Generalized MPLS
The DRAGON Project implementation at SARA

Research report

July 17, 2006

UNIVERSITEIT VAN AMSTERDAM

Mark Meijerink
mark@os3.nl

Rob Prickaerts
rprickaerts@os3.nl

DOCUMENT DATA

	Date	**Author**	**Document Status**
0.1	01-02-06	Mark Meijerink Rob Prickaerts	Draft
1.0	04-07-06	Mark Meijerink Rob Prickaerts	Final version

Table 1: Version history

Name	**Function**	**Organization**
M. Meijerink	Researcher	University of Amsterdam
R. Prickaerts	Researcher	University of Amsterdam
R. van der Pol	Project Coordinator	SARA
A. Toonk	Project Coordinator	SARA
C. de Laat	Project Advisor	University of Amsterdam

Table 2: Distribution List

Preamble

This paper was written for the class Research Project 2 for the System and Network Engineering Master program at the University of Amsterdam. To find a research subject suitable for this class and of both our interest, we have looked for an organization in the Amsterdam area that would be able to provide us with a challenging and technically renewing assignment. Our preferences where in the field of internetworking and routing, and with this in mind we ended up at SARA Amsterdam.

SARA is an IT-services centre which, with their a advanced and complete set of products and services has become a nationally and internationally know standard over the last thirty years. SARAs main fields of expertise are high performance networking, infrastructure services and high performance computing and visualization. Over the last few years many research networks have deployed hybrid networks. A hybrid network is a network which partially consists of a routed IP section and partially of a *optical* or *lightpath* section. Lightpaths are layer 1 or layer 2 connections with dedicated bandwidth and Quality of Service properties. SARA uses lightpaths in its role as tier-1 site for the CERN Large Hadron Collider project and LOFAR.

In cooperation with the University of Amsterdam SARA is conducting a research on how lightpaths which at this point in time need manual configuration by the Network Operations Center, in the future can be set up automatically by end users and applications. This is done on behalf of SURFnet. Part of this research focusses on Generalized MPLS. With this specific Generalized MPLS research both SARA and the University of Amsterdam hope to obtain more in depth knowledge on GMPLS in relation with the Open Source implementation of the DRAGON Project GMPLS software suite. This in turn might be able to help them in their initial lightpath research.

Abstract

Hybrid networks consisting of routed IP and Optical or lightpath sections have become a field of interest for SARA Amsterdam. The manual configuration required for setting up lightpaths costs time because a administrator has to configure the lightpath. This manual configuration leaves room for errors. For the future SARA expects that more research groups and applications will require their own configurations. For SARA in the future to be able to keep on using lightpaths they need a method of automating this configuration process. Generalized Multi-Protocol Label Switching GMPLS is one of the many techniques researched by SARA to overcome this obstacle.

This research focus was on an Open-Source GMPLS product currently under development under by DRAGON Project. DRAGON is short for the Dynamic Resource Allocation via GMPLS Optical Networks and is the name for a software project for dynamic resource provisioning across heterogeneous network technologies and vendor equipment. The software suite was used to set up multiple test infrastructures in which the usability and RFC compliance had to be tested. Before tests could be performed, first the GMPLS technology needed to be researched.

A literature study was conducted so that all members of the research group had a good understanding of the GMPLS technology and the different protocols used to enable label switching. GMPLS is a superset of MPLS, a technology well supported by the industry for label switching on packet- and cell-based networks. With GMPLS Label switching was extended to be used on several other networking technologies such as Time-Division Multiplexing, lambda switched or fiber switched networks next to the already in MPLS available Packet switch and Layer-2 Switch networks. With the support on these networks GMPLS also overcame scalability problems as were the case on MPLS. Header inspection was no longer needed and the use of the generalized label offered new possibilities. The generalized label in GMPLS enables switches to switch complete ports or wavelengths.

GMPLS uses a control plane to share Traffic Engineering TE and topology information about the data interfaces. The data interfaces are part of the data plane and are used for setting up the Label Switched Paths LSP's and transport the user data. Where in MPLS the control and data plane resided on the same physical link, in GMPLS these are typically separated and do not need to have similar topologies.

On the control plane OSPF-TE or ISIS-TE is used to enable the Label Switched Routers LSR's to exchange TE information. The TE information enables GMPLS to apply Quality of Service to the LSP's. Signaling information is exchanged by the Resource reSerVation Protocol RSVP-TE and is used for LSP setup and tear-down process. For fault isolation and link maintenance the Link Management Protocol LMP was especially created for GMPLS. This protocol checks the availability of the Data links and is responsible for fault isolation and reporting to the control plane.

The DRAGON software was installed on four Scientific Linux servers and uses OSPF-TE on the control plane for routing and TE information exchange. For the data plane two Raptor ER1010 switches were used which were control via SNMP by two of the DRAGON servers called Virtual Labels Switched Routers VLSR's. These VLSR's enable DRAGON to configure and manage the non GMPLS Raptor Switches and are the main components of the DRAGON software suite. The two remaining DRAGON

servers were used as Client System Agents CSA's and were used to test connectivity over LSP's after configuring them in the DRAGON Command Line Interface.

Three different tests were conducted on the GMPLS infrastructures to test several RFC functionalities. First of all LSP creation which was the main goal of this research. Being able to set up and use a LSP across the GMPLS network consisting of multiple VLSR's and switches had proven to be a great achievement already. Next the ability of DRAGON to support multiple LSP's over a single tunnel was tested to explore the possibilities of multiple sources trying to connect to a single destination. Even though the tests were inconclusive, it looked as if DRAGON did support this feature of GMPLS. The third and final test was to abruptly severe one of the links in the LSP to check DRAGONs reaction to this action. No error or tear down messages were send out and only after manually deleting the LSP on the requesting CSA the tear down messages were send out. Sending tear down messages is crucial when using GMPLS because resources will not be released until the LSP is deleted or timeout values are reached.

The results of the tests as well as the documentation available on the DRAGON Project and the information received from members but the lack of the LMP protocol as well as some other the key features of GMPLS point out that the software still needs a lot of development is before it can provide all of the GMPLS feature set. The already available Network Aware Resource Broker NARB to enable inter domain GMPLS though, states that the DRAGON Project is already on its way to overcome some of the most difficult elements of GMPLS and speaks highly of the ingenuity of the developers.

Further developing of the DRAGON software suite will have to lead to a fully operational an RFC compliant Open Source software suite for GMPLS. The first and so far only available package that will allow the usage of non GMPLS hardware to perform GMPLS functions. SARA will need to further research the possibilities of DRAGON on their newly created GMPLS infrastructure. By adding more complexity to the network they could set up a pilot network to get hands-on experience with GMPLS and for example test the aspects of the StarPlane project. This in turn will benefit their initial lightpath research and maybe even lead to the implementation of DRAGON in the future.

Contents

1 Project Introduction **8**
1.1 Assignment 8
1.2 Contents 8

2 GMPLS **10**
2.1 History 10
2.2 GMPLS Background 11
2.2.1 MPLS Operation 11
2.2.2 GMPLS Operation 12
2.3 GMPLS Routing 14
2.3.1 Routing : OSPF-TE 15
2.4 Signaling and Management 16
2.4.1 RSVP-TE 17
2.4.2 Link Management: LMP 18

3 DRAGON **20**
3.1 Elements 20
3.1.1 CSA 20
3.1.2 NARB 20
3.1.3 VLSR 21
3.1.4 ASTB 22
3.2 Extendability and Developments 23

4 Testing **24**
4.1 Scenario 1: Basic infrastructure 24
4.1.1 Tests 25
4.2 Scenario 2: Extended infrastructure 25
4.2.1 Tests 26
4.3 Configuration 27
4.3.1 GRE tunnels 27
4.3.2 OSPF-TE 27

5 Results **30**

5.1 Creating a LSP 30

5.1.1 Execution 30

5.1.2 Results 31

5.2 Interruption of a LSP 31

5.2.1 Execution 31

5.2.2 Results 31

5.3 Setting up multiple LSP's over one link 32

5.3.1 Execution 32

5.3.2 Results 32

5.4 Other tests 33

6 Conclusions and recommendations **34**

7 Appendix **37**

7.1 Appendix A: Configuration files of the CSA on the left, host 1 37

7.2 Appendix B: Configuration files of VLSR 1 39

7.3 Appendix C: Configuration files of VLSR2 42

7.4 Appendix D: Configuration files of the CSA on the right, host 2 45

7.5 Appendix E: Packet dump VLSR1 setting up a LSP 47

7.6 Appendix F: Packet dump VLSR2 setting up a LSP 49

7.7 Appendix G: Packet dump VLSR1 of tearing down a LSP 51

7.8 Appendix H: Packet dump VLSR2 of tearing down a LSP 52

7.9 Appendix I: Packet dump VLSR1 setting up multiple LSP's over one link . 53

7.10 Appendix J: Packet dump VLSR2 setting up multiple LSP's over one link . 54

7.11 Appendix K Contact Information 56

1 Project Introduction

1.1 Assignment

SARA will provide the means to set up a test environment to run the DRAGON software. It will have to consist of either three or four scientific Linux server systems and one or two Raptor ER1010 switches, depending on the tests. The Generalized MPLS Label Switched Paths (LSP) will be realized using both the Linux control systems as well as the switches. The DRAGON software will have to prove itself worthy in this environment to be used later by SARA in future GMPLS research. Putting together this test environment will be the First part of the assignment.

Within GMPLS there are two protocols responsible for establishing LSPs. First OSPF-TE and second RSVP-TE. OSPF-TE is responsible for the forwarding decisions on the data plane and holds extension to the OSPF standard. Where OSPF itself is used to provide connectivity and network knowledge on the control plane, the TE extension which are the Traffic Engineering part of OSPF and enable GMPLS to apply Quality of Service QoS functionality on the available data plane links. RSVP-TE is responsible for signaling the LSP setup and tear-down process within GMPLS.

Second part of the assignment will be the testing of the DRAGON software and GMPLS protocols. A literature study will have to be the basis of this report and will be complemented with the findings from the test stage. Not only the compliance of DRAGON with the GMPLS standard but also the modularity and extensibility of the product will be researched.

1.2 Contents

This report consists of three main items; GMPLS, the DRAGON Software and the Test phase. These three items in turn will be further divided into smaller subsections in this report. First of the report will start of with a short introduction into the research and its background. The basic ideas, philosophies as well as the expectations will be described in Chapter 1. It contains the project assignments and an overview of the contents of the report. Readers interested in a brief but complete summary of this paper are urged to read the Abstract which holds a management summary of the project and its results.

The first of the three main items can be found in Chapter 2. The GMPLS technology will be described in this chapter as well as the RSVP-TE and OSPF-TE protocols. It will start with a brief history on how GMPLS came to be. Next GMPLS extensions and protocols will be described in detail.

Chapter 3 holds all the elements concerning the DRAGON Projects implementation of GMPLS. An overview of the DRAGON software suite will describe all elements used such

as the VLSR and NARB systems. Next the extensibility of the suite will be discussed. Finally the developments on the DRAGON software can be found in the last subsection of chapter 3. Upcoming developments as well as future plans by the DRAGON Project will be described here.

The third and last item will be described in Chapter 4. The different infrastructures and test scenarios will be described in detail in this chapter. Network drawings of the test environment and configuration files from the test setups can be found here. Configuration choices made during the setup will be described here.

Chapter 5 will describe the results of the tests conducted as described in Chapter 4. Furthermore the RFC compliance of the DRAGON software will be described in this chapter. Because the software is still under development, not all RFC elements have been implemented yet. This part of the report will try and describe the RFC elements available in the software and which elements are not.

Chapter 6, Conclusion and Recommendations will hold the findings and results of the project. Those who want quick insight in the outcome of the research can find all major results in this chapter and will get insight in the future work for SARA with the DRAGON software package.

Chapter 7 contains the appendixes which can provide extra in-depth information on all research items. All appendixes will be referred to throughout the report and will help give the reader a detailed understanding and insight on the for instance the configuration files of the test infrastructures.

2 GMPLS

This chapter will describe Generalized Multi-Protocol Label Switching or GMPLS. All information found in this chapter is based on the GMPLS RFC 3945 [1] and is complimented with information from the book *GMPLS Architecture and Applications* by Adrian Farrel and Igor Bryskin [2] as well as Li Yin's *MPLS and GMPLS* presentation [3] and the IEC's GMPLS tutorial [4]. First a brief history of MPLS and GMPLS will be presented here. MPLS operation will be explained to create a basic understanding of Label Switching. Next the GMPLS extensions will be discussed in GMPLS operation. Basic functionalities of the protocols used in GMPLS like OSPF-TE, RSVP-TE and LMP will be described and an overview of the Protocol stack will be shown.

2.1 History

As MPLS was used more and more throughout the internet its limitations started to show. In the late nineties MPLS was introduced and offered new and easy ways of controlling traffic distribution across packet-based and cell-based networks. The ease at which an administrator could set up what seemed to be a point-to-point connection from and to any end node in a network, had been a big step forward. The fact that one could set up multiple tunnels and apply traffic engineering properties to them was a major benefit of MPLS. Before MPLS it was only possible on a direct connection like for instance a lease line. Secondly, with the introduction of MPLS they had found a way to make two opposing technologies coexist next to each-other and establish end-to-end paths in both packet-based and cell-based networks.

Bandwidth consumption has grown rapidly and the technological solutions on which the data is being transported needs to scale up at that same or preferably even faster rate. MPLS limits started to show with this scaling. MPLS which is based on packet-based or cell-based switching technologies works exactly on their data unit boundaries. Therefore before any decisions can be made MPLS requires the inspection of every unit header to determine its origin and destination. It is only then that a router or switch can forward a packet or cell in the right direction. Not only did this cause the scalability problems but also limited this the usage of MPLS to control different network technologies.

In 2001 GMPLS came into the world to encompass all networking technologies as available today. GMPLS should not be seen as a new protocol but as a superset of MPSL, by taking MPLS and going beyond its limitations GMPLS has a set of five interface such as a Time-Division Multiplex capable, Lambda Switch capable or Fiber Switched capable interfaces next to the already in MPLS available Packet switch capable and Layer-2 Switch capable interfaces.

GMPLS moves the decision process to different boundaries. Were MPLS used the data unit headers of the packets and cells to make forwarding decisions, GMPLS can use physical

space, time division and wavelengths on which data forwarding decisions can be made. This makes GMPLS not only suitable for different data transmission technologies, but also makes it more scalable.

2.2 GMPLS Background

As pointed out before, GMPLS is not an adversary but a superset of MPLS. GMPLS takes MPLS moves it not one, but multiple steps forward by adding more functionality and usability to the protocol. One of the biggest advantages of GMPLS next to the diversity of networking technologies it supports, is that it can effectively eliminate the need of a Network operator when setting up end-to-end GMPLS connections throughout the GMPLS capable network. The entire network can be automated, and no human interference will be required in the tunneling process.

Of course the need for human interaction will always remain in moments of maintenance and failures. But by taking out this variable of the daily process the time-savings and decreased chances for human-error will immediately translate in themselves into lower costs and higher availability of the network, which in turn will be beneficial to both the end-user and the organization.

GMPLS would not be able to support all the different types of networking technologies, if it did not have a generalized label. This label enables all switches between the source and destination to activate the Label Switched Path LSP and forward data, no matter what technology they are using. The generalized label can represent a time-slot, physical space or even a single wavelength.

2.2.1 MPLS Operation

To understand the GMPLS protocol suite and its operation one first needs to understand how MPLS operates. MPLS bases its forwarding decisions on a label which is added to the header of a packet. Every QoS item is put in a table of Forwarding Equivalence Classes FEC's. Every flow with the same traffic-engineering requirements will be in the same FEC and therefore receive the same Label.

Upon entering a MPLS network a label will be assigned to a packet after being inspected by the Label Edge Router LER. From that point on, the labeled packets are traveling over the established Labeled Switched Path LSP. Within this path, all the Label Switched Routers LSR's will make forwarding decisions based on the label of the package. Each hop in this process will extract the label, make a forwarding decision, look up the corresponding label for that specific interface in its Label Information Base and insert the new label. To the end system this process is transparent and the connection seems to be a point-to-point link.

All this would not work without some kind of control mechanism guarding the process while setting up the LSP throughout the network. In MPLS this is done by a Label Distribution Protocol. The two most popular are the Resourse reSerVation Protocol RSVP-TE, which will be addressed in more detail in this report and the ConstRaint based Label Distribution Protocol CR-LDP. This last protocol was invented especially for MPLS control signaling but is no longer used on a large scale and is considered dead by many. In MPLS a LSP is set up as an uni-directional path on which the label distribution flows from the destination to the source from where upon arrival the LSP will be operational and data can flow. The LSP can operate function end-to-end under the lowest FEC within the path.

To make MPLS understand what the data network looks like it needs a protocol to map the entire network, its links and their QoS features. What better way to do so then to use already available routing protocols. MPLS used the basic OSPF and ISIS protocols but needed traffic engineering abilities as well. OSPF-TE and ISIS-TE extensions where made especially for usage with MPLS. Now the routing protocols could not only exchange network topology information, but also transport TE information to MPLS devices so they could make decisions based on the QoS settings on the links.

MPLS devices have knowledge of the contents-of-information unit which they are passing on. They have to check each header to determine the destination and outgoing label. The control plane and data plane are logically separated.

2.2.2 GMPLS Operation

GMPLS works in many ways identical to MPLS and like MPLS OSPF-TE and ISIS-TE can be used for routing on the control plane and RSVP-TE and CR-LDP can be used for signaling on the data plane. But just as MPLS required extensions to these protocols so did GMPLS. The extensions in general related to the enable GMPLS to support new network technologies such as SDH/SONET and DWDM.

Not only did GMPLS extend MPLS by supporting all these techniques and extending all those protocols, but it also brought with it a totally new protocol designed especially for the use in GMPLS. The Link-Management Protocol LMP is used to monitor and control both the data plane and control plane between neighboring nodes. It sends regular keep-alive messages to check the state of a link. On the data plane LMP can preform fault isolation. It locates the cause of the fault by examining information from multiple hosts. Some of the extensions to RSVP-TE and CR-LDP were made on behalf of LMP. LMP will be described in detail in chapter 2.4.2

Establishing LSP's within GMPLS is still very similar to MPLS. Even though many changes where made in both the supported technologies as well as the protocols, the main difference is that multiple LSP's in GMPLS can reside within other LSP's. A certain form of hierarchy is established. Since LSP's can only be formed between two identical interfaces, we need

multiple LSP's throughout a network to form one end-to-end LSP over multiple network technologies. To achieve this LSP can only be set up between identical interfaces.

When trying to set up a LSP between two end nodes, say host A and host B. The key is to be sure they operate on the same interface. If somewhere between them they are connected via a different technologies, a LSP over those technologies needs to be set up before any end-to-end path can be established. In this case a PATH/Label Request message will be send out by host A with destination host B. On LSR A which for instance connects the Packet Switched capable network on which host A resides to a TDM network, a PATH/Label Request message with destination host B will arrive. Upon receiving this message LSR A will first send out his own PATH/Label message request towards LSR B. After a TDM based LSP is activated between LSR A and LSR B, LSR A will forward the Path/Label Request message from host A to host B. This message will now arrive at host B and a LSP between the two hosts can be established.

Where in MPLS the Data-plan and control plane are logically separated, in GMPLS they can also be physically separated. This is partially needed because of diversity of technologies GMPLS can control. For instance, when using wavelength switching one can not have the control plane and data plane on the same physical link. On the control plane it is mandatory to have Layer 3 functionality to support OSPF or ISIS, where on the data plane one could actually have a fully layer 2 switched network. The control network does not need to have an the same topology as the data network and it will not not matter on what kind of network technology the data is transported.

Two new features of GMPLS are the usage of bi-directional LSP's and the Suggested label. Where within MPLS it was not possible to set up a bi-directional path and one would need a second uni-directional path to ensure availability to the enduser. When using GMPLS it is no longer necessary to use a second path to maintain availability. This can be achieved by setting up a bi-directional path. The usage of the Suggested Label helps GMPLS LSP setup time to go down. By using a suggested label a upstream system can already enable its own part of the configuration, without waiting on the label from the requesting side. This will speed up the process, but it only works if the requesting side agrees upon the suggested label when receiving it. If not, the label will be changed and the process will still take all the time necessary to complete the path. By using both bi-directional paths networks not only converge faster, but in case of fail-over the recovery time will be shortened as well.

Fail-over should always be the last resort no matter what technology one is looking into. Reliability is far more important then ability by fail-over to overcome failures. Still, having fail-over systems in place is preferable. One of the many improvements of GMPLS is the option for automated fault management. To be able to maintain end-to-end LSPs over multiple smaller LSP's one needs a way to detect, localize and resolve the failing channel or link. Notification about the fault is a key step in this process. Chapter 2.4.2, LMP will describe this process in more detail.

To protect against failing channels or links GMPLS holds two forms of protection. First we have span protection, in which OSPF-TE or ISIS-TE extensions advertise the link-protection-type parameter with span protection while computing the received routes. RSVP-TE then sends out a signal to establish the backup paths. The second form of path protection in GMPLS reserves an extra path through the network. Shared-risk link groups are optional mechanisms to make sure that no secondary link is part of the original LSP. Again OSPF-TE and ISIS play a big part in this since network topology knowledge is required.

2.3 GMPLS Routing

GMPLS can not exist without its capabilities of Traffic Engineering TE. It is this specific functionality which makes GMPLS and before that MPLS so strong. Routing within MPLS and GMPLS is not used for the same reason as in normal Layer 3 networks. The basic idea of routing in Layer 3 networks is the computation of the shortest path to any destination. Each router is then concerned for finding the best possible connection and sending the data to the next hop. From that point on it is that next hop who needs to decide what to do with the data.

Within GMPLS we see that routing has been given different responsibility. It is used to share information about the network topology and QoS configuration of the underlaying data plane. It is this data plane in which GMPLS is interested. The topology of the OSPF or ISIS network in the control plane is irrelevant to GMPLS but is at the same time imperative to its signaling and control elements. Since the control plane can be separated from the data plane, GMPLS uses the OSPF or ISIS topology knowledge to enable IP connectivity between the GMPLS nodes and deliver to them that what is important, TE and link state information concerning the data plane.

One of the few requirements that GMPLS makes towards the control plane network is that it needs to be an IP based network. This is because the routing protocols which GMPLS uses are IP based and cannot function elsewhere. TE information and link state information about the TE links is transported within special extensions to these protocols and enable Label Switched Routers LSR's to make decisions on LSP's. All this information is transparent to the routing protocols. To them this information is irrelevant and therefore will not influence their control plane routing decisions.

It speaks for itself to say that GMPLS networks are complex environments with many TE values to take into consideration. The fact that it can also span across multiple networking technologies does not help it either. To make sure all this information is distributed and processed correctly each control-host has to be able to receive, if required process and certainly forward all TE information. A host which processes TE information builds a Traffic Engineering Database TED. A TED holds the complete GMPLS network topology including all TE specifications that go with all links.

Since it is nearly impossible to describe all preferences and information on links, not one value can hold all information to do so. It is for this reason that TE extension came to be in the first place. The most common TE attributes are:

- Protection Type - This value describes which protection capabilities are in place for the specific link
- Shared Risk Link Groups - SRLG's are a set of links which are affected by failures of one of the other links in the group
- Link Switching Capabilities - LSC's state the type of interfaces the switch can pass on data
- Data Encoding Type - This value describes the encoding format of the user data
- Max unreserved - The Maximum unreserved LSP bandwidth field identifies the amounts of bandwidth which is available on a link for new LSP's. With this information LSR's can calculate the available bandwidth and make routing decisions
- Resource Class - This field can influence the computation of certain links within new LSP by qualifications

GMPLS can use all these and more TE values to influence the decision making process for setting up new LSP's but it is important to remember that these values are not all mandatory. Based on the requirements set when requesting a LSP, some or all of these values will be taken into consideration for path computation.

2.3.1 Routing : OSPF-TE

OSPF-TE as mentioned before is the TE version of OSPF which is can be used by GMPLS to transport TE information to GMPLS LSR's without requiring the design of a new routing protocol. OSPF-TE uses a special LSA type to transport GMPLS information to all LSR's. [5] The OSPF-TE LSA type is a type 10 LSA which means that it is limited to the area in which it resides. The LSA itself is transparent, or *opaque* to the OSPF routers but they all forward it to other OSPF routers in the area depending on the OSPF network. If there is a Designated router in the network, this will receive all these LSA's and then propagate them through the network. The payload of these LSA's consists of a set of Type-Length-Value blocks TLV's which in itself can nest one ore more TLV's themselves. There are two top-level TLV's defined at this point:

- Router Address
- TE Link

In any OSPF-TE LSA there can only be one top-level TLV present. The Router Address is used to advertise a LSR's routable IP address. This in most cases will be a Loopback address which will be always available. The address can be uniquely for OSPF so that the LSR, if also running ISIS-TE it can have a different address. Then both addresses can be used to set up a connection between OSPF and ISIS and exchange TE information. This will enable the LSR to build one TED which holds all information of the network. The TE Link TLV holds information on the TE links in the network. All attributes except for the top-level will appear as separate sub-TLV's.

There are some restrictions to the use of TLV's in OSPF-TE in regards to for instance the Router Address. Any LSR can only submit one Router Address per protocol, next the TE links advertised can not be separated from the Router ID which therefore makes it practically impossible to control more then one GMPLS data switch in the data plane. Of course there are workarounds to overcome these problems, but they require some high impact changes to the configuration of the LSR's.

There are two ways to overcome this problem. The first is to appear to the OSPF domain as multiple routing controllers. This can be achieved by configuring separate Router ID's and Router Addresses for both controllers. But the only way to configure this one needs to configure multiple virtual connections for the control plane which is fairly complex. The second way to control more then one switch is to make the GMPLS data switches appear as one virtual switch. The switches need to be directly connected in some way and the only addresses advertised in the OSPF-TE messages will be the outside addresses of the virtual switch. This configuration requires great care to make sure it will not affect the GMPLS process. The next RFC for GMPLS will probably contain a solution to this problem but until that time these are the only ways to control multiple switches with just one controller.

2.4 Signaling and Management

Routing is used to create an awareness of the network topology and its link specifications. Signaling on its turn is used to set up and tear-down the actual LSP's on top of that topology. Signaling will not work if it cannot reach the destination and all hops in between to set up the path. Therefore routing must work before signaling can work at all. For Signaling GMPLS uses one of two protocols as stated before, RSVP-TE or CR-LDP. The latter will not be described in this paper because it was not used in any of the tests. For the test configuration see chapter 4 and 5.

Where signaling is responsible for setting up and tearing down the actual LSP's, LMP is responsible for maintaining active knowledge of the state a LSP is in, the state of the connections and the options for fail-over. LMP is a new protocol introduced as part of GMPLS.

2.4.1 RSVP-TE

The Resource reSerVation Protocol RSVP [6]is capable of providing custom levels of QoS for traffic flows for hosts by making the necessary resource reservation in the network. RSVP is used by many real-time and interactive applications such as IP telephony and video conferencing to assure the necessary QoS on the links. It guarantees that the requesting host or application will get at the very least the minimum values of what it requested. These are exactly the features GMPLS needs to maintain its LSP's and the choice to use this existing protocol was therefore fairly obvious. Still some changes were needed to accommodate MPLS [7] and later on GMPLS [8].

In RSVP-TE in a tunnel the source of the tunnel defines the exit point of the tunnel by stating the destination. When requesting a tunnel a user or application will also apply for special QoS properties. The network will have to try to accommodate the tunnel as requested with the appropriate QoS values. To make sure a tunnel is not mistaken for another tunnel, every tunnel is identified by its destination with 16-bit Tunnel Identifier. This way multiple tunnels can run to the same destination. By using the Extended Tunnel Identifier the requesting system can prevent other sources to use this same tunnel for sending information to that same destination. In general the source will use one of its own IP addresses for this identifier. If a source does not mind sharing the tunnel it is possible to share this with other systems.

A session object is added to the RSVP-TE messages to identify to what session the RSVP-TE message belong. Important to realize is that a tunnel in RSVP-TE is not the same as a LSP. A tunnel can hold multiple LSP's and therefore to identify each unique LSP RSVP-TE uses a 16-bit LSP-ID. The Sender-Template object holds elements that are used to identify LSP's within different sessions. When including the Sender Template object in an RSVP-TE message, it restricts the message to just this one LSP. GMPLS uses the exact same object as MPLS.

RSVP-TE sends its signaling messages over TCP port 3455 between signaling controllers. Where in MPLS RSVP-TE messages automatically followed the data path, in GMPLS this tends to be a little more complicated. Since the Data- and control plane can be physically separated, it is important for RSVP-TE to make sure signaling information is send to the correct LSR's and finally the destination.

RSVP-TE signaling is used for several purposes. First of all the LSP Establishment. A LSR which wants to set up a LSP to another LSR will send out an LSP setup message. This is a RSVP Path message and it is send out to the next hop LSR between itself and the destination. A LSP can only become active after receiving a LSP Accept message. This is accomplished with a RSVP ResvMessage from the downstream node which holds the label to be used on the LSP and to confirm the TE properties as configured. The Accept messages can only be returned to the requesting LSR after the destination has handed out the first Accept message. It will be handed down hop by hop until it reaches the source.

From that point on the LSP can be Active and data can flow. By doing this, a failed hop in the chain will not cause the requester to first think the LSP is active just to be torn down again after being unable to set up the LSP. Furthermore it allows GMPLS to try different routes to a destination in case the first one fails.

After the LSP becomes active it is important to maintain the LSP and keep data flowing until no longer required. Signaling has to ensure the LSP is still available and if RSVP-TE notices that a LSP has gone down, it has to notify the control plane of the failing link. LMP can then try to locate the cause and start rerouting for the LSP. To achieve this, RSVP-TE will retransmit Path messages after LMP orders a new route to a destination. The new LSP will be established the same as before and traffic will start flowing again.

The LSP tear-down process has been extended for GMPLS as well. Under normal conditions the LSR which requested the LSP will send out an LSP Downstream release message telling all downstream hosts that the LSP will be deleted and all resources need to be made available again for new LSP's. This is a one-way flow of LSP tear-down messages. With GMPLS RSVP-TE extensions there is also the possibility to have an Upstream Release mechanism as well. This any upstream LSR of the LSP to do exactly what the source LSR could already do before, namely tear-down the LSP. This is achieved by the usage of the RSVP PathError message with a new flag called the *Path state removed*.

2.4.2 Link Management: LMP

The Link Management Protocol LMP [9] was introduced as a part of GMPLS. In its basic function it is use by a controller to manage data links but it also supports the routing controllers that manage the data switches. It performs fault localization and several protection mechanisms. LMP is a UDP port 701 point-to-point application protocol, which means that LMP's scope is only between two neighboring systems.

LMP was created to overcome the problem which administrators had caused by the manual LSR configuration to identify Data-channels on the data plane from Control-channels. This needs to be done so that the control plane protocols understand what links are available. LMP automated this process in such a way that it allows switches to discover the capabilities and identifiers of links on the data plane and perform a number of management and control functions.

Control-channels are used for carrying all control plane signaling and routing information. They are links connecting the control plane nodes on which LMP is used and require unique identifier addresses to be configured on the channel end points. For LMP to function a minimum of one active Control-channel between two control systems is required but multiple may exist. The LMP Node-ID has to be unique in the GMPLS network where as the Control-channel ID CCID only needs to be unique on the host on which it resides.

The first function of LMP is Control-channel Management. LMP automatically set up Control-channels by sending out LMP Config messages on all its Control-channel links. In case of receiving a responding ConfigAck or a Config message a handshake model will step into place where LMP will negotiate LMP parameters. Since multiple Control-channels can exist between any two nodes in a GMPLS network the ConfigAck messages will consist of Control-channel, node and message ID's to make sure messages are not misplaced. Once the LMP connection is running Hello messages will enable LMP to check on the availability of the channel. In case of the Hello dead-interval to expire LMP will start using a secondary Control-channel if available.

After nodes have a Control-channel connection with all their neighbors the process of Link Discovery and Link Verification will begin on the data links. In the beginning a host knows what data interfaces are configured on the system but has no clue what their state is. All interfaces are configured with local identifiers to maintain an overview but to successfully set up LSP's it will need to know this information from the connecting systems. The LMP link verification is used to verify the Data-channels existence and check if it is active, but it is also used to match source Link ID's to its own local Link ID's.

Once both the Control- and Data-channels are established the Link Property Summarization information will be exchanged. Since connecting links might have different QoS properties, it is important to agree on what values to use. Next this information can be used to verify the integrity of the link in case the Link ID's are configured as well as discoverer. In this case LMP can check if both ID's are the same. Information exchanged during this process next to the Link ID includes a link type, wether the link is currently allocated for user traffic, it is in failed-state or the specific QoS settings of a link.

An optional but very important component of LMP is obviously Fault Isolation. When errors occur on a network, it is important to know where it originated from. Since GMPLS uses technologies which do not check data integrity before forwarding it, think about optical switches, the chances of faults to occur long before being detected by the receiving end are quiet big. LMP can isolate and report faults to the control plane and allows GMPLS to reroute traffic over a different link. When a receiver detects that a fault has occurred the receiving system will start sending out LMP ChannelStatus messages upstream on the Control-channel. Upon receiving one of these an upstream node will start checking the data-channels health by inspecting its signal. This process will repeat itself until reaching the faulty link or the source of the initial data transport.

A final part of LMP but not required by the RFC standard is Authentication. LMP specifications advises upon using IPsec to ensure that message senders can not be spoofed.

3 DRAGON

In this chapter an introduction to the DRAGON software suite will be given and the elements the software suite is build on will be described. This chapter is a summary of multiple documents, papers written and presentations held about the DRAGON software suite. The references used are listed in the references. DRAGON is the abbreviation for Dynamic Resource Allocation via GMPLS Optical Networks. It is the name of a software project for dynamic resource provisioning across heterogeneous network technologies and vendor equipment. The main contributors to the DRAGON software suite are the University of Maryland UMD, Mid-Atlantic Crossroads MAX, University of Southern California Information Sciences Institute East USC/ISI and the George Mason University GMU . To make all this possible the project is funded by the National Science Foundation as part of the Shared Cyberinfrastructure Division SCI Experimental Infrastructure Network EIN program. In this chapter we will discuss the elements which the DRAGON software suite is build of and give a general description of the DRAGON software suite.

3.1 Elements

The DRAGON software suite was designed to operate within the control plane of GMPLS networks. The control plane architecture consists of four key elements. The Client System Agent CSA, Network Aware Resource Broker NARB , Virtual Label Switch Router VLSR and the Application Specific Topology Builder ASTB. All these elements will be briefly addressed in the following paragraphs. Within this research the main focus was on the VLSR and therefore it will be discussed in more detail in this chapter.

3.1.1 CSA

A CSA terminates data connections. In the context of GMPLS a CSA is the end system which could request a network service. This could be everything from a telescope to complete computer cluster.

3.1.2 NARB

A NARB is used represent an autonomous system AS or domain. The key functions of a NARB are mainly high-level functions like topology abstraction, inter-domain path computation and inter-domain routing. A subcomponent of a NARB is the Resource Computation Element RCE which executes path computation tasks. RCE supports a NARB with a raw resource database and path computation. A NARB exchanges topology information across domains for inter-domain routing and LSP provisioning. A NARB

shares its link state database containing OSPF and TE information about his network to other NARB's in neighboring GMPLS networks.

The functions of a NARB can be compared to the functions of an ASBR within an OSPF network. A NARB provides services to the other systems by sharing the information about the GMPLS network and the neighboring networks. It can be queried by other systems about the availability of TE paths between a certain source and destination. These systems can connect to a server port, set up a TCP connection and can query a NARB. A NARB itself has connections with the RCE, OSPF daemon and other NARB's. Figure 1 shows a NARB representing an AS and having connections with two peer NARB's. The figure is borrowed from the presentation *GMPLS Tutorial and R&E Network Implementation* held by Chris Tracy at the University of Amsterdam on April 19th 2006.

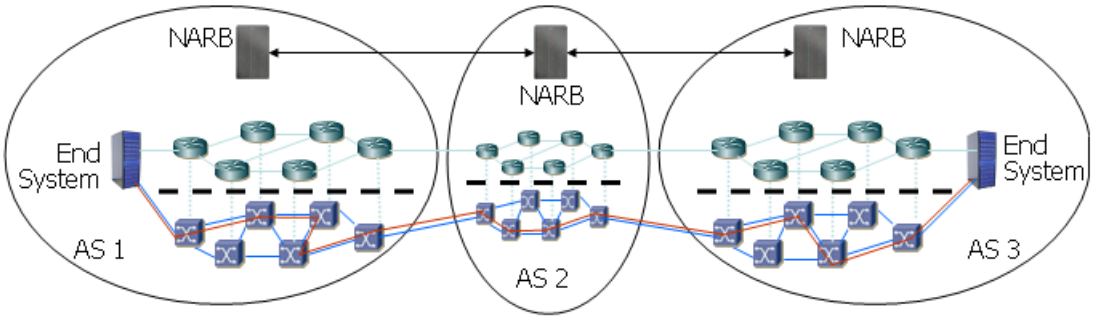

Figure 1: Overview of the NARB

3.1.3 VLSR

GMPLS has not yet been implemented on large a scale. There are still a lot of non GMPLS capable switches in use. To overcome this limitation the DRAGON protocol suite uses the VLSR. The V in VLSR stands for virtual and in this context this means that the VLSR acts as a GMPLS switch although in fact it is not. The VLSR has been developed for using vendor equipment in GMPLS networks which are non GMPLS capable. A VLSR is the combination of a PC running the DRAGON software suite and a non GMPLS capable switch.

A VLSR is used to control different kinds of switches like for instance Ethernet, TDM or Optical switches. What a VLSR does besides participating in the GMPLS protocols is translating GMPLS commands into switch specific commands like SNMP, TL1, CLI, XML, or similar protocol. By the use of these commands a VLSR can control the switch and for example set a switch port in the specific VLAN. To communicate with other VLSR's and CSA's a VLSR uses the routing protocol OSPF-TE and path signaling protocol RSVP-TE

which were explained in the chapter about GMPLS. Figure 2 displays the VLSR's controlling non-GMPLS capable switches within in a GMPLS network. The figure is borrowed from the presentation *GMPLS Tutorial and R&E Network Implementation* held by Chris Tracy at the University of Amsterdam on April 19th 2006.

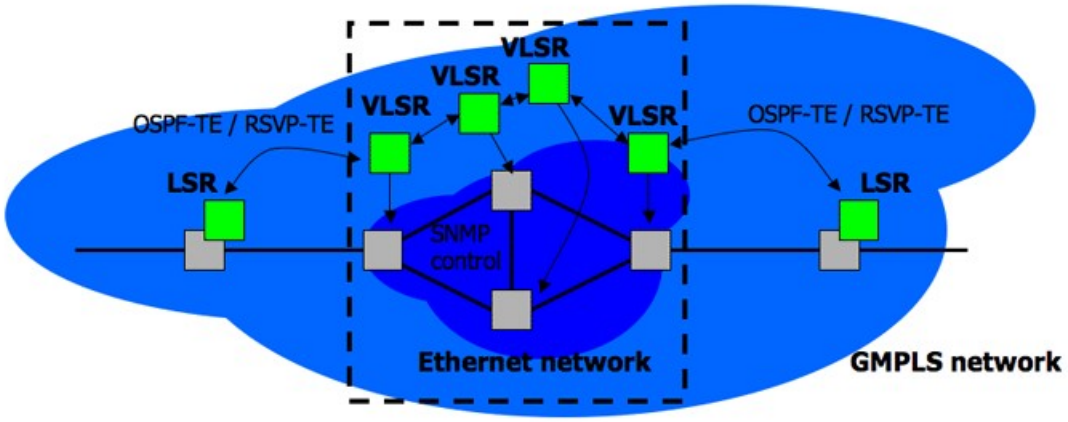

Figure 2: Overview of GMPLS network controlled by VLSR's

A VLSR uses OSPF-TE to get familiar with the control plane network and to inform the VLSR's and CSE's in the control plane about the TE network links. A VLSR uses the OSPF-TE LSA's to send information about the TE links. Information that could be send over the control plane is information about upcoming and down going LSP's. The OSPF-TE works with two daemons called OSPFD and zebra. Zebra, or GNU Zebra [11], is routing software for managing TCP/IP based routing protocols like RIP, BGP and OSPF. The DRAGON software extends the OSPF routing daemon with Traffic Engineering TE information like bandwidth, WDM and TDM used by GMPLS.

A VLSR uses RSVP-TE for signaling and setting up LSP's within the GMPLS network. The RSVP-TE protocol originates from the Technische Universitt Darmstadt's KOM-RSVP [12]. The DRAGON software extends the KOM-RSVP signaling protocol with support for RSVP-TE, GMPLS, Q-Bridge, SNMP and VLAN control.

3.1.4 ASTB

Some applications have specific demands concerning the LSP that needs to be created. The DRAGON software suite has an ASTB API which enables applications to request a dynamic path trough the DRAGON network between a source and destination pair identified by their IPv4 addresses. A ASTB utilizes the services of a NARB to determine if the requested network path is available and can offer the requested AAA and schedule

constraints. The DRAGON software suite includes a standard message set API and a web-interface.

3.2 Extendability and Developments

The DRAGON software suite is being developed under the GNU General public license [10]. The source code can be viewed, changed for own use. To extend the software suite with more functionality one can write their own add-ons. The DRAGON project is still very active and the software suite is being development as we speak. The latest version of the software suite can be downloaded at: http://Dragon.maxgigapop.net/public/Dragon-sw-vlsr-daily.tar.gz

At this moment the DRAGON Project is working on implementing Virtual switches as well as link Bundeling. Further developments are at the time of the release of this paper unknown.

4 Testing

In this chapter an overview will be given on the infrastructures we created to test the DRAGON software, the tests we performed and the configuration of these infrastructures. To test the DRAGON software suite multiple GMPLS infrastructures were created. In the short time available the DRAGON software was tested in two different setups. In the following paragraphs the configuration of the infrastructure and the performed tests will be described in detail. The systems in our network are currently running the DRAGON software under Scientific Linux. All Ethernet switches used for GMPLS are Raptor ER-1010 switches.

4.1 Scenario 1: Basic infrastructure

The basic infrastructure as presented in the VLSR implementation guide [22] available at the Mid-Atlantic Crossroads DRAGON Project website was used as a basis for this setup. In this infrastructure three machines and one Ethernet switch are being used. Two function as CSA's and the third as VLSR. Figure 3 shows an overview of the basic infrastructure used during this test scenario.

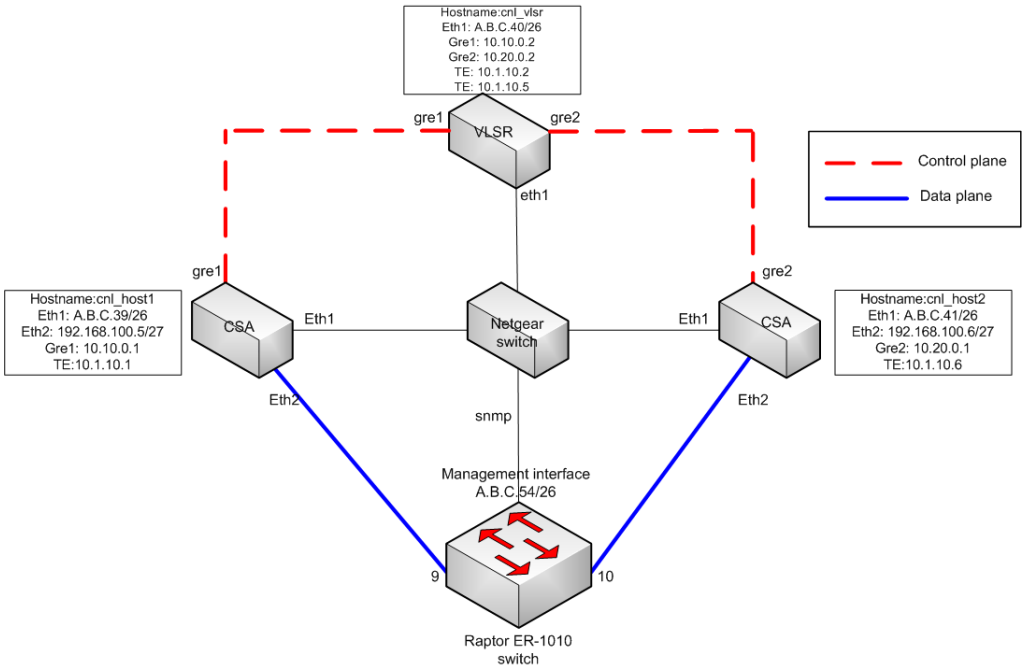

Figure 3: Overview of our basic infrastructure

The primary Ethernet devices of the CSA's are part of the control plane. The CSA's and

the VLSR's are all connected via a Netgear hub and are part of a bigger cluster which will not take any part in the tests. To separate the control plane from the rest of the cluster GRE tunnels were created between the CSA's and the VLSR. The Raptor switch is connected to the hub as well and enables the VLSR to control the switch by sending SNMP commands to the management interface of the switch.

The data plane exists of two CSA's and the switch. The secondary Ethernet interfaces of the two CSA's are part of the data plane and are connected to one of two switches. The CSA's are connected on port 9 and 10 of the switch.

4.1.1 Tests

Test 1: Setting up a LSP

In this test a LSP will be created between the two CSA's via the Ethernet switch. The DRAGON software suite and especially the VLSR will be researched in how it handles the request of the CSA to create the LSP.

Test 2: Interruption of a LSP

In this test a forced interruption of the LSP will be tested. The DRAGON software suite will be tested to see how it handles the sudden interruption of the LSP.

4.2 Scenario 2: Extended infrastructure

For the next tests the basic infrastructure was extended with a second switch. To accommodate the second switch another VLSR is needed. Now the infrastructure consists of four Linux servers and two Ethernet switches. Two of the servers function as CSA's, were the other two as function as VLSR's. Figure 4 shows an overview of the extended infrastructure used during Scenario 2.

In this infrastructure the control plane consists of two CSA's and two VLSR's. The primary Ethernet device of the CSA's are again part of the control plane. The CSA's and the VLSR's are connected via a Netgear hub. GRE tunnels were created between the CSA's and the VLSR's and between the VLSR's them selves similar to Scenario 1. Both switches have a connection with the hub to allow SNMP management by the VLSR's.

The data plane exists of the two CSA's and the two switches. The secondary Ethernet interfaces of the CSA's are part of the data plane and are each connected to one of two switches. The CSA's are connected on port 9 of the switch they are connected to. The coper link between the two switches are connected on port 1 on both switches.

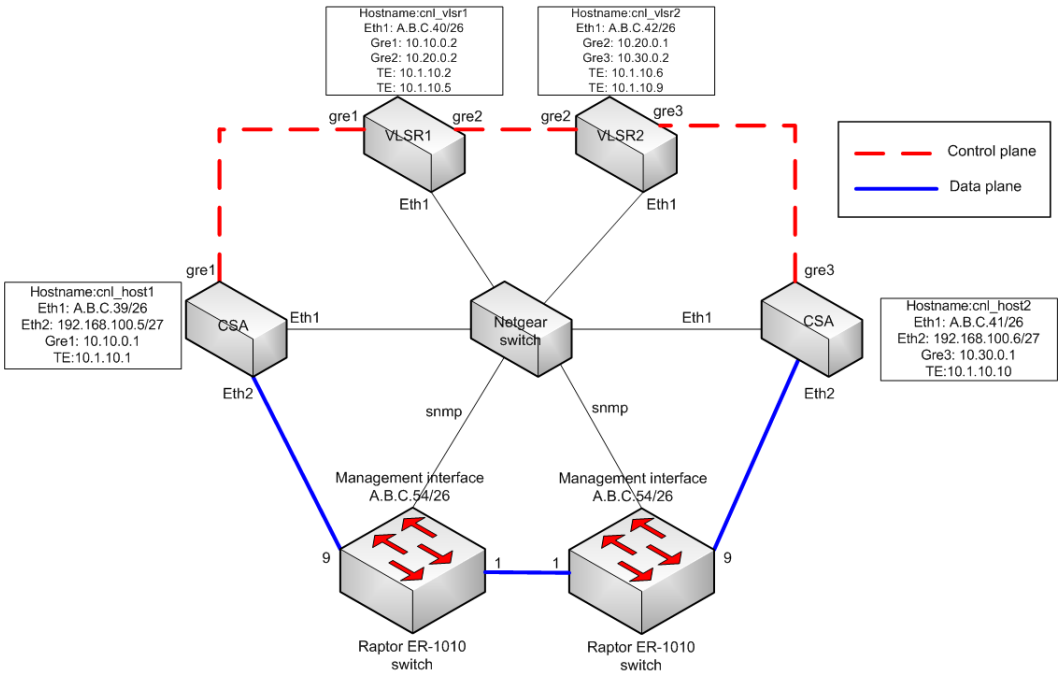

Figure 4: Overview of the extended infrastructure

4.2.1 Tests

Test 1: Setting up a LSP

In this test a LSP will be created between the two CSA's passing through both Ethernet switches. The DRAGON software suite and especially the VLSR's will be monitored to see how they handle the request of the CSA to setup the end-to-end LSP.

Test 2: Interruption of a LSP

During this test again a forced interruption of the LSP will be performed to see how the DRAGON software and the VLSR's handle the sudden interruption of the LSP.

Test 3: Setting up multiple LSP's over one link

Multiple LSP's will be set up between the CSA's over a single Ethernet connection between two switches. The test will show if it is possible to have multiple applications requesting a LSP to a CSA on the other side when there is only one single Ethernet connection between the two switches.

4.3 Configuration

In this section a short summary will be given of the configurations which are necessary to make the infrastructure work. The configuration files will be added in the appendixes. It will only discuss the creation of the GRE tunnels and the configuration of OSPF-TE. The configuration of the Zebra, DRAGON and the RSVP daemon are very basic and do not need further explanation. The configuration files used are the configuration files used to set up the basic infrastructure.

4.3.1 GRE tunnels

The first step in setting up the infrastructure is to separate the control plane from the rest of the cluster by using GRE tunnels. The bash script below will create a GRE tunnel between the CSA on the right and the VLSR. On every CSA and VLSR a similar script was created to set up the appropriate GRE tunnels.

```
#!/bin/sh
touch /var/lock/subsys/local
/sbin/modprobe ip_gre
/sbin/ip tunnel del gre2
/sbin/ip tunnel add gre2 mode gre remote A.B.C.40 local A.B.C.41 ttl 255
/sbin/ip link set gre2 up
/sbin/ip addr add 10.20.0.1/30 dev gre2
/sbin/ip route add 10.20.0.2 dev gre2
```

4.3.2 OSPF-TE

The most important part of the configuration is the configuration of the OSPF-TE daemon. This daemon will advertise control plane information and TE information to the control plane. In this paragraph both the configuration of the CSA and the VLSR will be discussed.

CSA

The following example is a part of the *ospfd.conf* configuration file for the CSA on the right.

```
!
router ospf
 ospf router-id A.B.C.41
 network 10.20.0.0/30 area 0.0.0.0
 ospf-te router-address A.B.C.41
!
ospf-te interface gre2
 level gmpls
 data-interface ip 10.1.10.6
```

```
 swcap l2sc encoding ethernet
 max-bw 125000000
 max-rsv-bw 125000000
 max-lsp-bw 0 125000000
 max-lsp-bw 1 125000000
 max-lsp-bw 2 125000000
 max-lsp-bw 3 125000000
 max-lsp-bw 4 125000000
 max-lsp-bw 5 125000000
 max-lsp-bw 6 125000000
 max-lsp-bw 7 125000000
exit
!
```

In this example the router-id will be set to A.B.C.41. This is the IP address of the management interface. The next step in the configuration file configures the daemon to advertise the 10.20.0.0/30. The 10.20.0.0/30 network is the subnet used within the GRE tunnel between the CSA and the VLSR. The next part of the configuration file is the OSPF-TE configuration. This information is used while setting up LSP's. The OSPF-TE router-interface must also be given a label and that label will be equal to the router-id.

The next step is configuring the interfaces which take part in the OSPF-TE network, in this case this interface gre2. To specify the route of a LSP an Explicit Route Object ERO will be created which is a list of data interfaces it will pass from end to end. The data interface is the the label for the connection to the data plane. The data interface in this infrastructure is set to be 10.1.10.6. In the next line the encapsulation is set to layer 2 switching l2c and the encoding is Ethernet because in the infrastructure Ethernet switches are used. Every LSP or link has a certain bandwidth. With the max-bw, max-rsv-bw and the max-lsp-bw configuration the maximum bandwidth, maximum reserved bandwidth and the maximum bandwidth for an LSP can be set.

VLSR

The OSPF-TE configuration of the VLSR is quite similar to the configuration of the CSA except from the data interfaces. The following example shows two parts from the *ospfd.conf* used by the OSPF-TE daemon.

```
ospf-te interface gre1
 level gmpls
 data-interface ip 10.1.10.2 protocol snmp switch-ip A.B.C.54 switch-port 9
 swcap l2sc encoding ethernet
!
ospf-te interface gre2
 level gmpls
 data-interface ip 10.1.10.5 protocol snmp switch-ip A.B.C.54 switch-port 10
 swcap l2sc encoding ethernet
!
```

The VLSR has two data interfaces. One data interface which is connected to the left CSA via switch port 9 and one data interface connected to the right CSA via switch port 10. As mentioned in the previous chapters the VLSR uses SNMP to control the switch. With these configuration lines the VLSR is configured to have two data interfaces with IP addresses 10.1.20.2 and 10.1.10.5 and can use the SNMP protocol to control the switch with IP address A.B.C.54.

5 Results

During the short time in which the research project was conducted, a small number of tests on the DRAGON software were completed. In two infrastructures LSP's were created and torn down. In this chapter the results of these tests will be described. Only the tests in the extended infrastructure will be discussed in detail because the tests in the first infrastructure had similar results.

5.1 Creating a LSP

In this test a LSP was created between the two CSA's passing through both Ethernet switches.

5.1.1 Execution

To set up a LSP the Command Line Interface CLI was used. To get into the CLI a telnet session had to be set up to the machine on which the DRAGON daemon is running on at port 2611. In the example below a LSP between the CSA on the left and on the right is created.

```
cln_host1-Dragon> edit lsp test
  cln_host1-Dragon(edit-lsp-test)# set source ip-address A.B.C.39 lsp-id 1000 destination \
                                   ip-address A.B.C.41 tunnel-id 2000
  cln_host1-Dragon(edit-lsp-test)# set bandwidth gige_f swcap l2sc encoding ethernet gpid ethernet
  cln_host1-Dragon(edit-lsp-test)# set vtag any
  cln_host1-Dragon(edit-lsp-test)# exit

cln_host1-Dragon> commit lsp test
```

First the new LSP has to be given a name, in this case *test*. In the next command the source IP address, LSP id, destination IP address and tunnel ID are defined for the new LSP. Next the switching type (swcap) and encoding type are defined. In the infrastructure only Ethernet is used so the encoding type is set to Ethernet and the switching type is set to Layer 2 Switching L2sc. With the DRAGON software it is possible to chose which VTAG should be used for the LSP. In this test the DRAGON software decided which VTAG it wanted to use by using the command *set vtag any*. Finally the LSP was committed. To show wether the LSP is in service the command *show lsp* is used.

```
cln_host1-Dragon> show lsp

                          **LSP status summary**

Name        Status     Dir   Source (IP/LSP ID)  Destination (IP/Tunnel ID)
------------------------------------------------------------------------------
test        In service  =>   A.B.C.39        A.B.C.41
                             1000                2000
```

5.1.2 Results

During the tests packet dumps were made of the data crossing the control plane at the CSA's and the VLSR's. The packet dumps are added in the appendixes. After the commit statement was executed the CSA on the left send out a RSVP path message to the first VLSR with the destination set to the target CSA. Both VLSR's forwarded the path message because they were not the destination for the message. When the CSA on the right received the path message it replied to the message with a resv message and send it to the closest VLSR. After receiving the resv message the VLSR send SNMP commands to the switch to place the ports in the correct VLAN. The VLSR forwarded the message to the other VLSR because it again was not the destination of the message. This VLSR also sends SNMP port configuration commands to the switch and finally sends a resv message to the left CSA. From this point the LSP was active and the LSP could be used. To test if the LSP had been successfully created, a ping request was executed from the CSA on the left to the IP address 192.168.100.6 which is the IP configured at the secondary Ethernet interface of the right CSA. This interface is connected to the data plane. The CSA on the right responded to the ping request was only succesful after the LSP was in service.

5.2 Interruption of a LSP

During this test the a forced interruption of the LSP was performed to see how the DRAGON software and the VLSR's handle the sudden interruption of the LSP.

5.2.1 Execution

To force the breakdown of the LSP the switch port 9 of the left Ethernet switch was placed in a different VLAN. The second part of the test was to delete the LSP in the CLI on the CSA who requested the LSP. The command used for deleting the LSP was *delete lsp test*.

5.2.2 Results

During the tests packet dumps were made of the data crossing the control plane at the CSA's and the VLSR's. The packet dumps are again added in the appendixes. When the switch port was placed in a different VLAN the DRAGON software did not send any RSVP path tear messages. When the LSP was deleted via the CLI however the CSA send a RSVP path tear message. This message was send to the VLSR and the VLSR on its turn send the path tear message to the other VLSR and that VLSR to the destination CSA. After sending the path tear message both the VLSR's send SNMP commands to the switches to the set the switch ports back to VLAN 1.

5.3 Setting up multiple LSP's over one link

In this test multiple LSP's were set up between the CSA's over the single Ethernet connection between the two switches.

5.3.1 Execution

In this test first a LSP was created between the CSA on the left and the CSA on the right when this was active, the LSP from the VLSR on the left to the CSA on the right was set up. The following commands were executed on the CSA on the left to create a LSP to the CSA on the right.

```
cln_host1-Dragon> edit lsp
cln_host1-Dragon> edit lsp csa
cln_host1-Dragon(edit-lsp-csa)# set source ip-address A.B.C.39 lsp-id 1000 destination ip-address \
                                A.B.C.41 tunnel-id 2000
cln_host1-Dragon(edit-lsp-csa)# set bandwidth gige_f swcap l2sc encoding ethernet gpid ethernet
cln_host1-Dragon(edit-lsp-csa)# set vtag any
cln_host1-Dragon(edit-lsp-csa)# exit
cln_host1-Dragon> comm
cln_host1-Dragon> commit ls
cln_host1-Dragon> commit lsp csa
```

The following commands were executed on the VLSR on the left to create a LSP to the CSA on the right.

```
cnl_vlsr-Dragon> edit lsp vlsr
cnl_vlsr-Dragon(edit-lsp-vlsr)# set source ip-address A.B.C.40 lsp-id 1020 destination ip-address \
                                A.B.C.41 tunnel-id 2000
cnl_vlsr-Dragon(edit-lsp-vlsr)# set bandwidth gige_f swcap l2sc encoding ethernet gpid ethernet
cnl_vlsr-Dragon(edit-lsp-vlsr)# set vtag any
cnl_vlsr-Dragon(edit-lsp-vlsr)# exit
cnl_vlsr-Dragon> comm
cnl_vlsr-Dragon> commit ls
cnl_vlsr-Dragon> commit lsp
cnl_vlsr-Dragon> commit lsp vlsr
```

5.3.2 Results

During the tests packet dumps were made of the data crossing the control plane at the CSA's and the VLSR's. The packet dumps are once more added in the appendixes. After the LSP between the two CSA's was created, a ping was started between the two CSA's. After the other LSP was created and committed, the CSA on the left kept receiving ping responses from the CSA on the right. We were not able to test if the LSP between the VLSR and the CSA on the right was working but when we showed the LSP's that were active at the CSA on the right and both the LSP's were in service. To perform a full test

about setting up multiple LSP's over a single link and tunnel another CSA should be added to the network and then set up a LSP from this CSA to the CSA on the right as well.

5.4 Other tests

Besides the tests described above other tests were done also. In one of the tests multiple LSP's were set up over the single connection between the two switches. In this test LSP's were set up from the CSA of the left to the CSA of the right and from the VLSR on the left to the CSA on the right. When creating the LSP the same tunnel-id was used but the LSP ID's were unique. The results of these test were conclusive because in the test it was not possible to test if data could be send between the VLSR and the CSA. The LSP's were set up and the CLI at the CSA on the right showed both the LSP's were in service. Because the lack of time this test could not be fully completed the way it should. The use of another machine running the DRAGON software and functioning as a CSA from which a LSP was set up to the CSA on the right could prove if it is possible to set up and use multiple LSP's over a single connection between the two switches.

6 Conclusions and recommendations

Hybrid networks consisting of routed IP and Optical or lightpath sections have become a field of interest for SARA Amsterdam. The manual configuration required for setting up lightpaths and tunnels through these networks cause great concern within SARA when looking at the rate at which these networks are being used. For the future SARA expects that more research groups and applications will require their own configurations. For SARA in the future to be able to keep on using lightpaths and tunnels they need a method of automating this configuration process. Generalized Multi-Protocol Label Switching GMPLS is one of the many techniques researched by SARA to overcome this obstacle.

This research focus was on an Open-Source GMPLS product currently under development under by DRAGON Project. DRAGON is short for the Dynamic Resource Allocation via GMPLS Optical Networks and is the name for a software project for dynamic resource provisioning across heterogeneous network technologies and vendor equipment. The software suite was used to set up multiple test infrastructures in which the usability and RFC compliance had to be tested. Before test could be performed, first the GMPLS technology needed to be researched.

DRAGON was originally written to be used with FreeBSD but in this research it would be used with a Scientific Linux operating system. The limited amounts of documentation available on the DRAGON software suite were written for DRAGON running under freeBSD since this was the prefered Operating System. This meant the example commands were not immediately working on the machines used in this project and some reconfiguring was needed. Because of this it took more time to install the software suite and set up the test infrastructure.

During the project external factors like the delayed delivery of the Ethernet switches and usage of non compliant switches we were not able to create the test infrastructure within the first two weeks. Besides this factor the limited documentation which was available delay the setup of the infrastructure even more because it was not clear how to set up and configure the DRAGON software suite.

During the project more and more insight was gained about the software suite and with the help of Chris Tracy, a member of the DRAGON Project team, we were able to set up the DRAGON test infrastructures and create LSP's within these networks. The DRAGON software suite has a lot of potential but because the limited documentation the current features are difficult to oversee. The test showed that not all of the features of GMPLS are implemented into DRAGON. Within the DRAGON software suite link bundling is not possible yet and LMP has been implemented either. Because of this features such as like link protection and fail over techniques are not yet possible in the DRAGON software suite.

The test infrastructures created have shown a few features of the DRAGON software suite like setting up multiple LSP's over one physical link. Because of the lack of time the full

possibilities could not be tested. SARA can expand the infrastructure build in this research project and test the more advanced features. One of the elements SARA could add is the NARB to expand their infrastructure and set up paths trough other GMPLS networks outside the SARA network.

Further developing of the DRAGON software suite will have to lead to a fully operational an RFC compliant Open Source software suite for GMPLS. The first and so far only available package that will allow the usage of non GMPLS hardware to perform GMPLS functions. SARA will need to further research the possibilities of DRAGON on their newly created GMPLS infrastructure. By adding more complexity to the network they could set up a pilot network to resemble the current hybrid networks and their operational demands. This in turn will benefit their initial lightpath research and maybe even lead to the implementation of DRAGON in the future.

List of abbreviations

Abbreviation	Meanings
ASTB	Application Specific Topology Builder
AS	Autonomous System
CR-LDP	ConstRaint based Label Distribution Protocol
CCID	Control-channel ID
CLI	Command Line Interface
CSA	Client System Agent
DWDM	Dynamic Wavelength Division Multiplexing
ERO	Explicit Route Object
FEC	Forwarding Equivalence Classes
GMPLS	Geralized MultiProtocol Label Switching
ISIS-TE	ntermediate System-to-Intermediate System - Traffic Engineering
L2SC	Layer 2 Switching
LMP	Link Management Protocol
LSC	Link Switching Capabilities
LSA	Link State Advertisement
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	MultiProtocol Label Switching
NARB	Network Aware Resource Broker
OSPF-TE	Open Shortest Path First - Traffic Engineering
QoS	Quality of Service
RCE	Resource Computation Element
RSVP-TE	Resource ReserVation Protocol - Traffic Engineering
SNMP	Simple Network Management Protocol
SRLG	Shared Risk Link Groups
TDM	Time-Division Multiplexing
TE	Traffic Engineering
TED	Traffic Engineering Database
TL1	Transaction Language 1
TLV	Type-Length-Value
VLSR	Virtual Label Switching Router
XML	eXensible Markup Language

7 Appendix

7.1 Appendix A: Configuration files of the CSA on the left, host 1

```
################################################################################################
# Bash script to create the GRE tunnels
################################################################################################
#!/bin/sh
touch /var/lock/subsys/local

sudo /sbin/modprobe ip_gre

sudo /sbin/ip tunnel del gre1
sudo /sbin/ip tunnel add gre1 mode gre remote A.B.C.40 local A.B.C.39 ttl 255
sudo /sbin/ip link set gre1 up
sudo /sbin/ip addr add 10.10.0.1/30 dev gre1
sudo /sbin/ip route add 10.10.0.2 dev gre1

sudo /sbin/ifconfig


################################################################################################
# Bash script to create the packetdump
################################################################################################
#!/bin/sh
sudo /usr/sbin/tcpdump -i gre1 -s 0 -w /var/log/host1_gre1_03_07 &


################################################################################################
# Configuration file of dragon
# dragon.conf
################################################################################################
hostname cln_host1-dragon
password uva


################################################################################################
# Configuration file of the zebra ospf daemon
# ospfd.conf
################################################################################################
!
!zebra-ospfd configuration file for cnl_host1
!
hostname cnl_host1-ospf
password uva
log file /var/log/ospdf.log
!
interface gre1
 description GRE tunnel between cnl_host1 and cnl_vlsr1
 ip ospf network point-to-point
!
router ospf
 ospf router-id A.B.C.39
 network 10.10.0.0/30 area 0.0.0.0
 ospf-te router-address A.B.C.39
 ospf-te interface gre1
  level gmpls
```

```
  data-interface ip 10.1.10.1
  swcap l2sc encoding ethernet
  max-bw 125000000
  max-rsv-bw 125000000
  max-lsp-bw 0 125000000
  max-lsp-bw 1 125000000
  max-lsp-bw 2 125000000
  max-lsp-bw 3 125000000
  max-lsp-bw 4 125000000
  max-lsp-bw 5 125000000
  max-lsp-bw 6 125000000
  max-lsp-bw 7 125000000
 exit
!
line vty
!


##############################################################################################
# Configuration file of the RSVP daemon
# RSVPD.conf
##############################################################################################
interface gre1 tc none mpls
api 4000


##############################################################################################
# Configuration file of zebra
# zebra.conf
##############################################################################################
! -*- zebra -*-
!
! zebra sample configuration file
!
hostname cln_host1-zebra
password uva
enable password uva
!
! Interface's description.
interface lo
interface gre1
!
line vty
log file /var/log/zebra.log
```

7.2 Appendix B: Configuration files of VLSR 1

```
################################################################################################
# Bash script to create the GRE tunnels
################################################################################################
#!/bin/sh
touch /var/lock/subsys/local

sudo /sbin/modprobe ip_gre

sudo /sbin/ip tunnel del gre1
sudo /sbin/ip tunnel add gre1 mode gre remote A.B.C.39 local A.B.C.40 ttl 255
sudo /sbin/ip link set gre1 up
sudo /sbin/ip addr add 10.10.0.2/30 dev gre1
sudo /sbin/ip route add 10.10.0.1 dev gre1

sudo /sbin/ip tunnel del gre2
sudo /sbin/ip tunnel add gre2 mode gre remote A.B.C.42 local A.B.C.40 ttl 255
sudo /sbin/ip link set gre2 up
sudo /sbin/ip addr add 10.20.0.2/30 dev gre2
sudo /sbin/ip route add 10.20.0.1 dev gre2

sudo /sbin/ifconfig


################################################################################################
# Bash script to create the packetdumps
################################################################################################
#!/bin/sh
sudo /usr/sbin/tcpdump -i gre1 -s 0 -w /var/log/vlsr_gre1_03_07 &
sudo /usr/sbin/tcpdump -i gre2 -s 0 -w /var/log/vlsr_gre2_03_07 &
sudo /usr/sbin/tcpdump -i eth1 not port 22 -s 0 -w /var/log/vlsr_eth1_all_03_07 &
sudo /usr/sbin/tcpdump -i eth1 port 161 -s 0 -w /var/log/vlsr_eth1_snmp_03_07 &


################################################################################################
# Configuration file of dragon
# dragon.conf
################################################################################################
hostname cnl_vlsr1-dragon
password uva


################################################################################################
# Configuration file of the zebra ospf daemon
# ospfd.conf
################################################################################################
!
!zebra-ospfd configuration file for cnl_vlsr1
!
hostname cnl_vlsr1-ospf
password uva
log file /var/log/ospdf.log
!
interface gre1
 description GRE tunnel between cnl_vlsr1 and cnl_host1
 ip ospf network point-to-point
!
```

```
interface gre2
 description GRE tunnel between cnl_vlsr1 and cnl_vlsr2
 ip ospf network point-to-point
!
router ospf
 ospf router-id A.B.C.40
 network 10.10.0.0/30 area 0.0.0.0
 network 10.20.0.0/30 area 0.0.0.0
 ospf-te router-address A.B.C.40
!
ospf-te interface gre1
 level gmpls
 data-interface ip 10.1.10.2 protocol snmp switch-ip A.B.C.54 switch-port 9
 swcap l2sc encoding ethernet
 max-bw 125000000
 max-rsv-bw 125000000
 max-lsp-bw 0 125000000
 max-lsp-bw 1 125000000
 max-lsp-bw 2 125000000
 max-lsp-bw 3 125000000
 max-lsp-bw 4 125000000
 max-lsp-bw 5 125000000
 max-lsp-bw 6 125000000
 max-lsp-bw 7 125000000
 metric 10
exit
!
ospf-te interface gre2
 level gmpls
 data-interface ip 10.1.10.5 protocol snmp switch-ip A.B.C.54 switch-port 1
 swcap l2sc encoding ethernet
 max-bw 125000000
 max-rsv-bw 125000000
 max-lsp-bw 0 125000000
 max-lsp-bw 1 125000000
 max-lsp-bw 2 125000000
 max-lsp-bw 3 125000000
 max-lsp-bw 4 125000000
 max-lsp-bw 5 125000000
 max-lsp-bw 6 125000000
 max-lsp-bw 7 125000000
 metric 10
exit
!
line vty
!


##############################################################################################
# Configuration file of the RSVP daemon
# RSVPD.conf
##############################################################################################
interface gre1 tc none mpls
interface gre2 tc none mpls
api 4000
```

```
##############################################################################################
# Configuration file of zebra
# zebra.conf
##############################################################################################
! -*- zebra -*-
!
! zebra sample configuration file
!
hostname cnl_vlsr1-zebra
password uva
enable password uva
!
! Interface's description.
interface lo
interface gre1
interface gre2
!
line vty
!
log file /var/log/zebra.log
```

7.3 Appendix C: Configuration files of VLSR2

```
##############################################################################################
# Bash script to create the GRE tunnels
##############################################################################################
#!/bin/sh
touch /var/lock/subsys/local

sudo /sbin/modprobe ip_gre

sudo /sbin/ip tunnel del gre2
sudo /sbin/ip tunnel add gre2 mode gre remote A.B.C.40 local A.B.C.42 ttl 255
sudo /sbin/ip link set gre2 up
sudo /sbin/ip addr add 10.20.0.1/30 dev gre2
sudo /sbin/ip route add 10.20.0.2 dev gre2

sudo /sbin/ip tunnel del gre3
sudo /sbin/ip tunnel add gre3 mode gre remote A.B.C.41 local A.B.C.42 ttl 255
sudo /sbin/ip link set gre3 up
sudo /sbin/ip addr add 10.30.0.2/30 dev gre3
sudo /sbin/ip route add 10.30.0.1 dev gre3

sudo /sbin/ifconfig


##############################################################################################
# Bash script to create the packetdumps
##############################################################################################
#!/bin/sh
sudo /usr/sbin/tcpdump -i gre2 -s 0 -w /var/log/vlsr2_gre2_03_07 &
sudo /usr/sbin/tcpdump -i gre3 -s 0 -w /var/log/vlsr2_gre3_03_07 &
sudo /usr/sbin/tcpdump -i eth1 not port 22 -s 0 -w /var/log/vlsr2_eth1_all_03_07 &
sudo /usr/sbin/tcpdump -i eth1 port 161 -s 0 -w /var/log/vlsr2_eth1_snmp_03_07 &


##############################################################################################
# Configuration file of dragon
# dragon.conf
##############################################################################################
hostname cnl_vlsr2-dragon
password uva


##############################################################################################
# Configuration file of the zebra ospf daemon
# ospfd.conf
##############################################################################################
!
!zebra-ospfd configuration file for cnl_vlsr2
!
hostname cnl_vlsr2-ospf
password uva
log file /var/log/ospdf.log
!
interface gre2
 description GRE tunnel between cnl_vlsr1 and cnl_vlsr2
 ip ospf network point-to-point
!
```

```
interface gre3
 description GRE tunnel between cnl_vlsr2 and cnl_host2
 ip ospf network point-to-point
!
router ospf
 ospf router-id A.B.C.42
 network 10.20.0.0/30 area 0.0.0.0
 network 10.30.0.0/30 area 0.0.0.0
 ospf-te router-address A.B.C.42
!
ospf-te interface gre2
 level gmpls
 data-interface ip 10.1.10.6 protocol snmp switch-ip A.B.C.58 switch-port 1
 swcap l2sc encoding ethernet
 max-bw 125000000
 max-rsv-bw 125000000
 max-lsp-bw 0 125000000
 max-lsp-bw 1 125000000
 max-lsp-bw 2 125000000
 max-lsp-bw 3 125000000
 max-lsp-bw 4 125000000
 max-lsp-bw 5 125000000
 max-lsp-bw 6 125000000
 max-lsp-bw 7 125000000
 metric 10
exit
!
ospf-te interface gre3
 level gmpls
 data-interface ip 10.1.10.9 protocol snmp switch-ip A.B.C.58 switch-port 9
 swcap l2sc encoding ethernet
 max-bw 125000000
 max-rsv-bw 125000000
 max-lsp-bw 0 125000000
 max-lsp-bw 1 125000000
 max-lsp-bw 2 125000000
 max-lsp-bw 3 125000000
 max-lsp-bw 4 125000000
 max-lsp-bw 5 125000000
 max-lsp-bw 6 125000000
 max-lsp-bw 7 125000000
 metric 10
exit
!
line vty
!

##############################################################################################
# Configuration file of the RSVP daemon
# RSVPD.conf
##############################################################################################
interface gre2 tc none mpls
interface gre3 tc none mpls
api 4000
```

```
##############################################################################################
# Configuration file of zebra
# zebra.conf
##############################################################################################
! -*- zebra -*-
!
! zebra sample configuration file
!
hostname cln_vlsr2-zebra
password uva
enable password uva
!
! Interface's description.
interface lo
interface gre2
interface gre3
!
line vty
!
log file /var/log/zebra.log
```

7.4 Appendix D: Configuration files of the CSA on the right, host 2

```
##############################################################################################
# Bash script to create the GRE tunnels
##############################################################################################
#!/bin/sh
touch /var/lock/subsys/local

sudo /sbin/modprobe ip_gre

sudo /sbin/ip tunnel del gre3
sudo /sbin/ip tunnel add gre3 mode gre remote A.B.C.42 local A.B.C.41 ttl 255
sudo /sbin/ip link set gre3 up
sudo /sbin/ip addr add 10.30.0.1/30 dev gre3
sudo /sbin/ip route add 10.30.0.2 dev gre3

sudo /sbin/ifconfig


##############################################################################################
# Bash script to create the packetdump
##############################################################################################
#!/bin/sh
sudo /usr/sbin/tcpdump -i gre3 -s 0 -w /var/log/host2_gre3_03_07 &


##############################################################################################
# Configuration file of dragon
# dragon.conf
##############################################################################################
hostname cnl_host2-dragon
password uva


##############################################################################################
# Configuration file of the zebra ospf daemon
# ospfd.conf
##############################################################################################
!
!zebra-ospfd configuration file for cnl_host2
!
hostname cnl_host2-ospf
password uva
log file /var/log/ospdf.log
!
interface gre3
 description GRE tunnel between cnl_host2 and cnl_vlsr2
 ip ospf network point-to-point
!
router ospf
 ospf router-id A.B.C.41
 network 10.30.0.0/30 area 0.0.0.0
 ospf-te router-address A.B.C.41
 ospf-te interface gre3
  level gmpls
  data-interface ip 10.1.10.10
  swcap l2sc encoding ethernet
  max-bw 125000000
```

```
  max-rsv-bw 125000000
  max-lsp-bw 0 125000000
  max-lsp-bw 1 125000000
  max-lsp-bw 2 125000000
  max-lsp-bw 3 125000000
  max-lsp-bw 4 125000000
  max-lsp-bw 5 125000000
  max-lsp-bw 6 125000000
  max-lsp-bw 7 125000000
 exit
!
line vty
!


##############################################################################################
# Configuration file of the RSVP daemon
# RSVPD.conf
##############################################################################################
interface gre3 tc none mpls
api 4000


##############################################################################################
# Configuration file of zebra
# zebra.conf
##############################################################################################
! -*- zebra -*-
!
! zebra sample configuration file
!
hostname cln_host2-zebra
password uva
enable password uva
!
! Interface's description.
interface lo
interface gre3
!
line vty
log file /var/log/zebra.log
```

7.5 Appendix E: Packet dump VLSR1 setting up a LSP

```
No.     Time        Source      Destination     Protocol Info
210 55.315140   10.10.0.1       10.10.0.2       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
213 55.345050   A.B.C.40        A.B.C.54        SNMP     GET SNMPv2-MIB::sysDescr.0
214 55.345688   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-MIB::sysDescr.0
215 55.345841   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
216 55.346638   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
217 55.346734   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
218 55.347544   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
219 55.347631   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
220 55.348417   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
221 55.348468   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
222 55.349096   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
223 55.349779   10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
227 55.529803   10.20.0.1       10.20.0.2       RSVP     RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                          277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
228 55.558129   A.B.C.40        A.B.C.54        SNMP     GET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
229 55.559048   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
230 55.559276   A.B.C.40        A.B.C.54        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.2
231 55.578444   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.2
232 55.578603   A.B.C.40        A.B.C.54        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
233 55.593738   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
234 55.593817   A.B.C.40        A.B.C.54        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
235 55.597948   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
236 55.598077   A.B.C.40        A.B.C.54        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
237 55.603808   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
238 55.603876   A.B.C.40        A.B.C.54        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
239 55.607250   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
240 55.608017   10.10.0.2       10.10.0.1       RSVP     RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
289 71.986920   10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
293 75.750262   10.10.0.2       10.10.0.1       RSVP     SREFRESH Message.
339 91.398629   10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
340 91.492894   10.10.0.1       10.10.0.2       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
341 91.494346   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
342 91.495179   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
343 91.495259   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
344 91.496057   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
345 91.496108   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
346 91.496919   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
347 91.496970   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
348 91.497759   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
349 91.497828   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
350 91.499406   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
351 91.499489   A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
352 91.500126   A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
372 97.465278   10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
393 108.691202  10.10.0.2       10.10.0.1       RSVP     SREFRESH Message.
396 110.628252  10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
457 135.053514  10.10.0.1       10.10.0.2       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
458 135.055099  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
459 135.055931  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
460 135.056007  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
461 135.056805  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
462 135.056858  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
463 135.057725  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
464 135.057784  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
465 135.059485  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
466 135.059538  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
467 135.060335  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
468 135.060460  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
469 135.061100  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
495 136.855367  10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
503 141.347945  10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
527 150.683320  10.10.0.1       10.10.0.2       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
528 150.684829  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
529 150.685670  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
530 150.685744  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
531 150.686541  A.B.C.54        A.B.C.40        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
532 150.686593  A.B.C.40        A.B.C.54        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
```

```
533 150.687409  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
534 150.687478  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
535 150.688268  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
536 150.688330  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
537 150.689128  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
538 150.689178  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
539 150.689816  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
541 151.051124  10.10.0.2      10.10.0.1      RSVP    SREFRESH Message.
585 168.091700  10.10.0.2      10.10.0.1      RSVP    SREFRESH Message.
594 174.355440  10.20.0.2      10.20.0.1      RSVP    PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                      277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
601 177.258259  10.20.0.1      10.20.0.2      RSVP    SREFRESH Message.
610 180.742913  10.10.0.1      10.10.0.2      RSVP    PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                      277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
611 180.744426  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
612 180.745254  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
613 180.745327  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
614 180.746124  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
615 180.746191  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
616 180.747006  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
617 180.747066  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
618 180.747855  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
619 180.747906  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
620 180.748701  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
621 180.748752  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
622 180.749390  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
632 185.104419  10.10.0.2      10.10.0.1      RSVP    SREFRESH Message.
672 199.913317  10.10.0.1      10.10.0.2      RSVP    PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                      277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
673 199.914877  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
674 199.915711  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
675 199.915781  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
676 199.916575  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
677 199.916625  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
678 199.917457  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
679 199.917511  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
680 199.918301  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
681 199.918373  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
682 199.919174  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
683 199.919231  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
684 199.919868  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
689 201.661765  10.10.0.2      10.10.0.1      RSVP    SREFRESH Message.
699 206.155112  10.20.0.2      10.20.0.1      RSVP    PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                      277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
751 221.688178  10.20.0.1      10.20.0.2      RSVP    SREFRESH Message.
770 233.602858  10.10.0.1      10.10.0.2      RSVP    PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                      277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
771 233.604527  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
772 233.605367  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
773 233.605438  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
774 233.606232  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
775 233.606278  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
776 233.607375  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
777 233.607421  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
778 233.609058  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
779 233.609108  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
780 233.609901  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
781 233.609943  A.B.C.40       A.B.C.54       SNMP    GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
782 233.610574  A.B.C.54       A.B.C.40       SNMP    RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
793 239.215501  10.20.0.2      10.20.0.1      RSVP    PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                      277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
802 241.171585  10.10.0.2      10.10.0.1      RSVP    SREFRESH Message.
842 258.348808  10.20.0.1      10.20.0.2      RSVP    SREFRESH Message.
```

7.6 Appendix F: Packet dump VLSR2 setting up a LSP

```
No.     Time        Source      Destination     Protocol Info
207 56.556381   10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
210 56.583934   A.B.C.42        A.B.C.58        SNMP     GET SNMPv2-MIB::sysDescr.0
211 56.591363   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-MIB::sysDescr.0
212 56.591513   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
213 56.593036   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
214 56.593106   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
215 56.597726   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
216 56.597835   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
217 56.599832   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
218 56.599888   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
219 56.600928   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
220 56.600976   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
221 56.601951   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
222 56.601999   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
223 56.603889   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
224 56.604613   10.30.0.2       10.30.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
226 56.633880   10.30.0.1       10.30.0.2       RSVP     RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
227 56.659563   A.B.C.42        A.B.C.58        SNMP     GET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
228 56.660205   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
229 56.660369   A.B.C.42        A.B.C.58        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3
230 56.671233   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3
231 56.671374   A.B.C.42        A.B.C.58        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
232 56.725110   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
233 56.725174   A.B.C.42        A.B.C.58        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
234 56.726751   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
235 56.726855   A.B.C.42        A.B.C.58        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
236 56.734000   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
237 56.734056   A.B.C.42        A.B.C.58        SNMP     SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
238 56.735626   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
239 56.736338   10.20.0.1       10.20.0.2       RSVP     RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
287 73.193455   10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
334 92.605157   10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
339 93.815812   10.30.0.2       10.30.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
349 97.071802   10.30.0.1       10.30.0.2       RSVP     SREFRESH Message.
353 98.671877   10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
354 98.673379   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
355 98.674222   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
356 98.674296   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
357 98.675092   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
358 98.675152   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
359 98.675950   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
360 98.676018   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
361 98.676831   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
362 98.676891   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
363 98.677681   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
364 98.677737   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
365 98.678526   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
366 98.678569   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
367 98.679356   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
368 98.679399   A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
369 98.680029   A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
394 111.834780  10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
402 116.632631  10.30.0.1       10.30.0.2       RSVP     SREFRESH Message.
426 126.426113  10.30.0.2       10.30.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
475 138.061961  10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                         277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
476 138.063485  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
477 138.064324  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
478 138.064401  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
479 138.065199  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
480 138.065253  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
481 138.066055  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
482 138.066106  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
483 138.066920  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
484 138.066975  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
485 138.067767  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
486 138.067818  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
487 138.068608  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
```

```
488 138.068668  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
489 138.069462  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
490 138.069544  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
491 138.070178  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
502 142.554469  10.20.0.1      10.20.0.2      RSVP     SREFRESH Message.
510 145.913228  10.30.0.1      10.30.0.2      RSVP     SREFRESH Message.
557 164.286153  10.30.0.2      10.30.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
576 175.562036  10.20.0.2      10.20.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
577 175.563554  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
578 175.564392  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
579 175.564463  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
580 175.565256  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
581 175.565323  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
582 175.566126  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
583 175.566178  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
584 175.566997  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
585 175.567054  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
586 175.567852  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
587 175.567903  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
588 175.568696  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
589 175.568752  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
590 175.569551  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
591 175.569602  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
592 175.570243  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
599 178.464777  10.20.0.1      10.20.0.2      RSVP     SREFRESH Message.
607 181.343115  10.30.0.1      10.30.0.2      RSVP     SREFRESH Message.
620 184.115620  10.30.0.2      10.30.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
673 207.361703  10.20.0.2      10.20.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
674 207.363162  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
675 207.364015  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
676 207.364095  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
677 207.364893  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
678 207.364965  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
679 207.365771  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
680 207.365831  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
681 207.366651  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
682 207.366709  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
683 207.367504  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
684 207.367568  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
685 207.368363  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
686 207.368441  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
687 207.369243  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
688 207.369298  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
689 207.369936  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
717 215.976200  10.30.0.2      10.30.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
737 221.033294  10.30.0.1      10.30.0.2      RSVP     SREFRESH Message.
741 222.894688  10.20.0.1      10.20.0.2      RSVP     SREFRESH Message.
770 240.422090  10.20.0.2      10.20.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
771 240.423556  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
772 240.424629  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
773 240.424709  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
774 240.425509  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
775 240.425564  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
776 240.426362  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
777 240.426427  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
778 240.427245  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
779 240.427305  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
780 240.428102  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
781 240.428161  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
782 240.428956  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
783 240.429006  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
784 240.429803  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
785 240.429854  A.B.C.42       A.B.C.58       SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
786 240.430491  A.B.C.58       A.B.C.42       SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
829 257.196020  10.30.0.2      10.30.0.1      RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                       277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
834 259.555322  10.20.0.1      10.20.0.2      RSVP     SREFRESH Message.
```

7.7 Appendix G: Packet dump VLSR1 of tearing down a LSP

No.	Time	Source	Destination	Protocol	Info
94	3.804148	10.30.0.2	10.30.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
136	18.642621	10.20.0.1	10.20.0.2	RSVP	SREFRESH Message.
178	20.900070	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
179	20.901530	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
180	20.902373	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
181	20.902443	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
182	20.903246	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
183	20.903293	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
184	20.904104	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
185	20.904150	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
186	20.904970	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
187	20.905020	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
188	20.905813	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
189	20.905858	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
190	20.906651	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
191	20.906699	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
192	20.907493	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
193	20.907539	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
194	20.908178	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
204	24.635961	10.20.0.2	10.20.0.1	RSVP	PATH TEAR Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
205	24.636457	10.30.0.2	10.30.0.1	RSVP	PATH TEAR Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
206	24.636929	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
207	24.642042	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
208	24.642179	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
209	24.645140	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
210	24.645197	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
211	24.646544	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
212	24.646678	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
213	24.651768	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
214	24.651836	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
215	24.654781	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
216	24.654832	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
217	24.656185	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
218	24.656352	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3
219	24.668286	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3

7.8 Appendix H: Packet dump VLSR2 of tearing down a LSP

No.	Time	Source	Destination	Protocol	Info
94	3.804148	10.30.0.2	10.30.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
136	18.642621	10.20.0.1	10.20.0.2	RSVP	SREFRESH Message.
178	20.900070	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
179	20.901530	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
180	20.902373	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
181	20.902443	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
182	20.903246	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
183	20.903293	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
184	20.904104	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
185	20.904150	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
186	20.904970	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
187	20.905020	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
188	20.905813	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
189	20.905858	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
190	20.906651	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
191	20.906699	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
192	20.907493	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
193	20.907539	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
194	20.908178	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
204	24.635961	10.20.0.2	10.20.0.1	RSVP	PATH TEAR Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
205	24.636457	10.30.0.2	10.30.0.1	RSVP	PATH TEAR Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
206	24.636929	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
207	24.642042	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
208	24.642179	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
209	24.645140	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
210	24.645197	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
211	24.646544	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
212	24.646678	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
213	24.651768	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
214	24.651836	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
215	24.654781	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
216	24.654832	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
217	24.656185	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
218	24.656352	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3
219	24.668286	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3

7.9 Appendix I: Packet dump VLSR1 setting up multiple LSP's over one link

No.	Time	Source	Destination	Protocol	Info
379	141.433778	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
381	141.600839	10.20.0.1	10.20.0.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
471	169.288496	10.10.0.1	10.10.0.2	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
474	169.316432	A.B.C.40	A.B.C.54	SNMP	GET SNMPv2-MIB::sysDescr.0
475	169.323529	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-MIB::sysDescr.0
476	169.323761	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
477	169.327948	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
478	169.328029	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
479	169.329733	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
480	169.329802	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
481	169.330597	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
482	169.330654	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
483	169.331290	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
484	169.332079	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
485	169.407331	10.20.0.1	10.20.0.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
486	169.408026	A.B.C.40	A.B.C.54	SNMP	GET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
487	169.416563	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
488	169.416702	A.B.C.40	A.B.C.54	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.2
489	169.434762	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.2
490	169.434895	A.B.C.40	A.B.C.54	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
491	169.457163	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
492	169.457261	A.B.C.40	A.B.C.54	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
493	169.458673	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
494	169.458790	A.B.C.40	A.B.C.54	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
495	169.464370	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
496	169.464431	A.B.C.40	A.B.C.54	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
497	169.465784	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
498	169.466584	10.10.0.2	10.10.0.1	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
515	176.261140	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
530	186.346365	10.20.0.1	10.20.0.2	RSVP	SREFRESH Message.
546	197.575352	10.10.0.1	10.10.0.2	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
547	197.576919	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
548	197.577748	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
549	197.577829	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
550	197.578629	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
551	197.578682	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
552	197.579497	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
553	197.579563	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
554	197.580356	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
555	197.580408	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
556	197.581200	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
557	197.581265	A.B.C.40	A.B.C.54	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
558	197.581903	A.B.C.54	A.B.C.40	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
566	201.297397	10.10.0.2	10.10.0.1	RSVP	SREFRESH Message.
569	204.551398	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
578	208.798724	10.20.0.2	10.20.0.1	RSVP	SREFRESH Message.

7.10 Appendix J: Packet dump VLSR2 setting up multiple LSP's over one link

No.	Time	Source	Destination	Protocol	Info
373	140.648324	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
377	140.676524	A.B.C.42	A.B.C.58	SNMP	GET SNMPv2-MIB::sysDescr.0
378	140.677147	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-MIB::sysDescr.0
379	140.677314	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
380	140.678113	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
381	140.678195	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
382	140.678991	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
383	140.679122	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
384	140.679936	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
385	140.680013	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
386	140.680806	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
387	140.680862	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
388	140.681653	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
389	140.681712	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
390	140.682349	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
391	140.683116	10.30.0.2	10.30.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
392	140.712226	10.30.0.1	10.30.0.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
393	140.738307	A.B.C.42	A.B.C.58	SNMP	GET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
394	140.747217	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
395	140.747387	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3
396	140.794322	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.3
397	140.794494	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
398	140.801967	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
399	140.802043	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
400	140.803892	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
401	140.804021	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
402	140.811647	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
403	140.811713	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
404	140.814490	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
405	140.815312	10.20.0.1	10.20.0.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
501	164.709905	10.30.0.1	10.30.0.2	RSVP	SREFRESH Message.
509	168.546621	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
510	168.548157	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
511	168.549000	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
512	168.549073	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
513	168.549868	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
514	168.549920	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
515	168.550721	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
516	168.550815	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
517	168.551627	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
518	168.551685	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
519	168.552475	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
520	168.552533	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
521	168.554156	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
522	168.554200	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
523	168.555000	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
524	168.555040	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
525	168.555672	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
526	168.556443	10.30.0.2	10.30.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
527	168.560303	10.30.0.1	10.30.0.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
528	168.560974	A.B.C.42	A.B.C.58	SNMP	GET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
529	168.561575	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
530	168.561680	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.4
531	168.591114	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.4
532	168.591214	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
533	168.609530	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
534	168.609610	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
535	168.611023	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.1
536	168.611141	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
537	168.619491	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
538	168.619559	A.B.C.42	A.B.C.58	SNMP	SET SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
539	168.621050	A.B.C.58	A.B.C.42	SNMP	RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.5.1.1.9
540	168.621800	10.20.0.1	10.20.0.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 277ca9c3. FILTERSPEC: IPv4-LSP, Tunnel Source: A.B.C.39, LSP ID: 1000.
556	175.475684	10.20.0.2	10.20.0.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID 287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
557	175.477176	A.B.C.42	A.B.C.58	SNMP	GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2

```
558 175.478021  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
559 175.478125  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
560 175.478931  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
561 175.479049  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
562 175.479854  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
563 175.479917  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
564 175.480724  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
565 175.480809  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
566 175.481632  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
567 175.481697  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
568 175.482491  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
569 175.482552  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
570 175.483365  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
571 175.483423  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
572 175.484223  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
573 175.484278  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
574 175.485079  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.4
575 175.485132  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.4
576 175.485770  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
577 175.486440  10.30.0.2       10.30.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                          287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
595 183.581867  10.30.0.1       10.30.0.2       RSVP     SREFRESH Message.
601 185.560833  10.20.0.1       10.20.0.2       RSVP     SREFRESH Message.
610 192.547271  10.30.0.2       10.30.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                          287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
625 202.810931  10.30.0.1       10.30.0.2       RSVP     SREFRESH Message.
626 203.765938  10.20.0.2       10.20.0.1       RSVP     PATH Message. SESSION: IPv4-LSP, Destination A.B.C.41, Tunnel ID 2000, Ext ID
                                                          287ca9c3. SENDER TEMPLATE: IPv4-LSP, Tunnel Source: A.B.C.40, LSP ID: 1020.
627 203.767481  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2
628 203.768320  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
629 203.768390  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.1
630 203.769187  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
631 203.769239  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.2
632 203.770044  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
633 203.770109  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.3
634 203.770917  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
635 203.770974  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.2.4
636 203.771794  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.3.1
637 203.771855  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4
638 203.772654  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
639 203.772718  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.1
640 203.773790  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
641 203.773845  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.2
642 203.775576  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
643 203.775638  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.3
644 203.776446  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.4
645 203.776504  A.B.C.42        A.B.C.58        SNMP     GET-NEXT SNMPv2-SMI::mib-2.17.7.1.4.3.1.4.4
646 203.777144  A.B.C.58        A.B.C.42        SNMP     RESPONSE SNMPv2-SMI::mib-2.17.7.1.4.3.1.5.1
655 207.336121  10.30.0.2       10.30.0.1       RSVP     SREFRESH Message.
657 208.013258  10.20.0.2       10.20.0.1       RSVP     SREFRESH Message.
```

7.11 Appendix K Contact Information

Contact Information University of Amsterdam

Researcher	
Name	BIct M. Meijerink
E-mail	mark@os3.nl
Researcher	
Name	BIct R. Prickaerts
E-mail	rprickaerts@os3.nl
Project Advisor	
Name	dr.ir. C.Th.A.M. de Laat
E-mail	delaat@science.uva.nl

Contact Information SARA

Project Coordinator	
Name	MSc. R. van der Pol
E-mail	rvdp@sara.nl
Project Coordinator	
Name	Msc. Andree Toonk
E-mail	andree@sara.nl

Address Information SARA

Kruislaan 415
1098 SH Amsterdam
Tel: 0031 02 5293000
Fax: 0031 02 6683167
Randstad 22 room 153
1316 BM Almere
Tel: 0031 36 5238000
Fax: 0031 36 5238030

References

[**GMPLS References**]

[1] RFC 3945 Generalized Multi-Protocol Label Switching (GMPLS) Architecture, http://www.ietf.org/rfc/rfc3945.txt

[2] GMPLS Architecture and Applications by Adrian Farrel and Igor Bryskin, This book was written by two of researchers at the forefront of the development of GMPLS and brings a complete and detailed overview of the GMPLS standard.

[3] MPLS and GMPLS by Li Yin, http://bnrg.eecs.berkeley.edu/ randy/Courses/CS294.S02/MPLS.ppt

[4] Generalized Multiprotocol label Switching (GMPLS) by Alcatel, Cisco Systems and IEC.org, http://www.iec.org/online/tutorials/gmpls/

[5] The OSPF opaque LSA option, http://www.ietf.org/rfc/rfc2370.txt

[6] Resource Reservation protocol (RSVP), http://www.ietf.org/rfc/rfc2205.txt

[7] SVP-TE: Extensions to RSVP for LSP Tunnels, http://www.ietf.org/rfc/rfc3209.txt

[8] Fast Reroute Extensions to RSVP-TE for LSP Tunnels, http://www.ietf.org/rfc/rfc4090.txt

[9] Link Management Protocol (LMP), http://www.ietf.org/rfc/rfc4204.txt

[**DRAGON References**]

[10] GNU General Public License, http://www.gnu.org/copyleft/gpl.html

[11] GNU Zebra, http://www.zebra.org/

[12] KOM RSVP Engine, http://www.kom.tu-darmstadt.de/en/downloads/software/kom-rsvp-engine/

[13] DRAGON: A Framework for Service Provisioning in Heterogeneous Grid Networks, IEEE Community Magazine March 2006 by T. Lehman (Inst. of Inf. Sci., Southern California Univ., CA, USA), J. Sobieski, B. Jabbari.

[14] DRAGON project site of The Mid-Atlantic Crossroads, http://DRAGON.maxgigapop.net/twiki/bin/view/DRAGON/WebHome

[15] DRAGON project site of USC/ISI, http://DRAGON.east.isi.edu/twiki/bin/view/Main/WebHome

[16] Policy-Based Resource Management and Service Provisioning in GMPLS Networks by Xi Yang, Tom Lehman, Chris Tracy, Jerry Sobieski, Shujia Gong, Payam Torab and Bijan Jabbari

[17] Generalized Multiprotocol Label Switching (GMPLS) by the International Engineering Consortium (IEC), http://www.iec.org/online/tutorials/gmpls/

[18] GMPLS Tutorial and R&E Network Implementation given by Chris Tracy at the University of Amsterdam April 19th 2006

[19] Deliverable DJ3.2.2:Initial Review of Technologies Related to the Provision of Bandwidth-on-Demand (BoD) Services by George Alyfantis (GRNET/UoA), Maarten Bchli (DANTE), Emilie Camisard (RENATER), Mauro Campanella (GARR), Vangelis Gazis (GRNET/UoA), Gabor Ivanszky (HUNGARNET), Eoin Kenny (HEANET), Simon Muyal (RENATER), Jan Radil (CESNET), Roeland Nuijts (SURFNET), Sarantis Paskalis (GRNET/UoA), Giannis Priggouris (GRNET/UoA), Esther Robles (REDIRIS), Laura Serrano (REDIRIS), Afrodite Sevasti (GRNET), Chrysostomos Tziouvaras (GRNET).

[20] Implementation of a GMPLS-based Network with End Host Initiated Signaling, by Xiangfei Zhu, Xuan Zheng, Malathi Veeraraghavan Zhaoming Li, Qiang Song, Ibrahim Habib Nageswara S. V. Rao

[21] Setting up the Environment for DRAGON Software

[22] DRAGON VLSR Implementation Guide, http://DRAGON.maxgigapop.net/twiki/pub/DRAGON/VLSR/DRAGON_VLSR_Implement_v02.pdf