RSVP-TE Path message. The receiver of such a Path message (transit nodes) should also attach this object to the outgoing Path message. So the request is propagated. The terminator node of the LSP may respond with a Resv message which also carries the Notify Request object for a bi-directional LSP. So the notification may be required in both directions. A

node receiving the message records the *Notify Node Address* in the protocol soft state (for the RSVP soft state, see the RSVP introduction in the previous sections).

The Notify message provides a mechanism to inform non-adjacent nodes of LSP related events. It is different from the RSVP error message. The RSVP error message must be forwarded one by one along the nodes of the LSP, which is too slow for fault notification and not necessary. Notify message can be "targeted" to a particular node, e.g., the traffic-switch-over trigger node. By "targeted" it means the destination address of the IP packet carrying the Notify message is set to the IP address of the target node, which is specified by the *Notify Node Address* received from the Notify Request object. So it does not need to travel along the hops of the original LSP. Because after a failure in the network, the network topology likely has changed and there is another path that is optimal for the Notify message (see Figure 3.18). Nodes receiving a Notify message, which is not the destination of the message, must forward the message, unmodified, to the target.

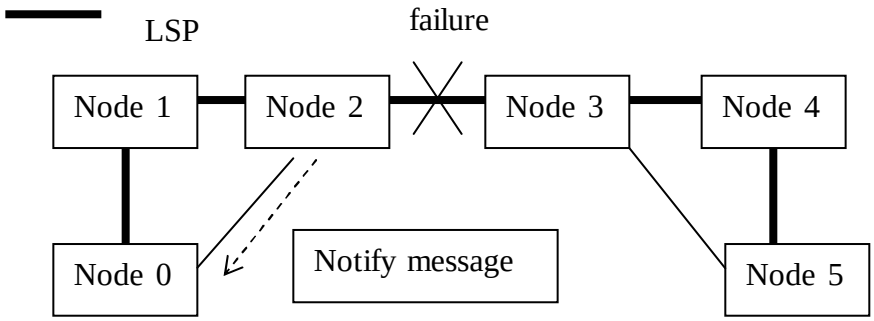


Figure 3.18: the Notify message is sent to the targeted node directly

The Notify message contains an ERROR_SPEC object, which specifies the IP address of the node that detects the failure or the link that has failed. Optionally it may carry other RSVP-TE objects that describe the LSP, e.g., the LSP SESSION object. Notify messages are normally generated only after a Notify Request object has been received.

It is not necessary for the local recovery to use such a notification mechanism. But other mechanisms need it, for example, the end-to-end LSP protection. Section 5 will specify which recovery mechanism needs it and when.

4.3.2 GMPLS signaling: CR-LDP with extensions

RSVP-TE, as a label distribution protocol, was built on the existing control protocol RSVP (RFC2205) [40]. Label Distribution Protocol (LDP) [41] was originally designed for label distribution.

LDP also uses the TLV structure to encode messages, which allows for future extensions. At first, LDP discovers its peers by multicasting an LDP Hello message onto the network. The nodes running LDP that receive the message are triggered to establish an LDP

session with each other. After the session is successfully created, they become LDP peers, and the session is maintained. Then the LDP peers can exchange label distribution messages. If there is any error during label distribution, the LDP Notification messages are used to provide error information, which could tear down the LDP session between LDP nodes. LDP uses TCP as the reliable transport mechanism to deliver all messages except the LDP Hello message, which uses UDP.

LDP has been extended to support Traffic Engineering, which is named *Constraint-Based LSP Setup using LDP* (CR-LDP) [42]. CR-LDP defines a new set of TLV structures to support explicit routed signaling, traffic parameters, LSP set-up/holding priority, etc. It also defines a means for resource reservation.

The constraint-based route TLV structure contains a sequence of IP prefixes or a sequence of Autonomous Systems. The contents of the constraint-based route TLV are computed by CSPF, which tells the routing mechanism where to forward the CR-LDP messages.

The LSP signaled by CR-LDP is initiated by the head node of the LSP. How it works is illustrated as below.

Router 1 is going to establish an explicit-routed LSP (R1, R2, R3, R4, R5). R1 constructs the constraint-based route TLV to have the sequence of nodes (R1, R2, R3, R4, R5). Each node can be represented by an IP address. Then R1 sends out the CR-LDP Label Request message carrying the constraint-based route TLV. Before the message is sent, R1 checks the top of the TLV, and it finds out that the next hop is R2. R1 removes itself from the TLV and sends the message to R2. The Label Request message may carry the Traffic Parameter TLV, which specifies the traffic parameters to be sent. If so, R1 reserves the resource before the message is sent out. R2 receiving the message also checks the top of the TLV, and it finds out R3 is the next hop. R2 removes the address of R2 from the TLV and sends out the Label Request message. It may also reserves the resource for the LSP if the Traffic Parameter TLV is carried. R3, R4, and R5 follow the same algorithm as R2 does. When R5 receives the message, the TLV only has one address, which is R5 itself. Along the message path, the LDP protocol state should be created.

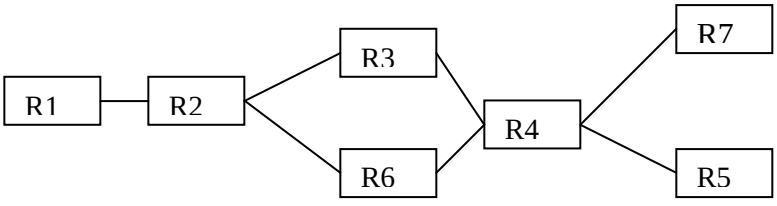


Figure 3.3.2.1: ER-LSP from R1 to R5

R5, as the ending node of the LSP, programs the label forwarding table, reserves the resource if needed, and responds with a CR-LDP Label Mapping message, which carries a

Label TLV. The Label TLV contains the label that the downstream node wants the upstream node to use. The protocol state on the node tells R5 to send the Label Mapping message to R4. R4, R3 and R2 do the same thing as R5 does. R1, as the head node of the LSP, does not need to allocate label any more, but simply receives the label and programs the label forwarding table.

The head node of an LSP transmits a Label Release message to a peer when it is no longer needs a label previously received from that peer. This takes place when the LSP is torn down (see the following figure).

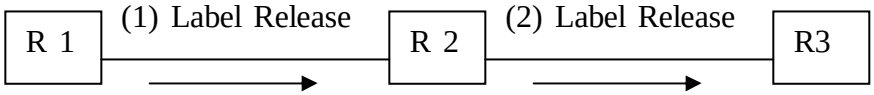


Figure 3.19: LSP (R1, R2, R3) is torn down by Label Release message

Unlike the RSVP-TE, CR-LDP is not a soft state protocol. By this it means the LSP created by CR-LDP does not need the signaling refresh periodically. The LSP must be torn down explicitly.

CR-LDP is also being extended to support GMPLS [43]. The information that is needed to support generalized label in RSVP-TE is also needed for CR-LDP. For example, the label format in the Label TLV is also generalized to support different types of "label", e.g., the port number, wavelength, etc.

CR-LDP is also required to support bi-directional LSP set up. The idea is to add the Upstream Label TLV in the Label Request message (see Figure 3.20).

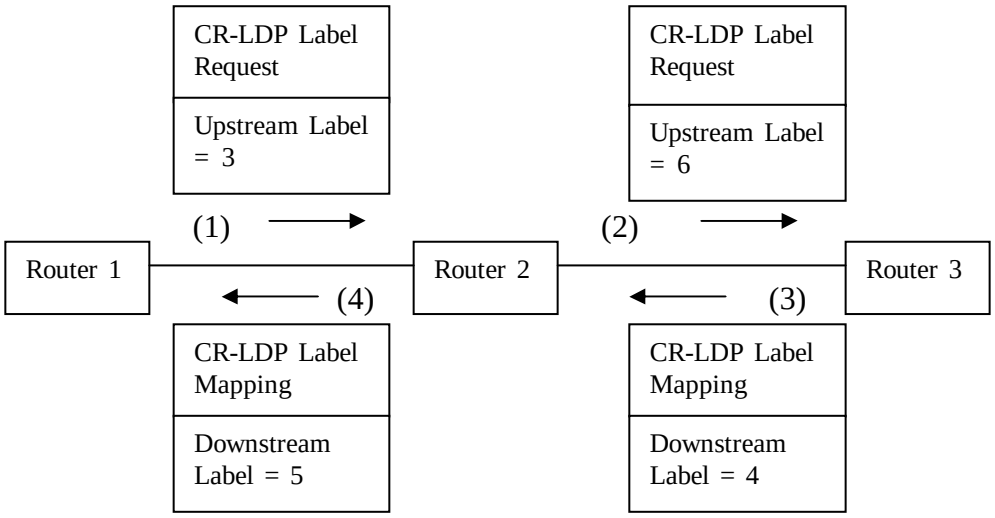


Figure 3.20: CR-LDP signals a bi-directional LSP

To support explicitly routed LSP in the context of GMPLS, just the IP address or the identifier of an autonomous system may not be adequate. RSVP-TE defines the Label object as a sub-object in the ERO, and CR-LDP defines the Explicit Label Control TLV as a sub-TLV following the IP address or the Autonomous System ID in the constraint-based route TLV.

In order to improve network survivability, the protection information is considered in GMPLS signaling. Like RSVP-TE, CR-LDP defines the Protection TLV, which includes:
(1) link protection type;
(2) indication of whether the path is primary or backup.

Can *CR-LDP with extensions* [25] do whatever *RSVP-TE with extensions* [26] can do so as to support GMPLS signaling? No, as this report is being written. The RSVP-TE [26] has got the fault notification mechanism (see Section 4.3.1.1) to notify a responsible node when a link/node failure occurs. But CR-LDP [25] does not have a similar mechanism yet. CR-LDP [25] has its own Notification message, but it does not provide the same function as the one does in RSVP-TE.

From now on in this report, RSVP-TE is used to illustrate the GMPLS signaling support for LSP protection/restoration.

4.4 The Hello Protocol

In fact, there is no protocol called Hello. OSPF, RSVP-TE, LMP and other protocols define a software method to detect failures, which is the Hello messaging. The idea of the Hello messaging is simple. Two nodes exchange a short message named Hello periodically. The interval can be configured, e.g., the recommended interval for OSPF Hello is 5 ms (see RFC2328). If a number of messages are missed, e.g., 4, then the node can determine that the other node is down or the link between the two is broken.

Although many protocols provide this method to detect failures, using software to detect a failure is very slow and usually does not meet real-time application requirements. Furthermore, it is difficult for the software detection to deal with the shaking problem. A node does not receive OSPF Hello messages from its neighbor for several times, and it determines its neighbor is down. But just after that it can receive Hello again due to the unstable situation in the network. The problem keeps repeating like that for a while, which is called *shaking*.

However, the software fault detection is still useful in some situations. An example is the Ethernet, where nodes are connected by a bus (multiple access media). A node can detect its peer’s fault by the Hello messaging.

5. The Recovery Mechanism in GMPLS

There are two recovery mechanisms: protection and restoration.

Protection: A dedicated protection path is established for a connection, and the connection switches from the working (primary) path to the protection (backup) path when a failure occurs on the primary path.

Restoration: The establishment of a backup path does not occur until a failure occurs in the primary path. Then the traffic is switched to the backup path. Such a mechanism is called restoration. But the backup path can be selected (calculated) in advance.

Restoration and protection are different mechanisms. They operate on different time scales; protection requires redundancy of resources, while restoration relies on dynamic resource reservation - hence restoration takes more time [44].

Protection/Restoration can be classified into the following categories according to the recovery scope (see [12] and [45]):

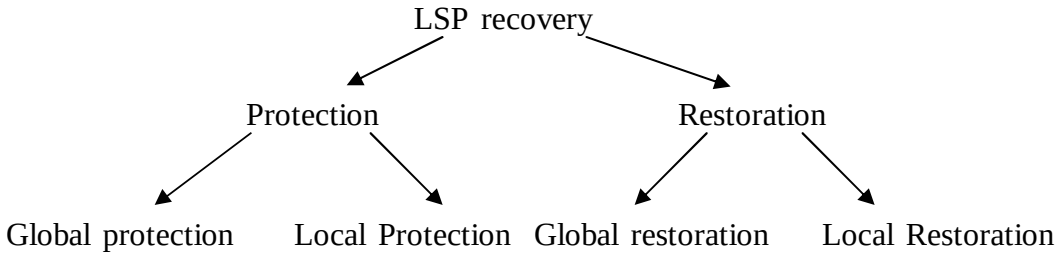


Figure 4.1: LSP recovery

The objective of local recovery is to protect against a link or neighbor node fault and to minimize the amount of time required for fault notification. The local recovery includes link recovery and node recovery. Local recovery is initiated by the immediate upstream node of the faulty link or node, which may be an transit node or the source node of an LSP.

The objective of global recovery is to protect against any link or node fault on an LSP or on a segment of an LSP except for the source or the destination node. Global recovery is also called end-to-end path recovery, because only the source or destination node initiates the recovery process.

5.1 Protection Mechanisms

The protection mechanisms are described in the following. The ideas can be applied on paths as well as links. These mechanisms can be used in any network that may have different switching technologies at any level of the GMPLS hierarchy, for instance, ATM networks, IP networks, optical (e.g., OXCs) network, etc.

• 1+1 protection

Two disjoint paths have been established and both of them have resources allocated. By "disjoint", we mean none of the links or nodes constructing these paths are overlapped (except the starting node and the terminating node of these paths). The same user data is transmitted simultaneously over the two paths, and the receiver can pick the best signal from one of these two paths. An example can be seen in Figure 4.2. In the example, Path 1 and 2 provide a 1+1 protection for the data transport from Node 1 to Node 5. If Path 1 is broken, for instance, then the receiver at Node 5 can pick the signals from Path 2.

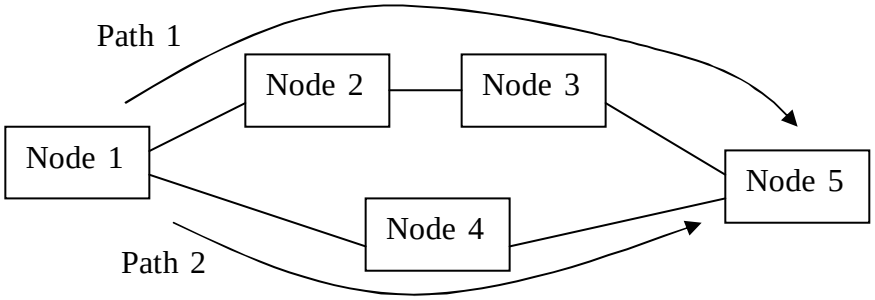


Figure 4.2: 1+1 path protection

The benefit of 1+1 protection is short recovery time and the lost data can be very small. But it requires two pre-established paths, double resources and the traffic is copied and sent over both paths. It is expensive.

• M : N protection

There are M pre-established backup paths that protect N primary paths. But user traffic is not transported by any of the backup path until a failure occurs. When one of the primary paths fails, the nodes connecting to the faulty link or the faulty node notify the end-nodes of the path (source and destination nodes). Then the end-nodes allocate the resource required by the traffic traveling that primary path on one of the backup paths. In the end, the traffic is switched over. Note that the backup paths can protect any of the primary paths. An example can be seen in Figure 4.3. In the example, 2 backup paths (Path 1 and 2) are protecting 2 primary paths (Path 3 and 4). If Path 4 is broken (for instance, the link between Node 7 and 8 is broken), then Node 7 notifies Node 1 to do the protection switch (and maybe Node 8 notifies Node 5 as well - depending on how the signaling protocol works). Node 1 allocates the resource on Path 1, which is required by the traffic traveling on Path 4, and switches the traffic from Path 4 onto Path 1.

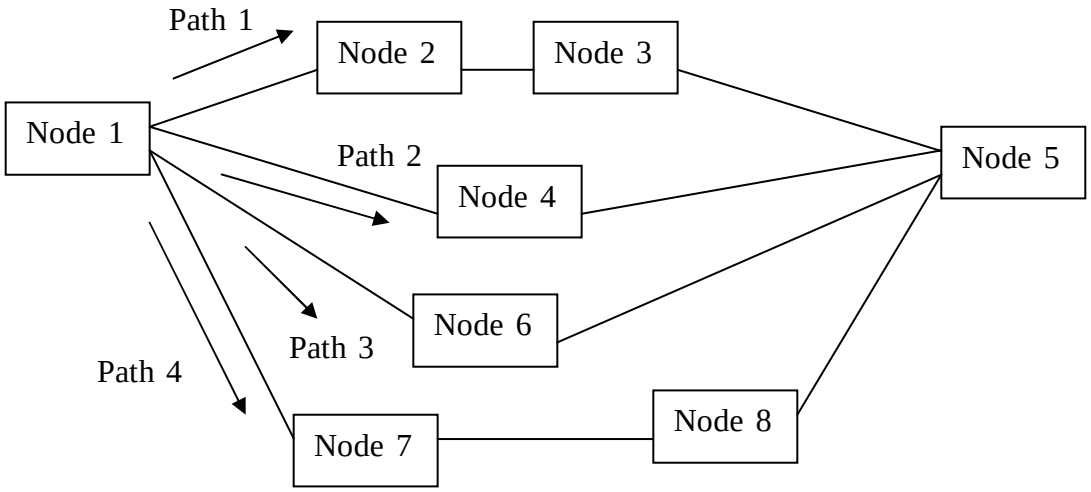


Figure 4.3: 2:2 path protection

It is not recommended that the links constructing different primary paths belong to the same Shared Risk Link Group. For example, both primary path (L1, L2, L3) and (L4, L2, L5) have the same link L2, and they would better not share the same backup path. Otherwise, if L2 goes down, it is possible that one of the primary paths would not have any protection.

- 1 : N protection

It is a special case of M : N protection - only one pre-established backup path provides protection for N primary paths. Let us look at Figure 4.3 again, and assume that Path 1 is protecting other paths. If the link between Node 7 and Node 8 is broken, Node 7 notifies Node 1 (and maybe Node 8 notifies Node 5 as well) to do the protection switch. Then Node 1 allocates resource required on Path 1, and switches the traffic onto Path 1.

If there are two primary paths that are broken simultaneously, then a policy is used to decide which one will get protected. One of the policies is taking priorities. For example, Path 3 and 4 are broken, if the traffic traveling on Path 3 is deemed to have higher priority than the traffic traveling on Path 4, then Path 3 will get protected by Path 1. Another simple policy can be First-Come-First-Service.

The links constructing the primary path should not belong to the same Shared Risk Link Group. Otherwise, if the link constructing both of the paths is cut, then one of the primary paths does not have any protection.

- 1 : 1 protection

It is also a special case of M : N protection - one dedicated backup path is pre-established for one primary path. For optimization, the resource may be pre-allocated if it is known in advance. But the user traffic is not inserted onto the backup path. So the resource pre-allocated on the backup path may be used by other LSPs that have lower

priorities. When the primary path fails, the signaling protocol notifies the end-nodes of the primary path. Then the traffic is switched over from the primary path and the LSPs that are using the resource of the backup path are preempted.

Summary of Protection Mechanisms
When a failure occurs, the nodes involved in the recovery need not notify the end-nodes of the route (path) in the 1+1 protection mechanism; but in the M:N, 1:1 and 1:N protection mechanisms, the nodes neighboring the failure must notify the end nodes so that the end-nodes will switch the traffic. So the 1+1 protection mechanism provides fast recovery because it does not need fault notification time. However, the other mechanisms utilize the resources more efficiently.

5.1.1 Local Protection
Local protection includes link protection and node protection.

5.1.1.1 Link Protection
Link protection switches the traffic from the primary link to a backup link between the same nodes when link failure occurs. It occurs between two adjacent nodes and only these two nodes are involved in the whole process.

As we specified in the sub-section *Enhancements in MPLS Signaling to Support GMPLS* (see Section 2.4.3), the requested link protection type is carried by the signaling protocol when an LSP is set up. The node that receives such a request must check the outgoing interface to see whether the request can be satisfied. If the link protection request is not satisfied, then the signaling for the LSP establishment cannot continue.

For RSVP-TE signaling, the Path message carries the link protection type for the LSP. The protection object of RSVP-TE signals the link protection type and the role of the LSP (see Figure 4.4). The S bit signals the role of the LSP being established and the "link flags" signal which link protection type is desired. If bit S is set, it means the LSP is a secondary (backup) one; otherwise, it is a primary LSP. The link protection flag contains one of the codes specified in Section 4.1.3.3.

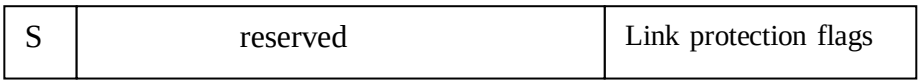


Figure 4.4: the protection object in RSVP-TE

An example is shown in Figure 4.5. In the example, the Path message carries a protection object to establish an LSP. The protection object signals the link protection type is "Dedicated 1+1" and the LSP being established is a primary one. The Path message arrives at the node. The node must check if there is a link connecting the next hop, which has link protection capability "Dedicated 1+1". Because the link protection type is distributed by the routing protocol, for example, the node checks the link state database maintained by the routing protocol. According to the definition of protection type

Dedicated 1+1, we know that the protecting link must not be in the same Shared Risk Link Group (SRLG) as the primary link. If there is such a link, signaling continues.

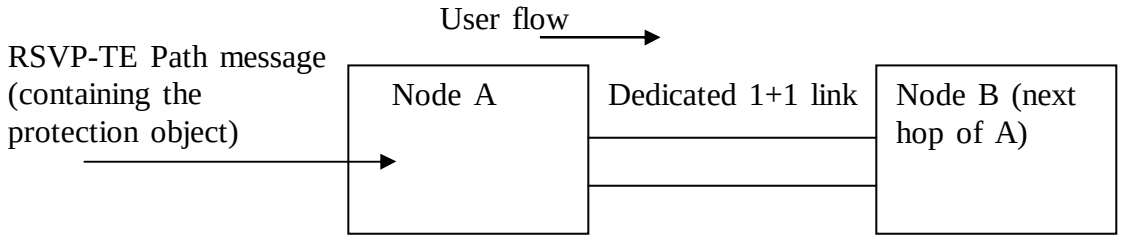


Figure 4.5: the link protection type must be honored to continue signaling

When the RSVP-TE Resv message arrives at Node A, it reserves the resource (e.g., bandwidth) on both of the links. After the LSP is created, the node (in this example, Node A) will copy the traffic and insert it into both links. The receiver selects the healthy traffic from any of the links. For example (see Figure 4.6), after initialization, the receiver takes the traffic from the primary link. When the primary link fails, LMP Fault Management (see the sub-section about LMP) is used to localize the failure. For example, all the nodes following Node B can detect loss of light if the nodes are in the optical network. LMP tells Node A and B that the faulty link is between them. Node B simply selects the traffic from the backup link.

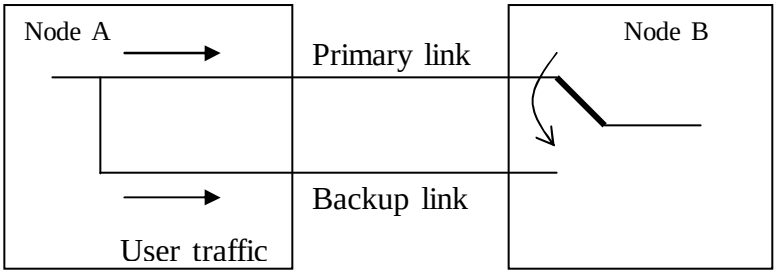


Figure 4.6: Dedicated 1+1 link protection

If the link protection type is shared, e.g., 1:N (or 1:1), then the Resv message also reserves the resource on the backup links. But the backup links will not transport traffic. And the resource reserved for the backup links can be used by other LSPs that have lower priorities. The reason is that these lower-priority LSPs will be preempted when the primary links fail, and the traffic is switched over.

With link layer protection, the LSP may stay there even though there is a link failure, and LSP recovery mechanism is even unaware of the problem. The failure will be reported by alarms signaled by the network management in the nodes connecting the faulty link. In the example, the alarms will be displayed on Node A and B.

Summary of Link Protection
Because the point that initiates the recovery is close to the failure, there is no need to have fault notification - it provides fast recovery. Only the nodes connecting the faulty link are involved in the recovery. And it does not require any changes in the GMPLS LSP.

But the protection ability is limited. If the entire LSP requires link layer protection, it is expensive and the control becomes a big overhead, because every node along the whole LSP needs to keep monitoring links.

Therefore, usually link layer protection is only used in an area that is deemed to be unreliable.

5.1.1.2 Node Protection
In fact, there is no protection mechanism in the GMPLS LSP level that is dedicated to locally provide single node failure protection. If some nodes in a network are deemed to be unreliable, then the path computation should compute a path that will work around those nodes. On the other hand, global path protection and restoration can recover the traffic affected by node failure. These mechanisms will be introduced later.

5.1.2 Global Protection
From the previous sub-section, we can see that link layer protection provides the link protection under the GMPLS LSP layer.

With the end-to-end path protection (global protection), multiple disjoint hierarchical LSPs are pre-computed and established between the initiator and the terminator nodes of a client LSP. Dedicated resources are allocated for these LSPs. So the nodes and links of the entire primary hierarchical LSP are protected except for the initiator and terminator nodes. In order to avoid the contention of multiple layer protection mechanisms, the LSPs may require “unprotected” Link Protection Type during signaling. Thus the protection is only built on the MPLS-based layer and contention will not occur. When a failure occurs, the nodes that detect the failure notify the end nodes (the initiator and terminator nodes). The end nodes initiate switching the traffic to the alternate path.

The illustration is shown in Figure 4.7. The logical view of the 1:1 LSP protection is shown in the figure. The LSP (Node 1, Node 3, Node 5, Node 7) is the primary one; LSP (Node1, Node2, Node4, Node6, Node8, Node7) is the backup. Both may be hierarchical LSPs, e.g., the link (Node3, Node5) is a FA-LSP (TE link), so is link (Node4, Node6). Traffic is sent along the primary LSP. If a failure occurs, the nodes that detect the failure notify the end nodes: Node 1 and 7, then the end nodes will switch the traffic to the backup LSP.

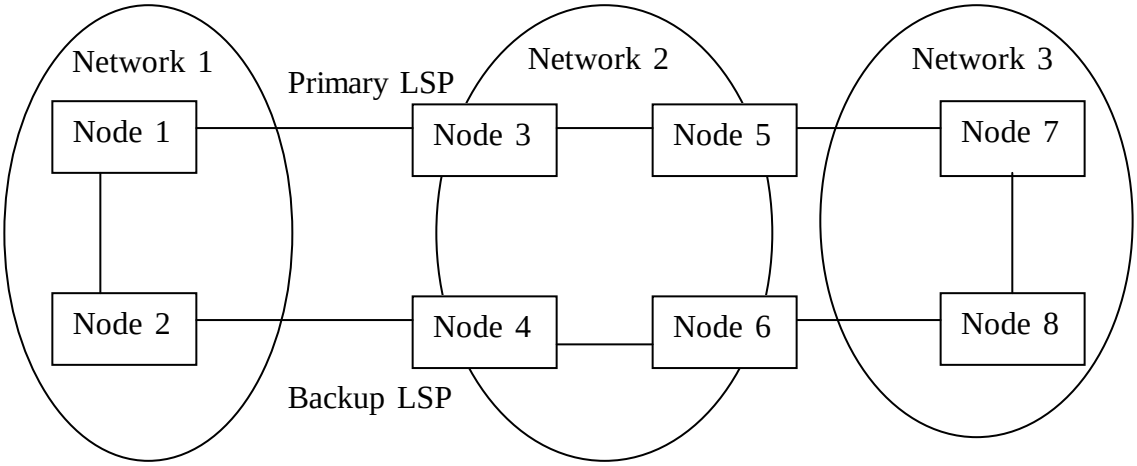


Figure 4.7: The logical view of the 1:1 LSP protection.

Summary of Global Protection

Global protection can provide a fast protection mechanism against any link or node fault on an LSP with the exception of the failure occurring at the initiator and terminating node (end nodes) of an LSP. Usually, the end nodes are far away from the failure, and need to be notified by the node that detects the failure, which takes time. Also, it is expensive because the backup path is pre-computed and pre-established. The resource is pre-allocated as well, but it may be used by low priority traffics.

5.2 Restoration Mechanisms

Restoration is implemented by rerouting. Some papers even use the term rerouting [31]. Rerouting is referred to as establishing new paths (global restoration) or path segments (local repair) on demand for restoring traffic after a failure occurs.

Rerouting follows the “make-before-break” principle. The “make-before-break” means the original path is used while the new path is set up, then the node performing the reroute switches the traffic to the new path and the original path is torn down.

5.2.1 Local Restoration

Local restoration includes link restoration and node restoration. When a link failure occurs between two adjacent nodes, with link restoration, the upstream node switches the traffic on an alternate route in which there are additional intermediate nodes. In the case of node failure, the immediate upstream node of the faulty node initiates an alternate route, which bypasses the faulty node. Then traffic is switched over to the alternate route. Such rerouting also provides the local restoration for node failure.

Upon detecting a failure, paths or path segments to bypass the failure are established using signaling. The idea is shown in Figure 4.8. Assuming that the path is (Node 0, Node 1, Node 2, Node 3, Node 4). If Node 2 is down, Node 1 creates a path segment which bypasses the faulty node – (Node 1, Node 5, Node 3). The new path segment goes

through another interface of Node 1 and arrives at Node 3 through another interface. For example, using RSVP-TE, because the message carries an identification (e.g., the Session object and the Sender Template object in RSVP-TE) for each LSP, the Path message can re-create the protocol state in Node 3 and re-program the label forwarding table. The original path segment will be torn down eventually.

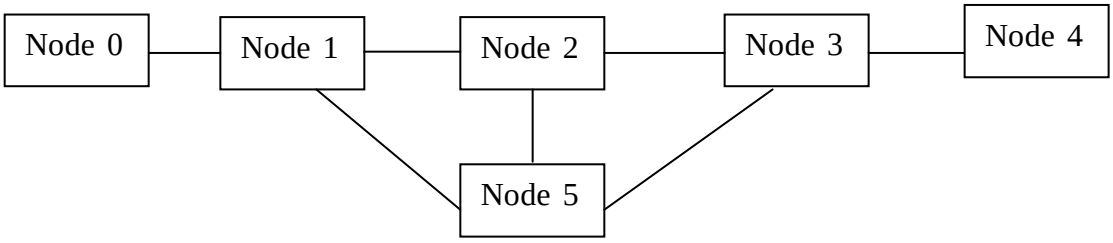


Figure 4.8: reroute

According to the position of the faulty node in the LSP, there are three cases.
Case 1: The failure does not occur at the end node of the hierarchical LSP.
In this case, there is no difference between link and node restoration from the rerouting point of view. An example is shown in Figure 4.9. In this example, if OXC 3 fails or the link between OXC 3 and OXC 4 fails, Case 1 occurs.

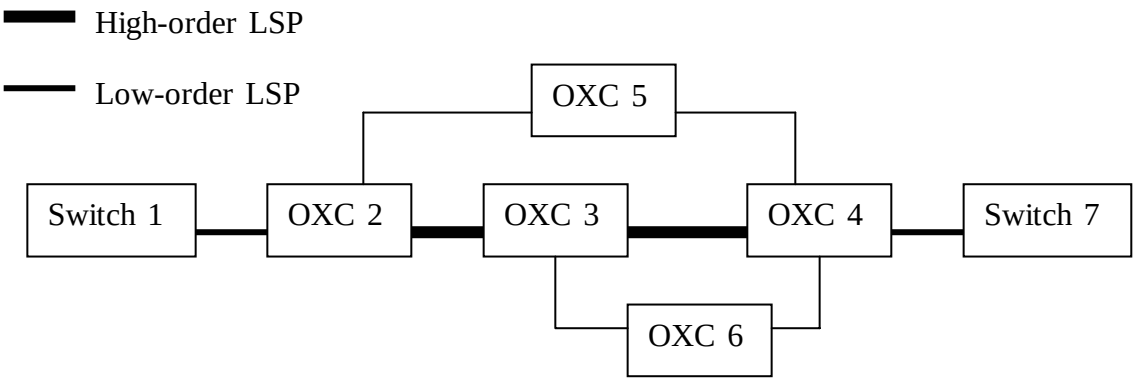


Figure 4.9: a situation where reroute Case 1 applies

The recovery steps are:
(1) The failure detection mechanism detects the failure.
(2) The fault localization mechanism localizes the failure. Meanwhile, the node that is the immediate upstream node of the failure knows about the failure.
(3) The node initiates the process to establish a new path or path segment that bypasses the failure.
(4) And the node switches the traffic to the alternate path.

As in all the recovery mechanisms, the failure detection usually is done by hardware at the physical layer (or link layer). After that, the fault localization mechanism is triggered,

which can find out where the failure is. For example, LMP fault management (see the Section 4.2) can localize a link failure. The fault localization mechanism does not stop running until the node that is the immediate upstream node of the failure finds out the failure. So there is no need to have explicit fault notification. For a faulty node, the immediate upstream node of the faulty node detects the problem. Then signaling is used to create a reroute.

Using RSVP-TE, the reroute initiator node sends out the Path refresh message, which will consult the routing component for a feasible route. In the example in Figure 4.9, if OXC 3 is down, OXC 2 detects the failure, e.g., by OSPF Hello messaging or other means, and sends out the Path refresh message, which can travel to OXC 5 to get to OXC 4. OXC 4 responds with a Resv refresh message, and the LSP between OXC 2 and OXC 4 is fixed. If the link between OXC 3 and OXC 4 is broken, LMP fault management can localize the failure. OXC 3 sends out the Path refresh message, which can travel to OXC 6 to get to OXC 4 and repairs the FA-LSP.

Case 2: The terminator node of a FA-LSP fails. An example is shown in Figure 4.10. In this example, if OXC 4 fails, Case 2 occurs.

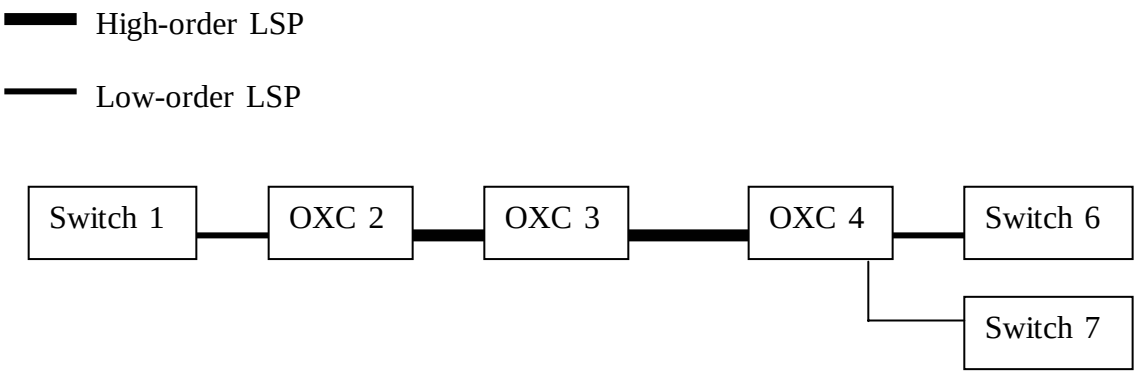


Figure 4.10: the situation where reroute Case 2 applies

Note that the high-order LSP between OXC 2 and 4 is an FA-LSP, which tunnels low-order LSPs. As the OXC 4 is the terminator node of the FA-LSP, it is impossible to rebuild this FA-LSP so that it meets the requirements of the tunneled LSPs. For example, the tunneled low-order LSPs go to multiple different destinations after the FA-LSP, like, Switch 6, Switch 7, etc. The FA-LSP does not know the information, so there is no sense for OXC 2 to reroute FA-LSP and there is no need to repair the FA-LSP.

However, local recovery can still be useful. The OXC 2 is the node that tunnels (aggregates) a number of low-order LSPs. If the FA-LSP is broken, the OXC 2 can trigger all the tunneled LSPs to reroute individually. For example, a low-order LSP, which was tunneled by the FA-LSP at OXC 2, can re-establish a path segment that bypasses the failure and reaches the desired destination, e.g., Switch 6. Let us see how it works.

When OXC 4 fails, it is as if the "link" *FA-LSP* failed. Because OXC 2 and OXC 4 have the "Forwarding Adjacency" (FA), OXC 2 should be notified according to the link restoration mechanism. The Notification mechanism of RSVP-TE is useful here.

OXC 2, as the initiator node of the FA-LSP, can attach the Notify Request object to the Path message when the FA-LSP is established, and the target node address in the object is OXC 2 itself. When OXC 4 fails, fault detection, e.g., OSPF Hello messaging, enables OXC 3 to detect the neighbor failure. Then OXC 3 notifies OXC 2.

Another way is by administration. During signaling, OXC 3 knows that it is the penultimate node of the FA-LSP, e.g., routing tells OXC 3 that it is directly connected to OXC 4. Let us assume that we have such an administration policy that the penultimate node of the FA-LSP must notify the initiator node of the LSP. The Notify message destination address can be configured. In the example, OXC 3 can send out the RSVP-TE Notify message targeted to the initiator node - OXC 2 in this example.

After the initiator node of the FA-LSP is notified, it tells all the tunneled low-order LSPs to re-establish the LSP segment (e.g., maybe another hierarchical LSP) that bypasses the fault.

Case 3: The initiator node of a FA-LSP fails.
If the initiator node of a client LSP fails, then there is no general LSP protection/restoration mechanism.

If the initiator node of a FA-LSP fails, then the immediate upstream node of the faulty node will re-establish a new LSP segment that bypasses the failure. An example is shown in Figure 4.11.

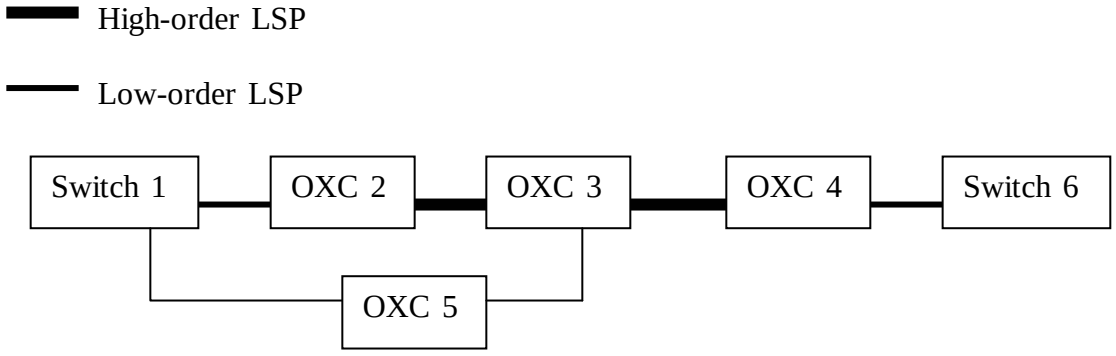


Figure 4.11: the situation where reroute Case 3 applies

In the example, a low-order LSP goes through (Switch 1, OXC 2, OXC 3, OXC 4, Switch 6). And the high-order LSP (FA-LSP) goes through (OXC 2, OXC 3, OXC 4). If OXC 2 is down, Switch 1 detects the neighbor failure, and it will initiate the reroute. It may trigger establishing another high-order LSP (FA-LSP) that bypasses OXC 2, e.g.,

FA-LSP (OXC5, OXC3, OXC4). And the low-order LSP is tunneled by the new FA-LSP.

The problem of multiple layer protection contention can also occur when using local restoration. For example, the link between two adjacent nodes is broken. If there is link layer protection there, e.g., that link is *Dedicated 1+1* protected link, it can provide faster recovery and the reroute should not be needed. So a coordination mechanism should be needed, e.g., the hold-off timer.

In the above cases, how does the reroute initiator node find the next hop to send out the signaling message so as to create the reroute path segment? The conventional RSVP [40] must consult the routing table. It expects that the changed topology has been shown in the routing table. But this does not happen right away after the fault in the network. So the conventional reroute to locally repair the link/node failure suffers packet loss. Let us see what is the problem.

When a link/node failure occurs in a network, routing protocols exchange the routing messages in the network to reflect a new topology. The routing information on different nodes may be temporarily inconsistent. And even a forwarding loop could be created. The situation causes packet loss. The longer the inconsistency lasts, the more packets are likely to be lost. The time consists of three periods: (1) the time the node needs to detect the failure, (2) the time a node needs to distribute the information across the network and (3) the time to reconstruct the routing table. Among these factors, period (2) is the major one (see [46]). To reduce the time it takes to detect link failure, we can use mechanisms in the link layer, e.g., in SONET, it is possible to detect link failure in less than 10 ms by SONET-specific mechanisms, such as Loss of Frame detection. But with regard to (2), the distribution nature of IP routing and the need for all the nodes to converge to consistent routing place limitations on how much (2) can be reduced. In practice, the time to converge within a single routing domain may be on the order of seconds. That means the packet loss may last on the order of seconds. Let us have an example (see Figure 4.12). Let us assume that the link between Router 1 and 2 is broken. Router 1 detects the failure. With the current routing table, Router 1 can find out that there is an alternative path (R1, R3, R4, R5, R2). Then, to create the reroute path segment, Router 1 sends out the RSVP-TE Path refresh message destined for Router 2 to Router 3. But due to the routing information distribution delay (2) mentioned above, Router 3 things R1 should be the next hop because it only takes 2 hops (R1 and R2). So Router 3 forwards the message back to Router 1. Thus the forwarding loop is created for a short period of time. The message is discarded. Packets would be lost for seconds and signaling fails until the next refresh time.

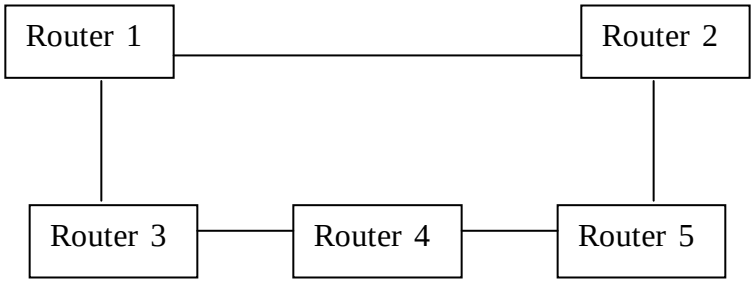


Figure 4.12: local reroute suffers packet loss for seconds

The conventional RSVP [40] suggests the signaling protocol wait a period of time, named W, before consulting the routing table to signal the bypass route. The recommended default value for W is 2 seconds. But this delay is not acceptable for many applications.

Yakov Rekhter and Bruce Davie in their book [47] suggest using an explicit-routed LSP as the reroute LSP segment to bypass the failure. Instead of using hop-by-hop, destination-based forwarding, the immediate upstream node of the faulty link/node constructs an explicit-routed LSP that bypasses the fault. The explicit-routed LSP merge with the original LSP at the node that is the immediate down stream node of the fault. Such an LSP uses the label stacking capability of MPLS to tunnel all the LSPs that used to going through the faulty link/node. And the rest of the original LSP does not need to be torn down or modified. Let us have an example to illustrate the idea.

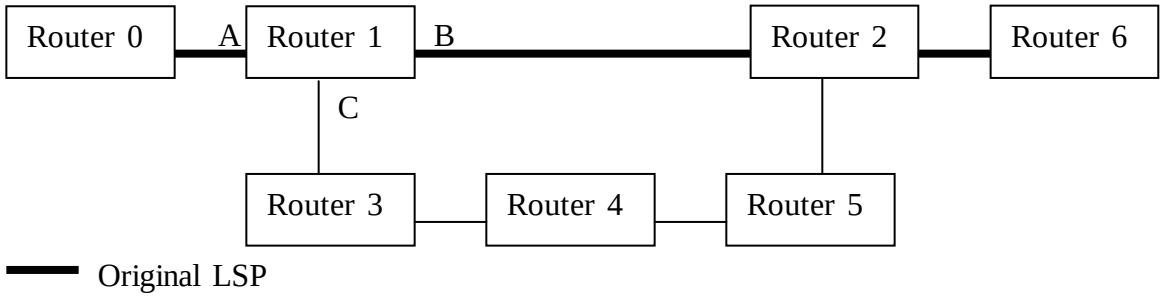


Figure 4.13: explicit-routed LSP bypasses the fault

In Figure 4.13, an LSP from Router 0 and Router 6 that is routed over Router 1 and 2. In the label forwarding table of Router 1 for that LSP, incoming label 10 and interface A corresponds to outgoing label 11 and outgoing interface B. It means, the packets from Router 0 with (incoming) label 10 through interface A will be replaced with 11 and forwarded to Router 2 through interface B. When Router 2 receives any packet with label 11 from Router 1, it will continue label forwarding, e.g., it forwards the packet to Router 6.

Let us assume that the link between Router 1 and Router 2 is broken. Router 1 detects the link failure, and it can construct an explicit-routed LSP right away, which is (Route 1, Router 3, Router 4, Router 5, Router 2), because its routing table tells it that there is such a route from Router 1 to Router 2. The topology change does not have an effect on constructing such an explicit-routed LSP. Now how to tunnel the original LSP? The Path message carries the ERO containing (Router 1, Router 3, Router 4, Router 5, Router 2), which specifies the explicit-routed LSP. The ERO drives Path message from Router 1, Router 3, Router 4, Router 5, and finally to Router 2. Router 2 responds with a Resv message, which allocates label 20 to Router 5. Similarly, Router 5 allocates label 21 to Router 4, Router 4 allocates label 22 to Router 3, and Router 3 allocates label 23 to Router 1. Router 1, receiving the label, re-programs the label forwarding table. First, it adds one more operation to the label forwarding process, which is to push label 23 on the packet that is from Router 0, and this operation is after replacing label 10 with 11 on the packet. Second, the outgoing interface is not B any more, but C. Router 2, as the ending node, may support penultimate hop popping.

Now, assuming that the packet with label 10 arrives at Router 1. Router 1 replaces label 10 with label 11 (as it did before the link failure), furthermore it pushes 23 on top of label 11. And it forwards the packet to interface C. Router 3 forwards the packet to Router 4 by replacing label 23 with 22. Similarly, Router 4 forwards the packet to Router 5 by replacing label 22 with 21. Router 5 forwards the packet to Router 2 by replacing label 21 with 20. When the packet arrives at Router 2, label 20 is striped off, and the label 11 becomes the top label. As before, Router 2 understands how to forward packets with label 11. The label allocation can be seen in Figure 4.14.

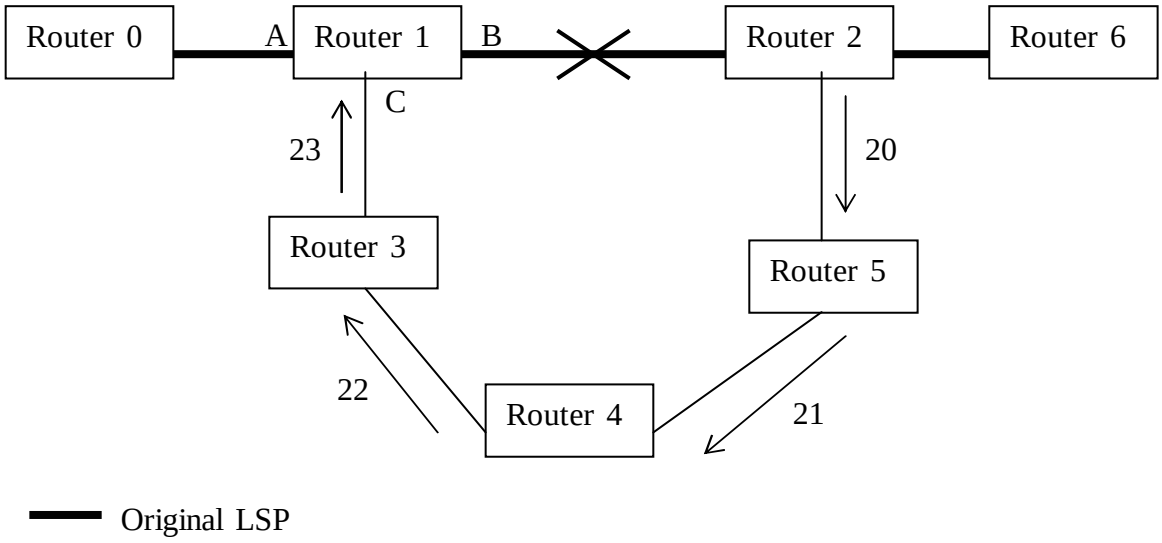


Figure 4.14: explicit-routed LSP bypasses the fault

The advantage of this solution is that the immediate upstream node of the fault does not need to wait for the routing information distribution or routing database synchronization. And it does not need to wait for reconstructing the routing table because the routing

database can still tell the reroute initiator node if there is another route to repair the path even after the link/node failure. Note that the reroute initiator node is the immediate upstream node of the failure. Thus, the waiting period of time (2) and (3) suffered by conventional IP routing can be eliminated. So this solution, which uses explicit-routed LSP to reroute, is faster.

From the above analysis, we can also see that this solution does not need to change the current signaling protocols, but it requires the nodes implement the intelligence to support this solution.

Summary of Local Rerouting
Local restoration eliminates the need to propagate fault information across networks. But its application is limited.

As specified in Section 4.3.1 of this report, an explicitly routed LSP (ER-LSP) is pre-computed, which usually meets some traffic engineering goals. If a user's LSP is an ER-LSP, it is highly desired not to be rerouted. For example, the user's ER-LSP can route away from network congestion and bottlenecks. The local restoration mechanism works for the hop-by-hop routed LSP recovery very well, and it also works for the loosely specified portion of an ER-LSP, but not for a strictly routed ER-LSP. The local reroute mechanism is dynamic – it repairs the LSP by bypassing the failure after the failure occurs, and the new LSP travels some nodes/links that may not be desired. Such an LSP may not be optimal. Therefore, if we use local reroute mechanism for a user's ER-LSP, then after the local repair for a strict ER-LSP or the strictly specified portion of a loose ER-LSP, the initiator node of the LSP must be notified. And it should re-compute the LSP, and establish a new ER-LSP to meet the original requirements.

Using conventional local reroute takes a lot of time to wait for the routing information synchronization, and the local reroute using Yakov Rekhter and Bruce Davie's proposal (see [47]) provides a solution to solve the problem. But the signaling for creating the reroute path still takes time.

Because of the network topology, local repair may not succeed.

5.2.2 Global Restoration
With global (end-to-end) path restoration, the backup path is not established until the failure on the path occurs. After the failure is detected, the initiator node of the faulty LSP is notified of the failure. And it establishes the alternate path destined for the destination node and switches the traffic to the new path.

When a failure occurs, the fault detection triggers the fault localization mechanism. After the location of the fault is found, the node that is closest to the failure distributes the fault information by the routing protocol. In the meantime, it notifies the node that initiates the LSP establishment.

The faulty LSP should be torn down and the resource allocated for the faulty LSP should be freed. The information is also distributed by the routing protocol.

The LSP initiator node should wait for the routing information synchronization. After that, it re-establishes another LSP that bypasses the failure and the traffic is switched over onto the new LSP.

Summary of Global Restoration
Compared to end-to-end path protection, the end-to-end path restoration is slow because the fault notification and the routing information synchronization would take seconds. So it does not work for real-time applications such as voice. It is resource efficient, because the alternative LSP is established on demand and the resource is allocated on demand.

In order to eliminate the time for routing information synchronization, Yakov Rekhter and Bruce Davie's proposal (see [47]) can be also used for end-to-end path restoration.

6. Case Studies

6.1 Case Study 1: The end-to-end LSP Protection

The network topology is shown in Figure 5.1. The switches in the client networks are SONET switches and the OXCs in the optical core network operate on a single lambda level. Let us assume that the edge OXCs have the capability to convert electrical signals to optical signals. They have interfaces that can provide SONET signals and interfaces that can provide WDM capability. There are two OC192 links that connect edge nodes, e.g., SONET switch S3 and OXC O1. The optical fiber between two OXCs can contain 16 lambdas, each of which can provide capacity equivalent to one OC192 capacity.

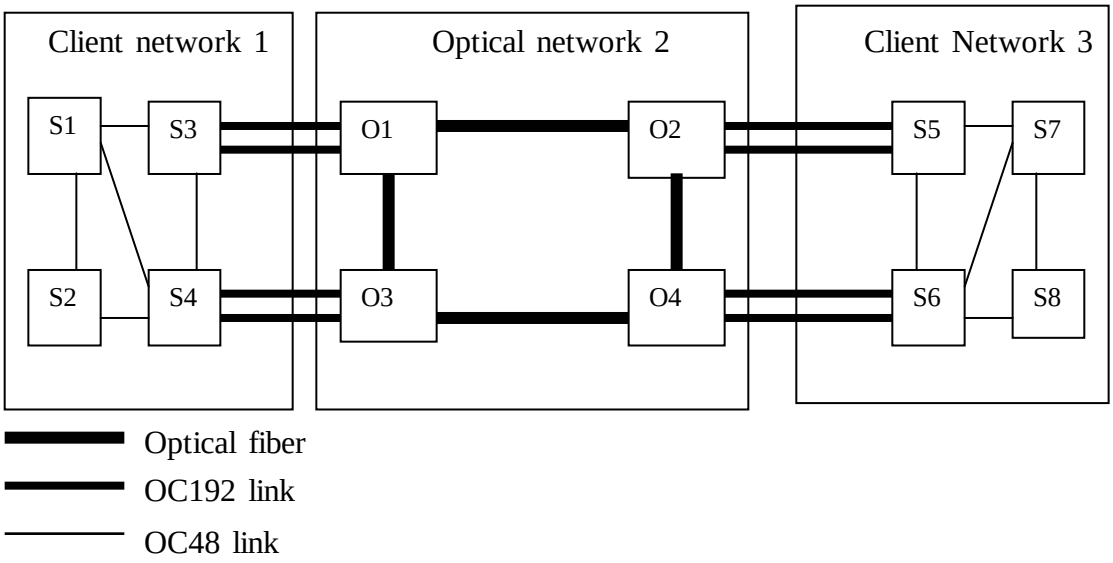


Figure 5.1: the network to show end-to-end 1:1 LSP protection

A client LSP is going to be established between Switch 1 of client network 1 and Switch 7 of client network 3, which requires 1:1 LSP protection. Switch 1 is the client LSP

initiator. This client LSP requires 622Mb/s (OC-12) bandwidth, and it is required to use links whose administration color is “red”. Note that a link is usually colored to indicate which administration group it belongs to. Here we assume that the client wants the links that belong to the administration group “red”.

To support traffic engineering, the primary path is an ER-LSP and it is pre-computed. Because the primary and the backup path are disjoint, the backup path should be also pre-computed. The database (TE-LSDB) stores the link TE information of the network. Let us assume that it has the information in Table 5.1 (Table 5.1 is on the coming page). For simplicity, the data encoding type contained by the Interface Switching Capability Descriptor (ISCD) is ignored. We only consider the interface switching type and the maximum reservable bandwidth of the ISCD. And we also assume that the TE information is the same for both directions of a link.

Because of the administration constraint, we only consider “red” links. So link (S1, S4) is excluded. Link (S6, S7) only has 500 Mb/s available, which is less than the required bandwidth. So it is also not considered. If we use a link whose link protection type is not “unprotected”, e.g., “dedicated 1+1”, then we must configure the coordination mechanism at each node of the path so as to avoid multiple-layer protection contention. If a node has intelligence to configure itself (e.g., the auto-configuration mechanism), then manual configuration is not needed. For example, we set a policy in each node – if the link protection type is not “unprotected”, then the lower layer protection has higher priority and the hold-off timer is one second. When the signaling message arrives at the node, the node configures itself based on link protection type. Another choice is to only use links whose link protection type is “unprotected”, and disable the link layer protection (e.g., set the hold-off timer to zero). Let us take this choice. Therefore, the topology we will consider becomes as in Figure 5.2. We calculate the metric (cost) of the link by (10^8 / available bandwidth) and we have the cost of the link, which is also shown in Figure 5.2.

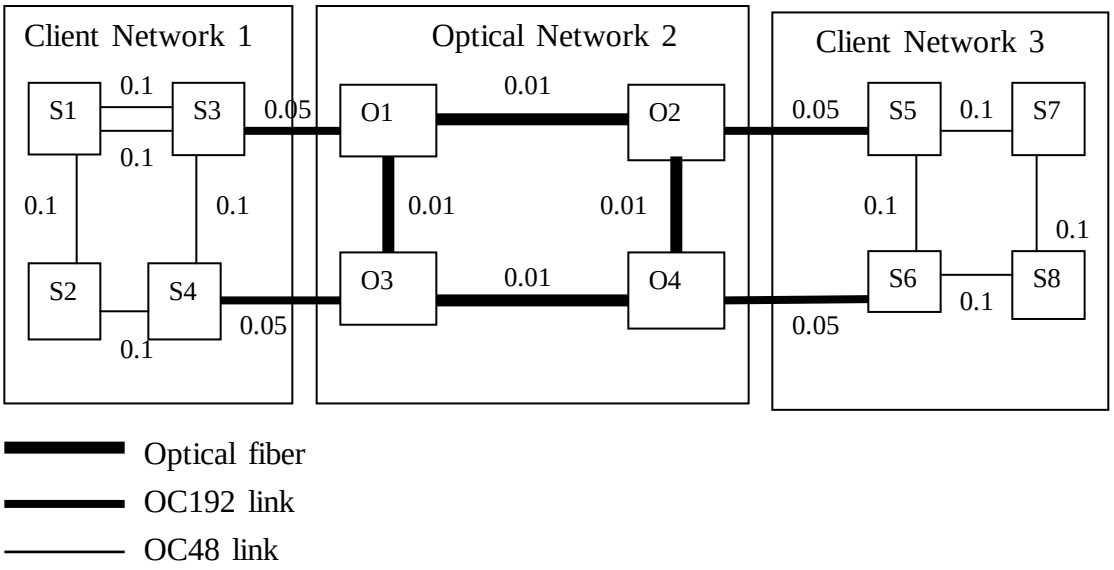


Figure 5.2: the topology that does not violate the constraints

Links (S1-1, S3-1) and (S1-2, S3-2) are equivalent links. By some policy (e.g., random), the first one is chosen. Then we can use the SPF algorithm to calculate the "shortest" path, which is (S1-1, S3-1, O1, O2, S5, S7). This is the primary LSP.

Legend in Table 5.1
SRLG: Shared Risk Link Group
ISCD: Interface Switching Capability Descriptor
MRB: Maximum Reservable Bandwidth

Local Address	Remote Address	SRLG	ISCD	Unreserved Bandwidth	Link Protection Type	Admin. Color
S1-1	S3-1	11	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S1-2	S3-2	11	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S1	S2	12	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S1	S4	13	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	green
S2	S4	14	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S3	S4	15	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S5	S7	31	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S5	S6	32	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S6	S7	33	TDM, MRB =2.5Gb/s	500Mb/s	unprotected	green
S6	S8	34	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S7	S8	35	TDM, MRB =2.5Gb/s	1 Gb/s	unprotected	red
S5-1	O2-1	231	TDM, MRB =10Gb/s	2 Gb/s	unprotected	red
S5-2	O2-2	231	TDM, MRB =10Gb/s	2 Gb/s	Dedicated 1+1	red
S6-1	O4-1	232	TDM, MRB =10Gb/s	2 Gb/s	unprotected	red
S6-2	O4-2	232	TDM, MRB =10Gb/s	2 Gb/s	Dedicated 1+1	red
S3-1	O1-1	121	TDM, MRB =10Gb/s	2 Gb/s	unprotected	red
S3-2	O1-2	121	TDM, MRB =10Gb/s	2 Gb/s	Dedicated 1+1	red
S4-1	O3-1	122	TDM, MRB =10Gb/s	2 Gb/s	unprotected	red
S4-2	O3-2	122	TDM, MRB =10Gb/s	2 Gb/s	Dedicated 1+1	red
O1	O2	21	LSC, MRB =160 Gb/s	10 Gb/s	unprotected	red
O1	O3	22	LSC, MRB =160 Gb/s	10 Gb/s	unprotected	red
O2	O4	23	LSC, MRB =160 Gb/s	10 Gb/s	unprotected	red
O3	O4	24	LSC, MRB =160 Gb/s	10 Gb/s	unprotected	red

Table 5.1: the TE link information for path calculation

Because link (S1-2, S3-2) has the same SRLG as the link (S1-1, S3-1), and the latter has been chosen for the primary LSP, it should not be considered when calculating the backup LSP. The primary and the backup LSPs should be disjoint, so the topology we can consider for the backup LSP becomes:

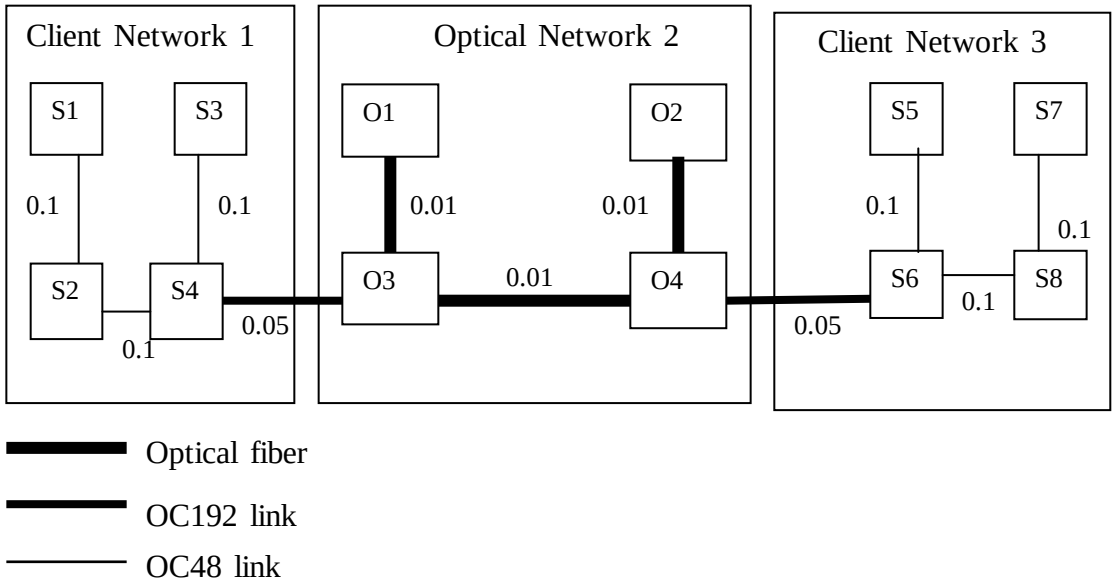


Figure 5.3: the topology for the backup LSP

Using the SPF algorithm, we have the "shortest" path (S1, S2, S4, O3, O4, S6, S8, S7) for backup LSP.

Now the Switch 1 can signal both LSPs, e.g., using RSVP-TE. Explicit-routed LSP (ER-LSP) signaling is used. The signaling protocol carrying the ERO establishes the LSP starting from Switch 1. When signaling arrives at SONET Switch 3, Switch 3 finds out that it is at the boundary for a hierarchical LSP by the Interface Switching Capability Descriptor. Let us assume that there is no existing FA-LSP that meets the requirement of the LSP being set up. So Switch 3 establishes a new FA-LSP starting from Switch 3 and terminating on Switch 5. Switch 3 initiates the new FA-LSP. When the signaling arrives at OXC1, OXC 1 finds out it also needs a new FA-LSP between OXC 1 and 2. Let us call this FA-LSP *F1w*, which has Link Protection Type "unprotected". After that, FA-LSP between Switch 3 and 5 is tunneled through *F1w*. We call this FA-LSP (between Switch3 and Switch5) *F2w*. It also has Link Protection Type "unprotected". This FA-LSP tunnels the client LSP. Eventually, the client LSP between Switch 1 and Switch 7 is established. It is the primary path we want, which we call *Pw*. A hierarchical LSP establishment is illustrated in Section 2 of this report.

Similarly, for the backup LSP, the FA-LSP between OXC 3 and 4 is called *F1b*; the FA-LSP between SONET Switch 4 and 6 is called *F2b*. Both of them have Link Protection Type "unprotected". And the backup LSP is tunneled through these FA-LSPs, and let us call it *Pb* (see Figure 5.4). LSP *Pw* and *Pb* construct the 1:1 LSP protection as desired.

The FA-LSP *F1w/b* and *F2w/b,* which have Link Protection Type "unprotected", will be advertised by the routing protocol. And their unreserved bandwidth is the difference between the maximum reservable bandwidth and the share used for LSP *Pw* (or *Pb*). For example, the FA-LSP *F2w* advertises that it has bandwidth 9.178 Gb/s available, and the FA-LSP *F1w* advertises that it has 15 lambdas available, each of which has OC192 bandwidth.

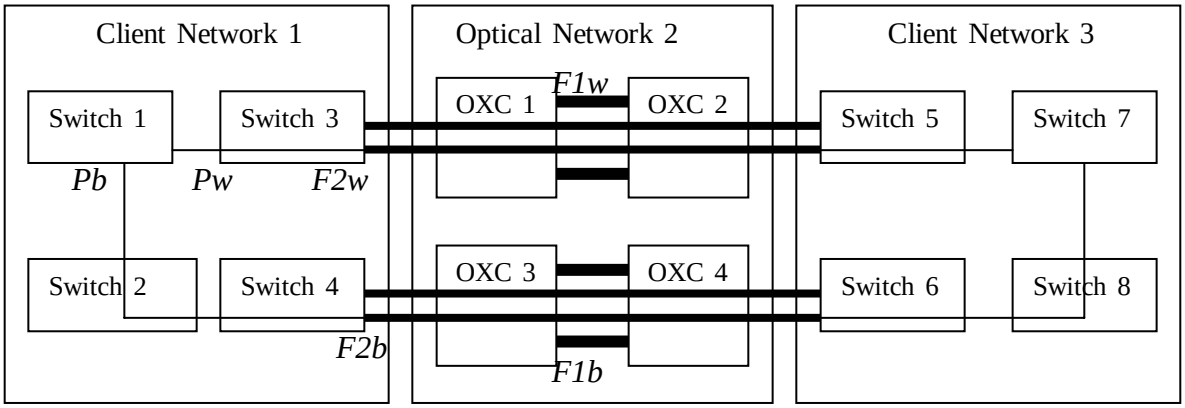


Figure 5.4: the 1:1 LSP protection

During the signaling, the resources are reserved. When the signaling takes place, the RSVP-TE Path message carries a flag that tells the nodes the LSP being signaled is the primary LSP or the backup one. Because the user requires 1:1 LSP protection, the user's traffic is not transported over the backup LSP until a failure occurs. The resource of the backup LSP may be used by other LSPs that have lower priorities.

When the RSVP-TE Path message is sent out, it carries the Notify Request object. It has the "targeted" node IP address, which is Switch 1 in this case. Every node along the path records this IP address. This is the end-to-end LSP protection. It is not necessary for the node that is responsible for triggering the traffic switch to know exactly where the failure occurs on the path. So it is not necessary to localize the failure. All nodes that detect the failure report the failure to the LSP initiator node. They send out a RSVP-TE Notify message destined for the targeted node – Switch 1. The LSP initiator node can trigger the traffic switch as soon as it receives the first notification, e.g., even one RSVP-TE Notify message.

Such an LSP protection can protect any failure on the LSP. But it takes a long time for the fault notification to travel the networks to reach the LSP protection initiator. For many real-time applications, e.g., voice over IP, it is highly desirable to be able to recover in 10s of milliseconds [48]. Fault notification may not work so fast. Therefore protection needs to improve for real-time applications. If what the user requires is the end-to-end restoration, the protection LSP is not pre-established. The primary LSP initiator does not start signaling the protection LSP until the failure occurs and the initiator is notified. So

end-to-end restoration is even slower and obviously it does not meet the requirement of real-time applications.

We will see how another end-to-end protection scheme can improve the recovery time in the next section.

6.2 Case Study 2: The Domain-specific Protection

A recent proposal [49] describes a GMPLS LSP protection scheme that is based on different network domains. It is called *subnetwork protection*.

The network across which a hierarchical LSP travels is partitioned into subnetworks. The nodes constructing the subnetwork have the same multiplexing capacity. Within each subnetwork, there is a pre-established backup LSP to protect the primary LSP. And the resource may also be pre-allocated. Because all nodes in a subnetwork have the same multiplexing capacity, the primary and the backup LSP are at the same level in the LSP hierarchy. The protection mechanism in each subnetwork can be M:N or 1+1. If there is a failure, for M:N protection mechanism, the traffic switchover occurs from the primary LSP to the backup LSP; for 1+1 protection mechanism, the LSP terminator node selects the traffic from the backup LSP. The protection is only performed within the subnetwork where the failure occurs. There is no need to do anything in other subnetworks across which the hierarchical LSP travels. The logical view of this idea is shown by the 1:1 protection mechanism in Figure 5.5. There is a protection LSP in the subnetwork for the primary LSP segment that goes over that subnetwork.

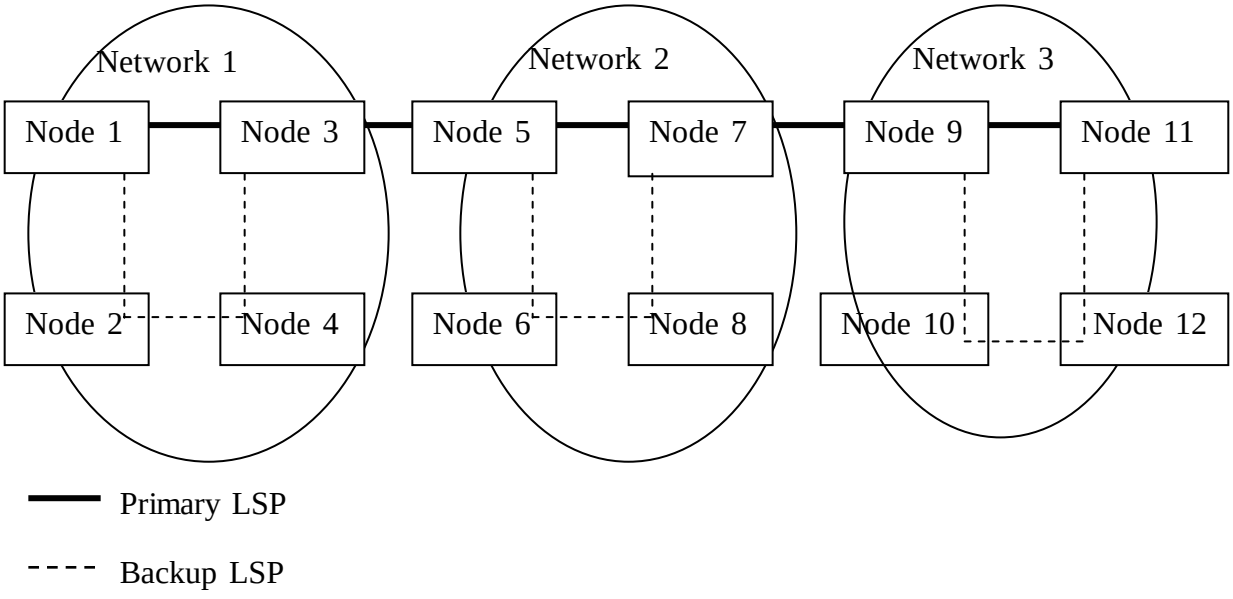


Figure 5.5: The logical view of the subnetwork protection

In Figure 5.5, the primary LSP is (Node 1, Node 3, Node 5, Node 7, Node 9, Node 11). If the link between Node 5 and Node 7 is broken, the protection LSP (Node 5, Node 6,

Node 8, Node 7) takes over the traffic, and there is no action in other networks. Traffic goes from Node 1 to Node 11 by (Node1, Node3, Node 5, Node 6, Node 8, Node 7, Node 9, Node 11).

In this subnetwork protection mechanism, the segments of the primary LSP are protected by the protection LSPs in different subnetworks. Compared to end-to-end LSP protection introduced in the previous section, this protection mechanism requires shorter time for fault notification as the fault notification only travels to the nodes within a subnetwork. Compared to local reroute, it is simpler. But, such a protection mechanism does not protect the nodes/links that are at the border of the subnetworks. The links at the borders can be protected by the link layer mechanism. However, the border nodes do not have protection. For example, there is no protection if Node 5 goes down in Figure 5.4. Fortunately, in practice, usually the nodes at the border of the network are very powerful and reliable.

Let us see how to implement such a protection scheme for the case we mentioned in the previous section. The client LSP has the same requirements as that in the previous section. Here let us re-use the network shown in Figure 5.1. Because the link (S6, S7) does not have enough available bandwidth and link (S1, S4) violates the constraint, we have the topology as in Figure 5.6 to consider.

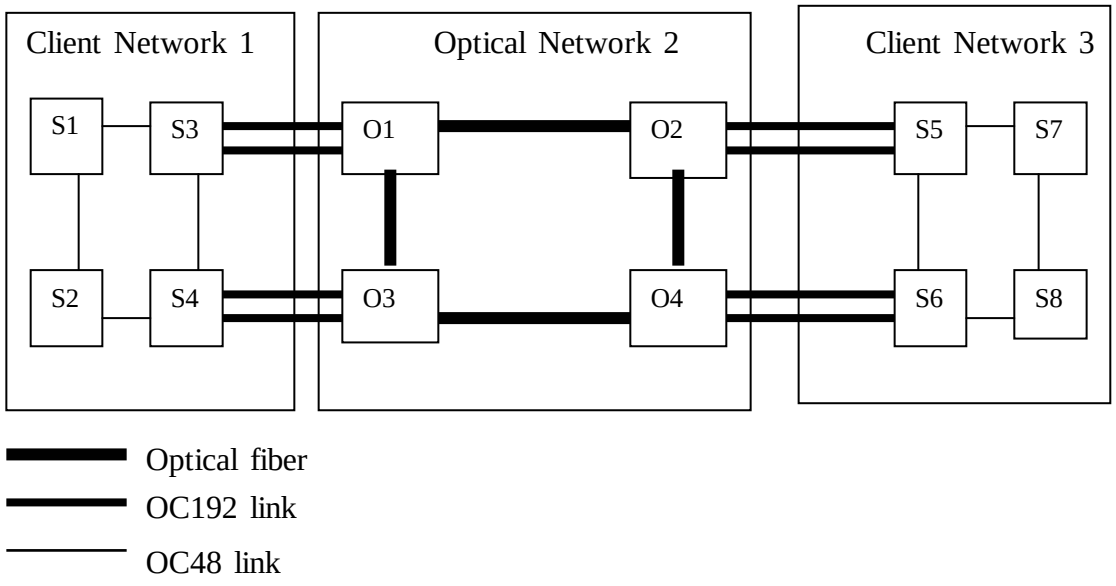


Figure 5.6: the network to show the subnetwork LSP protection

After the implementation, the LSP protection should be as follows. Switch 1 initiates the LSP, and this client LSP is tunneled by the high-order LSP from Switch 3 to Switch 5, which in turn is tunneled by the higher-order LSP from OXC 1 to OXC 2. Finally the client's LSP terminates at Switch 7. Because the 1:1 LSP protection is required, we choose such a method - all the links within the networks have link protection type "unprotected", but the link between Switch 3 and OXC 1 has link protection type

"Dedicated 1:1", e.g., the SONET APS link layer protection. So is the link between OXC 2 and Switch 5. The primary LSP is (link (S1, S3), link (S3-2, O1-2), link (O1, O2), link (O2-2, S5-2), link (S5, S7)). Within Client Network 1, the LSP segment from Switch 1 and 3 is protected by LSP (Switch 1, Switch 2, Switch 4, Switch 3). Within the optical network, the LSP segment from OXC 1 to OXC 2 is protected by LSP (OXC 1, OXC 3, OXC 4, OXC 2). And within network 3, the LSP segment from Switch 5 to Switch 7 is protected by LSP (Switch 5, Switch 6, Switch 8, Switch 7). The resource has been allocated on these protection LSPs, but the protection LSPs do not transport traffic. Thus, the entire user LSP has 1:1 LSP protection except the edge nodes, like, OXC 1, OXC 2, Switch 3 and Switch 5, and except the initiator and terminator nodes (see the following figure). The protection mechanism can protect any failures between the edge nodes within each subnetwork.

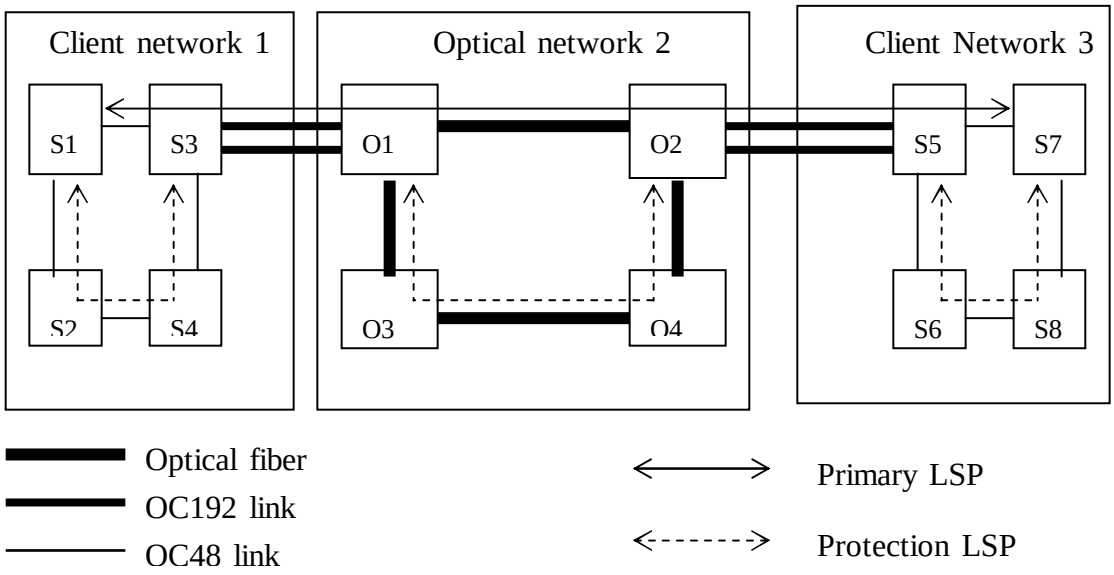


Figure 5.7: the network to show the subnetwork LSP protection

How to signal such a protection scheme? As this report is being written, there is no automatic mechanism proposed in IETF yet. Let us discuss what we need to do.

(1) The primary LSP and protection LSP should be disjoint within each subnetwork. It means the protection LSPs must be pre-computed, so they are explicit-routed LSPs

(2) Different protection mechanisms should be allowed within the subnetwork, e.g., 1+1 or 1:1. The LSP initiator node can be configured to create one of these protection mechanisms, but how to tell the ingress node (a node at which the working LSP enters a subnetwork) about the desired protection mechanism so that the ingress node signals the protection LSP? For example, in Figure 5.6 (on the previous page), how does the signaling protocol tell OXC 1 or Switch 5 to establish the protection LSP? And which protection mechanism is wanted, e.g., 1+1 or 1:1? The current signaling protocols do not provide any support yet, but it is possible to add some extensions to support this

subnetwork protection scheme, e.g., a new object in RSVP-TE. This new object is only processed by the nodes of the primary LSP that are at the border of different subnetworks. For example, Switch 1, Switch 3, OXC 1, OXC 2, Switch 5, Switch 7 in Figure 5.6 (see the previous page).

(3) The protection LSP should be pre-established so as to provide fast recovery. Resources may be pre-allocated as well. For M:N protection, lower priority traffic should be allowed to use the resource if the protection LSP is not protecting.

(4) Coordination mechanisms should be used to avoid the multi-layer protection contention if there is any. For example, "unprotected" link protection type may be used to signal both of the primary and backup LSPs.

(5) There is a problem concerning the incoming interface. Within each subnetwork, the primary LSP segment and the backup LSP merge at the edge node. The incoming interface may be regarded as a "label" and involved in label switching, e.g., in a network constructed by nodes that is fiber-switch capable, the incoming port may determine the outgoing port. Another example is an MPLS router that is packet-switch capable uses interface-based label space. The problem is illustrated in the following figure.

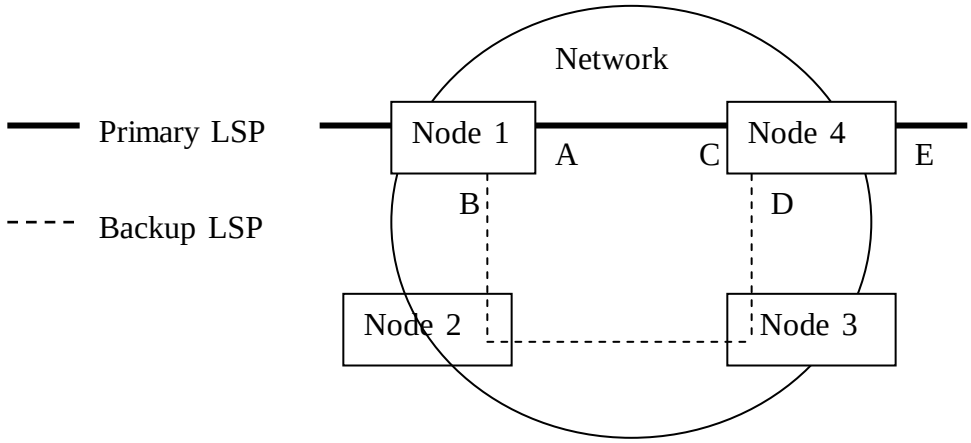


Figure 5.8: the incoming interface problem in the subnetwork protection

Node 1 switches the traffic from interface A to interface B if a failure between itself and Node 4 occurs. Then the traffic arrives at Node 4 through interface D, instead of C. If the label is unique node-widely (per-node label space), then there is no problem for Node 4 to work as usual, and the discussion can be stopped here. But in many situations, this is not the case. In order to reuse the label, usually per-interface label space is used. For example, a fiber can transport multiple wavelengths (lambdas), and another fiber on a different port can transport all the same wavelengths (lambdas). Let us assume that Node 4 has such an entry (see Figure 5.9) in its label forwarding table in the example shown in Figure 5.8:

Incoming information	Outgoing information
Incoming label	Outgoing label
Incoming interface C	Outgoing interface E
	…

Figure 5.9: the label forwarding entry in the example

Now the incoming interface has changed for Node 4, and how to tell it to accept the traffic from another interface and continue the label forwarding? One solution is to signal Node 4 to change the incoming interface C to D in its label forwarding entry after the failure is detected. It takes time and this protection would lose much of its value. Another solution is to tell Node 4 about it when the protection LSP is being established. In order to support this solution, a selector may be implemented in Node 4 that can select traffic from one of the multiple ports. Node 4 monitors the traffic from interface C and D, and it selects the healthier traffic from one of the two. The incoming interface may be programmed in the label forwarding entry before label switching occurs for optimization (see Figure 5.10) if the protection type allows. Or Node 4 can change the incoming interface in its label forwarding entry just before it is going to select the traffic from another interface.

Incoming information	Outgoing information
Incoming label	Outgoing label
Incoming interface C	Outgoing interface E
Incoming interface D	…

Figure 5.10: the interfaces for primary and backup LSPs are pre-programmed

(6) How to set up the multi-layer protection scheme like the link layer protection between nodes S3 and O1, O2 and S5? It is done usually by configuration. So is the set-up of coordination mechanism to avoid the multi-layer protection contention.

The other way to establish the entire *subnetwork protection* is by configuration. For example, on the network manager, the network administrator can configure such an LSP protection scheme. At the beginning, the network administrator requires the path computation component in the primary LSP initiator node to calculate the primary LSP. Then the LSP protection type and the primary LSP information (e.g., the nodes traveled

by the primary LSP) are sent to the ingress node of the primary LSP segment within each subnetwork, for example, node O1 in the optical network in Figure 5.6. At each ingress node, the protection LSP is calculated to protect the LSP segment that travels within that subnetwork. Note that the protection LSP must be disjoint with the primary LSP segment and the protection type should be honored. After that, the link layer protection (if needed) and the coordination for avoiding multi-layer protection contention can be done by configuration. How to solve the incoming interface problem? The egress node may provide an interface to network management for query and configuration. Such an interface allows the network administrator to manually query and configure the label forwarding table. We can see that using configuration to create such a protection scheme is tedious and error-prone.

Summary of the Subnetwork Protection Scheme

If the link between subnetworks fails, then the link layer protection is triggered. And it is expected that the link layer protection takes a short time to recover, e.g., the SONET APS just takes less than 50 ms to recover. If there is a failure (not the edge nodes) in a subnetwork, fault notification just needs to notify the head node of the LSP segment within that network. So the notification message travels only within that subnetwork. Compared to end-to-end LSP protection, it takes less time. The paper [50] proves that, in theory, it is possible to guarantee the 50 ms recovery time in large mesh networks by properly partitioning the network and applying subnetwork protection.

This subnetwork protection scheme also has another advance – it can protect a number of LSPs (see Figure 5.11). If a failure between Node 1 and Node 4 occurs, the protection LSP, which has the same level as the primary LSP segment within the subnetwork, is activated to protect the primary LSP. The tunneled low-order LSPs, e.g., LSP 1, 2 and 3 in the example, are not affected, and they are not even aware of the failure.

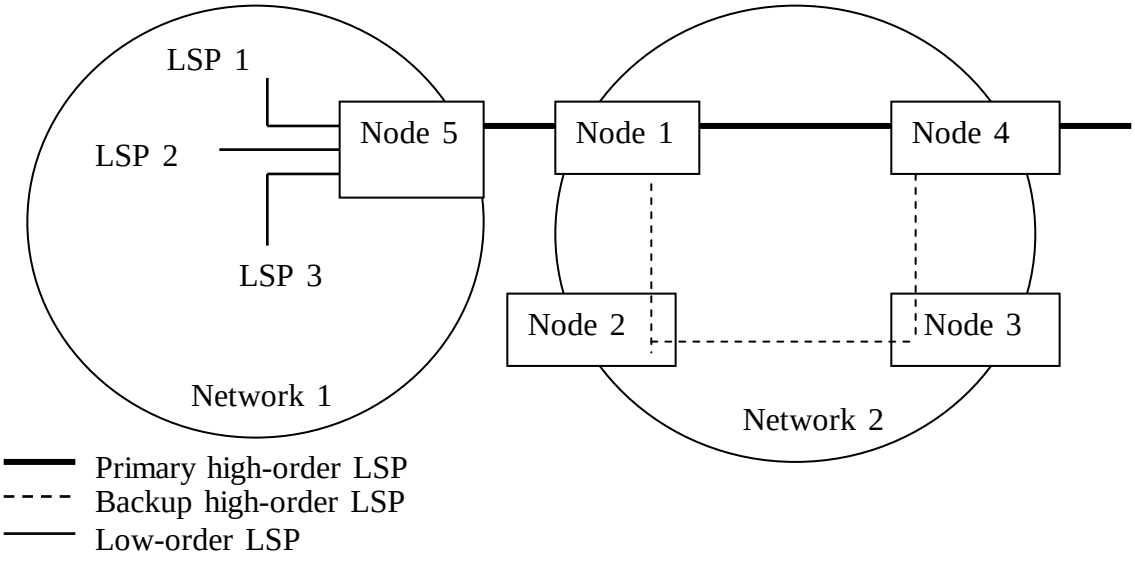


Figure 5.11: the subnetwork protection idea protects multiple low-order LSPs

This subnetwork protection scheme is resource-efficient. For example, the dedicated 1:1 end-to-end LSP protection mechanism doubles the resource. But in the subnetwork protection scheme, the resource for 1:1 LSP protection is shared - the protection LSP can be shared by multiple low-order LSPs.

Compared to local/global restoration, the protection LSP in the *subnetwork protection* is pre-established. So it provides faster recovery. But as other protection mechanisms, it requires more resource than restoration.

The signaling issues to solve the incoming interface problem in this subnetwork protection scheme needs further study.

6.3 Case Study 3: Link-layer Protection and Local Reroute

In the mesh network shown in Figure 5.12, photonic switches construct the core network. At the edge, devices O1 and O2 are optical switches. The optical switch has interfaces that provide WDM capabilities for photonic switches, and interfaces that provide SONET section level signals. SONET switches are connected to O1 and O2. They provide OC-192 capacity interface. Between O1 and P1, it is the WDM multiplexing of 16 OC-192 signals which remain intact through to O2. All lines have dedicated 1+1 link protection (the dedicated protection link is not shown in the figure). The links between SONET switches are OC48 links, like the link between S1 and S3, the link between S5 and S7. The optical switches O1, O2, O3 and O4 are IP-over-WDM nodes. So are the photonic switches.

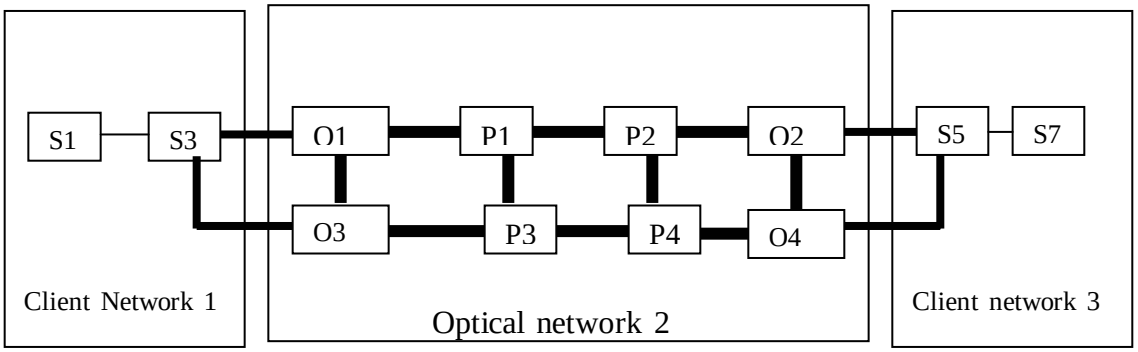


Figure 5.12: an LSP requiring 1+1 protection is built in the mesh network

A client LSP is going to be established between Switch 1 of client network 1 and Switch 7 of client network 3. It requires fault recovery in the optical network. The LSP will be used to transport real-time applications and the recovery should be done quickly if there is a failure, e.g., in 10s of milliseconds.

Link layer protection is one of the solutions for fast failure recovery. Let us study it here to see if 1+1 link layer protection can work in this case. If we build an LSP whose links all have “Dedicated 1+1” link layer protection type, the whole LSP has link protection. But what happens if a node goes down? Let us see an example in Figure 5.13. All the

nodes are IP-over-WDM nodes. If node N3 goes down, how to recover the failure even if all the links have 1+1 link protection? So just link layer protection cannot work. Other recovery mechanisms are needed to complement the link protection.

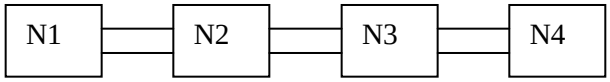


Figure 5.13: all links have 1+1 link protection between nodes

Because an LSP which has 1+1 link protection has doubled the resource for the traffic, further expensive recovery mechanisms are not desired any more. One of the solutions is to use local reroute. Let us consider if this recovery mechanism can work.

When establishing the primary LSP, the RSVP-TE Path message carries the RSVP-TE protection object, which signals "Dedicated 1+1". To avoid multiple layer protection contention, the coordination mechanism must be set during the signaling. Let us use hold-off timer. Note that, in order to meet the recovery time requirement, the hold-off time set-up must consider the time needed for MPLS-based recovery in case the link layer protection fails. We use LSP local reroute as the MPLS-based recovery in this case. And the link layer protection has higher priority than the MPLS-based recovery. Let us assume that the primary LSP is (S1, S3, O1, P1, P2, O2, S5, S7). The primary LSP contains: FA-LSP1, which is from O1 to O2; and FA-LSP2, which is from S3 to S5. When establishing FA-LSP1, P2 knows that it is the penultimate node of this FA-LSP, e.g., routing tells P2 that it is directly connected to O2. Let us assume we have such an administration policy that the penultimate node of the FA-LSP must notify the initiator node of the FA-LSP. The "target" address for the Notify message can be configured. In this case, P2 can send out the RSVP-TE Notify message targeted to O1. O2 is the penultimate node of FA-LSP2, and similarly it knows it will send a Notify message to S3 if S5 fails.

Figure 5.12: the primary LSP

At first, we consider Case 1 (see the section about local restoration for what the different cases are), for example, P2 goes down. P1 detects its neighbor's failure, e.g., by the Hello protocol (the Hello is exchanged between the neighbors every 5 ms). The link layer protection is triggered. Unfortunately, after the hold-off time, P1 finds out the failure is still there. So the hold-off timer triggers the LSP local restoration. The routing information database (LSDB) in P1 still shows that there is a route (P1, P3, P4, O2) to O2. Without waiting for routing information synchronization, P1 constructs an ER-LSP to reach O2, whose RSVP-TE ERO object contains P1, P3, P4 and O2. Because all of interfaces connecting these nodes have the same interface switching type – Lambda Switch Capable, there is no higher-order LSP needed. The reroute ER-LSP (P1, P3, P4, O2) has the same level as FA-LSP1 (O1, P1, P2, O2). When the signaling RSVP-TE

Path message driven by the ERO object arrives at O2, based on the PSVP-TE Session object and Sender template object, O2 understands the LSP has to be modified. So O2 modifies its label forwarding table and responds with a RSVP-TE Resv message. The message arrives at P1. And P1 understands that the reroute succeeds. So it also modifies its label forwarding table and switches over the traffic onto the reroute ER-LSP. If node P1 goes down, the reroute process is similar as both P1 and P2 are transit nodes of FA-LSP1.

If O2 fails, then reroute Case 2 occurs. P2 detects its neighbor's failure. As P2 is configured to notify the FA-LSP1 initiator O1, it sends out the RSVP-TE Notify message destined to O1. O1 is notified, and it tells all the tunneled low-order LSPs to reroute as it is the border of the hierarchical LSP. For example, it tells node S3 of FA-LSP2 to reroute. S3 consults its current routing database and builds the ERO object to signal a reroute ER-LSP. It understands it must cross the optical network to reach SONET switch S5. So the ERO object (S3, O3, P3, P4, O4, S5) is built and part of this ER-LSP (O3, P3, P4, O4) is a higher order LSP compared to FA-LSP2. The ERO drives the signaling. When it arrives at node O3, the higher-order FA-LSP is triggered to set up – let us call it FA-LSP1'. After that the reroute ER-LSP reaches S5. And the FA-LSP2 is tunneled by this FA-LSP1'. The reroute bypasses the faulty O2.

If O1 fails, the reroute Case 3 occurs. S3 detects its neighbor's failure and S3 triggers the reroute. S3 consults its current routing database and builds the ERO object to signal a reroute ER-LSP. The process is like what the S3 does in reroute Case 2 (see the preceding paragraph).

When we use reroute as the recovery method, we need to carefully consider the network topology. Due to the network topology, reroute may not work. For example, in the case we just described, if the user wants the fault recovery from end to end, reroute cannot work if node S5 goes down.

7. Conclusion

We have talked about the objectives for the LSP protection/restoration in Section 3. We note that the objective to *be cost-effective* may involve non-technical factors, but we do not discuss them here in this report. We compare the LSP protection/restoration mechanisms in GMPLS networks in the following table.

LSP recovery mechanisms	Resource requirements	Speed of recovery	Complexity	Application scope
Conventional local restoration	No resource is pre-allocated, the repaired LSP requires same resource	Very slow as it waits for the routing synchronization	No change to the current signaling protocols	Limited as the user's strict ER-LSP is not desired to be rerouted.
Local restoration with ER-LSP [47]	No resource is pre-allocated, the repaired LSP requires same resource	Fast. It does not wait for the routing synchronization to signal the reroute path. The path computation takes little time.	No change to the current signaling protocol, but it requires extra intelligence	Limited as the user's strict ER-LSP is not desired to be rerouted.
End-to-end restoration	No resource is pre-allocated, the repaired LSP requires same resource	Very slow. Fault localization is performed, fault notification takes time to travel across networks, and the reroute LSP is not set up until the failure occurs.	No change to the current signaling protocol	Can be used in any situations and the recovery meets traffic engineering goals
Local protection	Double resource is pre-allocated	Very fast as it is done at the link/physical layer	Additional configuration is needed to set up	It cannot easily provide node protection.
End-to-end protection	Additional resource is pre-allocated, dedicated 1+1 LSP protection requires double resource	1+1 LSP protection does not need fault notification but M:N LSP protection does.	Additional configuration may be needed to set up the protection on the end nodes of the LSP	It can be used in any situations

Table 6.1: comparison of recovery mechanisms

All protection/restoration mechanisms sacrifice resource to achieve fast recovery. Because additional resource is pre-allocated in the protection mechanism, the protection mechanism is expected to provide faster recovery than restoration. So objectives for LSP recovery

(1) *to optimize the use of resources* and (2) *to provide fast recovery and minimize the disruption to data traffic of any failure* are conflicting. Many protection/restoration mechanisms require signaling at the time of failure. The more signaling is required, the more time the mechanism takes to recover, and the less likely the recovery is timely.

We can achieve fastest recovery if we pay double resource, e.g., using the link/physical layer protection. The 1+1 LSP protection requires double resource, which is the most expensive LSP protection, and it can provide fastest LSP recovery. Any other protection mechanisms that share backup resource require fault notification. For example, the M:N, 1:N or 1:1 end-to-end LSP protection requires that fault notification travels across a number of nodes, which may cost time. The subnetwork protection mechanism tries to shorten the fault notification time but the nodes at the network boundary do not have any protection.

Many restoration mechanisms require a lot of signaling, so they usually do not meet real-time applications' requirement. The local restoration using ER-LSP proposed by [47] does not need fault notification and it does not need to wait for routing information synchronization. Although it needs to compute the ER-LSP to reroute, it does not give a burden to today's CPU. So it may be a fast restoration solution. However, the application scope of local restoration is limited.

Restoration mechanisms allocate resources after failure occurrence so they are resource effective but it takes time for them to provide recovery. Protection mechanisms provide fast recovery but they require additional resources. We should carefully consider the trade-off to choose the appropriate recovery mechanism so as to meet the requirements of users and network administration.

Compared to lower layer recovery mechanisms, the recovery mechanisms at the GMPLS level are relatively slow and may require more resources. Lower layer recovery mechanisms can provide fast recovery. But they have their limitations and disadvantages. For example, WDM networks may require complicated implementation and configuration for protection/restoration. And link layer protection cannot easily provide node protection.

In practice, usually a single type of protection mechanism does not satisfy the complicated working environment or user requirements. So a combination of recovery mechanisms is often the solution. When we choose a recovery solution, we need to achieve the balance between required resources and recovery time and the balance between cost and high survivability.

Nowadays, a lot of proposals have come up for LSP protection/restoration. GMPLS extends MPLS, but the LSP protection/restoration mechanisms that work in MPLS networks may not always work in GMPLS networks. For example, the "detour" proposal [48] makes the LSP very fault-tolerant in MPLS networks, but the current method described is only suitable for unidirectional LSPs. That is not applicable for GMPLS as bidirectional LSPs are recommended in GMPLS. Furthermore, the proposal places strict

constraints to the GMPLS network nodes when the "detour" LSP for protection is set up (see [51]). It is likely that the proposals that only work in MPLS networks but not in GMPLS networks would be dropped by IETF, e.g., [53] has been dropped, because of its limited scope.

Some proposals for LSP protection/restoration require the current signaling protocol to have more extensions, e.g., the one described in [48]. IETF considers these proposals very carefully as they would have a side-effect or put too much burden on the protocol. Some of these proposals are dropped, e.g., [54]. Therefore some researchers suggest that recovery mechanisms should be split from signaling protocol extensions (see [52]).

For local reroute, the aid from the signaling protocol is inevitable. But for the time being, none of the proposals in this area gets majority support. The issue is still being discussed in IETF.

With the further development of GMPLS, it is expected that more and more solutions are coming up for LSP protection/restoration in GMPLS.

References:

[1] E. Rosen, A. Viswananthan, et al., *Multiprotocol Label Switching Architecture*, RFC3031, IETF, http://www.ietf.org.

[2] E. Mannie, et al., *Generalized Multi-Protocol Label Switching* (*GMPLS) Architecture*, draft-ietf-ccamp-gmpls-architecture-02.txt, work in progress, IETF, http://www.ietf.org.

[3] P. Newman, G. Minshall, T. Lyon and L. Huston, *IP switching and gigabit routers*, IEEE communication magazines, January, 1997, pp.64-69.

[4] P. Newman et al., *Ipsilon's General Switch Management Protocol Specification*, RFC1987, IETF, http://www.ietf.org.

[5] P. Newman, W. L. Edwards, et al., *Transmission of Flow Labelled IPv4 on ATM Data Links*, RFC1954, IETF, http://www.ietf.org.

[6] Y. Rekhter, B. Davie, et al., *Cisco Systems' Tag Switching Architecture Overview*, RFC2105, IETF, 1997.

[7] C. Metz, *An overview of IP Switching Technology*, IBM Corporation, http://www.networking.ibm.com/isr/ip/ipswp1.htm.

[8] Professor R. Jain, Department of Computer and Information Science, The Ohio State University, http://www.cis.ohio-state.edu/~jain/cis788-97/ip_switching.

[9] D. Awduche, Y. Rekhter, J. Drake, R. Coltun, *Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control With Optical Crossconnects*, draft-awduche-mpls-te-optical-03.txt, Work in Progress, April, 2001, IETF, http://www.ietf.org.

[10] P. Ashwood-Smith et. al, *Generalized MPLS - Signaling Functional Description*, draft-ietf-mpls-generalized-signaling-02.txt, IETF Draft, Work in Progress, March, 2001, IETF, http://www.ietf.org.

[11] E. Mannie et al., Section 3.2 of GMPLS Architecture, *draft-ietf-ccamp-gmpls-architecture-02.txt*, work in progress, IETF, http://www.ietf.org.

[12] A. Banerjee, J. Drake, et al., *Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques.* July 2001, IEEE Communication Magazine.

[13] B. Rajagopalan, et al., Abstract of *IP over Optical Networks: A Framework*, draft-ietf-ipo-framework-01.txt, work in progress, IETF draft, http://www.ietf.org.

[14] B. Rajagopalan, et al., *IP over Optical Networks: A Framework*, draft-ietf-ipo-framework-01.txt, IETF draft, http://www.ietf.org.

[15] B. Rajagopalan, J. Luciani, et al., Section 3 of draft-many-ip-optical-framework-03.txt, work in progress, IETF.

[16] D. Awduche, J. Malcolm, et al., Section 2 of *Requirements for Traffic Engineering Over MPLS* (RFC2702), IETF, http://www.ietf.org.

[17] P. Srisuresh, P. Joseph, *TE LSAs to extend OSPF for Traffic Engineering*, draft-srisuresh-ospf-te-02.txt, work in progress, IETF, http://www.ietf.org.

[18] D. Cheng, *OSPF Extensions to Support Multi-Area Traffic Engineering*, draft-cheng-ccamp-ospf-multiarea-te-extensions-00.txt, work in progress, IETF, http://www.ietf.org.

[19] K. Kompella, Y. Rekhter, et al., *Routing Extensions in Support of Generalized MPLS*, draft-ietf-ccamp-gmpls-routing-04.txt, work in progress, IETF, http://www.ietf.org.

[20] K. Kompella, Y. Rekhter, *LSP hierarchy with Generalized MPLS TE*, draft-ietf-mpls-lsp-hierachy-06.txt, work in progress, IETF, http://www.ietf.org.

[21] K. Kompella, Y. Rekhter, A. Banerjee, et al., *OSPF Extensions in Support of Generalized MPLS*, draft-ietf-ccamp-ospf-gmpls-extensions-07.txt, work in progress, IETF, http://www.ietf.org.

[22] K. Kompella, Y. Rekhter, A. Banerjee, et al., *IS-IS Extensions in Support of Generalized MPLS*, draft-ietf-isis-gmpls-extensions-13.txt, work in progress, IETF, http://www.ietf.org.

[23] B. Rajagopalan, J. Luciani, D. Awduche, et al., Section 8.2 of *IP over Optical Networks: A Framework*, draft-ietf-ipo-framework-01.txt, work in progress, IETF, http://www.ietf.org.

[24] K. Kompella, Y. Rekhter, A. Banerjee, J. Drake, et al., *Routing Extensions in Support of Generalized MPLS*, draft-ietf-ccamp-gmpls-routing-04.txt, working in progress, IETF, http://www.ietf.org.

[25] P. Ashwood-Smith, L. Berger, et al., *Generalized MPLS Signaling - CR-LDP Extensions*, draft-ietf-mpls-generalized-cr-ldp-06.txt, work in progress, IETF, http://www.ietf.org.

[26] L. Berger, P. Ashwood-Smith, A. Banerjee, et al., *Generalized MPLS Signaling - RSVP-TE Extensions*, draft-ietf-mpls-generalized-rsvp-te-07.txt, work in progress, IETF, http://www.ietf.org.

[27] L. Berger, P. Ashwood-Smith, A. Banerjee, G. Bernstein, et al., *Generalized MPLS - Signaling Functional Description*, draft-ietf-mpls-generalized-signaling-08.txt, work in progress, IETF, http://www.ietf.org.

[28] L. Berger, P. Ashwood-Smith, A. Banerjee, G. Bernstein, et al., Section 4 of *Generalized MPLS - Signaling Functional Description*, draft-ietf-mpls-generalized-signaling-08.txt, work in progress, IETF, http://www.ietf.org.

[29] B. Davie, Y. Rekhter, Section 2 of *MPLS Technology and Applications*, Morgan Kaufmann Publishers, 2000, ISBN 1558606564.

[30] Y. Suemura, A. Kolarov, T. Shiragaki, *Protection of Hierarchical LSPs*, draft-suemura-protection-hierarchy-00.txt, work in progress, IETF, http://www.ietf.org.

[31] V. Sharma, F. Hellstrand, et al., *Framework for MPLS-based Recovery*, Section 2, draft-ietf-mpls-recovery-frmwrk-04.txt, work in progress, IETF, http://www.ietf.org

[32] B. Doshi, S. Dravida, P.Harshavardhana, O. Hauser, Y. Wang, *Optical Network Design and Restoration*, Bell Labs Technical Journal, Jan-March, 1999, see http://www.lucent.com/minds/techjournal/pdf/jan-mar1999/paper04.pdf

[33] G. Maier, S. De Patre, M. Martinelli, et al., *Resilience schemes in WDM networks*, Italy, June 2001, see http://leos.unipv.it/Pattavina.pdf

[34] V. Sharma, F. Hellstrand, et al., *Framework for MPLS-based Recovery*, Section 1.2, draft-ietf-mpls-recovery-frmwrk-04.txt, work in progress, IETF, http://www.ietf.org

[35] W. S. Lai, D. McDysan, et al., Section 5.5, *Network Hierarchy and Multilayer Survivability*, draft-ietf-tewg-restore-hierarchy-00.txt, work in progress, IETF, http://www.ietf.org.

[36] V. Sharma, B. Crane, et al., *Framework for MPLS-based Recovery*, draft-ietf-mpls-recovery-frmwrk-03.txt, work in progress, IETF, http://www.ietf.org.

[37] D. Katz, D. Yeung, et al., *Traffic Engineering Extensions to OSPF*, draft-katz-yeung-ospf-traffic-06.txt, work in progress, IETF, http://www.ietf.org.

[38] J. Lang, et al., *Link Management Protocol*, draft-ietf-ccamp-lmp-04.txt, work in progress, IETF, http://www.ietf.org.

[39] L. Berger, P. Ashwood-Smith, A. Banerjee, et al., *Generalized MPLS Signaling - RSVP-TE Extensions*, draft-ietf-mpls-generalized-rsvp-te-07, work in progress, IETF, http://www.ietf.org.

[40] L. Zhang, et al., Resource ReSerVation Protocol (RFC2205), IETF, http://www.ietf.org.

[41] L. Andersson, P. Doolan, et al., *LDP Specification* (RFC3036), IETF, http://www.ietf.org.

[42] B. Jamoussi, L. Andersson, et al., *Constraint-Based LSP Setup using LDP* (RFC3212), IETF, http://www.ietf.org.

[43] P. Ashwood-Smith, L. Berger, et al., *Generalized MPLS Signaling - CR-LDP Extensions*, draft-ietf-mpls-generalized-cr-ldp-06.txt, working in progress, IETF, http://www.ietf.org.

[44] A. Banerjee, J. Drake, et al., Section "GMPLS Protection and Restoration Techniques" of *Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques.* July 2001, IEEE Communication Magazine.

[45] V. Sharma, F. Hellstrand, et al., *Framework for MPLS-based Recovery*, Section 3.4, draft-ietf-mpls-recovery-frmwrk-04.txt, work in progress, IETF draft, 5, 2002.

[46] Jeff Doyle, *Routing TCP/IP*, Volume 1, Cisco Press, 1998, ISBN 1578700418.

[47] B. Davie, Y. Rekhter, Section 7 of *MPLS Technology and Applications*, Morgan Kaufmann Publishers, 2000, ISBN 1558606564.

[48] Ping Pan, Der-Hwa Gan, et al., *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*, draft-ietf-mpls-rsvp-lsp-fastreroute-00.txt, work in progress, July 2002, IETF, http://www.ietf.org.

[49] Y. Suemura, A. Kolarov, T. Shiragaki, *Protection of Hierarchical LSPs*, draft-suemura-protection-hierarchy-00.txt, work in progress, IETF, http://www.ietf.org.

[50] C. Ou, H. Zang, B. Mukherjee, *Sub-Path Protection for Scalability and Fast Recovery in WDM Mesh Networks*, Dept. of Computer Science, Univ. of California, Davis, CA, 2001.

[51] B. Miller, E. Harrison, A. Farrel, *An examination of the methods for protecting MPLS LSPs against failures of network resources*, Data Connection, Oct 2001, http://www.dataconnection.com/.

[52] D. Papadimitriou, et al., *Restoration Mechanisms and Signaling in Optical Networks*, Proceedings of the Fiftieth Internet Engineering Task Force, March 18-23, 2001, http://www.ietf.org/proceedings/01mar/slides/ccamp-7/.

[53] C. Huang, V. Sharma, S. Makam and K. Owens, *A Path Protection/Restoration Mechanism for MPLS Networks*, draft-chang-mpls-path-protection-01.txt, work in progress, July, 2000, http://www.ietf.org.

[54] C. Huang, V. Sharma, S. Makam and K. Owens, *Extensions to RSVP-TE for MPLS Protection*, IETF, work in progress, IETF draft, June, 2000, http://www.ietf.org.