Internet Engineering Task Force
INTERNET-DRAFT

Cristel Pelsser
FUNDP
Olivier Bonaventure
UCL
October, 2002

RSVP-TE extensions for interdomain LSPs

Abstract

We propose extensions to RSVP-TE to allow the establishment of traffic engineered LSPs with fast restoration requirements. We first discuss the problem of establishing explicitly routed interdomain LSPs and show that the current subobjects found in RSVP-TE are not sufficient to establish interdomain LSPs because they do not take into account the policy constraints of the interdomain environment. We then show how to extend the fast-reroute and detour objects to protect interdomain links and ASBRs on interdomain LSPs. We also discuss the establishment of disjoint interdomain LSPs for restoration and load balancing purposes in the appendix. Finally, we describe the necessary RSVP objects and flags and discuss the impact of the proposed solution on the syntax of existing RSVP-TE objects and the syntax of new required objects are presented.

1 Introduction

Today, most of the work on MPLS has focussed on its utilization inside a single domain. When considering traffic engineering, most of the existing solutions with MPLS assume that the domain is organized as a single IGP area. Interarea traffic engineering with MPLS is still an open problem.

In addition to MPLS-based traffic engineering inside a single area, there are several other important applications of MPLS that are not limited to a single domain. A first application is that a domain organised as a BGP confederation could be interested in using MPLS for traffic engineering and fast restoration purposes accross is subASes. This is not possible with the existing protocols. A second application are the MPLS-based VPNs that cross interdomain boundaries. In this case, interdomain LSPs need to be setup between domains. Given the reliability and performance requirements of VPNs, it can be expected that those interdomain LSPs will need to be traffic engineered and will require fast restoration in case of failures. Given the large BGP restoration time, a solution based only on BGP would not meet the requirements of the VPN users. A third application is the utilization of MPLS to establish virtual peerings through inter-AS LSPs. An example of virtual peerings with MPLS is given by the MPLS-IX architecture presented in [NEN02]. A fourth application is the establishment of optical LSPs that may cross interdomain boundaries [ea01].

This document is organized as follows. We first discuss the problem of establishing explicitly routed interdomain LSPs and show that the current subobjects found in RSVP-TE are not sufficient to establish interdomain LSPs because they do not take into account the policy constraints of the interdomain environment. We then look at the possibility of protecting segments of interdomain LSPs. We consider the protection of interdomain links and ASBRs since links and routers inside an AS may be protected by techniques exposed in [PGS+02].The protection of these resources requires extensions to the detour object from [PGS+02] and the introduction of a new object. Other extensions to the PCS protocol introduced in [VIZ+02] are left for further work. We also discuss the establishment of disjoint interdomain LSPs for restoration and load balancing purposes in the appendix. Finally, we describe the necessary RSVP objects and flags and discuss the impact of the proposed solution on the syntax of existing RSVP-TE objects and the syntax of new required objects are presented.

2 Establishment of inter-domain LSP

To setup an intradomain Label Switched Path (LSP), or an intradomain LSP segment, with RSVP-TE, the initiating Label Switching Router (LSR) needs to know the destination of the LSP. The destination of the LSP is either known through the Interior Gateway Protocol (IGP), as a BGP Next Hop, or by manual configuration. The initiating LSR computes a strict or a loose path towards the destination of the LSP depending on the topology information flooded by the IGP. If the domain is divided into areas, the initiating LSR may not be able to compute a strict route toward the destination since it only possesses limited information concerning the topology of the other areas in the domain. But, when the domain only consists of one area, a strict route may be computed. Even when it is possible to compute a strict route, a loose route may be computed instead, depending on manual configuration.

The situation is different when considering interdomain LSPs. In this case, the source of the LSP tunnel does not know the detailed interdomain topology. It only possesses information given by the IGP, concerning the domain to which it belongs, and the interdomain routes distributed by BGP. Therefore, it cannot determine precisely the path of the LSP all the way to the AS destination. This is not a problem because the Explicit Route Object (ERO) may be updated by intermediate LSRs on the way of the Path message. It follows that the source of the tunnel may be able to specify the path toward the next area or the next domain. Routing inside the area or the domain will be based on the ERO. A loose route may be given for the rest of the path. The border router will then compute, eventually based on the ERO, the route for the next area (domain) and update the ERO.

Another problem to consider in the dynamic establishment of interdomain LSPs is that the tunnel source usually does not know the IP address of the remote tunnel end point before establishing the tunnel. Based on its BGP routing table, the source of the LSP only has information about the destination prefixes and their AS paths. And, the remote end points of dynamically established interdomain LSPs cannot be configured manually since the need for such LSPs may not be known in advance. For routing purposes, the prefix information is much more useful than the AS path information, but the AS path information can be used to build an ERO object for the interdomain LSP.

To solve the problem of the remote tunnel end point address, we propose to enable the establishment of LSPs based on a prefix or on an AS number and a prefix. For the establishment of an LSP based on a prefix destination, the Path message should be forwarded through the network until it reaches an LSR that has an IP address that is part of this prefix. The Path message itself will be routed on the basis of its destination IP prefix and possibly along an explicit route defined by an ERO object.

The second type of destination that we propose is composed of two parts : an AS number and an IP prefix. In this case, the Path message should be forwarded through the network on the basis of the destination prefix until it reaches an LSR that is part of the specified AS. The path followed by the Path message can also optionally be specified with an ERO object.

Figure 1 shows the difference between a Path message with an AS plus prefix and a Path message with a prefix destination. When the destination of the Path message is an AS number, the node initiating the LSP chooses a prefix inside the AS destination and routing of the path message is based on the chosen prefix. Once a node inside the AS destination is reached the Path message stops, independently of the prefix used for routing purposes. A Path message with a prefix destination, is routed on the basis of this prefix. The Path message stops once it reaches a node inside the specified prefix.

Another issue to consider concerns the refresh messages. For the first Path message, we have proposed to use the AS+prefix or prefix destinations. These destination types are necessary to send the first Path message. However, once the first Resv message is received, the source LSR of the LSP knows the IP address of the destination LSR. A possible solution in this case would be to establish a new interdomain LSP with the found destination IP address and to cancel the establishment of the LSP with the AS+prefix or AS destination. However, this would mean that two LSPs with different identifiers are first established before tearing off the one with prefix or AS destination. This is not desirable and could create problems like multiple reservations of the same resources. Tearing down the LSP established with a prefix or AS destination before

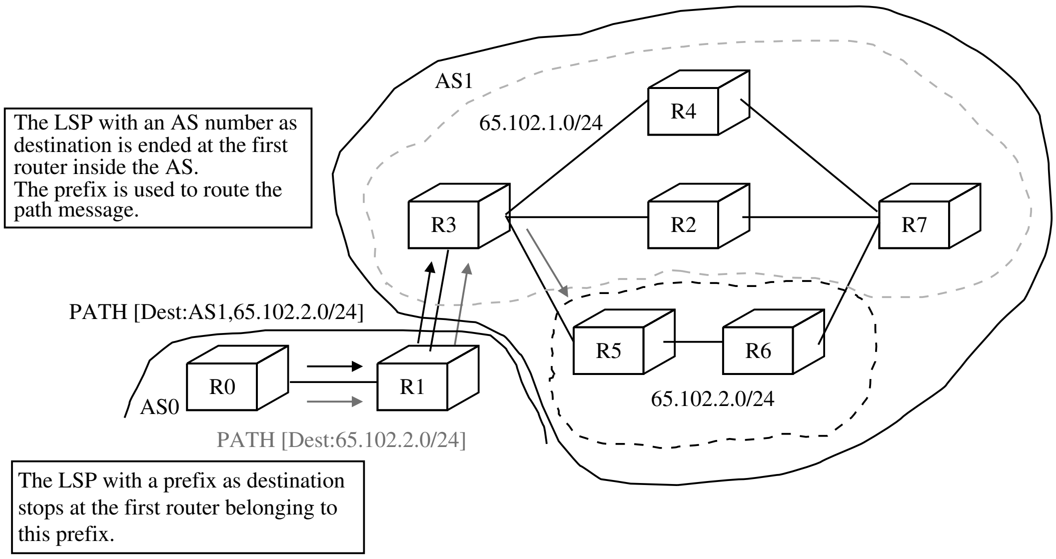

Figure 1: Establishment of LSP with AS+prefix or prefix destination

establishing the LSP with the corresponding IP destination address does also not ensure the final establishment of the LSP since the needed resources may meanwhile be allocated to other LSPs. To avoid these problems, a new object containing the identifier of the first Path message could be inserted into the first following refresh message. This object will be used to identify the path-state of the LSP and update it with the new identifier based on the now known remote tunnel end-point. This solution requires that the new Session object types, corresponding to AS+prefix and prefix destinations, be supported by all intermediate nodes as well as the object used to store the original identifier of the LSP which contains a prefix or an AS destination. We do not opt for this solution since, in addition to the required support of new objects, a method is needed to determine when the initial identifier does not need to be transmitted inside Path refresh messages to face the non reliability in the transmission of RESV messages. A last proposal, would be not to add any new types of Session objects. A prefix destination is then represented by an IP address terminated by zeros. Additionally, a subobject representing the prefix destination is inserted at the end of the ERO and a flag indicates that there is no need to establish the LSP beyond the first node belonging to the prefix subobject. Once a node that belongs to the last subobject in the ERO is reached, the Path message is ended and a Resv message is sent upstream. In the case of an AS destination, the Session object is also an IP address that is set to the prefix, used for routing the Path message, followed by zeros. And, the last subobjects of the ERO are the number of the AS destination and the prefix used for routing purposes with a flag indicating that the Path message is ended once a node belonging to the AS is reached. The AS number subobject is inserted to ensure that the AS destination is reached before terminating the Path message once the prefix subobject is treated. In this last proposal, all subsequent Path refresh messages will carry the same Session object. The identifier of the LSP will be carried in all those messages allowing a router to access the path-state of the corresponding LSP. This solution does not affect the Session object that must be supported by all routers along the path of the LSP.

2.1 Processing of the ERO and RRO objects

We expect that across interdomain boundaries, the ERO object will be often used to specify a strict or loose path for the LSPs being established. This object is often used in combination with the RRO object for route pinning purposes. Inside a single AS, the following situation typically occurs. The source LSR creates a new LSP with a loose ERO object and an RRO object. Once the LSP is established, the source LSR receives an RRO object with the complete list of the IP addresses of the LSRs traversed by this LSP. With this RRO object, the source LSR can then easily create a new strict ERO object that will be used to pin the route of the established LSP. The RRO object also enables the source LSP to compute a node disjoint LSP from the primary LSP. Furthermore, both the ERO and RRO objects are used to detect loops an LSP.

However, in the interdomain case, we must take care about transparency issues that do not occur inside a single AS. The two main problems are that the interdomain routing protocol does not distribute the detailed Internet topology and that an AS may not want to reveal its topology. For this reason, an AS may not agree to reveal the detailed path followed by an LSP by propagating the RRO object to external peers. To meet this transparency requirement while still being able to support route pinning, disjoint path computation and loop detection, we propose changes to the processing of the Record Route Object (RRO) at the AS border routers.

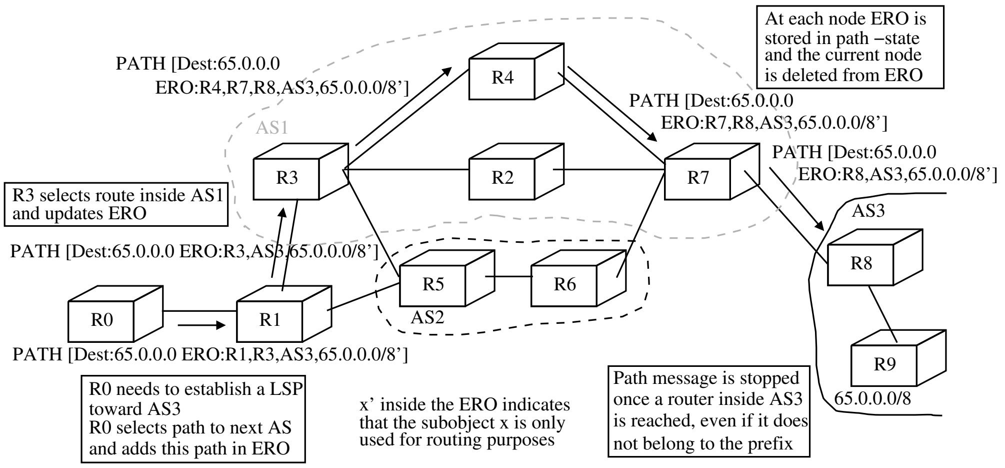

Figure 2: Establishment of an interdomain LSP

Figure 2 illustrates the establishment of an inter-domain LSP. In this case, router R0 determines that it needs to establish an LSP toward AS3. It selects a prefix that belongs to AS3 and creates a Path message with the destination set to the chosen prefix. The last subobject in the ERO represents the chosen prefix and the previous subobjects contains AS3 AS number. When R1 receives this Path message, it selects an interdomain path that verifies the constraints that may be optionally specified inside the Path message [VIZ$^+$02]. Then, it inserts the computed path inside the ERO and stores the ERO in the path-state. When receiving the Path message, R3 checks if it belongs to the first abstract[1] node in the ERO. Then, it computes an appropriate route inside AS1, based on the constraints, since it cannot reach AS3 directly. It updates the ERO by inserting the computed route segment. Finally, it stores the modified ERO in its path-state and forwards the Path message to the next abstract node in the ERO. R4 then removes its address from the ERO and forwards the Path message to R7. Similarly, R7 forwards the Path message to R8. And, finally, R8 is the LSP endpoint. The destination of the tunnel is reached because R8 belongs to AS3 that is specified as the destination for the LSP since the last subobject of the ERO is marked as only being used for routing purposes.

To support the transparency requirements for inter-domain LSPs, changes are required to the processing of the RRO object. This object may be part of the Path and Resv messages. It allows to record the addresses of the intermediate LSRs along the path of an LSP with,

[1]We talk about abstract nodes because it may represent a group of physical nodes or a single node.

optionally, the labels used along this path. As said previously, certain ASs may not want to let other ASs know their internal topology. Therefore, when using the RRO for interdomain LSPs, some information should be removed from the RRO before crossing AS boundaries. For this, we propose to allow AS boundary routers to summarize the path inside their AS as three elements : the IP address of the entry point, the AS number and the IP address of the exit point. This will allow us, as shown later, to support loop detection, route pinning and the establishment of disjoint LSPs.

To support the transparency requirements of ASs, we propose to modify the processing of the RRO object by the AS boundary routers (ASBRs). We do not change anything to the routers that are not ASBRs. To be able to hide topology information of an AS, the last router inside an AS, i.e. the exit point, needs to be able to determine the information that has to be aggregated. This may be done by parsing the RRO, in the reverse order, to determine for each subobject if it belongs to the current AS. This solution implies a non-negligible amount of processing. Therefore, it is interresting to mark inside the RRO the first router inside the current AS when inserting the corresponding subobject.. Hence, changes to the RRO processing are also required at the first router inside the AS, i.e. the entry point. When a Path/Resv message with RRO object enters an AS, the router stores its address with a flag, indicating that it is the entry point, inside the RRO. The exit point then removes the RRO subobjects starting after the last subobject marked with the “entry ASBR” flag. This set of subobjects is replaced by the current AS number and the exit point address. It follows that the AS topology information is summarized into the entry point inside the AS, the AS number and the address of the exit point. All routers along the LSP store the RRO once they added their address as in [ABG$^+$01]. An illustration of the processing of the RRO object is given in figure 3, where R3 is the ingress ASBR of AS1 and R7 is the egress ASBR of AS1.

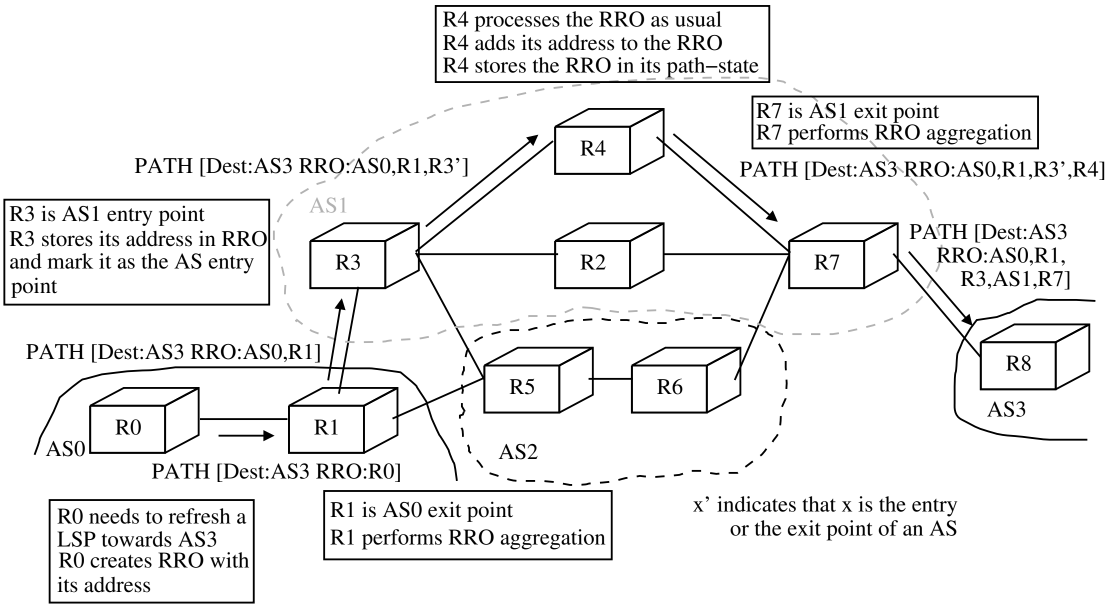

Figure 3: Processing of the RRO object

We notice that for a correct summarization of the RRO object, both the ingress and egress ASBR must support the modified processing of the RRO object. The insertion of the AS number inside the RRO object, when aggregation is performed, requires the definition of a new subobject for the RRO object. In addition to this new AS subobject, it might also be useful to change the IPv4 address and IPv6 address subobjects into IPv4 and IPv6 prefixes.

With our proposed solution, route pinning can be supported as follows. The RRO object stored in the path-state of an LSR is used to build the ERO of subsequent Path messages. Therefore, the RRO must be used in both Path and Resv messages to obtain a complete information about the path inside the AS and about the interdomain path. The source of the tunnel sets the ERO of the Path refresh messages to the RRO stored in its path-state. Once this Path message reaches the entrance of the next AS, the RRO of the path inside this AS is placed inside the ERO. This will force the Path refresh messages to follow the same path as the initial Path message inside the AS. This method is also valid for the downstream ASs because the ERO will be updated in a similar manner at the border of each AS. Figure 4 illustrates the processing of the RRO object for route pinning purposes.

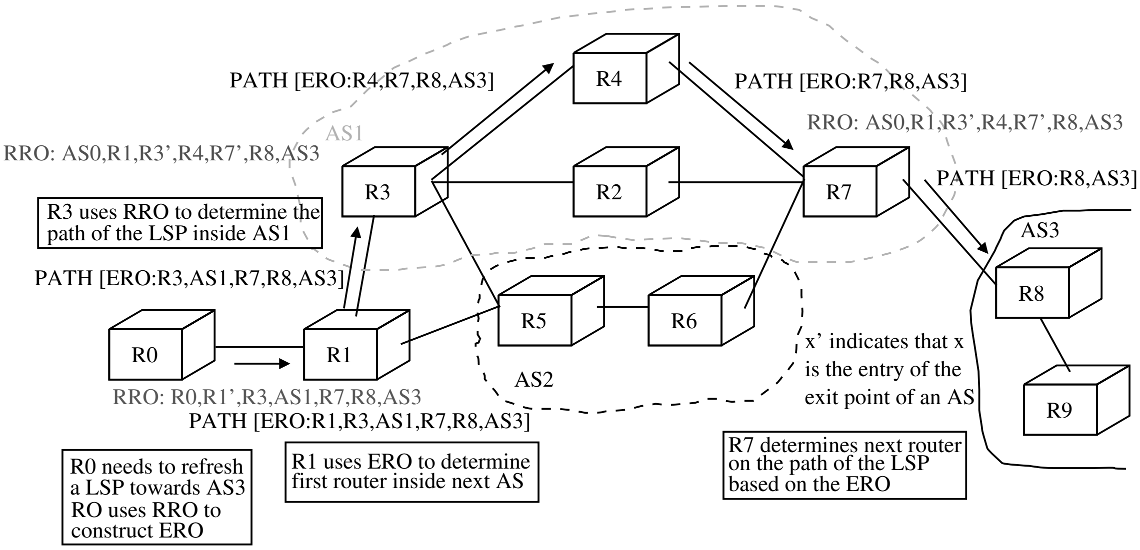

Figure 4: Establishment of an interdomain LSP with route pinning

Another utilization of the RRO object occurs in loop detection. This utilization is still possible with our proposed solution. The routers on the path of an LSP possess the RRO in their path-state with aggregated information concerning the path inside other ASs and detailed path inside the current AS. The loop detection will be performed at two different levels. The routers inside an AS will be responsible to detect intradomain loops by verifying that their address does not appear twice in the RRO. On the other hand, the ASBR will be responsible for the detection of interdomain loops. To detect such loops the ASBR verifies that its AS number is not already included inside the RRO object of a received RSVP message. This is possible since the summarization scheme that we propose for the RRO object has replaced all the IP addresses of a given AS by the entry/exit points and the AS number.

3 Protection of inter-domain LSPs

In this section, we discuss how the previous extensions can be used to provide protection of inter-domain links and protection of AS border routers. We will also refine the objects from [PGS+02] that are needed for SRLG protection and the required features of a Path Computation Server (PCS) and the communication protocol used with these PCSs. The solution we discuss requires the head-end LSR, of the LSP to protect, to indicate the required type of protection by using the appropriate flags inside the session attribute object of the path message. For example, it specifies that either link or node protection is required. Then, the downstream LSRs establish Detour LSPs or rely on Bypass tunnels, which may as well protect entire path segments, according to the protection policy of each AS.

We also consider the provision of SRLG protection. In order to indicate that SRLG protection is required, a flag inside the session attribute object or the fast reroute object [PGS+02] is required. Moreover, a flag is needed inside the RRO IP address/prefix subobjects to indicate if SRLG protection is provided.

A way to provide end-to-end protection of interdomain LSPs is given in appendix A

Before looking at the details of the proposed solution, it is useful to repeat the terminology defined in earlier documents [PGS+02, SH02] .

1. **Link protection** is provided by using a backup LSP that does not cross the link to be protected.
2. **Node protection** is provided by using a backup LSP that does not utilize neither the node to be protected nor the upstream link going to this node on the primary path[2].
3. **Point of Local Repair (PLR)** The head-end of a backup tunnel or a detour LSP[PGS+02].
4. **Path Switch LSR (PSL)** An LSR that is responsible for switching or replicating the traffic between the working path and the recovery path [SH02].
5. **Path Merge LSR (PML)** An LSR that is responsible for receiving the recovery path traffic, and either merging the traffic back onto the working path, or, if it is itself the destination, passing the traffic on to the higher layer protocols [SH02].
6. **Detour LSP** A Detour LSP provides one-to-one protection. A single LSP is established to protect another single LSP.
7. **Bypass tunnel** A Bypass tunnel provides many-to-one protection. It consists of a single tunnel that backups a set of protected LSPs by making use of label stacking[PGS+02].
8. **NHOP Bypass Tunnel** A backup tunnel which bypasses a single link of the LSP to be protected [PGS+02]. Such Bypass tunnel is used to protect the bypassed link.
9. **NNHOP Bypass Tunnel** A backup tunnel which bypasses a single node of the LSP to be protected [PGS+02]. NNHOP Bypass Tunnels protect against the avoided node failure and its upstream link.

Before considering in the next sections the various types of protection schemes in details, it is useful to summarize the main problems that arise when considering interdomain LSP protection compared to intradomain LSP protection. In both cases, each segment of the LSP to be protected will be protected through the utilization of a protection LSP that could be a Detour LSP or a Bypass tunnel established between the PLR and the PML. Of course, to be useful, this protection LSP needs to be disjoint from the segment of the primary LSP that it protects. Inside a single domain (organized as a single IGP area), each node on the path followed by a primary LSP knows the detailed path followed by this LSP and the complete topology of the domain distributed by a link-state IGP. Based on this information, the PLR can determine a path for a protection LSP that needs to be disjoint from a given segment of a primary LSP.

Across interdomain boundaries, the situation is more complex because the LSRs on the path of a primary LSP do not have such detailed information about the LSP and the interdomain network topology. As discussed in section 2, even with the utilization of the RRO and ERO

[2] An LSP that is node protected is protected against any link or node failure [PGS+02] except against the head-end and the tail-end LSR failures.

objects, a typical LSR will only know the list of transit AS, the IP addresses of the entry and exit ASBR inside each AS and the path followed by the LSP inside its own domain. Due to the incompleteness of the information about the path followed by a primary LSP inside an external domain, a LSR may have difficulties in locating the PML of interdomain LSPs. Since the PLR and the PML are located in different AS, we expect that the PLR will not be able to determine the address of the PML for interdomain LSPs. Instead the address of the PML will have to be determined by LSRs inside the downstream AS.

A second issue to be considered is the establishment of the protection LSPs. Inside its own domain, a LSR knows the entire network topology provided that the IGP is configured as a single area. However, the same LSR will only receive summarized information via BGP about the available interdomain paths. A single LSR will not usually be able to compute an explicit path for an interdomain backup LSP that needs to be disjoint from a segment of an existing LSP. The path to be followed by a backup interdomain LSP will be computed by several LSRs based on the information known by each LSR. This implies that a mechanism to communicate between LSRs will be required. For this, we rely on the mechanism described in [VIZ$^+$02]. The extensions required to [VIZ$^+$02] are left for further studies.

3.1 Link protection with a Detour LSP

In this section, we study the utilization of a Detour LSP to provide link protection for an interdomain link. Our reference environment is shown in figure 5. Assume that a primary LSP is being established between R11 inside AS1 and R23 inside AS2 and that the interdomain link between R13 and R21 needs to be protected by a Detour LSP. In this case, R13 will act as the PLR. To establish the Detour LSP, this LSR will need to obtain several informations as shown in figure 5.

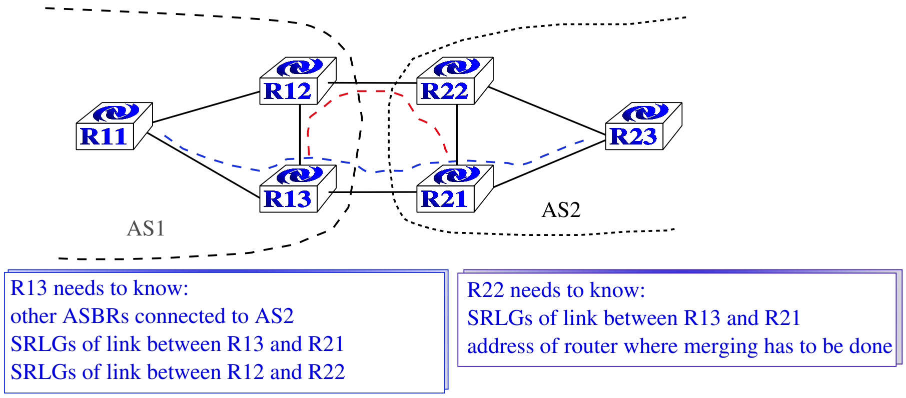

Figure 5: Link protection with Detour LSP

First, the PLR needs to determine which disjoint interdomain link can be used to reach the downstream AS on the path of the primary LSP. We assume that at least two disjoint links exist between each pair of AS on the path followed by a primary LSP[3]. To determine the usable interdomain links, the PLR can rely either on :

1. **manual configuration**. In this case, the PLR would know by configuration that link R12-R22 should be used to protect link R13-R21. Since a typical AS will usually only have a small number of external links towards a given AS, this can be a valid solution in practice.

[3]Otherwise, protecting an interdomain link would require the establishment of a Detour LSP through at least a third AS. We leave this case for further study and expect that in practice AS requiring interdomain LSP protection will be multiply connected.

2. **its BGP Routing Information Base (RIB)**. Since the PLR is an ASBR, it receives the routes selected by the other ASBR via iBGP. It could then parse its BGP RIB to determine the closest iBGP peer that advertised routes towards the downstream AS (or more precisely the routes with the downstream AS as the next-hop in their AS-Path attribute).

If the Detour LSP needs to be SRLG disjoint from the interdomain link to be protected, the PLR also needs to obtain information about the SRLG of the interdomain link. In the case of a manual configuration, the configuration can easily take the SRLG information into account. If the PLR relies on the information distributed by iBGP to determine the suitable interdomain links, then iBGP needs to distribute the information about the SRLG of each interdomain link. This could be done, for example, by configuring R12 to advertise with iBGP a /32 route towards R22 with an AS-Path of AS2 and to encode the SRLG of the interdomain link between R12 and R22 as a set of BGP extended communities. This route could be announced with the well known NO_EXPORT community to ensure that it is not redistributed across interdomain boundaries. The detailed encoding of the SRLG inside extended communities is outside the scope of this document.

Instead of distributing the SRLG information with iBGP, another solution would be to extend the communication protocol defined in [VIZ+02] to permit an ASBR to use it to request the SRLG of the interdomain links of another ASBR.

With this information, the PLR is able to determine the path of the Detour LSP inside its own AS. If the Detour LSP enters the downstream AS on the same entry ASBR as the primary LSP[4], then this ASBR can act as the PML. However, it can be expected that usually the Detour LSP will enter the downstream AS through a different entry ASBR than the entry ASBR of the primary LSP. In this case, the entry ASBR of the Detour LSP has to determine the address of the LSR where merging with the primary LSP has to be performed.

We expect that the Detour LSP will merge with the primary LSP inside the AS, but each AS may have its own policy concerning the location of the PML. Several solutions are possible. A first solution is to merge the Detour LSP with the primary LSP at the entry ASBR of the primary LSP (R21 in figure 5). In this case, the address of the PML is contained inside the summarized RRO of the primary LSP. This information can be specified by the PLR. A second solution is to merge the Detour LSP with the primary LSP at the exit ASBR of the primary LSP. In this case, the address of the PML may also be found in the summarized RRO of the primary LSP. A third solution is to merge the Detour LSP and the primary LSP at the closest LSR from the entry ASBR of the Detour LSP. In this, case, the entry ASBR of the Detour LSP needs to obtain the path of the primary LSP to determine the optimum location of the PML. This can be achieved with some communication between the entry ASBR of the Detour (R22) and the entry ASBR of the primary LSP (R21), known through the summarized RRO of the primary LSP. This communication may be performed through extensions to the path request and reply messages described in [VIZ+02]. These extensions are left for further study.

3.2 Node protection with a Detour LSP

In this section, we discuss the utilization of Detour LSPs to provide protection of an ASBR and its upstream link. We consider two distinct situations depending on whether the node to be protected is an exit or an entry ASBR for the primary LSP.

3.2.1 Node protection of the entry ASBR with a Detour LSP

Figure 6 shows a reference configuration and the information required at the different LSR to allow the establishment of a Detour LSP to provide protection of the entry ASBR.

To be able to establish the Detour LSP, the PLR (R13 in figure 6) needs to know the following information :

1. **the list of ASBRs connected to the downstream AS**. This list may be obtained as discussed section 3.1.

[4] This would be the case in figure 5 if there was a direct link between R12 and R21 with a different SRLG than link R13-R21.

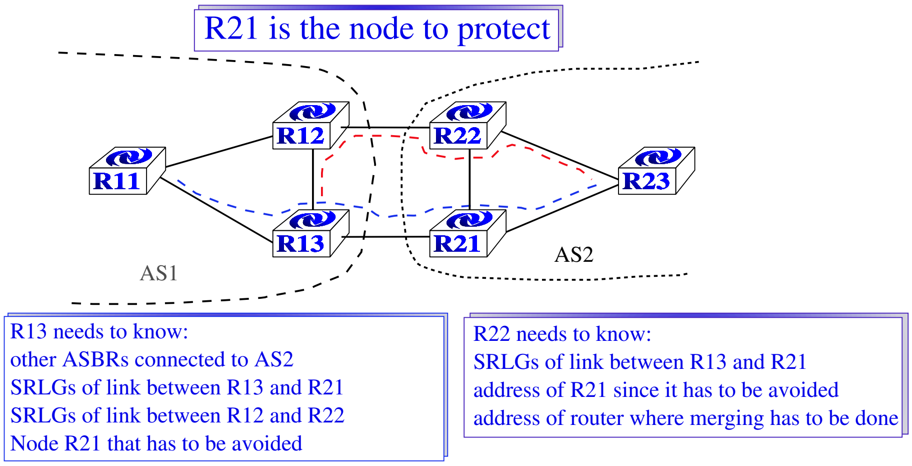

Figure 6: Node protection of the entry ASBR with a Detour LSP

2. **the SRLGs of the inter-domain link between R13 and R21**. These SRLGs can be manually configured.
3. **the SRLGs of the alternative inter-domain links**. This information can be obtained in discussed in section 3.1.
4. **the node to avoid with the Detour LSP** This node is known since it is stored inside the RRO.

Compared with the establishment of a Detour LSP to provide link protection, the situation is slightly different in the case of node protection. Here, the PML cannot obviously be the entry ASBR of the downstream AS (R21 in figure 6). The entry ASBR on the Detour LSP will thus need to determine the path towards the PML with the primary LSP. To compute this path, this ASBR needs to know the following information :

1. **the SRLGs of the inter-domain link between the PLR and the node to be protected** These SRLGs can be obtained through manual configuration or distributed with iBGP. It may also be carried inside the Path message of the Detour LSP. In section B.7, we show how the Detour object permits to specify SRLGs to avoid.
2. **the node to be avoided with the Detour LSP** The address of this node may be stored inside the Detour object defined in [PGS+02].
3. **the node where merging with the primary has to be done** The PML is obtained by communicating with a Path Computation Server (PCS) as explained in section 3.1.

3.2.2 Node protection of the exit ASBR with a Detour LSP

To protect a primary LSP from the failure of an exit ASBR, the situation is slightly more complex. Figure 7 shows a reference configuration and the information required at the different routers in order to provide this type of protection for the exit ASBR.

To protect an exit ASBR, the LSR upstream of the exit ASBR (R11 on figure 7) needs to be able to determine the path for the Detour LSP. For this, the PLR needs to find another ASBR inside its AS that is also connected with the downstream AS. This information can be obtained through manual configuration or distributed by iBGP as in the previous cases if the PLR receives routes via iBGP. If the PLR does not receive BGP routes, then it should communicate with another LSR to obtain the required information. This could be done via a dedicated PCS or by using the PCS protocol [VIZ+02] to contact the exit ASBR to be avoided.

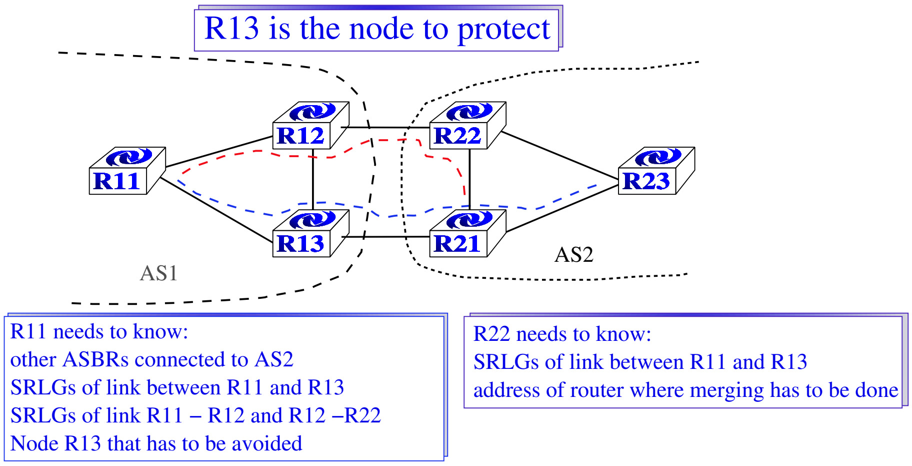

Figure 7: Node protection of the exit ASBR with a Detour LSP

If the Detour LSP also needs to be SRLG disjoint in addition to being node disjoint, then the PLR needs to obtain the SRLG information about the primary and the candidate Detour paths. The SRLGs of the link that leads to the exit ASBR, on the primary path is obtained from the conjunction of the information concerning the SRLGs flooded by the IGP and the RRO which enables to record the path of the working LSP. SRLGs of links to join alternative ASBRs connected to the downstream AS are also known through the IGP. And, the SRLGs of the alternative interdomain links to reach the downstream AS are either known by all ASBRs through manual configuration or by all BGP routers inside the AS through iBGP. Therefore, if the PLR is a BGP router it may possess the required SRLG information. Otherwise, communication with a PCS is required to get the SRLGs of the inter-domain links. Finally, the PLR needs to know the address of the node to be avoided. This information is stored inside the Detour object in the Path message of the Detour LSP.

The entry ASBR of the Detour LSP ($R22$ on figure 7) has to know the following information in order to complete the establishment of this LSP.

1. **the SRLGs of the link leading to the node to protect** These SRLGs are not available through the IGP and BGP to this router. Therefore, they should be carried inside the Path message of the Detour LSP. This is not currently possible with the Detour object defined in [PGS+02]. It follows that either extensions to this Detour object are required or the new Avoid Route Object (ARO)[5], specified in section B.3, should be used to store the SRLGs that should be avoided by the Detour.
2. **the address of the PML** If the PML is the entry ASBR on the primary LSP, then this address is known by at least the node to be protected. The PLR may known this information from the summarized RRO and place it inside the path message used to establish the Detour LSP. A PCS may also be used to obtain the address of the PML and the path to reach this PML.

3.3 Link protection with use of a Bypass LSP

Instead of protecting segments of a primary LSP with a dedicated LSP, in this section, we look at the possibility to protect several LSPs with a single Bypass tunnel. This kind of protection can be provided as soon as the LSPs to protect share a common PLR and downstream node.

[5]The ARO object has an additionnal use in the establishment of end-to-end disjoint LSPs. It permits to store the path of an LSP that has to be avoided.

In this section, we will first look at the way a Bypass tunnel is selected in order to provide protection for an interdomain link. Then, we will look at the establishment of a Bypass tunnel that is used to protect several primary LSPs.

When the protection of an interdomain link is considered, the PLR is the exit ASBR and, the common downstream router belongs to the downstream AS. Therefore, it is not easy to determine if different working LSPs can be protected by the same Bypass tunnel when they do not have a common entry point inside the downstream AS, since the path of these LSPs inside other ASs is not known by the PLR.

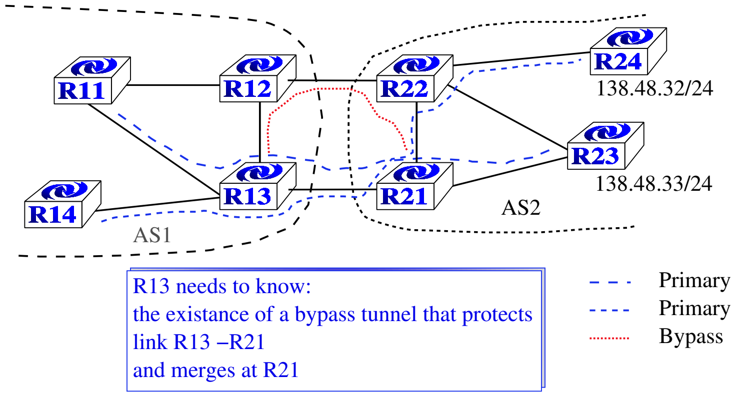

Figure 8: Link protection with Bypass tunnel

In the following examples (figures 8, 9,10), we assume that the primary LSP ("Primary 1") is already established as well as the Bypass tunnel that protects the interdomain link (R13-R21). In figure 8, a new LSP toward network 138.48.32/24 is established. This new LSP is called "Primary 2". Link protection needs to be provided for this LSP. Therefore, the PLR ($R13$) has to know that a Bypass tunnel toward the entry ASBR ($R21$) of the primary LSP inside the downstream AS (AS2) exists and that it protects against a failure of the interdomain link R13-R21. Additionally, the PLR ($R13$) has to be able to determine if there is enough bandwidth on the Bypass tunnel to protect the new LSP, if bandwidth protection is required.

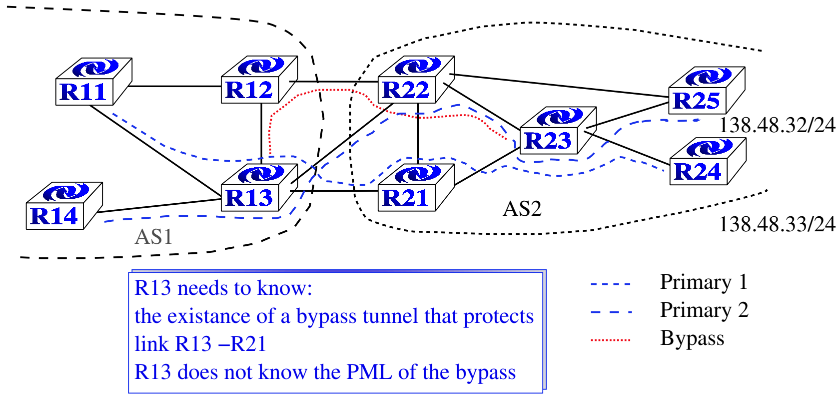

Figure 9: PML identification problem

When Bypass tunnels protecting the required interdomain link exist but do not terminate at an ASBR, it is more complex to determine if the Bypass tunnel is appropriate to protect the

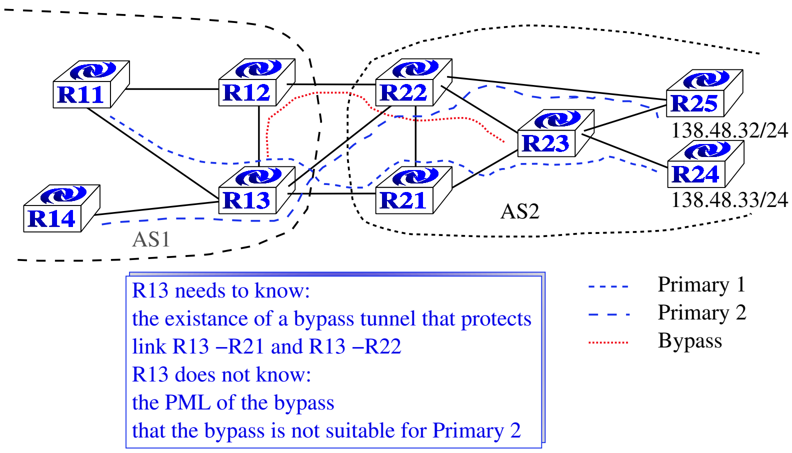

Figure 10: PML identification problem

new LSP being established. In this case, it is not possible for the PLR to know whether the destinations of established Bypass tunnels are on the path of the primary LSP.

Figures 9 and 10 illustrate the difficulty of choosing an adequate Bypass when the PML is not an ASBR. Among the candidate Bypass tunnels selected by the PLR (here the exit ASBR), some may not be adequate for the protection of a given LSP as shown on figure 10, where the PML of the existing Bypass is not on the path of "primary 2".

In figure 11, the required information and communication mechanisms between ASBR are exposed. In order to determine if the candidate Bypass tunnels for "Primary 2", known by the PLR (R13) are suitable for the protection of this LSP, the PLR needs to communicate with the entry ASBRs of the candidate Bypasses inside the downstream AS. These ASBRs, respectively, have to contact the entry ASBR ($R22$) of the LSP to protect, to obtain the path of the primary LSP. With this information, they will be able to determine the Bypass tunnels that cross the primary LSP and are usable for the protection of the working LSP; they will communicate the identifiers of these Bypasses to the PLR. When the first answer concerning an appropriate Bypass tunnel arrives, the PLR chooses this Bypass. If no positive answer is received, the PLR will have to establish a new Bypass tunnel as described below.

The question of the establishment of Bypass tunnels has now to be approached. These tunnels may be manually pre-configured but it is also interesting to be able to establish these LSPs dynamically. In this case, when an LSP with link protection required is established and no Bypass LSP is available for this LSP, a new Bypass can be established.

The Fast Reroute object defined in [PGS+02] is still useful in the set up of an interdomain Bypass tunnel. When a Bypass tunnel may be used, the "facility backup desired" flag is set inside the fast reroute object. In addition, a similar object to the Detour object is required in order to indicate the link that has to be avoided by the Bypass tunnel. This object should have another value for the C-type field to distinguish between a Bypass and a Detour LSP. If SRLG disjointness is required, the SRLGs of interdomain links may be obtained as exposed in section 3.1.

When a Bypass tunnel that protects an interdomain link needs to be established, the PLR and the entry ASBR of the Bypass tunnel inside the downstream AS have to get at least the same information as these routers need to have in order to establish a Detour that protects the same interdomain link (see figure 5). In addition, the PLR of a Bypass tunnel has to determine the bandwidth required by the Bypass.

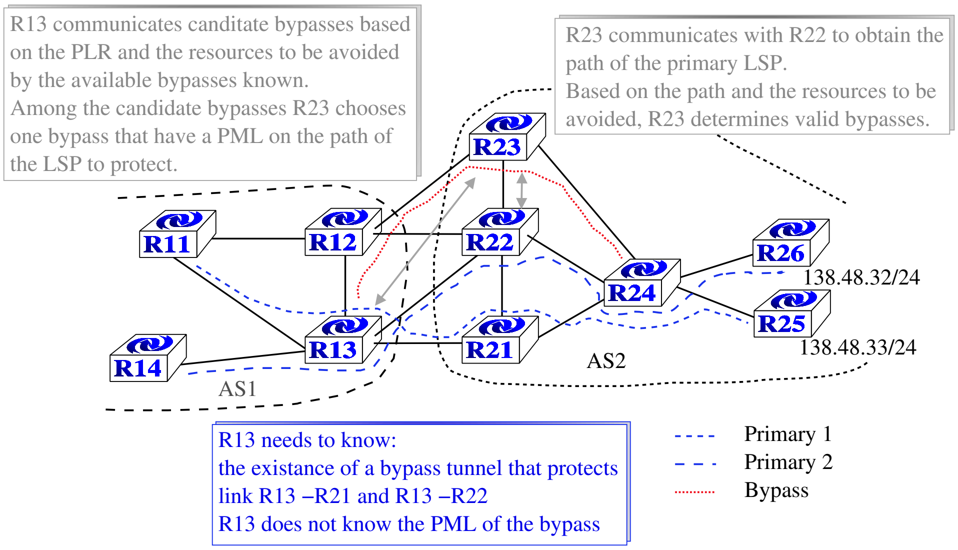

Figure 11: Choosing adequate Bypass

3.4 Node protection with use of a Bypass LSP

In this section, we first interest ourselves in the protection of an exit ASBR on a working path through a Bypass tunnel. Then, we look at the protection of entry ASBRs with Bypass tunnels. We first suppose that the required Bypass tunnel already exists and the PLR needs to determine the Bypass that it can use. Then, we suppose that there are no appropriate Bypass already established. In this case, we look at the establishment of Bypass tunnels that protect against ASBRs failures.

A Bypass tunnel can only be used by working paths that share the same PLR and PML. The PML may be any router inside the downstream AS but as explained in the previous section, it is easier to determine the candidate Bypass tunnels when the PML is either the entry or the exit point inside the downstream AS since this information is known by looking into the aggregated RRO. In figure 12, upon the establishment of the second primary LSP ("Primary 2"), the PLR ($R11$) has to know the existance of a Bypass that is a good candidate for the protection of the exit ASBR ($R13$) as well as SRLG disjoint from link R11-R13 and that it merges on the path of "Primary 2" LSP. If the merging router isn't an ASBR then inter-LSR communications as the onces shown on figure 11 should take place in order to determine a suitable Bypass. However, here the LSP that initiates such communication may not be an ASBR since we consider the protection of an exit ASBR.

When no candidate Bypass tunnel fits the requirements, a new Bypass tunnel has to be established. This requires that the PLR ($R11$) obtains the same kind of information as listed on figure 7. The information required at the entry ASBR ($R22$) for the establishment of the Bypass tunnel is also represented on figure 7. And, as in section 3.3, the PLR ($R11$) additionally has to determine the bandwidth to be allocated to the Bypass tunnel. The entry ASBR of the Bypass tunnel in the downstream AS ($R22$) obtains the SRLGs of the link that leads to the node to protect through the Bypass object and gest the address of the PML in the same way as described in section 3.2.

Concerning the protection of an entry ASBR with a Bypass tunnel, no new mechanism has to be introduced. The PLR needs to know the existence of a Bypass tunnel that protects the right node and eventually the SRLG of the link leading to that node. In order to identify if the candidate Bypass tunnels selected by the PLR merge on the path of the primary LSP and are disjoint from the primary LSP, the PLR communicates the identifiers of the selected tunnels

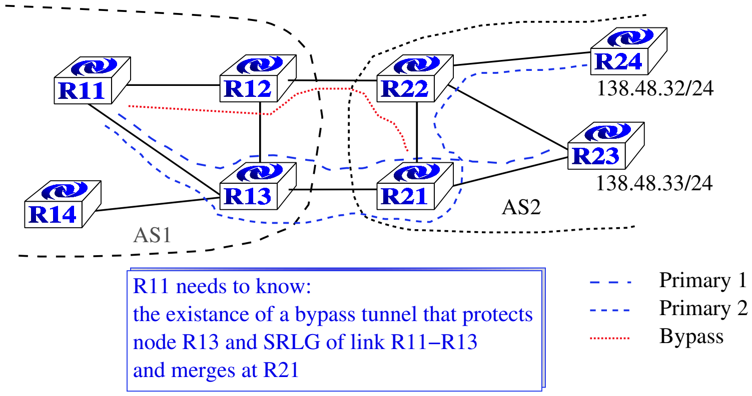

Figure 12: Bypass node protection

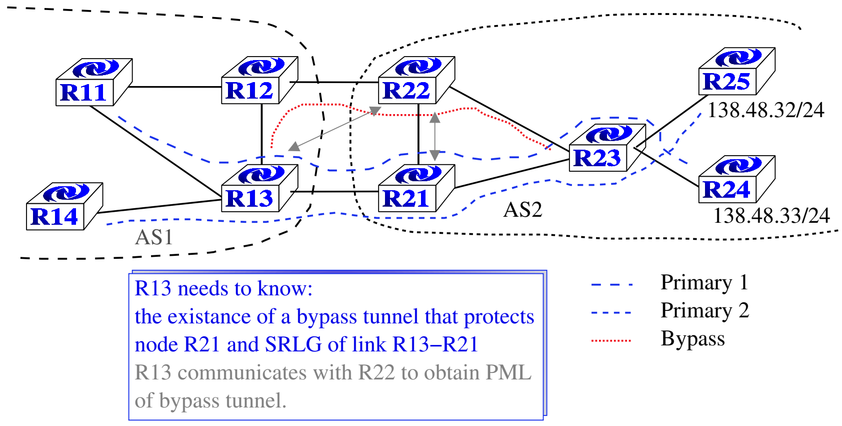

Figure 13: Bypass node protection

and the resources to be avoided by these tunnels to their entry point inside the next AS. These ASBRs determine if these Bypass tunnels are appropriate for the protection of the entry ASBR of the working path by communicating with the entry ASBR of the LSP to protect in order to obtain the path and the SRLGs of this working path. An example of the selection of a Bypass tunnel suitable for the protection of “Primary 2” is illustrated on figure 13.

In case no appropriate Bypass tunnel is available for the protection of the entry ASBR and its upstream link, a new Bypass tunnel needs to be established according to the mechanisms previously exposed. That is, the PLR needs to create a Path message with a Bypass object containing the downstream entry ASBR and the SRLG of the link to be avoided by the Bypass tunnel. The Bypass tunnel is then established in the same way as a Detour LSP protecting that same entry ASBR.

4 Security considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP-TE protocol [ABG+01] remain relevant.

5 Conclusion

In this document, we have proposed a method to establish interdomain LSPs that fulfills the transparency requirements of the interdomain environment while still supporting route pinning and the establishment of secondary LSPs which can be used for load balancing or to provide path protection in case of link or node failures. Our solution requires the definition of a few new objects and subobjects. An important advantage of our solution is that only AS border LSRs need to be modified to support the proposed extensions to RSVP-TE ; the LSRs inside an AS can still rely on the current RSVP-TE implementation.

Then, we looked at the establishment of Detour LSPs and Bypass tunnels for the protection of these interdomain working LSPs. More specifically, we payed attention to the protection of interdomain links and AS Border Routers, relying on existing solutions for the protection of intradomain links and core routers. The elaborated solution enables to takes into account the protection of SRLGs in the establishment of Detour LSPs and in the use or establishment of Bypass tunnels.

Finally, in appendix, we look at an other application of the mechanisms developed in the first two sections of the draft. That is the possibility to provide end-to-end protection of interdomain links as well as being able to establish disjoint LSPs to load balance the traffic on these LSPs.

These features require extensions to existing RSVP-TE objects by adding new subobjects and new flags. Some objects and flags from [PGS+02] are used and a new object is introduced. These modifications need only to be supported by ASBRs and the head-end LSRs who are now able to use these interdomain LSP establishment and protection features. The objects needed for local protection through Detour LSPs and Bypass tunnels need to be supported by all PLR on the path of the LSP to protect. This service however requires the support of the same objects by all PLR on the path of an intradomain LSP.

6 Acknowledgements

This work was supported by the European Commission within the IST ATRIUM project. We would like to thank Stefaan De Cnodder, Jean-Philippe Vasseur and Sebastien Tandel for their useful comments.

Authors' Addresses

Cristel Pelsser
Infonet group (FUNDP)
Rue Grandgagnage 21, B-5000 Namur, Belgium
Email: cpe@info.fundp.ac.be
URL : http://www.info.fundp.ac.be/ cpe

Olivier Bonaventure
Universite catholique de Louvain (UCL)
Email: Bonaventure@info.ucl.ac.be
URL : http://www.info.ucl.ac.be/people/OBO/

A Establishment of disjoint LSP

Another issue to consider is the establishment of a disjoint LSP either for backup or load balancing purposes. In this section, we show how it is possible to establish a new LSP that is path-disjoint from an existing LSP while still meeting the transparency requirements concerning internal AS topologies.

Inside a single domain organized as a single IGP area, the establishment of a path-disjoint backup LSP is simple. The source LSR can determine the entire path of the existing LSP thanks to the RRO object and use this information with the topology distributed by the IGP to select a new path that is disjoint from the existing one. When the AS is organised in several IGP areas, the situation is more complex since the source LSR does not know the detailed topology of the entire network. However, the source LSR can use the RRO object to determine the entire path of the existing LSP and to specify a list of the IP addresses to avoid for the new LSP as constraints in the RSVP Path message [VIZ+02]. When considering interdomain LSPs, this solution is not applicable since the source LSR will only receive a summarized RRO object.

To establish a backup path that is path disjoint from a primary path, we propose to use the new Avoid Route Object (ARO). It is used to specify the path of the existing LSP from which the new backup LSP should be path disjoint. It supports the following subobjects : IPv4 and IPv6 address prefixes as well as AS numbers. In the ARO, an AS number subobject is always preceded by the entry point address and followed by an exit point address.

When establishing a path disjoint backup interdomain LSP, an LSR can rely on the RRO object stored in its path state to determine the path of the primary LSP inside the current AS. Based on this information, the source LSR may compute a disjoint path. Two types of disjoint path can be envisaged. First, the path of the LSP could be disjoint when considering the intermediate AS. In this case, the source LSR needs to create an ERO object that is completely different from the ERO object of the LSP to protect. A second type of disjoint path is a path that passes through the same intermediate AS as the LSP to protect but through different routers inside these AS. We consider the latter type of disjoint paths in the remaining of this section. Within this second type of disjoint path, it is also possible to provide either end-to-end or segment protection.

To establish a disjoint path with the same AS path as the primary, the source LSR can proceed as follows. Since it knows from the stored RRO object the IP address of the entry point in the first downstream AS, it may easily choose another IP address to enter the downstream AS (e.g. based on its BGP table). The path inside the first AS will thus automatically be disjoint from the existing LSP. Once the Path message reaches the ASBR of the first downstream AS, this ASBR will have to compute a path inside this AS that will be disjoint from the path followed by the existing LSP. This ASBR does not have itself enough information to compute this new path. Instead, it will ask the ASBR that is the entry point for the primary LSP to compute a disjoint path. The address of this ASBR can be easily obtained from the ARO object that contains the summarised RRO of the primary LSP. The computation of such a disjoint path requires extensions to RSVP similar to those proposed in [VIZ+02] and a way to specify in the path computation message the identification of the primary LSP. Dedicated path computation servers as in [VIZ+02] may be envisaged for the computation of disjoint LSPs taking this functionality away from the ASBRs.

The primary LSP has to be identified in the Path message of the backup LSP such that the ingress ASBR of the primary path may identify the primary LSP and compute a disjoint path based on the RRO stored in the path-state of the primary LSP. In [ABG+01], a traffic engineered tunnel is identified by the session and the sender template objects. More precisely, the tunnel end point address, the tunnel ID, the extended tunnel ID and the tunnel sender address identify a tunnel while the LSP ID serves to reroute an established tunnel or to modify the bandwidth reserved for the tunnel. If we consider that establishing a backup path consists of rerouting the primary path, the identifier of the backup LSP is the same as the identifier of the primary path and this identifier is carried in the Path message of the backup LSP. No new object is required. Only the LSP ID changes. If both paths share a common link, which should not occur in this case, the resources will only be reserved once, when the Shared Explicit flag of the session attribute object is set[6]. The source of the tunnel has to refresh both paths such that they are both present in the network[7]. Figure 14 illustrates the establishment of a backup LSP, where Avoid LSP Identifier (ALSPId) denotes the identifier of the LSP to avoid, i.e. the identifier of the primary LSP composed of the tunnel end point address, the tunnel ID, the extended tunnel ID and the tunnel sender address. In case the disjoint LSP is established for load balancing purposes, we may not want to share resources between the LSPs. Therefore, different tunnel IDs are attributed to the primary and the disjoint LSP. And, it is necessary to carry a new object, the ALSPId object, that stores the identifier of the primary LSP, in the disjoint LSP establishment.

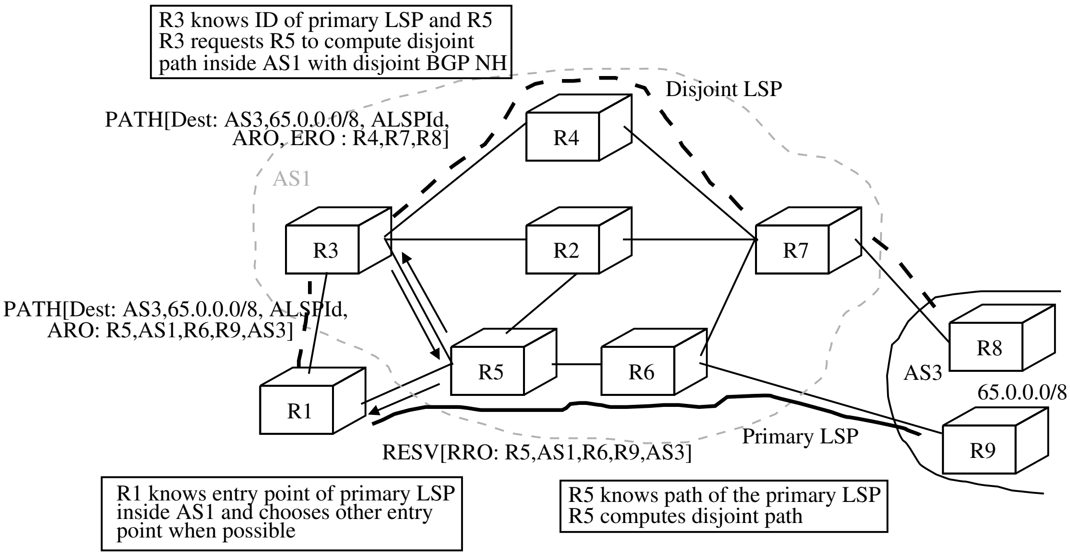

Figure 14: Inter-domain disjoint LSP

When an AS is composed of multiple areas, an ASBR may not be able to compute the path of an LSP through the whole AS. Therefore, it may be necessary to store the aggregated information concerning the primary path inside the path message of the disjoint LSP. We propose that the ingress ASBR of a primary path communicates the RRO of the primary path to the ingress ASBR of the backup path. The ingress ASBR of the backup path then stores the aggregated RRO object of the primary path into the Avoid Route Object (ARO). The computation of the backup path for the area is then either performed by the primary or the backup ingress ASBR. Once the Path message reaches an ABR, this ABR computes the path of the backup LSP for the next area based on the ARO or based on interarea techniques such as [VIZ+02]. When the Path message finally reaches the border of the AS, the information concerning the topology of the AS must be removed from the ARO in the same manner as aggregation of the RRO object is performed. Based on the ARO, each router inside an AS, in particular ingress ASBR, knows the route to avoid inside the AS as well as the BGP NH to avoid. Figure 15 illustrates the use of the ARO for the establishment of a path disjoint LSP.

[6] This should not happen if the paths are disjoint

[7] The source of the tunnel may stop refreshing the primary path when the backup path is in use if restoration is non revertive. The source of the tunnel may then establish a new path as backup of the used LSP.

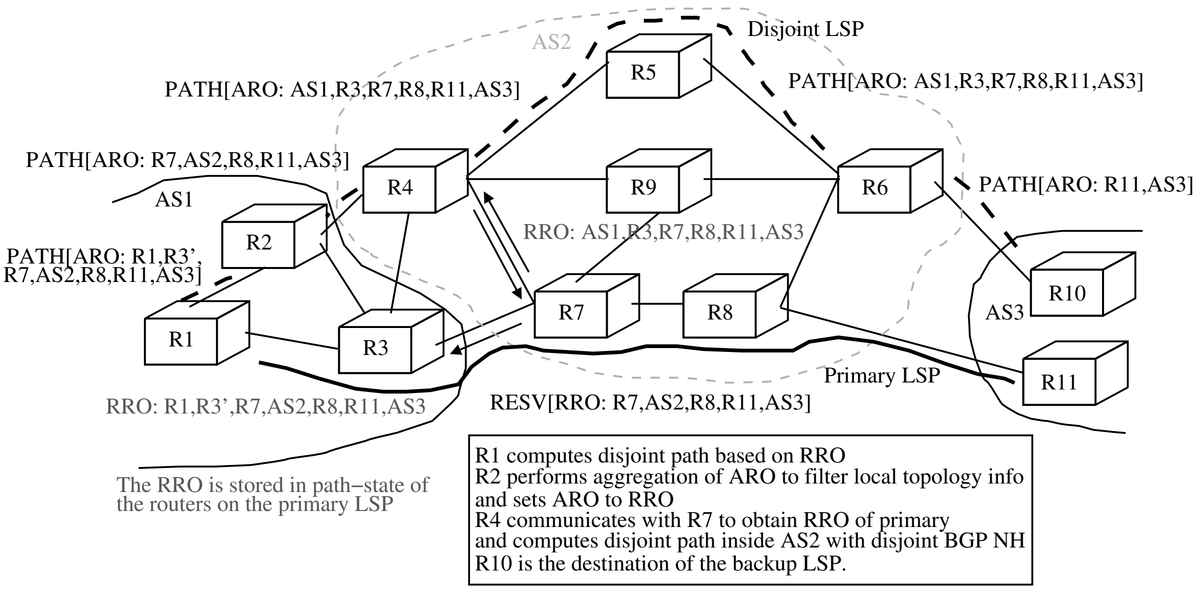

Figure 15: Role of the ARO object

B Inter-domain tunnels related message formats

Some new objects are defined for the support of inter-domain Traffic Engineered LSPs and their restoration. And, extensions to some objects defined in [ABG+01] are also introduced.

B.1 Explicit Route Object

The `EXPLICIT_ROUTE` object (ERO) has the following format:

```
Class = 20, C_Type = 1

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (Subobjects)                          //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This is unchanged from [ABG+01]. The ERO may be present in Path messages. As in [ABG+01], only the first ERO is meaningful when a Path message contains multiple EROs. Subsequent EROs MAY be ignored and SHOULD NOT be propagated.

B.1.1 Subobjects

The ERO is composed of a serie of variable length objects called subobjects. Each subobject has the form:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-------------//----------------+
|L|    Type     |     Length    | (Subobject contents)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-------------//----------------+
```

```
where


   L

      The L bit is an attribute of the subobject.  The L bit is set
      if the subobject represents a loose hop in the explicit route.
      If the bit is not set, the subobject represents a strict hop
      in the explicit route.

   Type

      The Type indicates the type of contents of the subobject. The
      values defined in \cite{Awduche:aug2001} are 1 (IPv4 prefix),
      2 (IPv6 prefix) and 32 (autonomous system number).

   Length

      The Length contains the total length of the subobject in
      bytes, including the L, Type and Length fields.  The Length
      MUST be at least 4, and MUST be a multiple of 4.
```

The syntax of the IPv4 prefix is as follows:

```
 0
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|    Type     |     Length    | IPv4 address (4 bytes)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv4 address (continued)      | Prefix Length |      Flags    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   L

      The L bit is an attribute of the subobject.  The L bit is set
      if the subobject represents a loose hop in the explicit route.
      If the bit is not set, the subobject represents a strict hop in
      the explicit route.

   Type

      0x01  IPv4 address

   Length

      The Length contains the total length of the subobject in bytes,
      including the Type and Length fields.  The Length is always 8.

   IPv4 address

      An IPv4 address.  This address is treated as a prefix based on
      the prefix length value below.  Bits beyond the prefix are
      ignored on receipt and SHOULD be set to zero on transmission.

   Prefix length

      Length in bits of the IPv4 prefix

   Flags

      TBD loose destination

          Indicates that the destination of the LSP may be any
          router inside this abstract node.

      TBD used for routing purposes

          Indicates that the prefix is used for routing purposes.
          The establishment of the LSP is stopped once the Path
          message enters the AS to which this prefix belongs.
```

The contents of an IPv4 prefix subobject are a 4-octet IPv4 address, a 1-octet prefix length, and a 1-octet flags field. The abstract node represented by this subobject is the set of nodes that have an IP address which lies within this prefix. Note that a prefix length of 32 indicates a single IPv4 node.

The syntax of the IPv6 prefix is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|L|    Type     |     Length    | IPv6 address (16 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv6 address (continued)      | Prefix Length |      Flags    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   L

      The L bit is an attribute of the subobject.  The L bit is set
      if the subobject represents a loose hop in the explicit route.
      If the bit is not set, the subobject represents a strict hop in
      the explicit route.

   Type

      0x02  IPv6 address

   Length

      The Length contains the total length of the subobject in bytes,
      including the Type and Length fields.  The Length is always 20.

   IPv6 address

      An IPv6 address.  This address is treated as a prefix based on
      the prefix length value below.  Bits beyond the prefix are
      ignored on receipt and SHOULD be set to zero on transmission.

   Prefix Length

      Length in bits of the IPv6 prefix.

   Flags

      TBD loose destination

          Indicates that the destination of the LSP may be any
          router inside this abstract node.

      TBD used for routing purposes

          Indicates that the prefix is used for routing purposes.
          The establishment of the LSP is stopped once the Path
          message enters the AS to which this prefix belongs.
```

The contents of an IPv6 prefix subobject are a 16-octet IPv6 address, a 1-octet prefix length, and a 1-octet flags field. The abstract node represented by this subobject is the set of nodes that have an IP address which lies within this prefix. Note that a prefix length of 128 indicates a single IPv6 node.

The syntax of the AS number subobject is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-------------------------------+
|L|    Type     |     Length    | AS number (2 bytes)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-------------------------------+

   L

      The L bit is an attribute of the subobject.  The L bit is set
      if the subobject represents a loose hop in the explicit route.
      If the bit is not set, the subobject represents a strict hop in
      the explicit route.

   Type

      0x20  AS number

   Length

      The Length contains the total length of the subobject in bytes,
      including the Type and Length fields.  The Length is always 4.

   AS number

      An AS number.
```

The contents of an Autonomous System (AS) number subobject are a 2- octet AS number. The abstract node represented by this subobject is the set of nodes belonging to the autonomous system.

Changes are required to the ERO subobjects syntax. The previous resvd field of the IPv4 and IPv6 prefix subobjects has become a flag field. The "loose destination" flag is used to indicate that the destination of the LSP is the first router inside the prefix crossed by the Path message. The other flag indicates that the prefix is used for routing purposes. In that case, the destination of the LSP may be any router inside the AS to which the prefix belongs. In case the "used for routing purposes" flag is used in a prefix subobject, this subobject MUST be preceded by an AS number subobject. This AS number subobject is used to determine if the AS destination is reached before removing the last subobject of the ERO. This last subobject is a prefix and carries the "used for routing purposes" flag. More precisions about the processing of the ERO due to the presence of these flags are given in SectionB.1.2.

These changes affect the handling of the ERO at the border routers of an AS. Additional optional changes are necessary for the support of the "loose destination" flag at routers that may be the end point of interdomain tunnels established with a prefix destination.

B.1.2 Handling of the ERO

The "loose destination" and "used for routing purposes" flags are exclusive. If both flags are present only the "used for routing purposes" flag is taken into account by a router. An IPv4 or IPv6 prefix subobject with these flags set MUST always be the last subobject inside the ERO. A prefix subobject (IPv4 or IPv6) with flag "used for routing purposes" set MUST be preceded by an AS number subobject to ensure that the AS destination is reached before stopping the LSP's establishement.

When a router encounters an IP prefix subobject with the "loose destination" flag set, during the processing of the ERO, it stops forwarding the Path message if it belongs to the prefix. Otherwise, the router updates the ERO with new subobjects in order to join the prefix.

A router that has to process an AS number subobject either removes the subobject if it belongs to the AS or adds new subobjects, that will be used for joining the next AS, based on the Path message's destination if necessary. Once an AS number subobject is removed, the following subobject to process may be an IP prefix with the "used for routing purposes" flag set. In that case, the Path message is terminated and a Resv message is generated since the destination of the LSP has been reached.

B.1.3 Non-support of the ERO or of its subobjects

The Class-Num of the `EXPLICIT_ROUTE` object is of the form `0bbbbbbb` where b represents a bit. An RSVP router that does not recognize the `EXPLICIT_ROUTE` object sends a PathErr with the error code "Unknown Object Class" toward the sender. This causes the path setup to fail. The sender should notify management that an LSP cannot be established and possibly take action to continue the reservation without the `EXPLICIT_ROUTE` or via a different explicit route.

As in [ABG+01], a node which encounters an unrecognized subobject during its normal ERO processing sends a PathErr with the error code "Routing Error" and error value of "Bad Explicit Route Object" toward the sender. The `EXPLICIT_ROUTE` object is included, truncated (on the left) to the offending subobject. The presence of an unrecognized subobject which is not encountered in a node's ERO processing SHOULD be ignored. It is passed forward along with the rest of the remaining ERO stack.

The modifications brought to the ERO subobjects are backward compatible with [ABG+01]. We added two flags to the IPv4 and IPv6 prefix subobjects.

A node that has to process a subobject with the "loose destination" flag, should stop forwarding the Path message and generate a Resv message if it belongs to the abstract node. If it does not support the flag and belongs to the abstract node, it will forward the Path message to another node on the way to the destination of the Path message. In this case, the Path message will not be ended at the entrance of the prefix destination.

The "used for routing purposes" flag indicates that the prefix subobject is only used for routing. In case this flag is not supported, the path message will be forwarded on the path to join the prefix. It should be ended inside this prefix depending on the destination of the Path message (i.e. the tunnel end point address inside the Session Object).

All nodes should forward the flags with the subobjects. They must not set the flags field to zero on transmission. This is a modification from [ABG+01]. In case this is no enforced, the setting of the flags back to zero leads to a similar situation as described in the previous paragraphs where the flags are not supported by the node that needs to deal with it.

B.2 Record Route Object

The `RECORD_ROUTE` object (RRO) has the same format as in [ABG+01].

```
Class = 21, C_Type = 1

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (Subobjects)                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The RRO can be present in both RSVP Path and Resv messages. If a Path message contains multiple RROs, only the first RRO is meaningful. Subsequent RROs SHOULD be ignored and SHOULD NOT be propagated. Similarly, if in a Resv message multiple RROs are encountered following a `FILTER_SPEC` before another `FILTER_SPEC` is encountered, only the first RRO is meaningful. Subsequent RROs SHOULD be ignored and SHOULD NOT be propagated.

B.2.1 subobjects

Two additional subobjects to the RRO are required. These are the Autonomous System (AS) number and the Shared Risk Link Group (SRLG) number subobjects. Therefore, two new types of subobjects have to be assigned. Furthermore, the IPv4 prefix and the IPv6 prefix subobjects are a generalization of the IPv4 and IPv6 address subobjects defined in [ABG+01]. A new flag, called the "entry ASBR" flag, is added inside the IPv4 and IPv6 address subobjects.

The IPv4 and IPv6 prefix subobjects are identical to the IPv4 and IPv6 address subobjects defined in [ABG+01] except that the prefix length field is not set to 32 and 128, respectively. This field may take any value in the interval 0-32 for the IPv4 prefix subobject and between 0-128 for the IPv6 prefix subobject. These subobjects are generalized in regards to future uses concerning the aggregation of information obtained by means of the RRO.

The Label subobject is unchanged from [ABG+01].

The syntax of the IPv4 address/prefix subobject is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     Length    | IPv4 address (4 bytes)        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv4 address (continued)      | Prefix Length |      Flags    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   Type

      0x01  IPv4 address

   Length

      The Length contains the total length of the subobject in bytes,
      including the Type and Length fields.  The Length is always 8.

   IPv4 address

      An IPv4 address.  This address is treated as a prefix based on
      the prefix length value below.  Bits beyond the prefix are
      ignored on receipt and SHOULD be set to zero on transmission.

   Prefix length

      Length in bits of the IPv4 prefix.

   Flags

      0x01  Local or segment protection available

            If prefix length is 32:
            Indicates that the link downstream of this node is
            protected via a local repair mechanism.  This flag can
            only be set if the Local protection flag was set in the
            SESSION_ATTRIBUTE object of the corresponding Path
            message.

            If prefix length < 32:
            Indicates that the segment of the LSP inside the
            abstract node is protected against link failures.  This
            flag can only be set if the segment protection flag was
```

```
              set in the SESSION_ATTRIBUTE object of the
              corresponding Path message.

        0x02  Local or segment protection in use

              If prefix length is 32:
              Indicates that a local repair mechanism is in use to
              maintain this tunnel (usually in the face of an outage
              of the link it was previously routed over).

              If prefix length < 32:
              Indicates that a local or segment repair mechanism is
              in use to maintain this tunnel (usually in the face of
              an outage of the link it was previously routed over).

        0x04  Bandwidth protection

              The Point of Local Repair (PLR) will set this when the
              protected LSP has a backup path which provides the
              desired bandwidth, which is that in the FAST_REROUTE
              object or the bandwidth of the protected LSP, if no
              FAST_REROUTE object was included.  The PLR may set this
              whenever the desired bandwidth is guaranteed; the PLR
              MUST set this flag when the desired bandwidth is
              guaranteed and the "bandwidth protection desired" flag
              was set in the SESSION_ATTRIBUTE object.

        0x08  Node protection

               When set, this indicates that the PLR has a backup path
               providing protection against link and node failures on
               the corresponding path section. In case the PLR could only
               setup a link-protection backup path, the "Local protection
               available" or the "Segment protection available" bit will
               be set but the "Node protection" bit will be cleared.

        TBD   SRLG protection

              When set, this indicates that the PLR has a backup path
              providing protection against SRLG failures on the
              corresponding path section.

        TBD   Entry ASBR

              Indicates that this subobject represents a router that
              is the entry point inside the current AS.
```

The flags "Bandwidth protection" and "Node protection" are introduced in draft [PGS+02]. In that draft, two objects (`FAST_REROUTE` and `DETOUR`), a few flags and the `MAX_PROTECTED_BANDWIDTH RRO` subobject are introduced. The "Local protection available" and "Local protection in use" flags are extended here to "Local or segment protection available" and "Local or segment protection in use" in order to indicate if link protection is available or in use on a path segment. This is useful when aggregation of the RRO into IP prefixes is performed for topology hiding purposes. The "SRLG protection" flag is added to indicate if a backup path that protects against SRLG failures is available. Moreover, the "Entry ASBR" flag is introduced here to be able to perform aggregation of the RRO at the border of an AS.

The syntax of the IPv6 address/prefix subobject is as follows:

```
 0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |      Type     |     Length    | IPv6 address (16 bytes)       |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | IPv6 address (continued)                                      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | IPv6 address (continued)                                      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | IPv6 address (continued)                                      |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | IPv6 address (continued)      | Prefix Length |      Flags    |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     Type

        0x02  IPv6 address

     Length

        The Length contains the total length of the subobject in bytes,
        including the Type and Length fields.  The Length is always 20.

     IPv6 address

        An IPv6 address.  This address is treated as a prefix based on
        the prefix length value below.  Bits beyond the prefix are
        ignored on receipt and SHOULD be set to zero on transmission.

     Prefix Length

        Length in bits of the IPv6 prefix.

     Flags

        0x01  Local or segment protection available

              If prefix length is 32:
              Indicates that the link downstream of this node is
              protected via a local repair mechanism.  This flag can
              only be set if the Local protection flag was set in the
              SESSION_ATTRIBUTE object of the corresponding Path
              message.

              If prefix length < 32:
              Indicates that the segment of the LSP inside the
              abstract node is protected against link failures.  This
              flag can only be set if the segment protection flag was
              set in the SESSION_ATTRIBUTE object of the
              corresponding Path message.

        0x02  Local or segment protection in use
```

```
          If prefix length is 32:
          Indicates that a local repair mechanism is in use to
          maintain this tunnel (usually in the face of an outage
          of the link it was previously routed over).

          If prefix length < 32:
          Indicates that a local or segment repair mechanism is
          in use to maintain this tunnel (usually in the face of
          an outage of the link it was previously routed over).

    0x04  Bandwidth protection

          The PLR will set this when the protected LSP has a backup
          path which provides the desired bandwidth, which is that in
          the FAST_REROUTE object or the bandwidth of the protected LSP,
          if no FAST_REROUTE object was included.  The PLR may set this
          whenever the desired bandwidth is guaranteed; the PLR MUST set
          this flag when the desired bandwidth is guaranteed and the
          "bandwidth protection desired" flag was set in the
          SESSION_ATTRIBUTE object.

    0x08  Node protection

           When set, this indicates that the PLR has a backup path
           providing protection against link and node failure on
           the corresponding path section. In case the PLR could only
           setup a link-protection backup path, the "Local protection
           available" bit will be set but the "Node protection" bit
           will be cleared.

    TBD   SRLG protection

          When set, this indicates that the PLR has a backup path
          providing protection against SRLG failures on the
          corresponding path section.

    TBD   Entry ASBR

          Indicates that this subobject represents a router that
          is the entry point inside the current AS.
```

The AS number subobject is composed of a 2-octet AS number, padding and flags. The total length of this subobject is 8 octets, including the type and the length fields. The type field is set to <TBD>.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     Length    |      AS number (2 bytes)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Padding                 |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

  Type

     TBD  AS number
```

```
Length

   The Length contains the total length of the subobject in bytes,
   including the Type and Length fields.  The Length is always 8.

AS number

   A 2-octet AS number (ASN).

Padding

   Zero on transmission. Ignored on receipt.

Flags

   0x01  Segment protection available

         Indicates that the path of the LSP inside the AS is
         protected. It indicates that the LSP is protected via a
         local or segment repair mechanism all the way inside the AS.
         This flag can only be set if the Local protection flag was
         set in the SESSION_ATTRIBUTE object of the corresponding
         Path message.

   0x02  Segment protection in use

         Indicates that a local or segment repair mechanism is in use
         to maintain this tunnel (usually in the face of an outage
         of the link it was previously routed over).

   0x04  Bandwidth protection

         The border router sets this flag when the primary LSP is
         protected by one or more backup segments, all the way inside
         the AS, and they provide the desired bandwidth, which is that
         in the FAST_REROUTE object or the bandwidth of the protected
         LSP, if no FAST_REROUTE object was included.  The border
         router may set this whenever the desired bandwidth is
         guaranteed; the border router MUST set this flag when the
         desired bandwidth is guaranteed and the "bandwidth protection
         desired" flag was set in the SESSION_ATTRIBUTE object.

   0x08  Global Node protection

          When set, this indicates that the path is protected against
          link and node failures on the path segment inside the AS. In
          case only link-protection backup paths could be setup, the
          "Segment protection available" bit will be set but the "Node
          protection" bit will be cleared.

   TBD   Global SRLG protection

         When set, this indicates that the path is protected against
         SRLG failures on the path segment inside the AS.
```

The SRLG number subobject contains a 4-octet SRLG identifier according to [PPD+02]. And, the total length of this subobject is 8 octets, including the type, the length and the padding fields. The type field is set to <TBD>.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |     Length    | SRLG identifier (4 bytes)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  SRLG identifier (continued)  |           Padding             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

B.2.2 Handling of the RRO

The RRO is used for loop detection, route pinning and disjoint path computation.

Route pinning is performed by using the RRO in the construction of the ERO. In [ABG+01], the RRO subobjects are put in sequence inside the ERO. The loose bit of the subobjects is not set since the RRO, in that draft, is used to record all nodes on the path. In that case, the RRO gives a complete and strict route of the LSP.

At the interdomain, since aggregation of AS topologies is necessary outside the ASs, the RRO may contain abstract nodes such as AS numbers and IP prefixes. Therefore, some changes are to be brought when composing the ERO. When an AS number (or an IP prefix) subobject is found inside the RRO, an AS subobject (an IP prefix, respectively) with the same AS number field (IP address field, resp.) is put inside the ERO and the loose bit of the following subobject is set.

The setting of the loose bit in the following subobject avoids the generation of a Path Error message when that suboject is treated since the current node probably does not belong to the abstract node. Indeed the aggregation of the RRO has suppressed some nodes. This results in some holes inside the recorded route. The holes encountered may be filled with the RRO stored locally at the node processing the ERO.

Aggregation of the RRO is performed by means of the "entry ASBR" flag. This flag is set when an entry ASBR, supporting the RRO aggregation, is stored inside the RRO. It marks the entrance inside the AS and is used to detect the nodes to remove from the RRO at the exit ASBR.

For IPv4 and IPv6 address subobjects, the flags are set as described in [ABG+01] and [PGS+02]. When we add IPv4 and IPv6 prefixes as well as AS number subobjects inside the RRO, the setting of the flags occurs as follows. These subobjects are used to replace IP addresses subobjects in order not to reveal the topology inside a network or an AS. This is what we call RRO aggregation. When aggregation is performed, the flags of the suppressed IP address subobjects are used to set the flags of the aggregated prefix or AS number subobject.

The "Link or segment protection available" flag is set when this flag is set inside all the replaced subobjects. The "Link or segment protection in use" flag is set when this flag is set in one of the replaced subobjects. The same is also applicable to the "Segment protection available" and "Segment protection in use" flags of the AS number subobject.

The "Bandwidth protection" and the "Node protection" flags are described in [PGS+02]. It is extended here to IP prefixes and AS number subobjects to indicate if the LSP is bandwidth or node protected all along the path inside the network or the AS, respectively.

To indicate that SRLG protection is provided for a downstream link or for the path segment inside a network, the "SRLG protection" flag is set inside the corresponding IPv4/IPv6 address subobject or IPv4/IPv6 prefix subobject, respectively.

Loop detection is performed by each node in the following way. Each node looks into the received RRO for subobjects that it may add. If such subobject is met, there is a loop. This principle has to be refined a little for interdomain LSPs. An interior router (i.e. router at the core of an AS) looks if it finds the address of one of its interfaces inside the RRO. In that case there is a loop and the establishement of the LSP is terminated. An ASBR (AS border router) checks whether one of its addresses is present in the RRO in addition to check whether the AS number is already present in the RRO. In both cases, loop is detected and the establishement of the LSP is ended. The same happens in case network aggregation is performed. At the entrance of the network, it is checked if the network prefix is already present inside the RRO.

An advantage of RRO aggregation is that it allows to reduce the length of the RRO, therefore causing less errors due the creation of packets larger than the MTU.

B.2.3 Non-support of the RRO or of its subobjects

The RRO object is to be used only when all routers along the path support RSVP and the RRO object. The RRO object is assigned a class value of the form 0bbbbbbb. RSVP routers that do not support the object will therefore respond with an "Unknown Object Class" error.

When processing an RRO, unrecognized subobjects SHOULD be ignored and passed on. When processing an RRO for loop detection, a node SHOULD parse over any unrecognized objects. Loop detection works by detecting subobjects which could be inserted by the node itself on an earlier pass of the object. This ensures that the subobjects necessary for loop detection are always understood.

A node that supports the aggregation of RRO information into entry point, AS number and exit point MUST support the flags defined in this draft. The same applies for a node that performs network aggregation. Therefore, these nodes are able to deal correctly with those flags. These flags are essentially useful for the nodes performing aggregation and for the node that initiates the LSP tunnel establishment. The other nodes on the path of the LSP do not need to support them they only need to transmit them inside the Path and Resv messages.

B.3 Avoid Route Object

The Avoid Route Object (ARO) is a new object that has the following format:

```
Class = <TBD>, C_Type = 1

 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        (Subobjects)                         //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The contents of an `AVOID_ROUTE` object are a series of variable-length data items called subobjects. These subobjects are the same as thoses of the RRO. There is an exception for the label subobject which has no use inside the ARO.

The ARO can only be present in RSVP Path messages. If a Path message contains multiple AROs, all the AROs are meaningful. This is not the case for the ERO and the RRO.

B.3.1 subobjects

As for the RRO, we have the IPv4 and IPv6 prefix subobjects as well as the AS number and the SRLG number subobjects. But, for the ARO, there is no label subobject. The IPv4 prefix, the IPv6 prefix, the AS number and the SRLG number subobjects have the same syntax as the corresponding subobjects of the RRO.

In these subobjects, there are no flags defined. The flag field is ignored on receipt and set to zero on transmission.

B.3.2 Handling of the ARO

The ARO is composed of single nodes (IP prefixes) or/and abstract nodes. The content of this object represents the path to be avoided by the LSP being established. The ARO is used by routers that need to complete the ERO in order to join the next abstract node in the ERO or the destination of the LSP. An example of the use of the ARO object is provided in appendix A.

As well as the RRO stored in the path state at each node, the ARO may contain holes. By holes we mean that the ARO may not contain the whole route of the primary LSP. This results from the fact that the ARO is formed from the RRO stored in the path state of nodes and all nodes may not have a global view of the topology.

To complete the ERO of a backup path, the ARO is used for disjoint path computation, if it contains information about single nodes inside the current routing domain. In case the path of the primary LSP is not available inside the ARO, then, a node on the path of the LSP to avoid is contacted in order to obtain that information. This is possible since the ARO contains at least the aggregated path of the primary LSP. Communication between a node on the backup and a node on the primary LSP is based on the Path computation request and reply messages defined in [VIZ+02].

B.3.3 Non-support of the ARO or of its subobjects

Routers that must compute the route or a segment of the route of an LSP must support the ARO if it is present in the Path message of the LSP. Routers that can forward the Path message without looking into the ARO, because the ERO does not need to be completed, do not need to support the ARO. When processing the ERO, if a router needs to add nodes into the ERO and at least an ARO is present, the router must take the AROs into account in the computation of the path and the ERO. In this case, if the router does not support the ARO, the router sends an Path Err message and the LSP is not established. Typically, ASBR and ABR need to support the ARO since these routers are the entry point into routing domains and routing area, respectively.

If new subobjects should be added in the future, only routers that are completing the ERO would need to support these new subobjects. A router that needs to compute a path based on AROs containing unknown subobject types should send a Path Err message to the node initiating the LSP. This message should contain the subobject types that are unknown and the address of the node that does not support them.

B.4 Session Attribute Object

The Session Attribute Class is 207. Two `C_Types` are defined, `LSP_TUNNEL`, `C-Type = 7` and `LSP_TUNNEL_RA`, `C-Type = 1`. The `LSP_TUNNEL_RA C-Type` includes all the same fields as the `LSP_TUNNEL C-Type`. Additionally it carries resource affinity information. The formats are as follows:

B.4.1 Format without resource affinities

```
SESSION_ATTRIBUTE class = 207, LSP_TUNNEL C-Type = 7

   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |   Setup Prio  | Holding Prio  |     Flags     |  Name Length  |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  //          Session Name      (NULL padded display string)      //
  |                                                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

     Setup Priority

        The priority of the session with respect to taking resources,
```

in the range of 0 to 7. The value 0 is the highest priority. The Setup Priority is used in deciding whether this session can preempt another session.

Holding Priority

The priority of the session with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority. Holding Priority is used in deciding whether this session can be preempted by another session.

Flags

0x01 Local protection desired

This flag permits transit routers to use a local repair mechanism which may result in violation of the explicit route object. When a fault is detected on an adjacent downstream link or node, a transit router can reroute traffic for fast service restoration.

0x02 Label recording desired

This flag indicates that label information should be included when doing a route record.

0x04 SE Style desired

This flag indicates that the tunnel ingress node may choose to reroute this tunnel without tearing it down. A tunnel egress node SHOULD use the SE Style when responding with a Resv message.

TBD SRLG recording desired

This flag indicates that SRLG information should be included when doing a route record.

0x08 Bandwidth protection desired

This flag indicates to the PLRs along the protected LSP path that a backup path with a bandwidth guarantee is desired. The bandwidth which must be guaranteed is that of the protected LSP, if no FAST_REROUTE object is included in the PATH message; if a FAST_REROUTE object is in the PATH message, then the bandwidth specified in there is that which must be guaranteed.

0x10 Node protection desired

This flag indicates to the PLRs along a protected LSP path that they must select a backup path that bypasses at least the next node of the protected LSP.

TBD SRLG protection desired

This flag indicates to the PLRs along a protected LSP path that they must select a backup path that bypasses the SRLGs of the downstream link of the protected LSP.

Name Length

The length of the display string before padding, in bytes.

Session Name

A null padded string of characters.

B.4.2 Format with resource affinities

SESSION_ATTRIBUTE class = 207, LSP_TUNNEL_RA C-Type = 1

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Exclude-any                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Include-any                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Include-all                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Setup Prio  | Holding Prio  |     Flags     |  Name Length  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//          Session Name      (NULL padded display string)      //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Exclude-any

A 32-bit vector representing a set of attribute filters associated with a tunnel any of which renders a link unacceptable.

Include-any

A 32-bit vector representing a set of attribute filters associated with a tunnel any of which renders a link acceptable (with respect to this test). A null set (all bits set to zero) automatically passes.

Include-all

A 32-bit vector representing a set of attribute filters associated with a tunnel all of which must be present for a link to be acceptable (with respect to this test). A null set (all bits set to zero) automatically passes.

Setup Priority

The priority of the session with respect to taking resources, in the range of 0 to 7. The value 0 is the highest priority. The Setup Priority is used in deciding whether this session can preempt another session.

Holding Priority

The priority of the session with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority. Holding Priority is used in deciding whether this session can be preempted by another session.

Flags

0x01 Local protection desired

This flag permits transit routers to use a local repair mechanism which may result in violation of the explicit route object. When a fault is detected on an adjacent downstream link or node, a transit router can reroute traffic for fast service restoration.

0x02 Label recording desired

This flag indicates that label information should be included when doing a route record.

0x04 SE Style desired

This flag indicates that the tunnel ingress node may choose to reroute this tunnel without tearing it down. A tunnel egress node SHOULD use the SE Style when responding with a Resv message.

TBD SRLG recording desired

This flag indicates that SRLG information should be included when doing a route record.

0x08 Bandwidth protection desired

This flag indicates to the PLRs along the protected LSP path that a backup path with a bandwidth guarantee is desired. The bandwidth which must be guaranteed is that of the protected LSP, if no FAST_REROUTE object is included in the PATH message; if a FAST_REROUTE object is in the PATH message, then the bandwidth specified in there is that which must be guaranteed.

0x10 Node protection desired

This flag indicates to the PLRs along a protected LSP path that they must select a backup path that bypasses at least the next node of the protected LSP.

```
          TBD   SRLG protection desired

                This flag indicates to the PLRs along a protected LSP
                path that they must select a backup path that bypasses
                the SRLGs of the downstream link of the protected LSP.

       Name Length

          The length of the display string before padding, in bytes.

       Session Name

          A null padded string of characters.
```

The flags “Bandwidth protection desired” and “Node protection desired” are defined in [PGS+02]. The “SRLG recording desired” flag indicates that SRLG should be recorded inside the RRO.

B.4.3 Handling of the session attribute object

This section concerns the handling of the session attribute and the session attribute object with ressource affinities.

We take a special look at the use of the flags since two flags have been added to these objects. We refer to [PGS+02] for the use of the “Bandwidth protection desired” and “Node protection desired” flags. Concerning the handling of the other fields see [ABG+01].

The “SRLG recording desired” flag is used to indicate that SRLGs should be recorded inside the RRO. These SRLGs will then be used for the computation of disjoint SRLG paths.

A node that gets a Path message with the “SRLG recording desired” flag set inside the session attribute object, should record the SRLG of the output link on which the Path message will be forwarded after the address of the node is recorded inside the RRO.

The “SRLG protection desired” flag is used to indicate that the LSP should be protected against SRLG failures. It requires that backup LSPs be SRLG disjoint from the segments of this LSP that they protect. A PLR that receives a Path message with this flag set in the session attribute object should establish a backup LSP that avoids the SRLGs of the protected segment.

B.4.4 Non-support of the session attribute object

All RSVP routers, whether they support the `SESSION_ATTRIBUTE` object or not, SHALL forward the object unmodified. The presence of non- RSVP routers anywhere between senders and receivers has no impact on this object.

A router that does not support the “SRLG recording desired” flag will not store the SRLG of its output link into the RRO. Consequently, it will not be possible to compute an SRLG disjoint path from this LSP based only on the RRO stored in path states.

The non-support of the “SRLG protection desired” flag is dealt in the same way as the non-suport of the “Bandwidth protection desired” and “Node protection desired” flags defined in [PGS+02].

B.5 Session Object

There are two C-Type session objects. One is used to specify an IPv4 destination of the LSP and the other is used when the destination has an IPv6 address. These two object types keep the same syntax as defined in [ABG+01]. The tunnel end point address however may be partially defined in that it may not be the effective end point of the LSP since we have added ways to indicate inside subobject of the ERO that the LSP may end at any router inside an AS or inside a prefix. However, the tunnel end point address must be part of the prefix destination or part of the AS destination.

B.5.1 LSP_TUNNEL_IPv4 Session Object

Class = SESSION, LSP_TUNNEL_IPv4 C-Type = 7

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   IPv4 tunnel end point address               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MUST be zero                 |      Tunnel ID                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Extended Tunnel ID                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 tunnel end point address

IPv4 address of the egress node for the tunnel.

Tunnel ID

A 16-bit identifier used in the SESSION that remains constant over the life of the tunnel.

Extended Tunnel ID

A 32-bit identifier used in the SESSION that remains constant over the life of the tunnel. Normally set to all zeros. Ingress nodes that wish to narrow the scope of a SESSION to the ingress-egress pair may place their IPv4 address here as a globally unique identifier.

B.5.2 LSP_TUNNEL_IPv6 Session Object

Class = SESSION, LSP_TUNNEL_IPv6 C_Type = 8

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                   IPv6 tunnel end point address               |
+                                                               +
|                            (16 bytes)                         |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  MUST be zero                 |      Tunnel ID                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                       Extended Tunnel ID                      |
+                                                               +
|                            (16 bytes)                         |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
IPv6 tunnel end point address

   IPv6 address of the egress node for the tunnel.

Tunnel ID

   A 16-bit identifier used in the SESSION that remains constant
   over the life of the tunnel.

Extended Tunnel ID

   A 16-byte identifier used in the SESSION that remains constant
   over the life of the tunnel.  Normally set to all zeros.
   Ingress nodes that wish to narrow the scope of a SESSION to the
   ingress-egress pair may place their IPv6 address here as a
   globally unique identifier.
```

B.5.3 Handling of the session object

Each node on the path of the LSP treats the session object as usual. But, the source of the LSP has to set the destination field in a consistent way such that this destination may be used to join the desired AS or network in case the end point inside either the AS or the network does not matter.

B.5.4 Non-support of the session object

The session object should be supported by all nodes on the path of the LSP. If it is not supported a Path Err message MUST be generated by the node that doesn't recognize it.

B.6 FAST_REROUTE Object

The FAST_REROUTE object is defined in [PGS+02]. The FAST_REROUTE object carries the control information, such as setup and hold priorities and bandwidth. A protected LSP uses the FAST_REROUTE object to specify the level of protection that is required during local repair. The FAST_REROUTE object can be used for both one-to-one and facility backup, and has the following format:

```
Class = TBD  (use form 11bbbbbb for compatibility)
C-Type = 1

       0             1              2             3
  +-------------+-------------+-------------+-------------+
  |       Length (bytes)      |  Class-Num  |   C-Type    |
  +-------------+-------------+-------------+-------------+
  | Setup Prio  | Hold Prio   | Hop-limit   |    Flags    |
  +-------------+-------------+-------------+-------------+
  |                  Bandwidth                            |
  +-------------+-------------+-------------+-------------+
  |                   Include-any                         |
  +-------------+-------------+-------------+-------------+
  |                   Exclude-any                         |
  +-------------+-------------+-------------+-------------+
  |                   Include-all                         |
  +-------------+-------------+-------------+-------------+
```

Setup Priority

The priority of the backup path with respect to taking resources, in the range of 0 to 7. The value 0 is the highest priority. Setup Priority is used in deciding whether this session can preempt another session. See [RSVP-TE] for the usage on priority.

Holding Priority

The priority of the backup path with respect to holding resources, in the range of 0 to 7. The value 0 is the highest priority. Holding Priority is used in deciding whether this session can be preempted by another session. See [RSVP-TE] for the usage on priority.

Hop-limit

The maximum number of extra hops the backup path is allowed to take, from current node (a PLR) to a MP, with PLR and MP excluded in counting. For example, hop-limit of 0 means only direct links between PLR and MP can be considered.

Flags

0x01 One-to-one Backup Desired

Indicates that the LSP should be protected via the one-to-one backup mechanism described in Section 5. This flag can only be set by the head-end LSRs.

0x02 Facility Backup Desired

Indicates that the LSP should be protected via the facility backup mechanism described in Section 6. This flag can only be set by the head-end LSRs.

Bandwidth

Bandwidth estimate (32-bit IEEE floating point integer) in bytes-per-second.

Exclude-any

A 32-bit vector representing a set of attribute filters associated with a backup path any of which renders a link unacceptable.

Include-any

A 32-bit vector representing a set of attribute filters associated with a backup path any of which renders a link acceptable (with respect to this test). A null set (all bits set to zero) automatically passes.

```
Include-all

 A 32-bit vector representing a set of attribute filters associated
 with a backup path all of which must be present for a link to be
 acceptable (with respect to this test). A null set (all bits set
 to zero) automatically passes.
```

The C-Class must be assigned in such a way that, for the LSRs that do not support the FAST_REROUTE objects, they MUST forward the objects downstream unchanged.

No changes are brought to the initial definition of the FAST_REROUTE object made in [PGS+02]. The two flags "One-to-one Backup Desired" and "Facility Backup Desired" are very useful for the establishment of detour LSPs or to indicate the use of bypass tunnels.

B.7 DETOUR Object

The DETOUR object is used in one-to-one backup to setup and identify detour LSPs. It has the following format:

```
Class = TBD  (to conform 0bbbbbbb format for compatibility)
C-Type = 7

       0             1              2             3
  +-------------+-------------+-------------+-------------+
  |       Length (bytes)      |  Class-Num  |   C-Type    |
  +-------------+-------------+-------------+-------------+
  |                      PLR ID  1                        |
  +-------------+-------------+-------------+-------------+
  |                    Avoid Node ID 1                    |
  +-------------+-------------+-------------+-------------+
 //                          ....                         //
  +-------------+-------------+-------------+-------------+
  |                      PLR ID  n                        |
  +-------------+-------------+-------------+-------------+
  |                    Avoid Node ID  n                   |
  +-------------+-------------+-------------+-------------+


PLR ID  (1 - n)

  IPv4 address identifying the beginning point of detour which
  is a PLR. Any local address on the PLR can be used.

Avoid Node ID  (1 - n)

  IP address identifying the immediate downstream node that
  the PLR is trying to avoid. Router ID of downstream node
  is preferred. This field is mandatory, and is used by
  the MP for merging rules discussed below.
```

There could be more than one pair of (PLR_ID, Avoid_Node_ID) entry in a DETOUR object. If detour merging is desired, after each merging operation (Section 5.3), the MP should combine all the merged detours in the subsequent Path messages.

The C-Class must be assigned in such a way that, for the LSRs that do not support the DETOUR objects, the LSRs MUST reject the message and send a PathErr to notify the PLR.

In order to establish detours that are SRLG disjoint form the portion of the working path that it protects, a new type of DETOUR object has to be defined. This object has the following format:

```
Class = TBD  (to conform 0bbbbbbb format for compatibility)
C-Type = TBD

       0             1              2             3
  +-------------+-------------+-------------+-------------+
  |       Length (bytes)      |  Class-Num  |   C-Type    |
  +-------------+-------------+-------------+-------------+
  |                         PLR ID                        |
  +-------------+-------------+-------------+-------------+
  |                       Avoid Node ID                   |
  +-------------+-------------+-------------+-------------+
  |                          SRLG 1                       |
  +-------------+-------------+-------------+-------------+
 //                          ....                        //
  +-------------+-------------+-------------+-------------+
  |                          SRLG n                       |
  +-------------+-------------+-------------+-------------+

 Avoid SRLG  (1 - n)

   SRLG of the link preceeding the node being protected.
   There may be more than one SRLG for a link since a link
   may belong to different Shared Risk Link Groups.
```

The PLR ID and the Avoid Node ID fields have the same meaning as in the DETOUR object of C-type equals to 7.

Note that merging of detour LSPs is not possible with this object since only one (PLR ID, Avoid Node ID) may be stored inside the DETOUR object. This results from the possibility of storing many SRLGs, corresponding to a single link, inside this object.

If merging of detour LSPs is desired, a DETOUR object of C-type TBD, for each detour LSP, should be present inside the Path message of the merged detour LSPs. Before merging these detours, it should be checked if each detour avoids the SRLGs that have to be avoided by each single detour.

An alternative to the use of the DETOUR object of type TBD is the use of the ARO object defined in a previous section. The SRLGs to be avoided are stored inside the ARO and the DETOUR object of C-type 7 is used to indicate the PLR of the detour and the node avoided by this detour LSP.

B.8 BYPASS Object

The BYPASS object is used in many-to-one protection to setup and identify bypass tunnels. It has the following format:

```
 Class = TBD  (to conform 0bbbbbbb format for compatibility)
 C-Type = 7

       0             1              2             3
  +-------------+-------------+-------------+-------------+
  |       Length (bytes)      |  Class-Num  |   C-Type    |
  +-------------+-------------+-------------+-------------+
  |                         PLR ID  1                     |
  +-------------+-------------+-------------+-------------+
```

```
 |                      Avoid Node ID                          |
 +-------------+-------------+-------------+-------------+
 |                       Avoid SRLG 1                          |
 +-------------+-------------+-------------+-------------+
//                          ....                              //
 +-------------+-------------+-------------+-------------+
 |                       Avoid SRLG n                          |
 +-------------+-------------+-------------+-------------+
```

```
   PLR ID

     IPv4 address identifying the beginning point of detour which
     is a PLR. Any local address on the PLR can be used.

   Avoid Node ID

     IP address identifying the immediate downstream node that
     the PLR is trying to avoid. Router ID of downstream node
     is preferred. This field is mandatory, and is used by
     the MP for merging rules discussed below.

%comment fait-on quand on protge une ligne et pas un noeud? On n'a pas
%besoin de cet object? En effet, je pense que l'on en a pas besoin.

   Avoid SRLG  (1 - n)

     SRLG of the link to protect or SRLG of the link preceeding the
     node being protected.  There may be more than one SRLG for a
     link since a link may belong to different Shared Risk Link
     Groups.
```

The C-Class must be assigned in such a way that, for the LSRs that do not support the BYPASS objects, the LSRs MUST reject the message and send a PathErr to notify the PLR.

An alternative to storing the SRLGs to avoid inside the BYPASS object is the use of the ARO object. These SRLGs are stored inside the ARO and the bypass is used to indicate the node to avoid and the PLR.

References

[ABG+01] D. Awduche, L. Berger, D.-H. Gan, T. Li, V. Srinivasan, and G. Swallow. RSVP-TE: Extensions to RSVP for LSP Tunnels, December 2001. RFC 3209.

[AEWX00] D. Awduche, A. Elmalid, L. Widjaja, and X. Xiao. A framework for Internet traffic engineering, July 2000. Work in progress, draft-ietf-te-framework-02.txt.

[AMA+99] D. Awduche, J. Malcom, J. Agogbua, M. O'Dell, and J. McManus. Requirements for traffic engineering over MPLS, September 1999. RFC 2702.

[dBPNP00] S. Van den Bosch, F. Poppe, H. De Neve, and G. Petit. Multi-objective traffic engineering of IP network using Label Switched Paths. In *Networks 2000*, Toronto, Canada, September 2000.

[ea01] G. Bernstein et al. Optical inter domain routing considerations. Internet draft, draft-ietf-ipo-optical-inter-domain-00.txt, work in progress, November 2001.

[FT00] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *INFOCOM2000*, March 2000. Available at http://www.ieee-infocom.org.

[NEN02] I. Nakagawa, H. Esaki, and K. Nagami. A design of a next generation IX using MPLS technology. In *2002 Symposium on Applications and the Internet, SAINT 2002*, Nara City, Japan, January 28 - February 1 2002.

[PGS+02] P. Pan, D.-H. Gan, G. Swallow, J. P. Vasseur, D. Cooper, A. Atlas, and M. Jork. Fast Reroute Extensions to RSVP-TE for LSP Tunnels, January 2002. Work in progress, draft-ietf-mpls-rsvp-lsp-fastreroute-00.txt.

[PPD+02] D. Papadimitriou, F. Poppe, S. Dharanikota, R. Hartani, R. Jain, J. Jones, S. Venkatachalam, and Y. Xue. Shared Risk Link Group Encoding and Processing, June 2002. Work in progress, draft-papadimitriou-ccamp-srlg-processing-00.txt.

[PPJ+01] D. Papadimitriou, F. Poppe, J. Jones, S. Venkatachalam, S. Dharanikota, R. Jain, R. Hartani, D. Griffith, and Y. Xue. Inference of Shared Risk Link Group, July 2001. Work in progress, draft-many-inference-srlg-01.txt.

[SH02] V. Sharma and F. Hellstrand. Framework for MPLS-based recovery, September 2002. Work in Progress, draft-ietf-mpls-recovery-frmwrk-07.txt.

[VIZ+02] J.-P. Vasseur, C. Iturralde, R. Zhang, X. Vinet, S. Matsushima, and A. Atlas. RSVP Path computation request and reply messages, June 2002. Work in progress, draft-vasseur-mpls-computation-rsvp-03.txt.

[Wan00] Z. Wang. Internet traffic engineering. *Special section of IEEE Network Magazine*, March-April 2000.

[WWZ01] Y. Wang, Z. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. In *INFOCOM2001*, April 2001. Available at http://www.ieee-infocom.org.