Master of Science in Internetworking

MINT 709 – Comparing Segment Routing vs. traditional traffic engineering

Supervisor: Prof. Juned Noonari

Author: Aleksei Petrov

2018-2020

Contents

1 MPLS TE

1.1 Introduction

In the first part of the capstone project on topic “Comparing Segment routing vs. traditional traffic engineering” I would like to discuss and analyze the use cases of Resource Reservation Protocol - Traffic Engineering (RSVP-TE) operating principles starting with the discussion of Multiprotocol Label Switching Architecture and finishing it with challenges in implementing traffic engineering.

Speaking about the use cases of MPLS traffic engineering, firstly, we have to consider what traffic engineering is. It is a technology that allows network engineers to fit different types of traffic into the existing network infrastructure to utilize its resources as efficiently as possible. For example, imagine that one company has two departments: IT and Accounting, the IT department usually generates more internet traffic than the second unit does. Therefore, it is illogical to provide both departments same network resources, because the problem of underutilization or deficiency of resources can arise and negatively affect all business processes

Now let us speak about real-life applications for MPLS TE. In a book, “Traffic Engineering with MPLS” written by Eric Osborne and Ajay Simha (1), three typical usage cases are identified:

- Optimizing network utilization – building a full mesh of MPLS TE Label Switch Paths (LSPs) between different routers, allocating resources such as bandwidth, to these LSPs. As a result, network traffic is spread in different paths and an engineer can get as much as it is possible out of the infrastructure he already has and delay upgrading it for a period of time.
- Handling unexpected congestion – this approach helps a network engineer to handle congestions occurring in the network. Working mostly on IGP protocols and determining congestion, routers can be configured to transfer traffic over MPLS TE tunnels in order to unload problem links and forward traffic through the paths, which were not used by IGP protocol.
- Handling link and node failures – this approach is used for a quick recovery of the network after link and node failures. MPLS TE has a feature called Fast Reroute, which noticeably minimizes packet loss in the situations described above. Fast Reroute is a technology, which switches the primary tunnel to the backup one in case of failures.

Having examined the above points, we can say that MPLS traffic engineering is a very important and useful part of modern network infrastructures because it is able to provide quick network convergence, high utilization of network resources and, to some degree, improve its cost-effectiveness.

1.2 MPLS technical concepts

This part of a project is devoted to the description of the key technical concepts of MPLS technology.

1.2.1 MPLS

MPLS is a routing technology that provides end-to-end reachability between nodes in a network with the usage of labels. MPLS is a fundamental and key aspect in terms of traffic engineering implemented with an RSVP-TE protocol, that is why it is very important to analyze its mechanisms and understand how it operates.

1.2.1.1 Labels

Label in MPLS terminology is a fixed-length integer identifier, which has a local significance and is used to identify Forwarding Equivalence Class (FEC). FEC is a set of properties that map any incoming packet in a network to the same outgoing label. Those properties can be an IP address and Quality of Service (QoS).

An example is provided in figure 1. Router A sends a packet to Router B. Router A has to determine if the packet belongs to FEC F, so it checks packets' network layer address and/or QoS parameters. After this step, Router A puts a label L in the packet (outgoing label) and Router B understands that packet belongs to FEC F by checking incoming label L. It is important to say, that label L has a local meaning within a segment Router A-Router B and may not be an indicator of FEC F on any other segment.

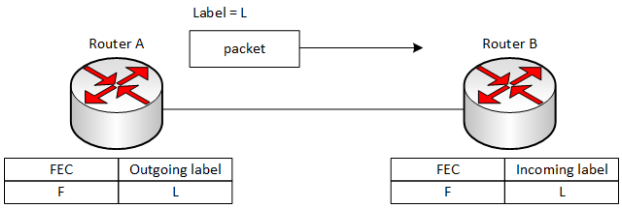

Figure 1 – FEC operation process

Sometimes it can be very hard for Router B to understand that label L was put by Router A and not by any other router. This situation often occurs if Label Switching Routers (LSR) are not direct neighbors. In such cases, Router B must make sure that FEC and label have one to one binding. For example, label L cannot be bounded to FEC G or FEC K. All LSRs must uniquely interpret its incoming labels.

Speaking about label assignment and distribution, we have to examine firstly MPLS terms "Upstream" and "Downstream" LSRs:

- Upstream LSR – a router that is located closer to the source of the packet, relatively to the second one;
- Downstream LSR – a router that is located farther from the source of the packet, relatively to the second one.

In Figure 1 Router A is an Upstream router and Router B is a Downstream router.

Label assignment and distribution in MPLS can be implemented in both directions. The default way, downstream, is described in IETF RFC 3031. In this case, the downstream router (Router B in our example) binds FEC and label and then informs upstream router (Router A) about this binding.

IETF RFC 5331 "MPLS Upstream Label Assignment and Context-Specific Label Space" introduces an optional, second way – upstream label distribution. In this case, upstream LSR (Router A) binds FEC and label and informs downstream LSR (Router B) about it. The typical use case for upstream label distribution is MPLS multicast because in some situations it requires upstream LSR to distribute labels simultaneously to multiple downstream LSR`s.

Besides the existence of two directions of label distribution, "Downstream" mode is also divided into two types:

- Unsolicited Downstream – as soon as a downstream LSR gets information about new FEC, it advertises its MPLS label for this FEC to all its MPLS neighbors;
- Downstream on demand – in this case, downstream LSR gets information about new FEC, but advertises its MPLS labels for this FEC only after its neighbor requests this information.

The first method is simple in implementation and understanding, but in some cases, it supplies LSRs with excess and useless information. The second one corrects this drawback, however, adds some complexity.

1.2.1.2 Label distribution protocols

As described in paragraph 1.2.1.1, all LSR`s in the MPLS domain have to clearly and definitely bind labels to Forward Equivalence Classes. Furthermore, neighboring routers must decide on common values for these parameters on a connected segment. Label distribution protocols perform this task. Here is the definition of label distribution protocols, given in Juniper TechLibrary (3):

"Label distribution protocol – is a set of procedures by which one LSR informs a peer LSR of the meaning of the labels used to forward traffic between them. Label distribution protocols create and maintain the label-to-FEC bindings along an LSP from MPLS domain ingress to MPLS domain egress. It enables each peer to learn about the other peer's label mappings. The label distribution protocol provides the information MPLS uses to create the forwarding tables in each LSR in the MPLS domain."

Nowadays there are three label distribution protocols used in MPLS:

- MPLS Border Gateway Protocol (MP-BGP) – this protocol can distribute labels for establishing Label Switched Paths (LSP) and traffic transport (if two adjacent LSRs are also BGP neighbors, we can avoid usage of any other protocol in terms of label distribution) and for separation of different services (i.e. VPN);
- Label Distribution Protocol (LDP) – this protocol can be used only for establishing LSP by mapping information from network layer routing tables to switched paths;
- Resource Reservation Protocol-Traffic Engineering (RSVP-TE) – this protocol can distribute labels for establishing LSP and also for resource reservations across the IP network.

From the brief description of three label distribution protocols, we can see, that MP-BGP, LDP, and RSVP-TE can perform different functions and there is no limitation on the use of only one of them in MPLS domain. Conversely, these protocols can operate together and be piggybacked one upon another.

1.2.1.3 Label stacking

Before the discussion of label stack technology, it is important to analyze an MPLS label format. It is shown in figure 2.

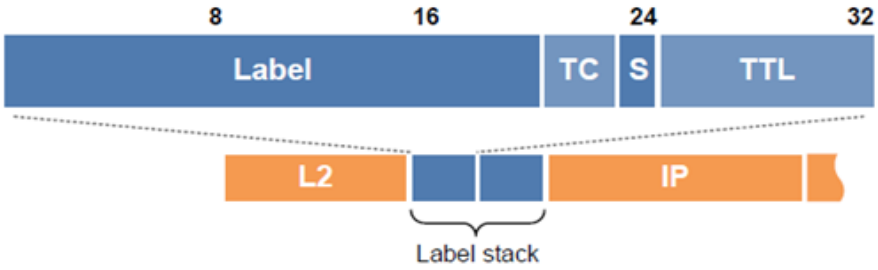

Figure 2 – MPLS label format and label stack

MPLS label is inserted between Layer 2 and Layer 3 headers in the IP packet. Label`s size is 32 bits and its field description is provided in Table 1.

Table 1 – MPLS labels' fields

Field	Description
Label	20 bits field, the possible value can be between 0 and 2^{20}
TC	3 bits used for QoS
S	Bottom of the stack, shows if this label is the last label in a stack: 1) S=0 – not last label in the stack 2) S=1 – last label in the stack
TTL	Time to live is decreased by one after each hope, prevents loops.

Not all the labels in the defined scope can be assigned by LSRs to packets. There are 16 special labels in MPLS architecture (7, 8, 9):

Table 2 – MPLS labels` scope

Label value	Meaning
0	IPv4 Explicit NULL Label – can be used only if there is no stacking, it tells the receiver to pop it upon receipt.
1	Router Alert label – when LSR receives a top label with a value 1, it delivers the packet to the local software module for processing
2	IPv6 Explicit NULL Label

Label value	Meaning
3	Implicit Null label never appears in the encapsulation
4-6	Unassigned
7	Entropy label – provides “entropy” to improve load balancing
8-15	Unassigned

Label stacking – is a process of adding an MPLS label to a packet “on top” of the existing label, in other words, creating an MPLS packet inside of another MPLS packet (4). By default, the maximum number of labels in the stack is three. The most popular and useful implementation of label stacking nowadays is the opportunity to use LDP and RSVP-TE in collaboration. In this case, LDP label is a “transport” label and RSVP-TE label is a “service” one. It is important to mention that switching in MPLS core is processed based only on the outer or “transport” label.

1.2.1.4 Forwarding information base

IETF RFC 3031 defines three components, which make up decision-making mechanisms for packet forwarding:

- Next Hop Label Forwarding Entry (NHLFE) – an entry, which defines:
  - Information about next-hop: outgoing interface, next-hop address;
  - The operation to perform with the label:
    - Pop the label stack;
    - Replace the top label in a stack with a new one;
    - Replace the top label in a stack with a new one and push new label/labels onto the label stack.
  - Label encoding information;
  - Layer 2 encapsulation information.
- Incoming Label Map (ILM) – this component maps labels in incoming packets to NHLFE entries. There can be multiple entries for a specific label, in case of load balancing, for example, but finally, only one entry must be chosen for a label.
- FEC-to-NHLFE Map (FTN) – this component maps each FEC to NHLFE entries. It is applied if the incoming packet has no label, for example, if a packet arrives from outside the MPLS domain (6).

Figure 3 illustrates the logic, described above.

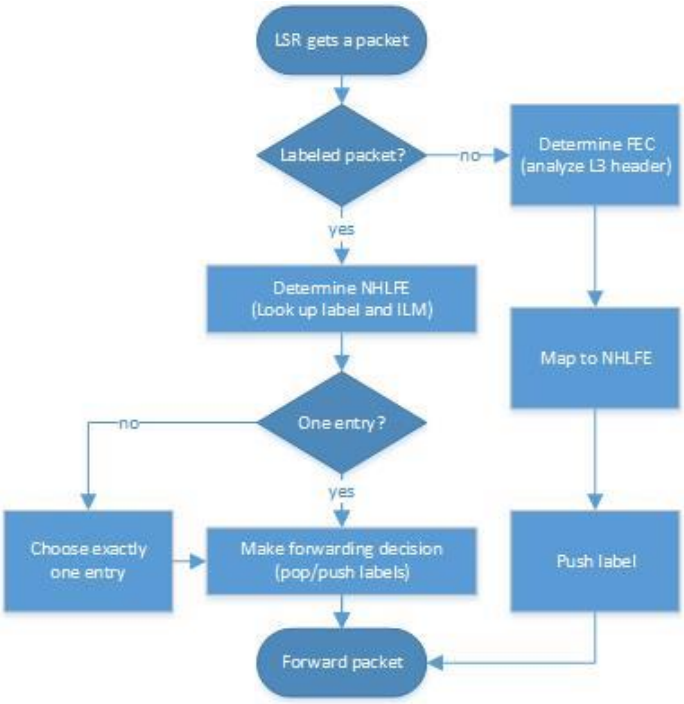

Figure 3 – Packet forwarding logic

1.2.1.5 Label Switched Path

Here I would like to cite a definition of LSP from IETF RFC 3031: Label Switched Path (LSP) for a particular packet is a sequence of routers in the MPLS domain. This path has some characteristics and properties:

- The first router in LSP path, that handles newly-arrived packet is called “LSP Ingress”, it pushes a level M label so that label stack in a packet has a depth of M (the previous level was L);
- During all the LSP, on each router, when a packet arrives, it has a depth of label stack equal to M;
- Each LSR in the path makes a forwarding decision with a means of MPLS by checking the packet`s top-level label (transport label);
- If there is an intermediate system between two peering LSRs in LSP path, switch, for example, this system makes forwarding decisions without top-level label analysis. This decision can be based on L2 or L3 headers;
- The Last router in the LSP path that makes a forwarding decision by analyzing the level L label in the packet is called “LSP Egress.”

Taking into account everything of the above, we can say, that if an LSR pushes a label onto the packet's label stack, it should be confident, that this new label belongs to FEC, which Egress LSR is the LSR that assigned a second label in the current stack.

1.2.1.6 Penultimate Hop Popping

In section 1.2.1.5, it was said, "During all the LSP, on each router, when a packet arrives, it has a depth of label stack equal to M." It is the default behavior of MPLS LSP. However, there is a feature that improves the MPLS operation process. It is called Penultimate Hop Popping (PHP). The key idea of it is to pop a top-level label on the LSR, before the LSP Egress router. As far as the main goal of LSPs, labeling and other MPLS processes and attributes is to deliver the packet to LSP Egress router, the top-level label has no meaning and functionality after being transferred from the router, previous to the LSP Egress router, to LSP Egress router.

This feature provides architecture with a noticeable technical advantage that results in better resource utilization and improved processing time. The key reason for it is a number of operations that LSP Egress router has to perform on a packet. With a default MPLS behavior, Egress router has to:

- Analyze the top label and determine if it is LSP egress;
- Pop the top-level label and, if there is another label, analyze it.

In this case, the router has to do two lookups, but if Penultimate Hop Popping is enabled, the maximum one label stack lookup is needed, because the top-level label was already popped by previous LSR. This is not a default behavior of MPLS, because not all hardware supports PHP. Negotiations on popping between LSRs have to be performed by means of label distribution protocol, enabled in the domain. In this process, downstream LSR (Egress router) should request hop popping from its upstream neighbor. LSR, before Egress router, must not do PHP without corresponding request.

1.2.1.7 Label Encodings

In order to proceed, a packet with a label stack in the network, labels have to be encoded using a special technique. This technique depends on hardware and software of the devices, which transmit and receive labeled packets. If we speak about the MPLS cloud, the MPLS-specific Hardware and/or Software is the solution. Here a method defined in IETF RFC 3032 and called MPLS-SHIM is used. The key concept is that a label (format of this label is described in paragraph 1.2.1.3) is inserted as a "shim" between the data link layer and the network layer of the transmitted

packet. This “shim” is a protocol-independent; in other words, it does not depend on network layer protocol.

1.2.1.8 Label Merging

Label merging is an MPLS technique, which allows saving the number of available labels when transferring packets by assignment of the different incoming labels to the same FEC. For example, LSR has multiple incoming interfaces and gets packets on them with different labels. It is not important for LSR that labels are different. Finally, the same label will be assigned to those packets and packets will be transferred through the same outgoing interface. Figure 4 shows an example of a Label Merging. Here packets with labels L1, L2, L3 arrive on different interfaces of the LSR. After analysis of FEC, LSR makes a decision to proceed with all three packets with a new, equal for all packets, label L4. Important to say, that once the packets are transmitted, the information that they arrived from different interfaces and with different incoming labels is lost (5).

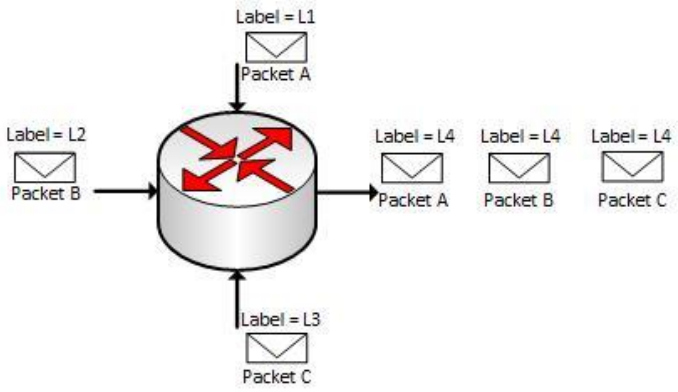

Figure 4 – Label merging

Some LSRs, due to its hardware or software, cannot perform label merging. In such a case, packets with N different incoming labels should have proceeded with M different outgoing labels. So in case of label merging, there can be N incoming labels and only 1 outgoing, while if LSR does not support this technology, it should use N=M number of incoming and outgoing labels.

Now I would like to analyze the interaction of downstream and upstream routers when Label merging is enabled. There are two possible situations:

- Both LSRs support Label Merging – in this case, the upstream router needs to be sent only one label per FEC;
- An upstream router does not support Label Merging – then the upstream router has to send explicit requests to the downstream router, asking for other labels for specific

FEC. If downstream LSR also does not support this technology, it similarly sends explicit requests to the next downstream routers.

It is important to mention, that some devices support Label Merging but with certain limitations in merging multiple labels in one. For example, LSR can merge only three incoming labels in one outgoing, but accordingly to architecture, five labels belong to one FEC. In such a case, five incoming labels can be merged into two outgoings.

1.2.1.9 TTL and Loop Control

In standard IP networks, every packet being transferred carries a special value – Time To Live. This value is decremented by one by each router in the packet`s path and if the TTL reaches 0, the packet is dropped. The key function of this value is loop prevention. It works similarly for MPLS. Packet, which is transferred through the MPLS domain, should emerge with the same TTL value as if it had passed the same number of routers in a standard IP network.

Mechanisms of handling TTL values in MPLS depend on the way, which is selected for label carriage. If the MPLS-SHIM technique (described in paragraph 1.2.1.7) is used in the MPLS domain, then:

- Ingress router copies a TTL value from a network-layer header and puts it in label`s field;
- TTL value is decremented by one at each LSR;
- Egress router copies a TTL value from a label`s field and puts it in a network layer header.

This approach significantly differs if we speak about MPLS architectures, where labels are encoded in the data link layer header (ATM, Frame Relay). In this case, data link layer switches cannot change labels. Such LSPs are called the non-TTL LSP segment. In the non-TTL segment, ingress LSR to such a segment has to be informed about the number of LSRs in LSP and decrement packet's TTL with this number. If the number of LSPs is greater than the value in the TTL field, it means that ingress LSR must not switch and label the packet.

Of course, TTL is not the only loop prevention mechanism in MPLS. In order to solve this problem, MPLS relies on label distribution protocol, which, in turn, relies on Interior Gateway Protocol (IGP), used in the MPLS domain. Although this is a reliable and good approach, MPLS has its own additional loop prevention mechanism. It is described in IETF RFC 3063. Short description of key steps are given below:

- If LSR understands, that it has one more available next-hop for a particular FEC, it creates a “thread”. Thread is a sequence of messages, used to set up an LSP (11). These messages contain “color”, hop count and TTL parameters;
- Each thread is assigned with a unique “color”. Color is a combination of an IP address and a unique event identifier from a node`s numbering space. The only information associated with a next-hop is a “color” and hop count;
- After reaching a destination node, acknowledgment is sent to the initial one;
- Destination node changes thread`s color to “transparent” (value = 0) and rewinds the thread back;
- If there is no loop in LSP, the thread is rewound to the node that created it with “transparent” color and extended hop count. Only after that labels are assigned;
- If there is a loop, the thread will come back to the initial node it and it will be obvious for this node that loop has been detected.

This mechanism is compatible with MPLS-SHIM, ATM, and Frame relay architectures and supports downstream-on-demand label distribution technology.

1.2.1.10 Security

Security has always been one of the most actual and acute topics for network engineers and designers. Thousands of threats for business data and information exist nowadays and nobody wants his data to be stolen or corrupted. That is why practically every protocol and technology has its own means of protection against various types of attacks and MPLS is not an exception. In this section, I would like to discuss security problems, which are unique for MPLS, and analyze some prevention mechanisms.

In order to simplify the discussion, typical MPLS topology is provided in figure 5. This topology has the following characteristics:

- Customers C1 and C2 can be connected to the public Internet by two service providers – AS-A and AS-B;
- Every connection between domains (C1 – AS-A; AS-A – Internet; AS-A – AS-B, etc.) is a “trust boundary”. “Trust boundary” is a link and each of the nodes of this link may not trust traffic coming from a neighbor. In other words, CE will trust traffic from the AS-A domain not as fully as traffic from the C1 domain.

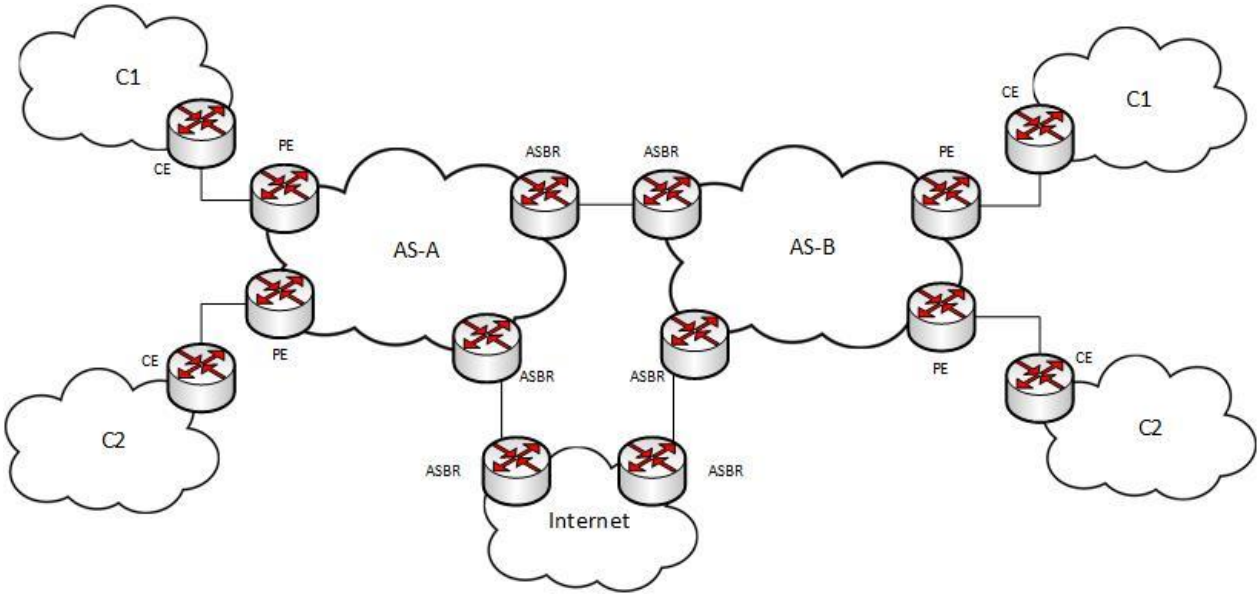

Figure 5 – MPLS architecture example

Speaking about threads, that threats, I would like to list issues, mentioned in “MPLS: next steps” by Bruce S. Davie and Adrian Farrel (13):

- Observation of a customer`s or provider`s data while in transfer;
- Modification or deletion of data in transit;
- A replay of previously transmitted data;
- Insertion of data in the traffic stream;
- Disruption of connectivity between customer and provider or between providers;
- Degradation of quality of services experienced by traffic of one or more customers;
- Unauthorized use or theft of provider network resources.

All possible threats mentioned above can be logically divided into two groups: Control plane threats and Data plane threats. MPLS control plane consists of protocols, which are used in the MPLS domain: label distribution protocols, interior gateway protocols, etc. Some of the possible Control Plane threats are described below:

- Creation of LSP – if an illegal device is attached to the MPLS domain, it can create and send MPLS messages and become a member of some LSPs. It can result in harmful resource consumption or traffic misrouting;
- Snooping of LSP messages – if an attacker has an ability to get LSP messages, he can track the label distribution process and finally can get a complete picture of the topology of the domain;

- Denial of Service (DoS) – network bandwidth or node`s CPU can be overloaded with LSP or other types of messages and become a reason for network failure;

If we speak about attacks on Data plane, its key targets are packets and information being transferred in these packets. In general, there is no big difference in attacks on the data plan of the usual IP packets and MPLS packets. The only thing which makes the MPLS packet`s data plan more vulnerable is labels. If an attacker has an ability to change the label in the packet`s data plan, he can easily route this packet to any tunnel or through the "trust-boundary" link.

We have discussed some possible MPLS security threats in the first part of this chapter. Now it is time to look at threats prevention mechanisms and strategies. Some basic methods are provided below (13),(14):

- Physical access control – It is the basic and most obvious safety precaution. However, it can be a very good first line of defense. For example, if cables or MPLS devices are placed in cabinets with locks, it becomes much more difficult for an attacker to have physical access;
- Use of Isolated infrastructure – use of physically separated devices especially for MPLS;
- Resource limits for services in aggregated infrastructure;
- Logical access control – In order to prevent getting some undesirable packets, control lists with different filters can be applied;
- Control plane authentication – This safety precaution can get rid of many possible problems mentioned above. The basic idea is to accept control plane messages only from devices, which can be verified;
- Monitoring, detection, and reporting of security attacks – as far as attackers often begin with probing defenses, systems can early detect such tries and collect information about further threats.

Control plane authentication the most complicated feature from the list and I would like to analyze its process a little deeper.

- MD5 standard is usually used for it;
- Neighboring LSRs have to agree on the use of MD5 Signature option;
- Each LSR working with MD5 is configured with a password for each potential neighbor;

- Before transferring a message to the potential peer, LSR applies the MD5 hash algorithm to compute MD5 digest for the TCP segment, which will be sent to a peer;
- When peer gets such message, it calculates expected MD5 digest and compares it with a digest in TCP segment;
- If values are not the same, messages are discarded.

Speaking about encrypting the data plane of the MPLS packet, it is not really needed and only applied if the high level of protection is required. In this case, the most obvious approach is IPsec technology. It allows authenticating and encrypting MPLS packets, which are encapsulated in an IP header.

1.3 RSVP-TE

In this part of the capstone project, I would like to discuss Resource Reservation Protocol-Traffic Engineering, its history, technical concepts, and implementation and analyze its performance in real-life situations.

1.3.1 Prior and similar technologies

Before companies came to use the protocol RSVP-TE in terms of traffic engineering, some technologies had already existed and been used for this purpose, as an example RSVP (IETF RFC 2205) and CR-LDP (IETF RFC 3472).

1.3.1.1 RSVP

Resource reservation protocol is used by hosts for requesting specific qualities of service from the network for particular application data streams and flows (14). Permanent Bandwidth for a video stream can be an example of such data. Those hosts send in special network messages, which format is defined by RSVP protocol. This message contains data about the type of transferred information and required for its bandwidth. Then it is transmitted between routers throughout the whole path. When the router gets such a message, it checks its resources for availability. If it is impossible to allocate needed resources, routers deny the request. If the requested bandwidth is available, the router configures its packet-processing algorithm in such a way that a flow, mentioned in the message, is allocated with it. Then the router transfers the message to its neighbor in the path. As a result, requested bandwidth is reserved on every network segment from the source to destination. This is a short description of the RSVP operation. In the following paragraphs, it is described deeper.

Now I would like to list shortly some RSVP attributes:

- RSVP requests resources only for one direction in a flow;
- RSVP is not a routing protocol. It works with current routing protocols;
- An RSVP session, the potential receiver initiates reservation;
- There is a feature “soft state” in RSVP. It provides a periodical refresh of reserved resources.

1.3.1.1.1 Key concepts

In RSVP, packets with the same destination port and same destination address are called a “flow”. Every such flow has its own flow descriptor. Flow descriptor is a combination of flowspec and filterspec. Filterspec is used for the identification of the packets with the same sender IP address and source port. Flowspec, in its turn, specifies the desired QoS for that flow and consists of three parts: service class (is defined by an application), Rspec (defines the desired QoS) and Tspec (describes the level of data flow. If this level exceeds an expected one, the router can drop it). Figure 6 illustrates the relationship between flowspec and filterspec.

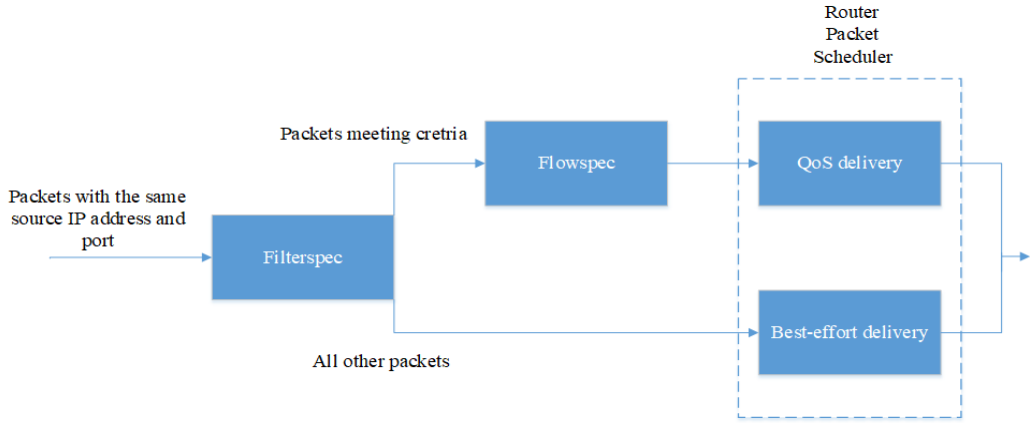

Figure 6 – Flowspec and Filterspec

Reservation messages in RSVP are created by the receiver and then passed upstream to the sender. The whole reservation process can be divided into two major steps:

- Reservation on the link– RSVP process transfers upstream requests for admission control and policy control. Admission control is required to check the availability of requested resources. Policy control checks the rights of the node to make a reservation request. If any of these two requests fails, RSVP sends a reject message to the originator. If both are successful, the route classifier is attached to the packet. Route classifier is a Quality of Service class and requests;

- Request forwarding – RSVP process transfers the request to upstream nodes until the sender receives it. Important to say, that this request can be different when transferring from node to node due to modifications done by traffic control mechanisms configured on nodes.

The model described above is called “one pass”. Despite the ease of implementation, there is a rather noticeable drawback: receiver nodes do not know the results of the reservation process. Here comes the RSVP enhancement called “One Pass With Advertising” (OPWA). In OPWA, control packets are sent downstream in order to gather hop-by-hop information about resources being reserved. As a result, the receiver node gets a reservation topology and, if it is needed, take a decision to make a new request in order to improve it.

There are seven types of messages, which are used in RSVP. All of them use the same header format with the fields, mentioned in table 3. The header is followed by a body, called “object”. Its fields are described in table 4.

Table 3 – Body`s fields

Name	Size	Value
Length, bytes	16 bits	Total length of object in bytes
Class-Num	8 bits	Identifies class of the object (Flowspec, filterspec, etc.)
C-Type	8 bits	Object type

Table 4 – Header`s fields

Name	Size	Value
Vers	4 bits	Protocol version
Flags	4 bits	Not defined yet
Msg Type	8 bits	1 = Path 2 = Resv 3 = PathErr 4 = ResvErr 5 = PathTear

		6 = ResvTear 7 = ResvConf
Checksum	16 bits	If all “0”, checksum was not transmitted
Send_TTL	8 bits	IP TTL
RSVP Length	16 bits	Total length of RSVP message in bytes

Two main types of RSVP messages are Resv and Path. The use of Resv messages was discussed above. It is only important to add, that these messages follow the reverse path of what data messages do. The second type is the Path messages. These messages are created by the sender (upstream node) and spread in the downstream direction. Path messages follow exactly the same path as data messages and collect a “path state” in each node along the way. Path state includes at least one address of the previous-hop node. This information is used later in a routing process of Resv messages. Besides an address of a previous-hop, Path messages can contain a “Sender template” (describes a format of packets, which will be created by sender), “Sender Tspec” (describes characteristics of the data, which will be created by the sender) and “Adspec” (package of OPWA).

The next type of RSVP message is “Teardown message”. It can be either PathTear or ResvTear. PathTear message is initiated by the upstream node, passes all downstream nodes and removes path states. ResvTear message, in its turn, is transferred in the upstream direction and deletes reservations of the resources on each node in the path. Teardown requests can be initiated by an application, user or network device.

RSVP Error messages can be ResvErr and PathErr. These messages indicate errors, which occur in the process of a reservation or of creating a path state.

The last possible type is a confirmation message. Here exists only ResvConf type. It provides an origin node with a confirmation of a successful result of the reservation process.

There is a feature in RSVP called “Soft state”. It has already been briefly described in section 1.3.1.1. Soft state feature allows a network engineer to maintain reservation of channel resources by refreshing it or deleting on routers and hosts. This state is created and refreshed by Resv and

Path messages. Soft state and, accordingly, all reserved resources on the node can be deleted because of two possible reasons:

- If the node does not get “refresh” messages during “clean-up timeout” interval before it runs out;
- If the node gets a teardown message.

Each node in RSVP has its own “refresh timeout”. Each time it expires, the node has to send “refresh” messages” to its next hop. As far as RSVP does not have a mechanism of acknowledgment reception of such “refresh” messages, possible packets loss has to be taken into consideration. That is why it is a good practice to set a “clean-up timeout” interval K times bigger (default value is 3), than the “refresh timeout” interval. If there is a need not in deleting the state, but in changing it, node, which initiates it just, has to send Path or Resv messages. As a result, corresponding changes in RSVP state on all nodes along the path will be implemented.

1.3.1.1.2 Reservation Styles

RSVP can work not only with unicast streams but also with multicasts. Traffic can be generated by multiple senders in multicast streams. Until this moment it was discussed that the receiver in RSVP initiates one reservation process per flow per sender. However, RSVP supports the feature that provides a more efficient way of reservation for multicast streams. It is called “reservation style”. There are two main options in this feature. The first one defines the receiver`s treatment of the reservation. The Receiver can establish either a distinct reservation (separate reservation for each sender) or shared (common reservation for all senders). The second option defines the selection process of the senders and there are two variants: explicit list or wildcard. In the explicit list, filterspec is used and identifies exactly one sender. Filterspec is not needed in a wildcard option. Below follows a short description of all possible “reservation styles”.

- Wildcard-Filter (WF) – here options “wildcard” and “shared” are used. This reservation style establishes a single reservation for all senders in a session. Reservations from different senders are merged together along the path so that only the biggest reservation request reaches the senders. A wildcard reservation is forwarded upstream to all sender hosts. If new senders appear in the session, for example, new members enter a videoconferencing, the reservation is extended to these new senders (16);

- Fixed-Filter (FF) – here options “distinct” and “explicit” are used. This means that a distinct reservation is created for data packets from a particular sender. Packets from different senders that are in the same session do not share reservations (16);
- Shared-Explicit (SE) - here options “shared” and “explicit” are used. This means that a single reservation covers flow from a specified subset of senders. Therefore, a sender list must be included in the reservation request from the receiver (16).

1.3.1.1.3 Problems

As it was said in paragraph 1.3.1.1.1, there are two types of error messages in RSVP: PathErr and ResvErr. In the case of PathErr, there are no big difficulties as far as there are not many variants of Path errors. This message is sent upstream and does not change RSVP state on nodes through which it passes.

If we analyze errors, which occur during the reservation process, there some complexity is added. It is important to say, that if reservation request fails, all possible receivers must know about it. Reservation requests can fail because of different reasons: lack of resources, lost reservation request message, admission control. In addition, there can be such a specific situation, called “killer reservation”, when one reservation request is a threat for another one:

- First killer reservation problem (KR-1) – If reservation Qa already exists on each node and another receiver asks for resources Qb, Qb>Qa, then the result of the merger of Qb and Qa can be rejected by any upstream node due to admission control. This situation must not influence existing reservation Qa, that is why upstream nodes are instructed to simply discard new request and leave Qa;
- Second killer reservation problem (KR-2) – Let`s assume that there is a persistent receiver in topology which requests Qb resources. Its request fails on some nodes. This situation must not influence new request Qa from the newly appeared receiver. It is solved by the technology called “blockade state”. Blockade state allows excluding request Qb, in our case, from the merge process, so that request Qa can be transferred and satisfied by the upstream nodes.

Despite the fact that some reservation requests can fail on any node through the RSVP path, as it has been just discussed, it is not mandatory that these reservations are deleted from all downstream nodes, where they were successful. There are multiple reasons exist for such behavior:

- Receiver wants to obtain requested quality of service as far in the path as it is possible, so he prefers to hold those reservations “alive”;
- If the reservation state is deleted, RSVP cannot take it into consideration in the course of further work. For example, it can try to unsuccessfully flap to such route and find it to be congested;
- If such a reservation is deleted, it can be very annoying for users, because RSVP will attempt to reserve every time out interval and then delete again.

1.3.1.1.4 Confirmation

In order to get an acknowledgment for its reservation request, the receiver puts a “confirmation-request” object with its IP address in Resv message. The logic of the RSVP, in this case, is that only the largest flowspec and corresponding confirmation requests are transferred upstream. If the newly arrived reservation request is smaller or equal to the existing one on the node, the confirmation message is sent to the receiver immediately.

Because of the algorithm described above, we can conclude the following:

- For each new request with biggest flowspec receiver get from each sender either ResvConf or ResvErr message;
- If a receiver gets ResvConf it does not guarantee a successful reservation of the resources. For example, if two new requests Qa and Qb arrive from Ra and Rb and Qb arrives second, these requests merge and Rb gets a ResvConf. However, Qa can fail on any other node in the path and, as a result, Qb will finally get ResvErr.

1.3.1.1.5 Timer values

There are two main timers in RSVP:

- Refresh timer – timer which shows an interval of time between successful state`s updates, generated by the neighbor. It is indicated by the letter R;
- Local state lifetime timer – a local timer that shows the local state’s lifetime. It is indicated by the letter L.

Each node in RSVP topology when generating Path or Resv messages puts a refresh timeout value in this packet in a specific object. This value R is then taken into consideration in the process of the receiver`s local state lifetime timer value calculation when the state is received. R and L values can be different from hop-by-hop. The default value for R is 30 seconds.

In case a refresh message is lost due to some reasons, the state can be lost prematurely. In order to fix this problem, the value of state lifetime timer must be greater than value R and has to be calculated by the following formula:

$$L \geq (K + 0.5) \times 1.5 \times R,$$

Where K is a small integer, default is 3.

This technology allows RSVP devices not to delete states after the loss of one refresh packet. Be default, four refresh messages can be lost before the state is removed.

1.3.1.1.6 Message exchange process

In this version of RSVP messages must occupy only one IP datagram. If the size of such IP datagram exceeds allowable MTU, the datagram will be broken into several parts by the sender and reassembled by the receiver. It can cause a problem in congested networks because a fragment of the datagram can be lost and the whole reservation request fails. RSVP specified in IETF RFC 2205 does not fix this problem.

RSVP messages are transferred hop-by-hop between RSVP devices as a “raw” IP datagram with a protocol number 46. In the case of raw IP datagrams application can get not only the payload, as it is in data datagrams, but also all the headers of the packet. In other words, Raw IP datagram technology allows an application to access information about lower layer protocols and not use protocol-specific transport layer formatting. However, not all network devices support this technology. In this case, RSVP messages can be encapsulated into UDP datagram.

Messages Path, PathTear, and ResvConf are transferred with a special label, which indicates the Router Alert option. This option requires all transit nodes in the path to analyze the packet`s content in more detail in case if the packet needs a special processing method.

If RSVP node gets a message containing information about the change of the state, it should immediately transfer it through all the interfaces except the one, which received this packet. This logic allows preventing packet storms on broadcast Local Access Networks (LAN).

In the case of packet loss RSVP uses its refreshing mechanisms in order to restore it. Loss of RSVP packets can be critical because it can lead to failure in the reservation process. Refreshing mechanisms provide packet retransmission after the timer expires. If the network is congested and default configuration of refreshing mechanisms does not help, its retransmission timer`s value can be increased.

1.3.1.1.7 Security

There are three main topics in terms of RSVP security: message integrity and node authentication (14), user authentication, secure data streams.

The first one, message integrity, and node authentication is very important because corrupted, stolen or spied RSVP messages can lead to different types of service failure, for example, service theft by third parties or service failure due to blocking network resources. Message loss or change can be tracked by analysis of its sequence number. Speaking about the problem of wiretap message content, RSVP protects against it by the specific encrypted hash function, described in IETF RFC 2747. It can be implemented by the usage of a specific RSVP object, called INTEGRITY, which carries Key Identifier value. This value has to be unique for each device, and each RSVP node is associated with this parameter. Actually, Key identifier is a combination of an address of sending interface and key number.

Each pair of neighbors should have at least one security association between them. There are some parameters, which are stored for each security association on the sending node in the RSVP process:

- Authentication algorithm;
- Key for authentication algorithm and its lifetime;
- Sending interface;
- Latest sequence number;

Receiving node in its turn should maintain the following values:

- Authentication algorithm;
- Key for authentication algorithm and its lifetime;
- The source address of the sending system;
- List of sequence numbers, received by the node.

Thus, due to the parameters mentioned above, the receiving system can identify the security association and sender by using the Key Identifier.

Basic IPsec technology is not recommended for RSVP due to following reasons:

- The logic of IPsec protection is based on the destination address, but control plane messages and data plan messages in RSVP may follow different ways, and as a result, an additional IPsec session can be required;

- RSVP systems are not limited to those that face one another across a communication channel;
- If the transport and higher-level headers are encrypted, RSVP's generalized port numbers cannot be used to define a session or a sender (14);
- RSVP is intended for IP packets carrying protocols that have TCP or UDP ports. IPsec does not have such port (18).

The second one, user authentication, it is closely related to policy control. If a reservation request comes from the user, which is not authenticated, this request will not pass the policy control. As far as IETF RFC 2205 does not specify any certificates for user authentication, this task can be completed again by the usage of the INTEGRITY option.

In order to solve a problem with secure data streams, an extension for IPsec, IPsec SPI (IETF RFC 2207), was invented. The key idea of this technology is to extend RSVP for the usage of the new parameter SPI (Security Parameter Index) instead of TCP and UDP ports. SPI is assigned depending on the destination address and associated with the sender. As a result, two senders to the same destination have different SPIs. This parameter is put in a new object called FILTER_SPEC. These innovations do not change the RSVP message format, it changes only the way packets are processed. In particular, each message with object FILTER_SPEC and SPI in it must be further additionally analyzed by the packet classifier.

1.3.1.1.8 Performance and shortcomings

Speaking about productivity, I would like to rely on studies conducted by Ursula Schwantag at the University of Oregon. For the experiments, the topology shown in figure 7 was chosen.

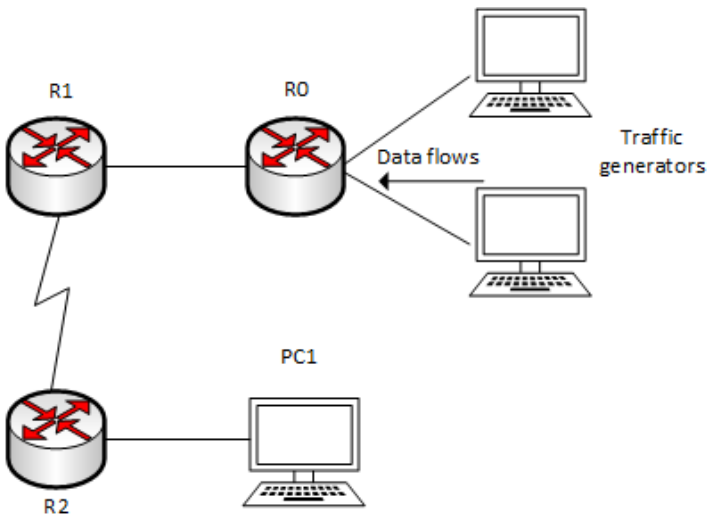

Figure 7 – Experimental topology

The experimental conditions were overloading router 1 with different data streams created by traffic generator. These data streams combined exceeded permissible throughput by 2 percent. Therefore, R1 had to drop 2 percent of all packets. In addition, RSVP was enabled for a particular stream on a serial link between R1 and R2. As a result, no packet loss for the stream with reserved resources was observed, while some packets from other streams were dropped by R1.

Exactly the same topology was used to provide experiments for analysis of jitter in delay for streams with reservation and without. Two hosts behind the R0 generate bursty flows of data. Each of these flows periodically (periods are different) changes its speed from 160 packets per second to 10 packets per second. In addition, there is a test flow of data. All these three flows occupy practically whole available Bandwidth of the serial link between R1 and R2 but do not overload it. The results of this experiment are shown in figure 8:

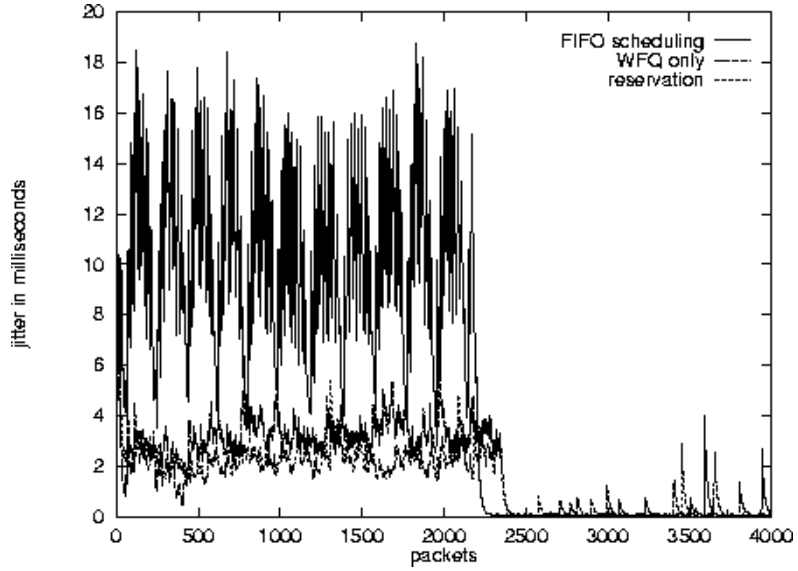

Figure 8 (source: http://ns.uoregon.edu/ursula/thesis/thesis.html) – Jitter

The top plot in figure 8 shows jitter of the test flow with FIFO (first in first out) scheduling scheme implemented on R1 and R2. Jitter is periodical because bursts are periodical. The middle plot shows jitter of the test flow with fair queuing (high rate flows cannot completely “crush” flows with a low rate) scheduling scheme implemented. The bottom plot shows the jitter after the reservation was implemented for test flow. We can see, that jitter`s value became significantly smaller.

Figure 9 shows the results of delay measurements.

Here we can see that packets proceeded with a FIFO logic have a greater delay, then packets having resources reserved and common delay variation was decreased from 40 milliseconds to 15 milliseconds.

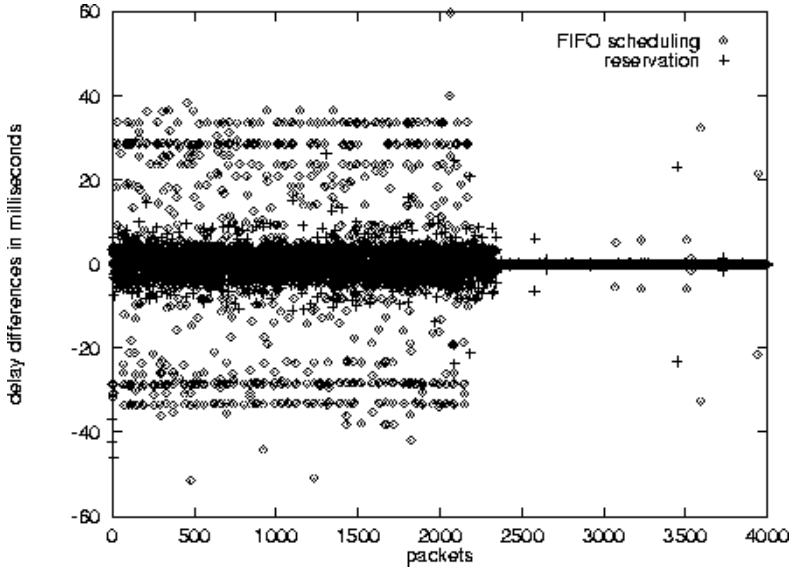

Figure 9 (source: http://ns.uoregon.edu/ursula/thesis/thesis.html) – Delay

After analysis of RSVP`s logic of work and performance in real situations, we can highlight the advantages and disadvantages of this protocol.

Advantages:

- RSVP provides particular flows with a prioritized quality of service and it is very useful in case of real-time applications;
- Flows with reserved resources have a significantly smaller delay and jitter.

Shortcomings:

- It is necessary to conduct routing at the level of flows - for each flow its state must be stored;
- Poor scalability;
- Bandwidth allocated to the source of information, while reducing the activity of the source cannot be used to transmit other information;
- Large overhead;
- In case of packet loss, long convergence time.

1.3.2 LSP tunnels

Standard RSVP, described in chapter 1.3.1, defines “session” as a data flow with a specific destination address and layer 4 protocol. However, if RSVP and MPLS protocols are combined, the definition of the “session” can be much more flexible and extended. LSP ingress device (1.2.1.5) is able to use various mechanisms in order to make a decision on label assignment. After multiple packets are assigned the same label, flow is considered to be created, and this label is used as an indicator of the flow. Such flow is also called LSP Tunnel because the packet`s content is not visible for intermediate devices in the LSP path.

Here new RSVP Session, Sender_Template, and Filter_Spec objects called LSP_TUNNEL_IPV4 and LSP_TUNNEL_IPV6 are introduced in order to support tunnels. As far as routing decision in MPLS is based on labels, IPV4 and IPV6 parts in the object`s names only specify the type of the destination address.

Some applications may require the existence of multiple paths instead of the only one. It can be very useful in case of rerouting operations and if an application wants to spread its traffic through different paths. Such sets of LSP tunnels are called LSP traffic-engineered tunnels (TE tunnels). In order to identify and associate such tunnels, two parameters are carried: a Tunnel ID and LSP ID. A Tunnel ID is a part of the Session object, while LSP ID is located in Sender_Template and Filter_Spec objects. Tunnel ID is a unique identifier of the TE tunnel, and LSP ID is an identifier of the label switched path.

There are six main features supported by RSVP-TE when working with LSPs:

- An ability to crate LSP tunnels with or without QoS requirements;
- An ability of dynamical change of the routes of the created LSP tunnels;
- An ability to check active route;
- An ability to identify and maintain LSP tunnels;
- An ability to put LSP tunnel under administrative policy control;
- An ability to perform actions with tags.

If a sender wants to create an LSP tunnel, it sends a Path message with an object LABEL_REQUEST. This object shows that the label assignment for the route is requested. In addition, it specifies the network layer protocol, which is used for the route. If the sender knows that the route most likely meets QoS requirements or efficiently uses network resources, it can use this route for some or all its sessions. In order to do it sending node puts an EXPLICIT_ROUTE object in Path message. EXPLICIT_ROUTE object is a chain of consecutive nodes in the path.

After the session is established and the sender finds out a new, more profitable route, it can dynamically switch path by changing the content of the EXPLICIT_ROUTE object. When Path messages contain an EXPLICIT_ROUTE object, it is transferred towards the destination through the hops, specified in this object. Each node collects it in the path state block. Transit devices also can change the chain, specified in the EXPLICIT_ROUTE object (ERO). In such a case not only the original ERO is stored but also a modified one. If any error occurs while establishing a new route (loops, hardware incompatibility), the sender will be notified about that.

A RECORD_ROUTE object is used when the sender wants to know which LSP tunnel does the active route goes through. It also can be used for routing error detection.

Object LABEL_REQUEST requires every node in the path, except the sender, to allocate a label for the initialized session. If any of the intermediate nodes or destination nodes cannot perform this task, it sends a PathErr message to the originator. So the sender will be immediately informed about such errors. If every node in the topology supports label assignment, the following process takes place:

- Destination node sends RSVP Resv message in response to a Path packet with a LABEL_REQUEST object;
- This Resv message contains an object LABEL, which is put into the filter spec list;
- RSVP Resv message is sent upstream and goes through the way that is reverse to the Path`s message one;
- Each RSVP node, which gets such Resv message, uses a label from the object LABEL for the outgoing traffic in this LSP Tunnel;
- If this node is not a sender, it allocates a new label for the session, puts it into the LABEL object and sends an updated RSVP message upstream. It will use this new label for an identification of the incoming traffic and associating with a session;
- RSVP nodes update ILM tables;
- RSVP path is established when the RSVP Resv message is delivered to the sender.

One of the most important tasks and requirements in traffic engineering is the ability to reroute existing and active tunnels due to various reasons. For example, it can be needed if a more efficient route is detected or if any component in an established path fails.

It is very important to understand that during the process of rerouting traffic flow should not be interrupted. That is why the best strategy for rerouting traffic is to create a new tunnel first, then to switch traffic flow to it and only after that to tear down an old LSP tunnel. However, this approach may pose a problem of the lack of resources in some nodes: if there are not enough

available resources for a new tunnel on the device, then newly arrived reservation requests will be simply discarded. RSVP-TE solves the problem by saying that on links that are common to the old and new LSPs, reservations should not be counted twice (20).

A similar situation may occur if Traffic Engineered tunnel requires increased bandwidth. If a new reservation is created, it will ask for the whole required bandwidth, while the only delta between the new value and the old one is needed to be added. Such "big" requests often fail because of admission control on the devices. The solution to this problem is quite similar to the previous one: sharing of resources on common for LSPs channels. In order to implement its reservation style, "SE" must be chosen. If an ingress node wants to carry out reroute or bandwidth-increase operations, it has to be presented as two different senders. LSP ID in object Sender_Template is used in order to fulfill this requirement. Next, follows the steps described below:

- Ingress node picks up a new LSP ID, creates new Sender_Template and ERO objects;
- Ingress node sends Path message using the original Session object, new Sender_Template, and ERO objects. Path messages with old LSP_ID are also constantly sent, in order not to lose the existing reservation;
- If the receiving link is not common, the new LSP tunnel is created;
- If the receiving link is common, shared Session object and "SE" reservation style allow the LSP to be established sharing resources with the old LSP.
- When the ingress node receives RSVP Resv message for a new LSP, it can switch traffic and tear down the old one.

1.3.2.1 LSP tunnel messages

To create LSP tunnels, RSVP-TE introduces five new objects:

Table 5 – RSVP-TE new objects

Object	Message type
LABEL	Resv
LABEL_REQUEST	Path
EXPLICIT_ROUTE	Path
RECORD_ROUTE	Path, Resv
SESSION_ATTRIBUTE	Path

1.3.2.1.1 Label Object

LABEL object can be transmitted only in Resv messages. It takes 32 bits and contains only the value of the label that can range from 0 to 1048575. For Fixed-Filter (FF) and Shared-Explicit (SE) reservation styles each label is allocated for a separate sender.

Downstream node chooses labels that will be associated with the flow. This choice can be influenced by specifying the label range in the LABEL-REQUEST message. If this range is mentioned, the downstream node must choose a new label from this range. If there are no available labels, the node sends a PathErr message with an error code “Label allocation failure”. When any intermediate node gets a Resv message with a label that is associated with a group of senders, it can also assign one common label or specify new unique labels for each sender.

After the label was allocated, the node should write this association down in its Registration State block. Only after that he can send a Resv message. If any upstream node cannot recognize the LABEL object, it sends a ResvErr message to the “receiver” with an error code “Unknown Object Class”.

1.3.2.1.2 Label request object

LABEL_REQUEST object can be transmitted only in Path messages. It can be represented in three variants:

- Label request without label range
- Label request with an ATM label range
- Label request with a Frame Relay label range

In the context of this work, only the first type will be analyzed.

LABEL_REQUEST object without label range specified takes 32 bits and consists of 2 parts: 16-bit “Reserved” field and 16 bit “L3PID” field. The “Reserved” field is set to all zeroes and must be ignored by the receiver. “L3PID” field contains an identifier of IP layer 3 protocol, which uses this path.

The presence of a LABEL_REQUEST object means that label binding for a labeled switched path has been requested. This object, similar to the LABEL one, also must be saved in Path State Block. When an upstream node sends a Path message with the LABEL_REQUEST object, it starts to wait for Resv message and, of course, must be able to handle it. The contents of the “L3PID” field must not be changed on any intermediate node. If any of these nodes do not support the protocol specified in “L3PID”, then PathErr message with code “Unsupported L3PID” is delivered to the sender.

It is obvious, that if there is a non-RSVP router in LSP, it will not process the message, nothing will be sent to the originator as an error message and the whole reservation process will fail. In order to avoid such a situation, RSVP routers are instructed not to send any RSVP messages to the neighbor if it is RSVP incapable. Instead, if RSVP router finds out that the next node in the path is non-RSVP (there are no Hello messages on the link), it generates a new PathErr message with the error code “MPLS being negotiated, but a non-RSVP capable router stands in the path”. Exactly the same message is generated when the LABEL_REQUEST object comes from a non-RSVP router.

1.3.2.1.3 Explicit route object

Only Path messages are used for transmission of the EXPLICIT_ROUTE object (ERO). This object specifies an explicit route. EXPLICIT_ROUTE object is intended only for work with unicast topologies. An explicit route is a path in the network topology usually defined by the ingress node. This path is defined by the node sequence in ERO. ERO is made up of subobjects. Format of such subobject is presented below:

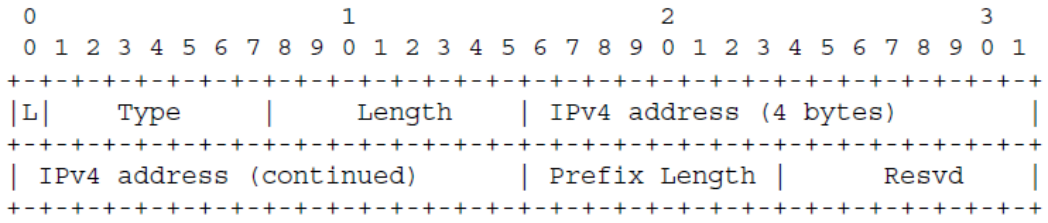

Figure 10 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE ERO subobject format

Table 6 – RSVP-TE ERO subobject`s fields

Field	Description
L	“Loose” or “strict” bit. Strict means that the router you are configuring has a direct connection to the preceding router. Loose means that the route taken from the previous router to this router need not be a direct path, it can include other routers, and can be received on any interface (21).
Type	If 1 = IPv4 prefix; If 2 = IPv6 prefix.
Length	Length of the subobject. It is always 8 bytes for IPv4 and 20 bytes for IPv6.

Field	Description
IP address	An Ipv4 or IPv6 address
Prefix Length	Length in bits of the IPv4 or IPv6 prefix
Resvd	All 0`s

Not only specific nodes can be mentioned in such subobjects, but also groups of nodes, which have to be passed along the pass. This useful feature allows working with the situation when the ingress router does not have exact information about network topology. Such a group of nodes is called an “abstract node”. If only one node specified (mask = 32), then it is called “simple abstract node”.

Therefore, now it is possible to give an exact definition of the explicit route. An explicit route is a specification of a set of abstract nodes to be traversed (20).

Since the EXPLICIT_ROUTE object has a limited number of subobjects and an explicit route is the final concept, loops can occur only because of the underlying routing protocol when a loose subobject comes across. This situation can be detected by the originator using a RECORD_ROUTE object. This object is designed to collect detailed information about the path information and detect routing loops.

A node that gets a Path message with an EXPLICIT_ROUTE object inside must determine next hop in this route. In order to complete this task, the following steps have to be performed:

- When a node gets a Path message with an EXPLICIT_ROUTE object inside, it must analyze the first subobject. If the node turns out to be not a part of the abstract node, mentioned in the first subobject, then the message was delivered by mistake and the node must send a response with an error code “Bad initial subject”. If there is now the first subobject at all, a message with error “Bad EXPLICIT_ROUTE object” is sent;
- If there is no second subobject in ERO, it means that the specified path ends. EXPLICIT_ROUTE object must be removed from the Path message. It does not necessarily mean that this node is the last one. On the contrary, it may be required to add a new ERO;
- The node analysis the second subobject. If it turns out to be a part of an abstract node mentioned in the second subobject, then it deletes the first subobject and returns to the second point. This action makes the second subobject to be the first one in the next iteration, allowing to finally get information about next hop;
- The node determines whether it is topology adjacent with an abstract node, described in the second subobject. If it is so, the node chooses next hop and deletes the first

subobject. This first subobject is then substituted with a subobject that contains information of the node having information about the next hop;

- If the node is not topology adjacent with an abstract node, described in the second subobject, then it chooses the next hop within the abstract node specified in the first subobject, i.e. along the path to the second object`s abstract node. If such a path does not exist, there are two possible situations:
    - If an abstract node from the second subobject is “strict”, then an error message with code “Bad strict node” is sent;
    - If an abstract node from the second subobject is “loose” and the way to it is not defined, then an error message with code “Bad loose node” is sent;
- Finally, the first subobject must be substituted with a subobject that contains information of the node having information about the next hop. This is necessary so that when the explicit route is received by the next hop, it will be accepted.

Figure 11 briefly illustrates the algorithm, described above.

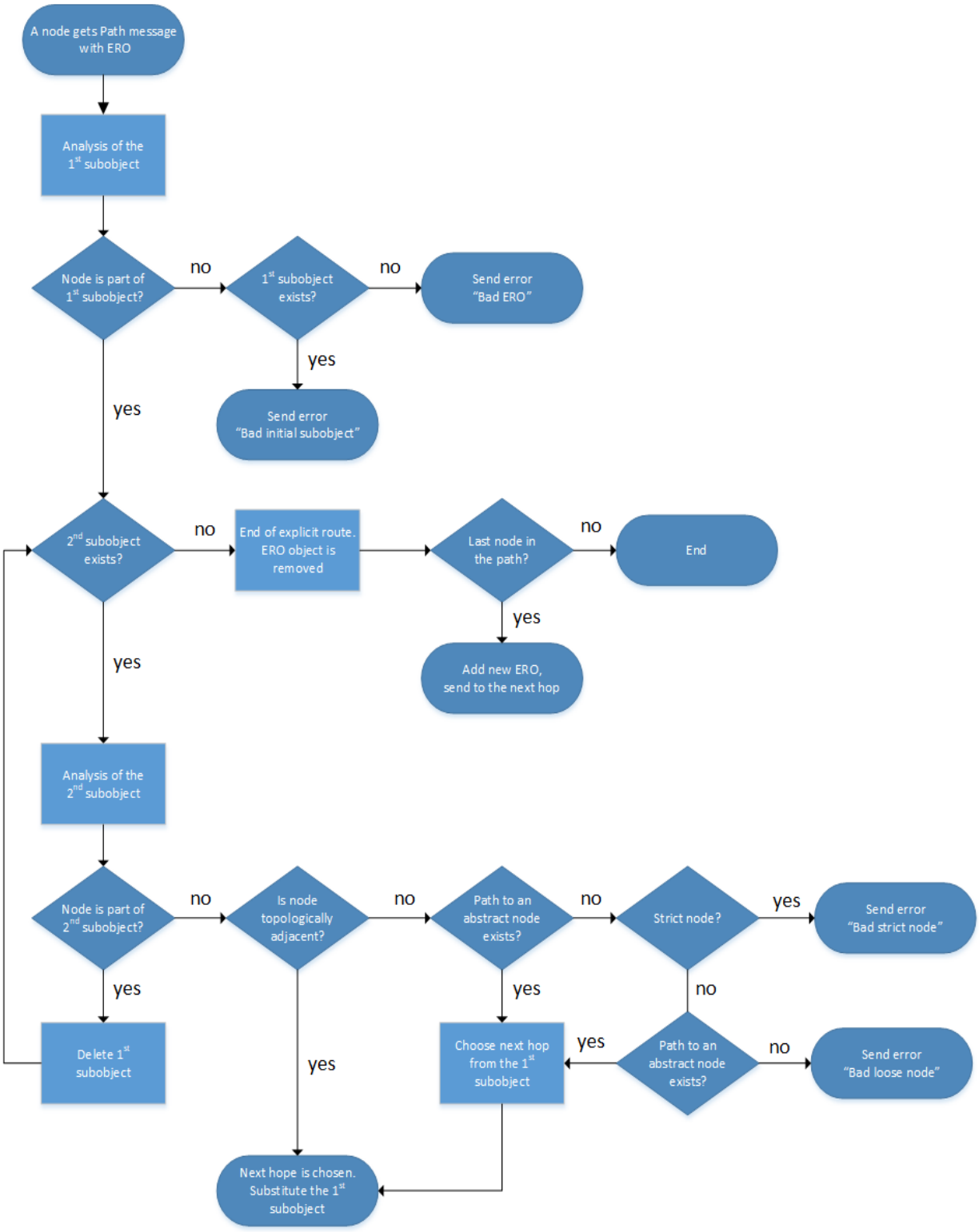

Figure 11 – Next hop selection algorithm

1.3.2.1.4 Record route object

RECORD_ROUTE (RRO) object can be transmitted by both types of messages, Path and Resv. If there are several RRO objects in Path or Resv messages, then the only first one is considered to be meaningful and only it should be analyzed. All the following RR objects must be

ignored and should not be propagated. Messages with the RECORD_ROUTE object should be used only in case all the nodes in the path support it.

The format of the RECORD_ROUTE object is very similar to the format of the EXPLICIT_ROUTE object. It also consists of subobjects. Subobjects are organized in a last-in-first-out stack. It means that the closest to the beginning of the RRO subobject is a top one and if any subobject is added, it is pushed on the top of the stack. Nowadays only three types of subobjects are defined:

- IPv4 subobject
- IPv6 subobject
- Label subobject

IPv4 and IPv6 subobjects are practically identical and differ only in the address field. Format of IPv4 and Label RRO`s a subobject are presented below on the figures 12 and 13. Fields description is provided in tables 7 and 8 correspondingly.

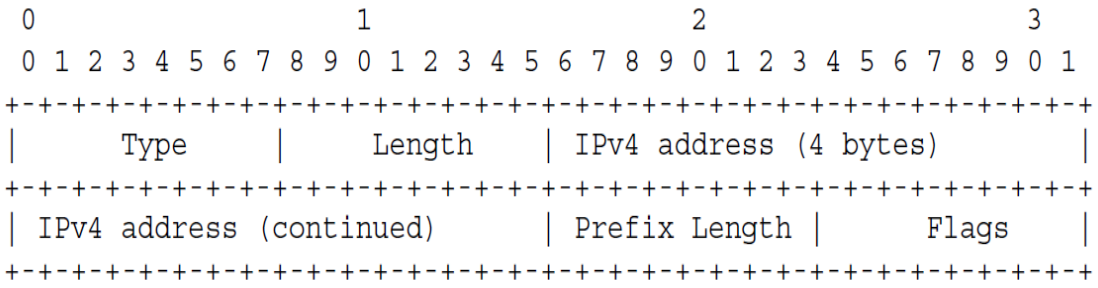

Figure 12 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE RRO IPv4 subobject format

Table 7 – RSVP-TE RRO IPv4 subobject`s fields

Field	Description
Type	If 1 = IPv4 prefix; If 2 = IPv6 prefix.
Length	Length of the subobject. 8 bytes for IPv4; 20 bytes for IPv6.
IPv4/IPv6 address	A host IP address that should be unicast and network-reachable. Loopback addresses should not be used.
Prefix length	Length of the prefix. 32 for IPv4; 128 for IPv6.

Flags	If value = 1, then it indicates that the link downstream of this node is protected via a local repair mechanism; If value = 2, then it indicates that a local repair mechanism is in use to maintain this tunnel.

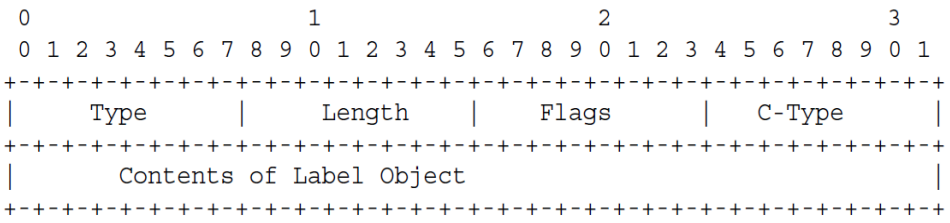

Figure 13 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE RRO Label subobject format

Table 8 – RSVP-TE RRO Label subobject`s fields

Field	Description
Type	3 = Label
Length	Length of the subobject in bytes
Flags	If value = 1, then it indicates the Global value, label will be understood on any interface.
C-Type	C-Type of the Label object, by default is 1
Contents of the Label Object	Copied value from Label object.

Nodes usually add RECORD_ROUTE objects in Path messages during the RSVP initialization session. Original RRO contains only one subobject with an IP address of the sender. If the node desires not only IP addresses to be recorded, but also labels, it should set Label_Recording flag in the SESSION_ATTRIBUTE option.

When Path message with RECORD_ROUTE object arrives on an intermediate node, this device saves a copy of it in its Path State Block. Then the intermediate router has to add a new subobject to the existing RRO and sends further. This subobject must be an address of this node. The address should be an address of the outgoing IP interface, through which Path message is sent. If the Label_Recording flag in the SESSION_ATTRIBUTE object is set, then the new Label Record subobject also has to be included. In this case Label subobject should be pushed firstly. There is should not be a situation in which node pushes on the RECORD_ROUTE object only

Label subobject and ignores IP subobject. If the node uses Global label space, the corresponding flag should be set.

If after the addition of subobjects into RECORD_ROUTE object it becomes too big to fit the size of the Path message, RRO object is simply discarded and the message is sent further. PathErr message with an error value “RRO notification” should be sent. When the sender receives such error it should exclude RECORD_ROUTE object from its Path messages.

When the destination node receives Path message with RRO, this router understands that the sender needs route recording. As an answer it sends RSVP-TE Resv message with a new RECORD_ROUTE object. The packet transfer process is similar to the one that was described above for the Path messages. The only difference is the direction, it is reverse. As a result of such exchanging processes, every node along the path knows the exact route to “sender” and “destination” in the topology. This can be very useful for network management.

There are three possible RSVP RRO`s use cases: loop detection mechanism and collection of up-to-date detailed path information. The first one is Layer 3 loop detection. When intermediate routes receive a message with RRO, it should analyze all subobjects in it. If the router finds itself in the list, it means that the routing loop exists.

There are two types of routing loops:

- Transient loop – normal condition when Layer 3 routing protocol tries to converge on a given path for all the destinations;
- Permanent loop – permanent loop. This means that some mistakes in network configuration were made.

If the router gets Path message with RECORD_ROUTE object, analysis it and finds routing loops, it must send PathErr message with an error code “loop detected” and drop this message.

If the router gets Resv message with RECORD_ROUTE object, analysis it and finds routing loops, it must simply drop this message, because Resv should not contain it if Path message was successfully delivered.

The second useful function, collection of up-to-date detailed path information, provides very important information about the route to the sender or receiver, so any changes in network topology can be detected.

1.3.2.1.5 Session object

As it has already been sent in previous chapters Labeled Switched Path in RSVP technology is defined by the Tunnel ID. SESSION object is used for carrying Tunnel destination IP address, Tunnel ID and Tunnel source IP address. It is transmitted in RSVP-TE Path messages during the setup of LSP Path. The format of the SESSION object for IPv4 is shown in figure 14.

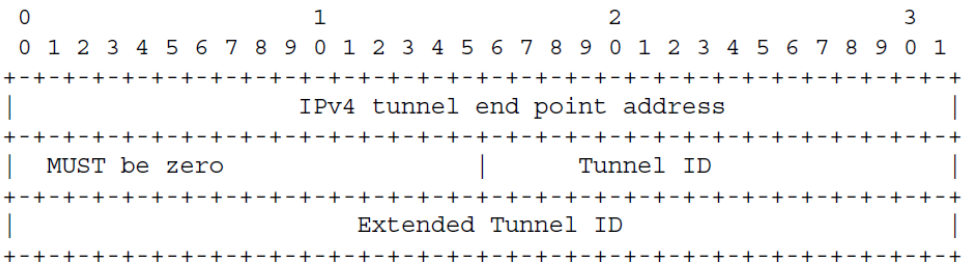

Figure 14 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE SESSION object format

Table 9 – RSVP-TE SESSION object fields

Field	Description
IPv4 tunnel endpoint address	An IPv4 address of the tunnel`s egress node.
Tunnel ID	A 16-bit identifier, which is unique and remains constant during the life of the tunnel.
Extended Tunnel ID	A 32-bit identifier. 32-bit identifier used in the session and stored throughout the life of the tunnel. Usually set to zero. Input nodes that want to narrow the session to an input / output pair can put their IPv4 address here as a globally unique identifier (20).

SESSION objects can also be used in IPv6 networks. The only difference between objects for IPv4 and IPv6 is an address format in the “tunnel endpoint address” field.

1.3.2.1.6 Sender Template object

SENDER_TEMPLATE object is used for identification of the traffic flows within the context of sessions. SENDER_TEMPLATE in RSVP-TE includes a source address of the sender and LSP ID. LSP ID should be unique within the source address`s owner and represent an instance of the tunnel, identified by the Tunnel ID in the SESSION object. The format of the SENDER_TEMPLATE object for IPv4 is shown in figure 15.

SENDER_TEMPLATE object can also be used in IPv6 networks. The only difference between objects for IPv4 and IPv6 is an address format in the “tunnel sender address” field.

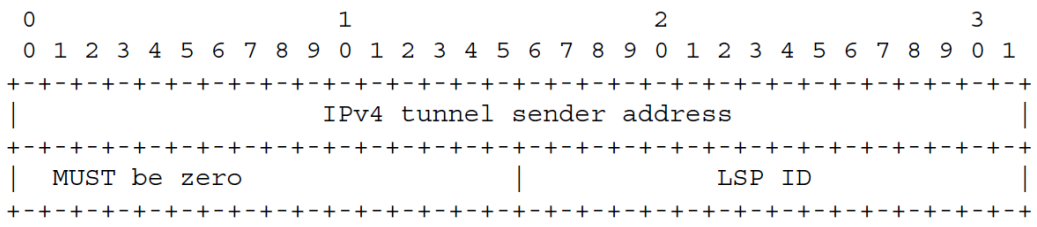

Figure 15 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE SENDER_TEMPLATE object format

Table 10 – RSVP-TE SENDER_TEMPLATE object

Field	Description
IPv4 tunnel sender address	IPv4 address of the sending node.
LSP ID	A 16-bit identifier, which is unique within the sender

In order to illustrate how the LSP path is uniquely identified with parameters LSP ID and Tunnel ID, I would like to show you figure 16. In this picture, one tunnel and two possible LSP paths can be noticed. Tunnel ID is the same for both LSP paths, but LSP ID is different.

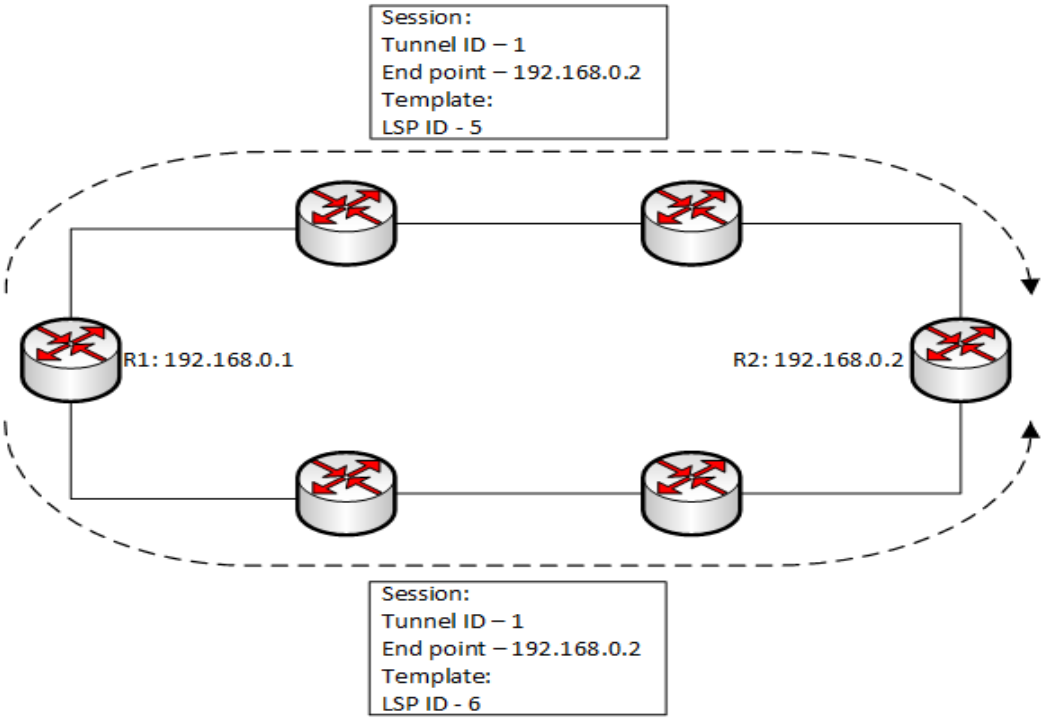

Figure 16 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE LSP paths

EXPLICIT_ROUTE object has the same format as a SENDER_TEMPLATE object.

1.3.2.1.7 SESSION_ATTRIUTE object

SESSION_ATTRIBUTE object can be transmitted only in Path messages. It is used for an indication of the session`s specific demands, some of which are listed below:

- The need to write labels in the object;
- Usage of SE reservation style;
- Bandwidth protection (need in Bandwidth guarantees in case of failures) (22).

The format of the SESSION_ATTRIBUTE object is shown in figure 17.

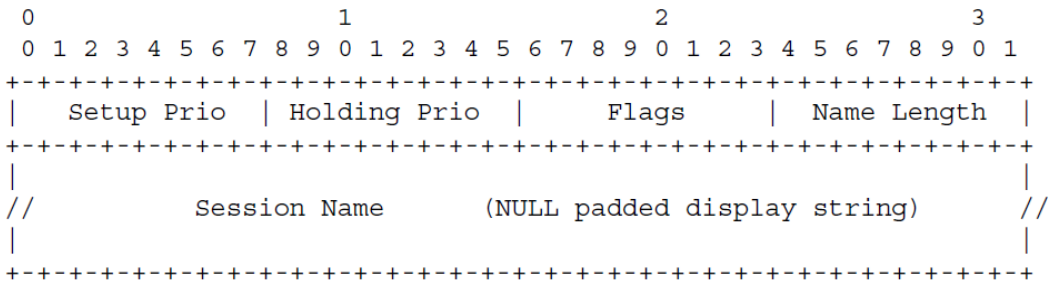

Figure 17 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE SESSION_ATTRIBUTE object format

Table 11 – RSVP-TE SESSION_ATTRIBUTE object

Field	Description
Setup Priority	This flag indicates the ability of the current session to take the resources, which have already been allocated to another session. Possible values are from 0 to 7, where 0 is the highest priority.
Holding priority	This flag indicates the ability of the resources of the current session to be taken by another session. Possible values are from 0 to 7, where 0 is the highest priority.
Flags	If the value is 1, then, in case of failures, transit routers are permitted to violate route defined in ERO and allowed to run its local repair mechanism; If the value is 2, it means that label recording is needed in RRO; If the value is 4, then the SE reservation style is the desired one.

Name Length	The length of the name in Session Name field, before padding;
Session Name	Name of the session, padded by zeroes.

Not all devices nowadays can process fields Setup Priority and Holding priority, that is why its support is defined as an optional feature. In case the transit node cannot process these fields, it just sends it further downstream. If it is able to process, the following algorithm is used:

- If there are both elements in the SESSION_ATTRIBUTE object, then Setup Priority is used. It is mapped to the value called “Preemption Priority” accordingly to table 12:
  - If the requested bandwidth value exceeds the available one, then PathErr message is sent with an error code “Requested bandwidth unavailable”;
  - If the requested bandwidth value is less then unused one, then reservation is successful;
  - If the requested bandwidth value is available, but already occupied by a session with a Holding Priority smaller than new session`s Setup Priority, then its resources can be preempted and reservation succeeds;

Table 12 – Preemption value mapping

Preemption priority	Setup Priority
0-3	7
4 – 15	6
16 – 63	5
64 – 255	4
256 – 1023	3
1024 – 4096	2
4096 – 16383	1
16384 – 65535	0

1.3.3 Hello messages

In RSVP-TE Hello messages are intended for the detection of neighbor`s failures. This mechanism is typically used only when the link layer failure detection mechanism cannot cope with an error due to some reasons.

Hello messages mechanism works in such a way that only one node in the session can initialize it, while the other one cannot. There are two possible objects in a Hello message:

- Request object;
- ACK object.

Usually Hello message mechanism is activated after the expiration of a preconfigured timer`s value. Every message with a Request object should be answered by the message with an ACK object. This mechanism can detect two types of neighbor`s failure:

- Neighbor loss;
- Neighbor`s restart.

If the neighbor does not respond on three Hello messages with a Request object in a row, then it is considered to be lost and RSVP-TE Hello process should be re-initialized.

The neighbor`s restart can be determined by collecting and tracking its “instance” value. If any change in this value is detected, it means that the neighbor had a reset. When such a situation occurs, a new “instance” value advertised to a neighbor is immediately created.

Hello messages are always exchanged between two immediate neighbors. Source IP address there is an IP address of a sending node and destination IP address – is an address of a destination node. The TTL value is always “1”.

Speaking about Hello message format, it is important to say that Request and ACK objects have the same format, shown in figure 18:

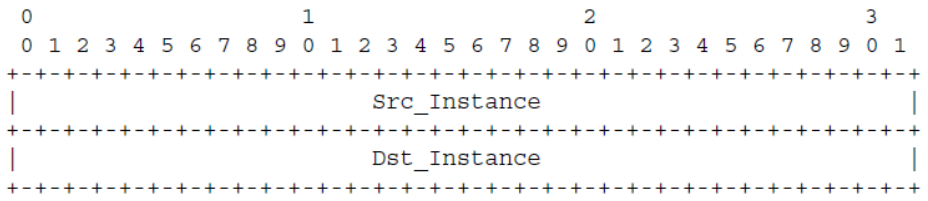

Figure 18 (source: https://tools.ietf.org/html/rfc3209) – RSVP-TE Request and ACK object`s format

Table 13 – RSVP-TE Request and ACK object`s fields

Field	Description
Src_Instance	This is a 32-bit value which indicates the sender`s instance. This field must not be a field with only zeroes. Each node should save the neighbor`s instance value and change it in case of reset or loss of the neighbor.
Dst_Instance	This is a 32-bit value, representing the most actual, latest instance value received from the neighbor.

The usage of a Hello message mechanism is optional. Corresponding messages can be simply ignored by the node that does not want to participate in the RSVP-TE Hello exchange process. In this chapter I would like to discuss only situations in which both of the neighbors participate in it. Hello Request messages are generated accordingly to the timer by the sender. This timer is called Hello_interval timer and by default it is set to 5 ms. Sender puts its instance in the Src_Instance field and neighbor`s instance value in the Dst_Instance field. If it is the first Hello message in the process and neighbor`s instance value is not known, then the Dst_Instance field is set to all zeroes.

After receiving of Hello Request message, the destination node must reply to it with an ACK message. It also must make sure that the neighbor`s instance value was not changed. In case the Dst_Instance field in Hello Request message is a field by all zeroes, the destination node puts there its instance value. As it has already been said, errors are detected by comparing newly arrived instance value to the recorded one. If it has been changed or set to zeroes, it means that reset or neighbor`s loss happened.

Hello message mechanism does not influence any other RSVP-TE process. It just provides tools for detecting errors in a faster way.

1.3.4 Security

Generally speaking, RSVP-TE security mechanisms are similar to mechanisms, discussed in chapter 1.3.1.1.7 for the original RSVP protocol. However, there is one difference. In RSVP security associations can be identified only using the Key Identifier value, which is based on the sender`s IP address. In RSVP-TE this identification process may use the label`s values.

1.4 Operation

To this moment, in previous chapters, the mechanisms of operation of the main parts of MPLS-TE have been analyzed in detail. These parts are RSVP-TE and MPLS protocols. This chapter describes how these protocols are combined to provide MPLS-TE technology.

1.4.1 Routing protocols

Nowadays three main types of routing protocols are defined: Link-state routing protocols, Distance vector routing protocols and Exterior gateway routing protocols. In terms of MPLS Traffic Engineering only the first type, Link-state routing protocols, can be used. Its reason will be explained in the next paragraphs.

Link state routing protocols are protocols in which all routers in topology have a complete map of it. In other words, every node knows about all network prefixes and devices. Routers working with Link state routing protocols keep all this information in Link State Database (LSDB). LSDB is then used for the construction of the routing table.

There are three types of information in LSDB:

- Topological information. It is also known as a “network map” and contains information about routers, links between routers and interface`s costs;
- Addressing information. Here the information about prefixes and its bindings to network devices is stored;
- Link information. It contains the next hop address for particular prefixes. Link Information has a local meaning within the device.

Topological information and addressing information must be the same on all the network devices. In other words, it should be synchronized. This synchronization can be achieved by using a technology, which is that each router creates its own “network record” and distributes it to all other routers. Network record consists of three parts:

- Router ID;
- Connected link identifier;
- Router interface`s costs.

When the router gets such records from other devices, it saves this information and combines it with its own record. If two routers are connected to each other with the same link, they can find it out by investigating link identifiers.

If the router fails, it cannot delete its network record from all other devices. This failure should be detected by neighbors by using the Hello mechanism, supported by Link State routing protocols. After the failure is detected, routers should generate refreshed network records in which the link to the failed router is not listed. Thus, a network section with a faulty device becomes isolated, and other routers do not consider it when building new routes.

The other two types of routing protocols, distance vector and exterior gateway, use different principles of work and, most importantly, do not imply that each router has full information about the topology. This criterion is a key one for traffic engineering, because it is needed for resource allocation information flooding through the network. That is why Link-state protocols are the only opportunity for MPLS-TE technology.

1.4.2 Path calculation

When all routers know information about the whole network about the topology, these routers start an algorithm, which calculates the shortest path between sender and receiver. Link state routing protocols in order to perform this task use “Dijkstra Shortest Path First algorithm”.

1.4.2.1 Dijkstra Shortest Path First algorithm

Dijkstra Shortest Path First algorithm is an algorithm that works on graphs. It finds the shortest paths from one point of the graph to all others. The shortcoming of this algorithm is that it does not support ribs with negative weight.

Description of steps in the Dijkstra algorithm is given below:

- It starts on the node A and initializes all distances to other nodes i as Distance [i] = ∞, while Distance [A] = 0;
- For every node B, that is adjacent to node A, a new distance is found by using the cost of the link that connects them. If newly found Distance [B] is smaller, than the previous one, then it will be used in further calculations;
- After this node B is considered to be the current one;
- If B is a destination node, then calculations are finished. If it is not, the algorithm returns to step 2.

On figure 19 an example of Dijkstra algorithm is provided:

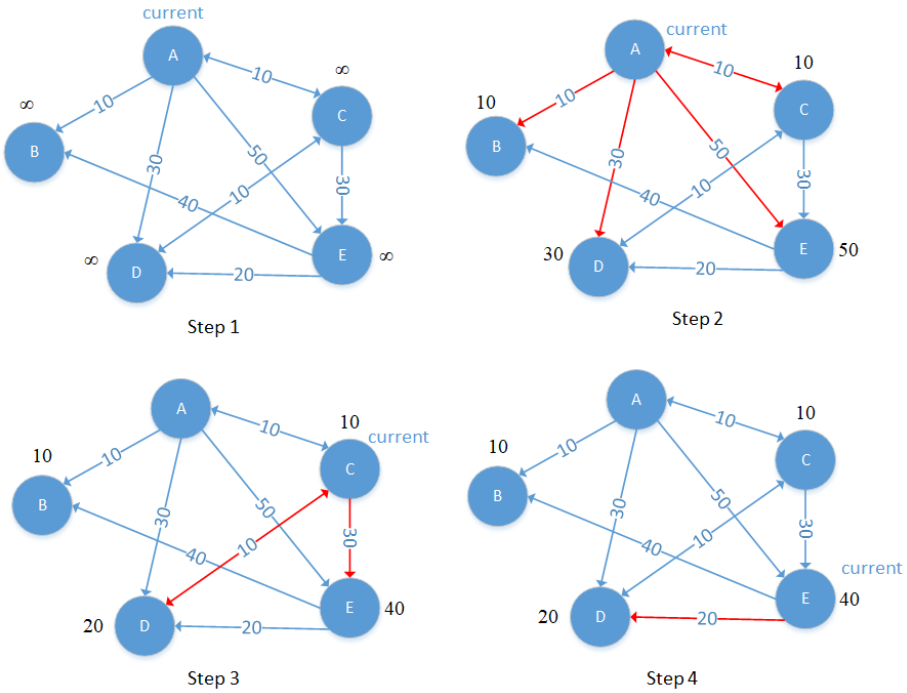

Figure 19– Dijkstra SPF algorithm

- Step 1: node A is current and distances to all the nodes are initialized with value = ∞;
- Step 2: nodes B, C, D, E are adjacent to node A, so first distances to them are found;
- Step 3: node C is the current one, it is not final, distances from it to neighboring routers are found. As far as these distances are smaller than previously calculated ones (20 < 30 for D and 40 < 50 for E), new values are in use. The point B has no outgoing paths, therefore it cannot be current and should not be considered;
- Step 4: node E is the current one, it is not final, distances from it to neighboring routers are found. As far as the new distance is bigger (60 > 20 for D), new values are not in use. Now we see that the shortest paths to all nodes are found: A-D with cost 20, A-B with cost 10, A-C with cost 10, A-E with cost 40.

1.4.2.2 CSFP

Despite the fact that the Dijkstra Shortest Path First algorithm is commonly used for the routing protocol and has several advantages, its expansion called Constrained Shortest Path (CSPF) first was invented for Traffic Engineering tunnel's installation. This process is very similar to the original SPF, however it has two key differences:

- CSPF finds the best route not to all routers in topology, but only to the Traffic Engineering tunnel's endpoint. This means that when an endpoint router gets into the Path list, the search for the best route stops;
- In SPF the only metric is a cost of the outgoing interface. In CSFP three parameters are used:
    - Bandwidth;
    - Link attributes;
    - Administrative weight.

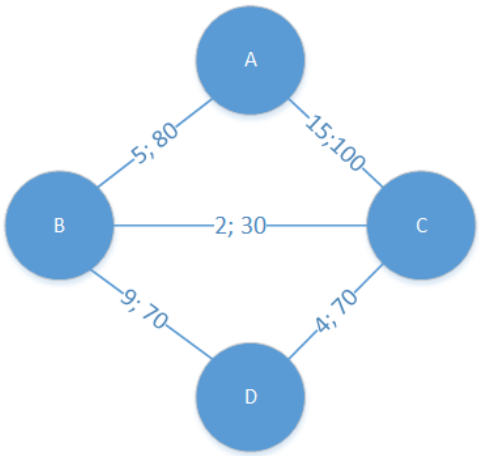

Figure 20– CSPF algorithm

The route, which is chosen by CSPF as the best one, is the shortest route that meets the established requirements (for example, minimum bandwidth). On the figure 20 an example of how CSPF process takes new metric parameters is given.

For this example, let us assume that only cost of a link (first number) and bandwidth are taken into consideration during the selection process. The only pre-requirement here is that minimum bandwidth of 60 Mbps should be supported by the path from node A to node D. It is seen, that if the best path was chosen using SPF algorithm, then route A-B-C-D had to be chosen, because of its lowest total cost. However, this path does not satisfy our requirements because link between routers B and C has a bandwidth of 30 Mbps. That is why this path will not be chosen by CSPF. In this case algorithm will choose route A-B-D for Traffic Engineering tunnel with a total cost of 14 and minimum bandwidth of 70.

If we speak about limitations, which can be introduced by “link attributes”, then these limitations are mostly about some particular links. There is an option in MPLS-TE, called “affinity bits”. This option is assigned to the traffic-engineering tunnel and tells it if this tunnel can use a particular link. Let us return to the example discussed above. If affinity bits were set in such way that link between routers B and D was prohibited, then A-B-D path would not be chosen and path A-C-D would take its place.

Administration weight is just a cost of a link in terms of Interior Gateway Protocols. For OSPF this cost is calculated using the formula:

$$\text{Cost} = \frac{\text{Reference BW}}{\text{Outbound Interface BW}}$$

, Where Reference Bandwidth is a default value of 100 Mbps. So if the outbound interface has a Bandwidth of 10 Mbps, then the link`s cost will be 10. In IS-IS all links by default have a cost value of 10.

Another important characteristic of the Constrained Shortest Path algorithm is that it does not allow load balancing. If CSPF finds more than one paths available with equal attributes, it uses its tiebreakers:

- Choose path with the largest Minimum BW available;
- Choose the path with the lowest number of hops;
- Choose the path that is on top of the list.

After CSFP algorithm is done and the shortest path is found, RSVP-TE picks up information about this path, puts in Path messages in EXPLICIT_ROUTE object (discussed in paragraph 1.3.2.1.3) and sends it downstream in order to reserve resources.

1.4.2.3 Path recalculation

Since any network is a dynamic environment and can be updated, the situation can occur in which new, more efficient and shorter path can be detected. MPLS-TE supports four path`s re-optimization options:

- Periodic;
- Manual;
- Event-driven;
- Lockdown.

Some vendors support periodic re-optimization technology. It means that there is a preconfigured timer and each its interval network devices try to find better paths for TE tunnels. By default, timer`s value is 60 minutes.

Manual re-optimization is about manual switching between paths if the network engineer is confident that new option is more efficient.

In case of Event-driven re-optimization, TE tunnels can be redirected in the event of a network topology change, for example, if any link comes up. This is disabled by default for reasons of network stability.

Some paths can also be locked down. This means that no other type of path`s re-optimization can be performed on this path. It can be used if there are very efficient links along the current path these links must be used in any case.

1.4.2.4 Inter-area TE tunnels

Link state routing protocols have an ability to divide a network topology logically into different zones. In the previous chapters of this capstone project only intra-area tunnel`s installation methods were discussed. However, sender and receiver are often located in different areas.

There are three possible path setup methods for inter-area traffic engineering tunnels are known nowadays (25):

- Contiguous LSP – LSP between head-end and tail-end nodes of the tunnel is setup using hop-by-hop signaling between neighboring network devices;
- LSP stitching – Separate LSPs are created here for each area. Then these LSP are connected with each other with a 1:1 relationship;
- LSP Nesting – Here LSP between head-end and tail-end nodes goes inside other, intra-domain tunnels.

The most efficient and popular variant nowadays is the first one. Contiguous LSP technology`s methods are discussed in more detail below.

Path calculation process here is initiated by the head-end router as it was in the case with an intra-area tunneling. The only difference is that the Contiguous LSP process is performed also on each Area Border Router (ABR) and on each LSR, which finds out that in its next-hop subobject in EXPLICIT_ROUTE object flag L is set (1.3.2.1.3). ABR is a router, which has interfaces connected to two or more different areas. This flag L means that next-hop is “loose,” and the current network device should independently find the path to the next ABR. If the path is found, following ERO subobjects will be passed through and new ones, describing the path to next ABR, will be added to ERO. This process is called EXPLICIT_ROUTE expansion.

Of course, errors can occur during the establishment of the inter-area TE tunnel. These error can be divided into two groups:

- Errors, which happen in area where the head-end device is located;
- Errors, which happen in any other area.

If failure occurs in the first case, then the head-end router should get a “path error” message. When it gets such message, it should try to reinitialize session only after the predefined time interval is over.

If a failure occurs in any other area, then ABR, which has just performed EXPICIT_ROUTE object expansion, has to try to find an alternative path to the next ABR to the tail-end node. If an

alternate path does not exist, the “path error” message must be sent to a head-end device. This algorithm is called “Crankback”.

Inter-area MPLS-TE, as an intra-area MPLS-TE, supports feature called “Fast Reroute”. The key idea of it is to provide additional reliability of TE tunnels in case of errors in the network. In order to implement it, second TE tunnel, called a backup one, should be pre-configured. When something fails in the primary TE tunnel, traffic is simply switched to the backup one and “path error” message is sent to the head-end router. Head-end router, in its turn, has to try to re-optimize existing tunnel.

If Fast Reroute feature is not activated, then the node, which is the closest one to the failed network segment, sends the same “path error” message. Upon receipt of such a message, the head-end network device must send PathTear message and afterward try to establish completely new TE tunnel.

Speaking about re-optimization feature for inter-area tunnels, it is said in (26), that “Work on this topic is still ongoing at the IETF”. Nevertheless, this functionality exists, although it is currently very complex and inefficient.

1.4.3 Traffic forwarding through the tunnel

After each network device got the whole network map by using Link State routing protocol, the shortest path between head-end and far-end routers was found by algorithm CSPF and required resources were reserved by protocol RSVP-TE, TE tunnel is considered to be established. At this point network engineer gets one more task: he must forward the necessary traffic through the tunnel. Three possible options exist here:

- Static routes;
- Policy-based routes;
- AutoRoute.

Static routes is the easiest way to perform this task. The engineer just has to point down the static route in right TE tunnel.

Policy-based routing is also quite simple. It works just like traditional IP forwarding based on policies, defined on the network devices. These policies can be routing maps, access lists and prefix lists.

The third option, AutoRoute, is unique for MPLS-TE technology. Its uniqueness lies in the fact, that the IGP routing protocol cannot be enabled at the entry point to the TE tunnel. This

limitation is due to the fact that Traffic Engineering tunnels are unidirectional and routing protocols do not work in such conditions. However, MPLS-TE technology needs a mechanism that will dynamically tell network device that tunnel can be taken as a direct link to tail-end node, and that corresponding traffic can be passed through it. AutoRoute feature is designed just for this. It works the following way:

- IGP routing protocol runs its SPF;
- If SPF algorithm sees, that destination node is a tunnel`s tail-end or it is a device that is located behind tunnel`s endpoint, then IGP treats TE tunnel as a route, not IGP path.

In other words, when AutoRoute works, traffic destined to the tunnel`s tail-end node is always routed through the TE tunnel, because in the routing table entry point to the TE tunnel replaces physical interface.

1.4.4 Quality of service

Quality of service (QoS) refers to any technology that manages data traffic to reduce packet loss, latency , and jitter on the network. QoS controls and manages network resources by setting priorities for specific types of data on the network (27).

Nowadays only two QoS architectures exist:

- Integrated Services (IntServ);
- Differentiated Services (DiffServ).

Briefly speaking, IntServ can be used only in small networks and is useful only when it is needed to provide QoS for end-to-end or host-to-host connections. DiffServ, in its turn, is very scalable and can be implemented in service provider networks.

1.4.4.1 Diffserv architecture

There are two main components in DiffServ architecture:

- Traffic preparation – this component includes policing, classification and shaping of the traffic. It is performed only on the tunnel’s head-end router;
- Per-hop treatment – this is how traffic is treated by all other devices. It can include queuing, shaping, or dropping.

Policing is a traffic processing process in which its characteristics are compared with those specified in the service contract. This is necessary because DiffServ architecture does not allow network overflow. If traffic exceeds allowed resources, it can be marked, shaped or dropped.

Marking is just about putting a special note on the packet, which indicates that it is “out-of-rate” or “in-rate”. Shaping is a process of delaying traffic on a network device in order to meet characteristics defined in the service contract. Dropping is an intentional packet loss in case shaping does not help.

Classification is the process of analyzing traffic and determining its type (voice, video, text, etc.). While IP packets are classified based on information in Layer 3 header, in case of MPLS packets, it is a little bit more complicated process, as far as network device cannot see Layer 3 header in MPLS packets. Here are used 3 bits from a TC field, shown on the figure 2. In addition, only the level label in a stack can be used for such analysis.

Queuing is a part of router`s normal behavior. It just manages the order, in which incoming packets will be sent further to its destination. Multiple techniques for traffic queueing exist nowadays, such as:

- First in first out (FIFO) – Packets leave network device in the same order as arrived;
- Round Robin – Here multiple processes get an equal number of resources and are polled for sending in turn;
- Class-based queuing – The queue for sending packets is based on their belonging to the class. The higher the priority of the class, the more resources the corresponding process gets;
- Low-Latency Queuing – Packets with delay-sensitive data are provided with preferential treatment.

Here arises the following question: how network devices can detect traffic type or its class. Firstly, traditional IP networks will be discussed.

In IP protocol header, defined in IETF RFC 3168, there is a field called “IP Prec or DSCP”. Its format is shown in figure 21.

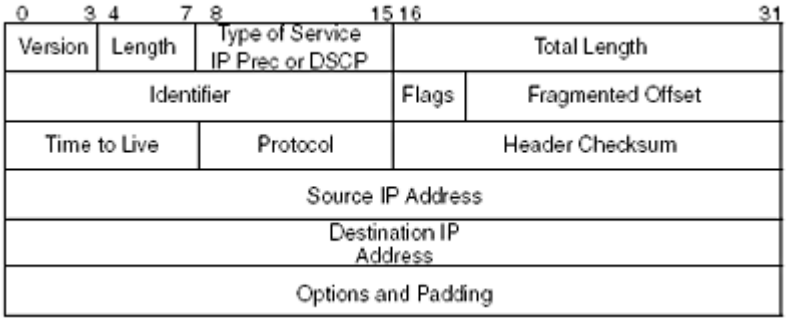

Figure 21 (source: https://searchnetworking.techtarget.com/tutorial/Routing-First-Step-IP-header-format) – IP header

This field is used for class definition and takes 6 bits. In previous versions of IP header, its place was taken by 3-bit field “IP Prec”. There is no big difference in DSCP and IP Prec, except that the first one defines more classes of service. All the classes from IP Prec architecture are uniquely mapped to DSCP values. New DSCP classes can be grouped into two types:

- Assured Forwarding – there are 13 possible values in this group. Its key idea is that Assured Forwarding class is used when packets loss is the most important criteria in the network, while latency is less important;
- Expedited Forwarding – there is only one class in this type. Expedited Forwarding guarantees required low-latency and low-jitter.

As it has already been said above, MPLS provides only 3 bits for classification in its headers. Here comes the difference between implementation of DiffServ architecture in MPLS and IP topologies. In order to support all possible classes from IP networks (64 value), MPLS supports the one-to-many matching scheme. However, despite the lack of bits in the header, it is not considered to be a problem nowadays because MPLS mostly works in production and service provider`s large networks and 8 available classes is still enough to satisfy all customer`s possible requirements.

In order to understand how this mapping works in real-life situations, I would like to discuss three typical cases:

- IP-to-MPLS – It happens when IP packet enters MPLS topology. In this case three most significant bits are copied from DSCP field to a top label`s TC field;
- MPLS-to-MPLS – Such a situation may happen when packet already has label stack and now some manipulations are carried out with it. For example, if a new label is pushed onto an existing label stack, TC field should be copied from the underlying label to a new one;
- MPLS-to-IP – Such a situation happens when packet leaves MPLS cloud and enters IP network. In this case the only manipulation is required to perform is a label stack deletion.

1.4.4.2 DiffServ-aware Traffic Engineering

Up to this point in terms of Quality of Service, only pure DiffServ for MPLS was discussed. It is important to understand that it has not much in common with QoS in MPLS-TE technology. The only kind of DiffServ that is used in MPLS-TE is a DiffServ-aware Traffic Engineering (DS-TE).

DS-TE provides MPLS-TE components with information about traffic`s Quality of Service characteristics and allows it to make reservations accordingly to pre-configured QoS requirements. Therefore, resource reservation is done at per-class level. Each DiffServ class of service gets here a separate LSP for its traffic. Resources of such LSPs are called “sub-pool”, while the whole TE tunnel’s Bandwidth is called “pool”. “Subpool” is configured to be more restrictive in order to achieve more efficient QoS performance.

If DS-TE runs in the topology, then the functionality of Link State Routing protocol and RSVP-TE protocol is extended:

- DS-TE adds some limitations to the CSPF process. These limitations are defined by the class of service and mostly effect Bandwidth`s requirements. In order to provide information about these classes to all network devices, Link State Routing protocol should send relevant LSA messages to routers. Such LSA`s contain information about required Bandwidth for each class;
- RSVP-TE, in its turn, defines a new object, called CLASSTYPE. CLASSTYPE object specifies needed class of service. It is sent only in Path messages. Each network device, which receives this message, records its content and performs admission control. If it cannot allocate required resources, PathErr message is sent to the originator.

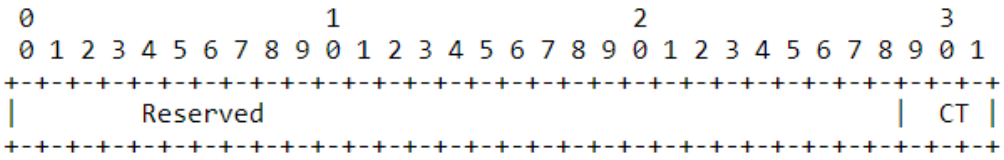

Figure 22 (source: https://tools.ietf.org/html/rfc4124) – RSVP-TE Classtype object

Table 14 – RSVP-TE Classtype object`s fields

Field	Description
Reserved	This field is not used, it must be set to all zeroes.
CT	Type of the class. Possible values are from 0 to 7

1.4.5 Protection

MPLS-TE usually works on the provider`s side and is considered to be a very important part of big companies’ network infrastructures. However, even at that level of responsibility, different types of errors and failures may occur, that is why MPLS-TE` mechanisms, which allow to reduce

the impact of such errors, play a great role in networks nowadays. These mechanisms can be categorized into three groups:

- Path protection;
- Node protection;
- Link protection.

The first one, Path protection, is the creation of a backup Label Switch Path, which starts to carry traffic only in case of some problems have happened with the primary one. Backup LSP, also known as Secondary and Standby, should have exactly the same constraints as an active LSP. Otherwise, it will not be able to perform a full replacement. In addition, it is very important for network engineer to understand that Standby LSP must not share the same network components with the primary LSP in order to minimize risks of common failure.

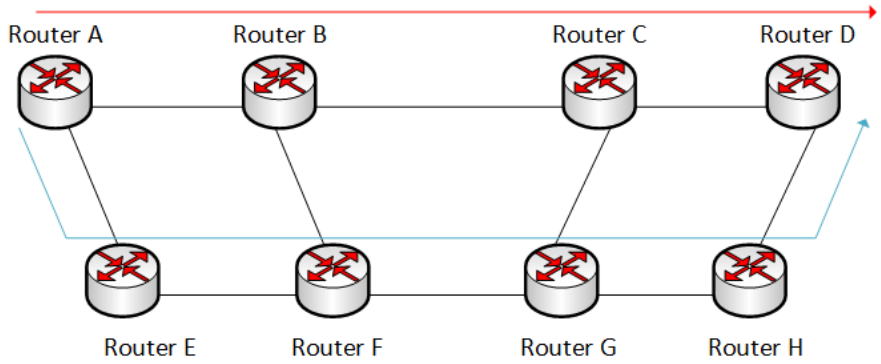

Figure 23 – Path protection

Figure 23 shows an example of Path protection technology. Here red line indicates Primary LSP (A-B-C-D) and blue one (A-E-F-G-H-D) indicates backup LSP. The relationship between LSP`s in Path protection is one-to-one.

The common name for Node and Link protection is a Local Protection. In local protection. If we compare it to Path protection, only some elements of the MPLS-TE tunnel have duplicate components, nodes or links. Here the relationship between the primary path and standby options is one-to-many, which means that more than one active Label Switch Path can be protected with only one standby component. Now we can make conclude that Local protection has better scalability than Path protection technology.

Link protection is a duplication of some LSP`s links. One of the most important concepts in terms of link protection is a “label stacking”. Label stacking is an addition of an extra label stack on top of the existing one. It is performed on the backup segment`s entry point. In order to make it clearer, I would like to provide one more example.

Figure 24 shows us an example of Link protection with a red line indicating Primary LSP (A-B-C-D) and blue one (A-B-F-G-C-D) indicating LSP, which backs up the link between Router B and Router C. The Algorithm is provided below:

- Router D starts the process of label assignment. It allocates Implicit NULL label (POP label) for an LSP;
- Router C, in its turn, allocates label 2 for this session. Allocation process ends is performed on every router in primary and backup LSPs;
- If link B-C fails, Router B puts label 2 and an additional one, label 3, in the packet. Label stack now is {3; 2};
- Router F receives packet with a label stack {3; 2} and changes it to {4; 2};
- Router G receives packet with a label stack {4; 2}. It has an instruction to pop one label before transmitting to Router C. Therefore, the outgoing packet has a label stack {2} and is treated by Router C in a way like it had been transmitted through a primary LSP.

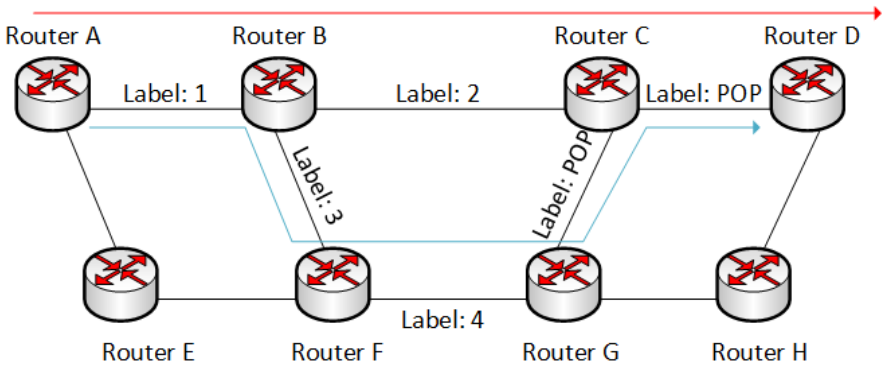

Figure 24 – Link Protection

Many companies and service providers logically separate the channels in order of importance. In order to avoid redundancy of the configuration, it is recommended to backup only links, which carry critical flows of traffic. In my example link B-C is supposed to be it.

However, not only links can go down. There are possible situations, when a network device such as switch or router fails. If, for example, the router fails, link protection can become useless. If we have a look at the last example again and imagine that something happened with Router G, then it will be noticed, that LSP A-B-F-G-C-D is not available any more. Here comes technology called Node Protection. Node protection is the presence of a backup device. In our example Router H can perform this function.

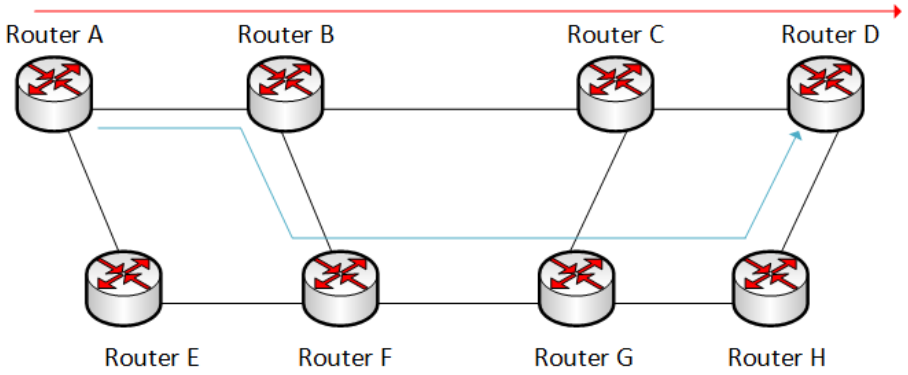

Figure 25 – Node Protection

1.5 Summary

In conclusion of the first part of the capstone project, I would like to summarize the analysis of MPLS-TE technology.

First, I want to say that MPLS-TE is a complex technology, the operation of which depends on many protocols and techniques, such as:

- Link State Routing protocol;
- RSVP;
- MPLS.

If at least one of the above components does not work correctly, then the MPLS-TE technology will also fail. I think it to be a drawback that affects the reliability of the entire network infrastructure. This minus translates into other possible problems. One of them is a long topology deployment process and an expensive maintenance process.

As mentioned earlier in this chapter, MPLS-TE is mainly applied on the provider side, i.e. at the core level of the network. Here come following drawbacks:

- Provider's network can be treated as a single point of failure;
- MPLS-TE security relies mostly on the provider`s core network infrastructure protection and do not have means for increased protection. It will be dangerous if someone hacks the provider's protection and gets access to infrastructure;
- In most cases, customer does not have an access to the whole topology and is tied to provider. This is known as a "lack of control" problem;

However, in my opinion, the biggest drawback of MPLS-TE technology is an overhead. Each component of this technology (RSVP, OSFP, IS-IS, etc.) has its own control plane and takes channel resources with control plane`s messages. Overhead is also present in data traffic, caused

by label stacks. Each of the label stacks can contain up to 5 four-byte labels, which results in significant additional traffic.

Of course, MPLS-TE has many advantages. If we speak about it, several points can be highlighted:

- Fast convergence – It is achieved by using such mechanisms as fast reroute and local or path protection in MPLS-TE and soft-state in RSVP-TE;
- QoS support, provided by the DiffServ-aware Traffic Engineering;
- Provisioning customer with a guaranteed channel bandwidth and other resources;
- Scalability – Hundreds of MPLS-TE processes can run simultaneously on the same device;
- Flexibility – a lots of path recalculation mechanisms are supported by MPLS-TE.

2 Segment routing

2.1 Introduction

As far as it has been noticed in the previous chapters, Multiprotocol Label Switching Traffic Engineering technology (MPLS-TE) is not perfect and has several key disadvantages, such as big overhead, complicated configuration and poor scalability. In order to improve these limitations and problems, Cisco Systems developed and introduced a new technology, called Segment Routing in 2017. Since that moment, many other vendors have started to release corresponding updates to its operating systems with a Segment Routing support feature.

As in the case with MPLS-TE, the main task of the Segment Routing is to reserve resources for different types of traffic along its route. This technology is intended for use in networks of service providers or networks of large corporations. Bell was the first Canadian telecommunications company, which implemented Segment Routing in its core network already in 2017.

Speaking about use cases of Segment Routing, it is similar to the cases described in chapter 1.1 for MPLS-TE. However, one more important application comes up with the usage of Segment Routing. It is Software Defined Networks (SDN). SDN allows companies to enable efficient and scalable network architecture with high performance.

The second part of the capstone projects is devoted to the analysis of the principles of Segment Routing technology.

2.2 Technical concepts

Segment Routing is a source routing technology, where the source device chooses a path and encodes its characteristics in packets. The path is represented as a stack whose elements are segments. In other words, segments are instructions for nodes for further actions. These instructions can have either local meaning for a particular device or global meaning for the whole topology. Segment with a local meaning, for example, may specify an outgoing interface for a flow. The segment with a global meaning, in its turn, may specify the whole path through the domain. Each segment can be identified by a special value called Segment Identifier (SID).

Practically every networking technology or protocol has data and control planes. Segment Routing supports three types of the control plane, specified in segments:

- Distributed – here segments are signaled by a routing protocol, either IGP Link State (OSPF, IS-IS) or BGP;
- Centralized – here, everything is signaled by a special device, called a controller. It gets all the information about the topology via protocol BGP-Link State (BGP-LS), chooses a path with a Path Computation Element Communication Protocol (PCEP) and advertises it to application servers;
- Hybrid – this type is a combination of distributed and centralized data planes. It can be used when, for example, source and destination nodes are located in different IGP domains.

Speaking about data plane, we have to mention that two types are supported in Segment Routing:

- Segment Routing over MPLS – in this architecture stack of segments is encoded as a stack of labels in MPLS. Top stack segment is a segment to process first.
- Segment Routing over IPv6 – here a new header called Segment Routing header is introduced. Shortly, segment is encoded as an IPv6 address.

2.2.1 Segment Routing with LSR IGP

In a Segment Routing domain IGP device, which supports SR, advertises segments for both connected prefixes and neighbors. These segments are called IGP SID. It is used to compose and characterize the path. However, standard versions of IGP LSRs do not support Segment Routing, that is why following extensions were created:

- IS-IS-SR-Ext (IETF RFC 8667);
- OSPF-SR-Ext (IETF RFC 8665);
- OSPFv3-SR-Ext (IETF RFC 8666).

IGP prefix SID is attached to an IGP prefix and has a global meaning within the SR domain. IGP prefix SID includes information about three components:

- Prefix – this is an identifier of a subnet;
- Topology – Link State Routing protocols, used in Segment Routing, support Multi-topology routing and Multi-instance routing. In the first case, there can be several logically different topologies (subsets of subnets) with specific paths. If Multi-instance topology is implemented, then one network device can have several Link

State Routing protocol`s processes. So, the Topology parameter specifies an affiliation of the SID to a particular instance or topology.

- Algorithm – this parameter shows which path calculation algorithm is used in the domain. There can be two options:
    - SPF – packet follows the way, defined by SPF. However, local policies on any intermediate device can put some changes in the path;
    - Strict SPF – packet follows only the way, defined by SPF, no changes can be implemented.

Several IGP prefix SID can be attached to the IGP prefix, but these SIDs must have a different set of parameters.

Most often in real-life topologies, s specific type of Prefix SID is used. It is called node SID and it identifies the specific node. Node SID is typically configured under the loopback interface with a loopback`s address as a prefix

If we speak about Segment Routing over MPLS, it is highly recommended to configure the same labels (SIDs) on all Segment Routing Devices in the domain. This approach simplifies troubleshooting and makes topology more understandable because the same prefixes get the same SIDs on all the nodes.

The following are some characteristics of the operation of Segment Routing over MPLS:

- SID (label) should include instructions for the neighbor on the next action. Possible variants are Next and Continue (Penultimate Hop Popping and Ultimate Hop Popping in MPLS correspondingly);
- As far as IGP prefix SIDs have a global meaning, they are not changed;
- Same IGP prefix SIDs must not be assigned to different prefixes;
- If node recognizes that an IGP prefix SID has a value, which is outside the SR Global Block (set of global segments in Segment Routing domain), it should send an error and must not use this SID.

Figure 26 shows an example of Segment Routing over MPLS topology. In this case, the following actions are performed in order to deliver a packet from router S to router R:

- Ingress router adds a stack of segments (SIDs) to an IP packet.
- Each node in the path deletes its entry in the stack and forwards to the next one, till the destination is reached.

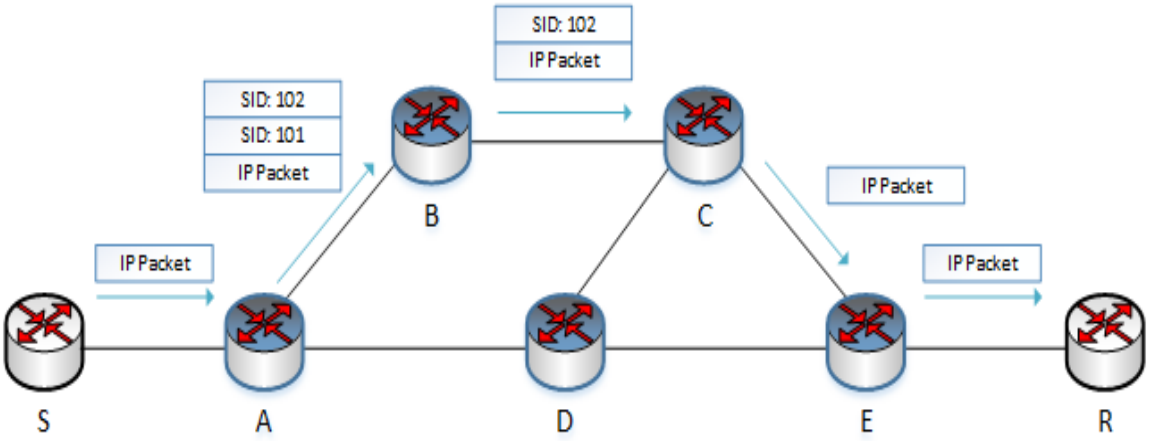

Figure 26 – Example of SR over IPv6 topology

Next Header	Hdr Ext Len	Routing Type	Segments Left
Last Entry	Flags		Tag
Segment List[0] (128-bit IPv6 address)			
. . .			
Segment List[*n*] (128-bit IPv6 address)			
Optional Type Length Value objects (variable)			

Figure 27 (source: https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-6/segment-routing/configuration/guide/b-segment-routing-cg-asr9000-66x/b-segment-routing-cg-asr9000-66x_chapter_011.pdf) – IPv6 Segment Routing header

Segment Routing over IPv6 is used in networks with IPv6 addressing. Here SID is represented as an IPv6 address. Such IPv6 address must be configured additionally, because standard IPv6 addresses are not SIDs by default. The headend router encodes the whole path in an ordered list of IPv6 addresses in an IPv6 packet. For this purpose a new header, called Segment Routing Header (SRH), is used. Its format is shown in figure 27.

Table 15 – IPv6 header`s fields

Field	Description
Next header	Type of header, which follows right after SRH;
Hdr Ext Len	The length of SRH;
Segments left	Points on the current active segment in SRH;

Field	Description
Routing Type	Value = 4, assigned by IANA;
Last entry	Index of the last segment in SRH;
Flags	8 possible optional values;
Tag	Defines routing class membership (packets sharing the same set of properties)
Segment List	Set of *n* 128-bit IPv6 addresses, which represent segments in the path. These IPv6 are stacked in the reverse order of the path, so in the bottom the IPv6 of the last segment is located

Every IPv6, which represents SID, contains three parameters:

- Locator – represents a node`s address or node`s ID;
- Function – represents an instruction, which is local for each device and specifies actions to be performed on this particular node;
- Args – represents optional function's arguments.

As far as SRH is transmitted in usual IPv6 packets, the source and destination addresses should be assigned to it. Segment Routing over IPv6 specifies that current segment identifier, which is highlighted by the "Segments Left" pointer, should be moved into the destination address field of the packet.

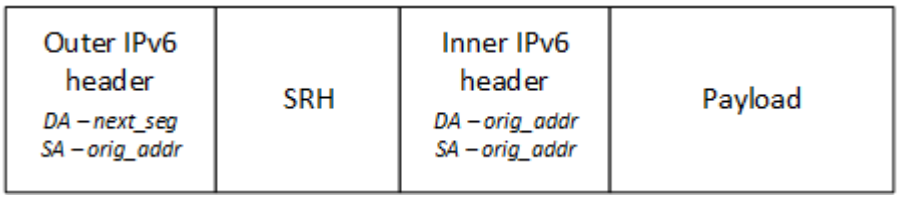

Figure 28 – SR over IPv6 Packet format

Figure 29 shows an example of Segment Routing over IPv6. In this case, the following actions are performed in order to deliver packet from router S to router R:

- Ingress router A adds an SRH header, which contains the Segment list (C, F, H, and R) and pointer to the next segment (marked in red). In addition, this node changes Destination Address in Outer IPv6 header to the address of the first segment in a segment list;

- Packet is sent to the first segment in a list. In this example, only routers marked in blue support Segment Routing over IPv6, so it is important to say, that such packet can pass devices without this support. This can be explained by the fact that only router, which address is put in the DA field, should analyze SRH field;
- When Router C receives packet, it should change the DA field and put there the highlighted segment from the segment list. In addition, it should increment pointer and move it to the next segment;
- When packet achieves router H, router H should delete SRH header and proceed packet to the destination, router R.

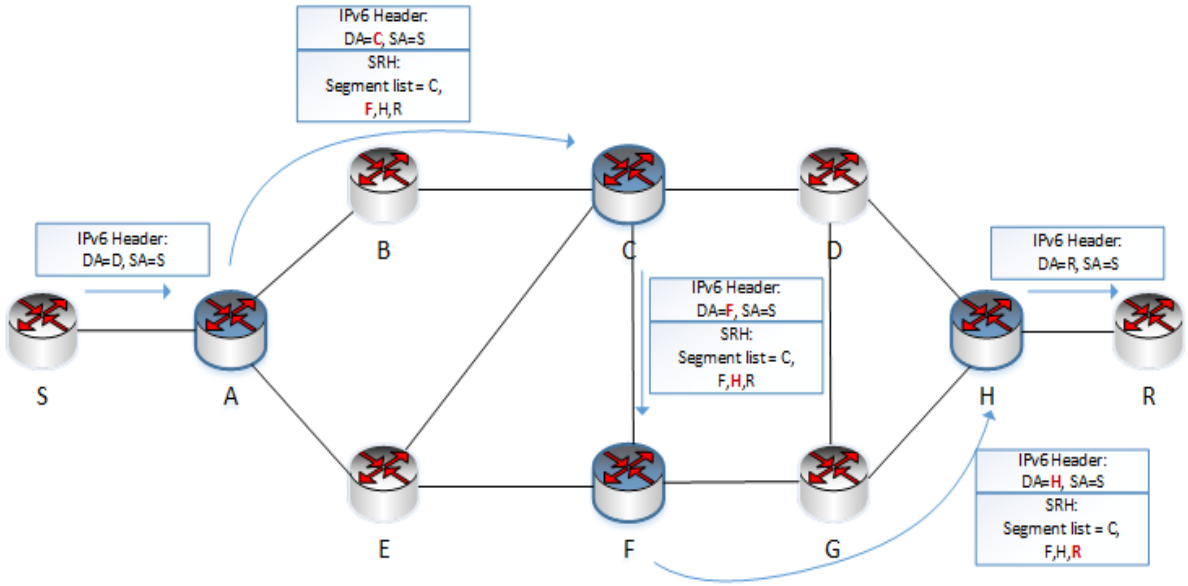

Figure 29 – Example of SR over IPv6 topology

Till this moment, only Prefix-SID`s have been discussed in this chapter. However, there is one more very important type of SID`s that is called IGP Adjacency Segment Identifier (Adj-SID). Adjacency-SID represents a link to a neighbor, has a local value and enforces switching of the packet through a particular outgoing interface. It is allocated from the Segment Routing Local Block. SRLB is a range of labels, which are only valid on the nodes that perform its allocation.

Let`s say there is a package on the network that has a header with a segment list {N, L}, where N is a Node-SID of the node N and L is an Adj-SID of the link, connected to device N. In this case packet will be forwarded through the shortest path, defined by SPF, to node N. Than it will be sent through the link L without any considerations with predefined path. If an Adj-SID a set of adjacencies, then node N performs load balancing.

In order to reduce the overhead and make a segment list smaller, globally defined Adj-SID can be used. In this case, segment list is represented in the following way {L}, where L is a link from

node N to its adjacency. Packet will be forwarded to the node N through the shortest path available and afterwards sent via link L.

When IGP Adj-SID is in use, the following flags can be set in the packet`s Segment Routing header:

- Eligible for protection flag – this protection allows retransmission of the packet through the link, which is different from the one, which is set in Adj-SID;
- Global or local scope – specifies the scope of Adj-SID. The default value is local;
- Indication of saving Adj-SID when restarting the control plane. Saving is a key attribute that prevents SR Policy from temporarily sending incorrectly because of the re-provisioning of the Adj-SID.

The typical use-case of IGP Adjacency-SID is a Segment Routing Traffic Engineering, because it allows determining explicitly, the whole path throughout the SR domain. Adj-SIDs are distributed by the Segment Routing Control data plane.

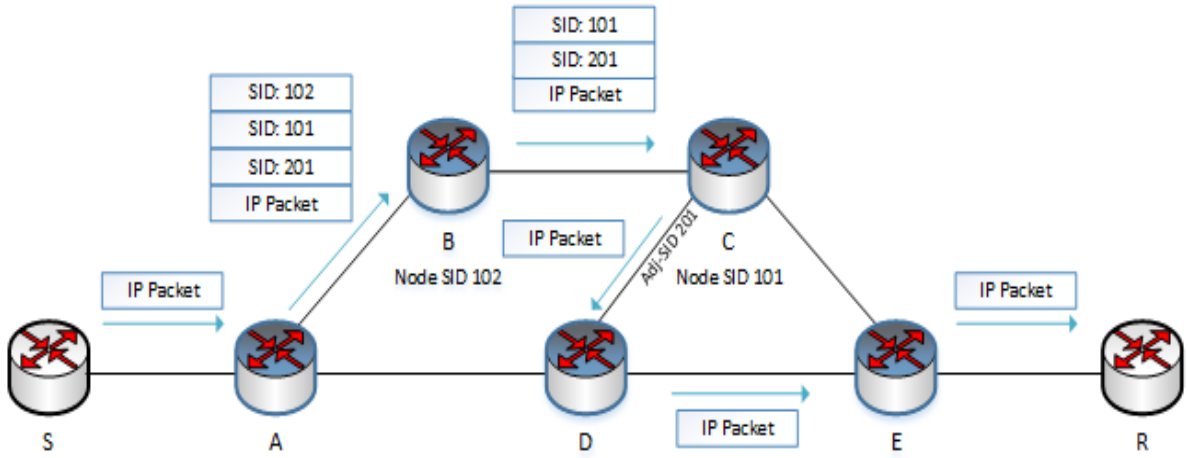

Figure 30 – Example of SR over IPv6 topology

Now I would like to have a look at the example in figure 30. There Segment Routing algorithm established the best path through routers A, B, C and E. This choice was based on IGP Prefix SIDs only. Figure 27 illustrates the same topology, but here comes a customer`s requirement to establish path through nodes A, B, C, D and E. In order to meet this requirement, one more SID should be used. This SID is an Adj-SID, which represents a link between routers C and D. As a result, needed path is established.

Let`s say there is a package on the network that has a header with a segment list {N, L}, where N is a Node-SID of the node N and L is an Adj-SID of the link, connected to device N.

Some additional characteristics of Adj-SID are provided below:

- A node with an enabled Segment Routing technology should allocate Adj-SID for all adjacencies;
- A node may allocate more than one Adj-SID for each adjacency. It can be useful, for example, if one Adj-SID allows protection while another one does not;
- Same Adj-SID may be assigned to different adjacencies;
- The Adj-SID means that router, which has allocated it, should override the routing decision made by SPF.

Adj-SID can represent parallel segments or links between two adjacent routers. Usually only one common Adj-SID is used for this purpose. Router, in its turn, creates a special parameter, called "weights" and associates this parameter with Adj-SID for each segment. These weights specify the load-balancing coefficient between links.

On figure 31 router A allocates different weights for Link 1 and Link 2:

- Link 1: Adj-SID 1000; weight 2
- Link 2: Adj-SID 1000; weight 1

Therefore, when the source device gets these announcements, it balances traffic in the ration 2:1.

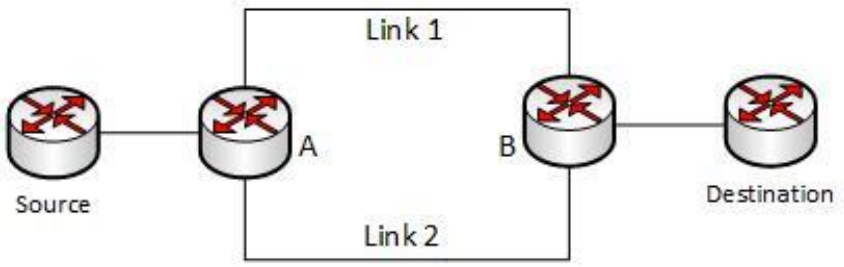

Figure 31 – Parallel agencies

2.2.1.1 Segment Routing in Inter-Area topologies

Links State routing protocols use the mechanism if dividing the topology into zones. I would like to analyze concepts of Segment Routing in such cases on a particular example.

In this topology, Segment Routing over MPLS is used. In order to advertise prefix 2.2.2.2/32 to router S following actions are performed:

- Node R allocates Node-SID 100 to its local prefix 2.2.2.2/32;

- Area Border routers C and E analyze if router R has advertised a SID. If R has advertised it, then routers C and E will send this SID in a backbone area in a Link State Announcement (LSA);
- ABRs B and D will perform the same actions as routers C and E and will send Prefix-SID 100 in area 1;
- As a result, when router S sends packets to router R, it pushes SID 100 and forward packets to A;
- When ABRs get these packets, they send it accordingly to their routing tables via the shortest available path to router R with SID 100;
- The process ends when the packet reaches node R.

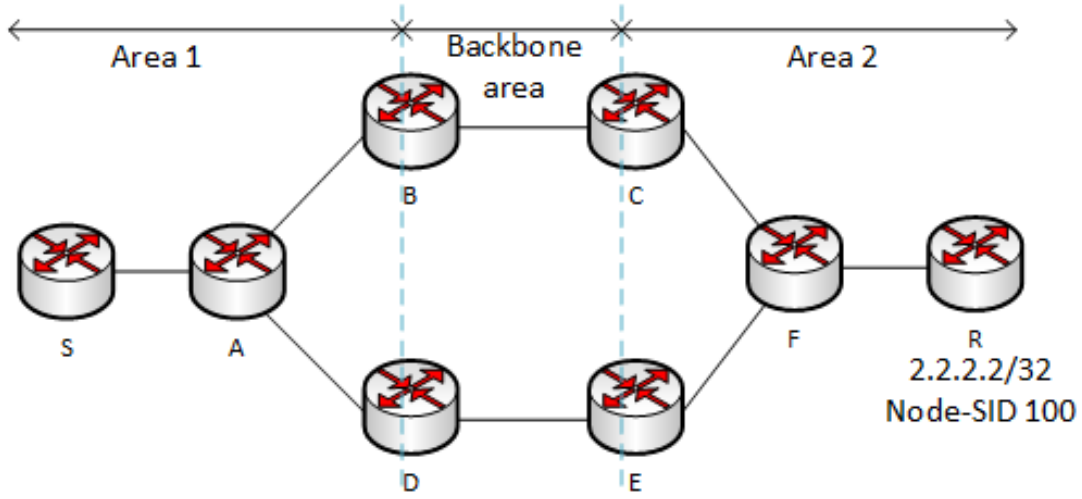

Figure 32 – Inter-Area Segment Routing

2.2.1.2 Binding segments

Binding Segment, also known as BSID, is a special segment identifier, which was introduced in Segment Routing for the purposes of Traffic-Engineering. When Segment Routing-Traffic Engineering policy is created, it is automatically identified by BSID. Binding Segment Identifier can have either a local or a global value inside a Segment Routing domain. However, it can seamlessly steer traffic between separate domains. In this case, each domain defines its own SR-TE policies, which are independent of the policies in other domains. These policies can be changed (rerouting procedure, for example) at any moment and implemented changes will affect only local domain`s topology and architecture. BSID is usually a reference to the SR-TE list of SID`s.

The basic principle of BSID`s operation is that when router receives a packet with a BSID as a top label, this packet is steered to the referenced Segment Routing-Traffic Engineering policy. Than this BSID is popped and SR-TE policies' list of SID is pushed.

There are three main use cases of Binding Segment Identifier:

- Multi-Domain topologies;
- Large single domains;
- Need in compression of the label stack.

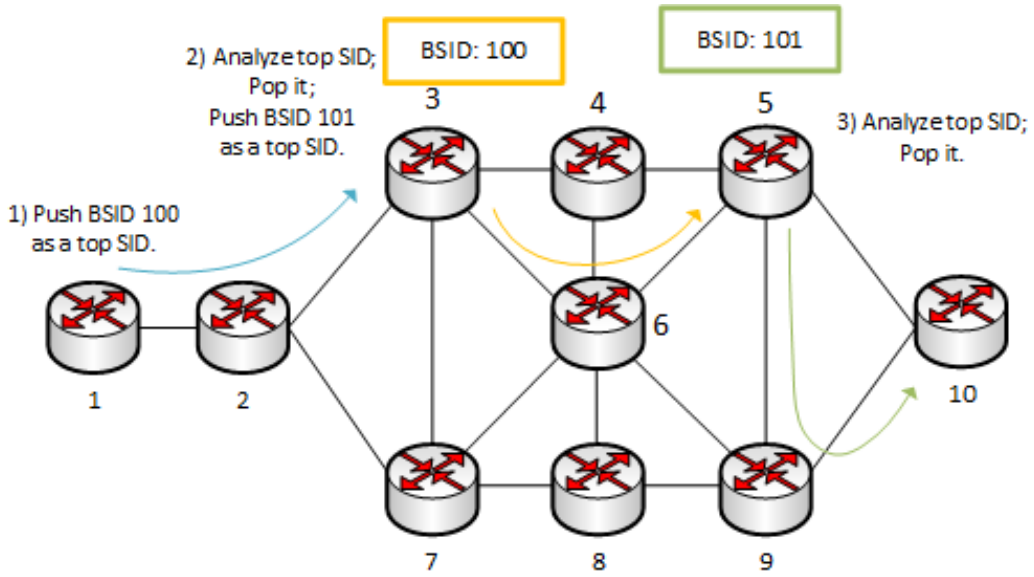

Figure 33 – BSID example

Figure 33 illustrates an example of the usage of BSID. There the following actions are performed:

- SR-TE policy representing a path from node 5 to node 10 is created (green line) with a BSID 101. When a packet arrives on it, the packet will be forwarded accordingly to SR-TE policy via routers 9 and 10;
- SR-TE policy representing a path from router 3 to router 5 (orange line) is created on router 3. This policy is identified by BSID 100. When a new packet arrives on router 3, a new one BSID (BSID 101) should be pushed on top of a label stack. This is the instruction for router 5 on the following actions.
- SR-TE policy representing a path from node 1 to node 3 is created on router 1. When a new packet arrives on router 1, a new one BSID (BSID 100) should be pushed on top of a label stack. This is the instruction for router 3 on following actions.

2.2.1.3 OSPF extensions for Segment Routing

As it has already been said previously, IGP Link State protocols are responsible for advertising Segment Identifiers in the domains. OSPF routing protocol uses a new class of link state advertisements called Opaque LSA`s for this purpose. Its format is provided on figure 34.

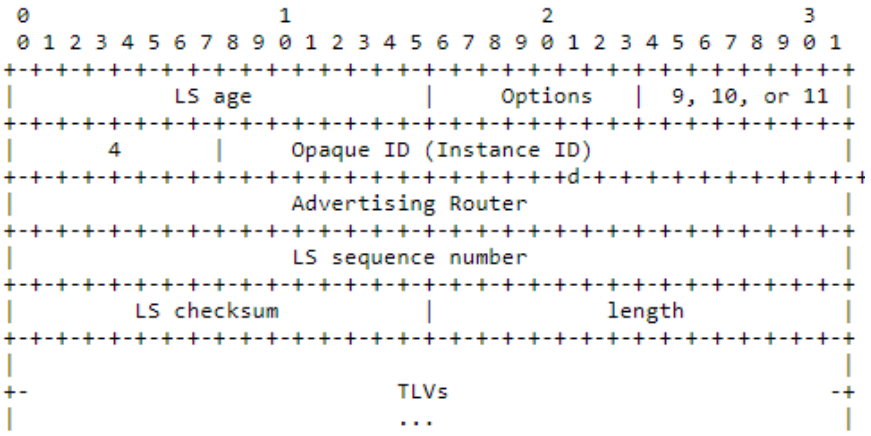

Figure 34 (source: https://tools.ietf.org/html/rfc7770#ref-OPAQUE) – OSPF Opaque LSA

Opaque LSA carries Segment Identifiers in a TLV field. There can be multiple TLVs in one Link State Announcement. Format of the SID/Label TLV is shown on figure 35.

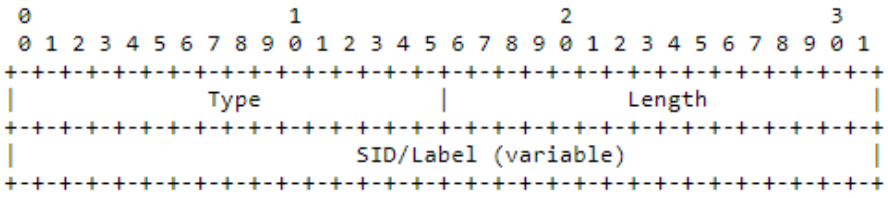

Figure 35 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – SID/Label TLV format

Table 16 – SID/Label TLV`s fields

Field	Description
Type	Always is set to 1;
Length	Length of the SID/Label field. Can be set to 3 (20 bits SID) or 4 (32 bits SID). If node receives packet with any other value in this field, it should ignore this TLV;
SID/Label field	Value of the SID.

Besides SIDs, TLVs can contain an information about additional parameters of the Segment Routing. The first one is a SR-Algorithm TLV. SR-Algorithm TLV is an optional value. This parameter specifies path calculation algorithm, which is used in the topology. There are two possible values of the algorithm:

- Shortest Path First
- Strict Shortest Path First

The key difference of these algorithms was discussed in detail in chapter 2.2.3. If one node receives SR-Algorithm TLV from the same router, it must use the first one SR-Algorithm TLV in Opaque LSA.

The second parameter is a SID/Label Range TLV. It specifies the allowed space of SID/Labels. It is important to mention, that one SID/Label sub-TLV must be included in the announcement of SID/Label Range TLV. This allows uniquely and completely specify the available SID/Labels space. The example of how it works is provided further. Let`s assume that originating node advertises the following SID/Label Range TLV:

Range 1: Size 200; SID/Label Sub-TLV 200,

Where SID/Label Sub-TLV 200 is the first label in specified range, then the range R {200; 399} will be added to a Segment Routing Global Block (SRGB).

Format of the SID/Label Range TLV is provided below:

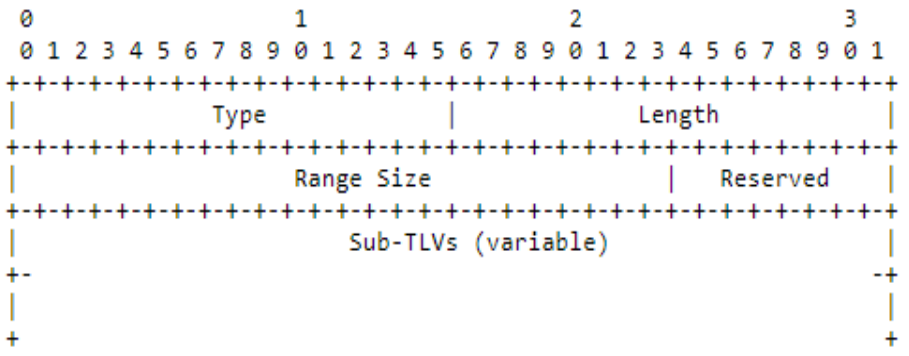

Figure 36 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – SID/Label Range TLV

Table 17 – SID/Label Range TLV`s fields

Field	Description
Type	Always is set to 9;

Field	Description
Length	Specifies the length of SID/Label Range TLV, depends on the length of SID/Label Sub-TLV;
Range Size	Number of SIDs in the range, must be greater than 0.
Reserved	Always is set to 0. If there is another value, TLV must be ignored;
Sub-TLV	Sub-TLV, which indicates the first SID in the range.

One router can advertise multiple SID/Label Range TLVs in order to specify different ranges. When the receiver gets an announced range, it must allocate SIDs only from the specified scope.

Similar to the scope of labels from SRGB, range of SRLB (Segment Routing Local Block) can also be advertised through the whole SR domain. In this case SR Local Block TLV is used, and it contains only labels from a Local scope. For example, Adjacency SIDs can be advertised in such way.

SR Local Block TLV can be very useful if SDN controller wants to instruct routers to allocate specific Local SIDs for specific purposes. Format of the SR Local Block TLV is provided on figure 37:

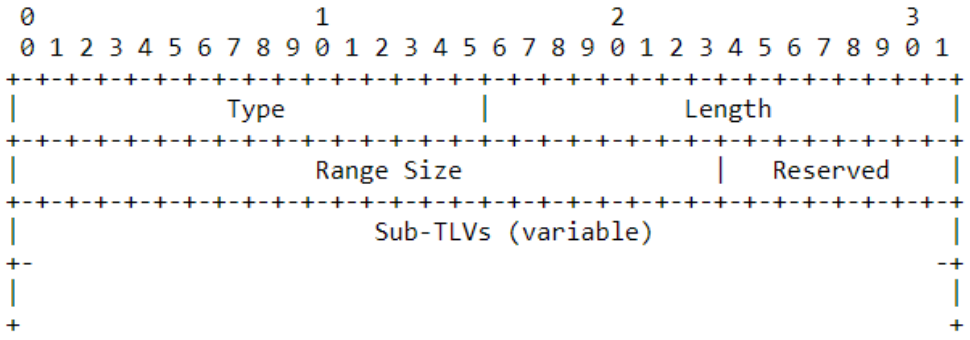

Figure 37 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – SR Local Block TLV

Table 18 – SR Local Block TLV`s fields

Field	Description
Type	Always is set to 14;

Field	Description
Length	Specifies the length of SR Local Block TLV, depends on the length of SID/Label Sub-TLV;
Range Size	Number of SIDs in the range, must be greater than 0.
Reserved	Always is set to 0. If there is another value, TLV must be ignored;
Sub-TLV	Sub-TLV, which indicates the first SID in the range. Only one Sub-TLV can be advertised in SR Local Block TLV.

It is important to say, that originating router may obtain multiple SRLB scopes of labels. However ranges of these scopes must not overlap. Every time when a label from advertised SR Local Block scope is allocated, this allocation should be advertised to all other components in Segment routing domain. This requirement allows to avoid collisions in the process of label allocation. In addition, it helps routers to obtain the most up-to-date information about the topology.

The next one option is Segment Routing Mapping Server (SRMS) Preference TLV. SRMS function allows allocating Prefix-SIDs to prefixes, which are connected either to SR-capable devices, or to SR-no-capable nodes. The most important and popular use case of this technology is SR-LDP network topologies. For example, if Segment Routing device has only LDP neighbors, then this node must be able to stitch LDP announcements in SR packets and vice versa. There are two main components in SRMS:

- Mapping Server;
- Mapping Client.

Mapping server is a network device, which advertises Segment Routing mappings. These advertisements define an allocation of the Prefix-SID to a specific prefix even if there is no information about its reachability (prefix is connected to LDP capable router, for example). So this prefix can be unreachable through any router in Segment Routing domain, but it will also have a Prefix-SID.

Mapping Clients receive and use Prefix-SIDs, generated by Mapping Server. If there is a situation, when Mapping Client receives both SR mapping instruction and usual Prefix-SID for

the same prefix at the same time, it must use usual Prefix-SID. If there is another situation, when Mapping Client receives SR mappings from multiple Mapping Servers, then some priority should be defined. This priority is advertised in SRMS Preference TLV.

One node can be Mapping Server and Mapping Client at the same time.

Format of the SRMS Preference TLV is provided on figure 38:

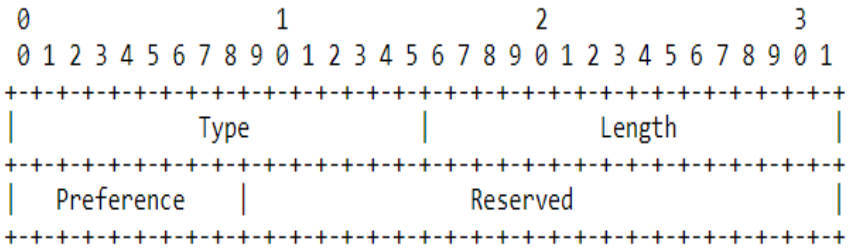

Figure 38 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – SRMS Preference TLV

Table 19 – SRMS Preference TLV`s fields

Field	Description
Type	Always is set to 15;
Length	Always is set to 32;
Preference	SR mapping preference. Can take value from 0 to 255;
Reserved	Always is set to 0. If there is another value, TLV must be ignored.

The last available option is Extended Prefix Range TLV. This feature is specific for an OSPF routing protocol. It allows advertising the same Segment Routing attributes for a scope of prefixes. Extended Prefix Range TLV can be used, for example, when the range of SIDs needs to be announced for a range of contagious prefixes.

Format of the OSPF Extended Prefix Range TLV is provided in figure 39:

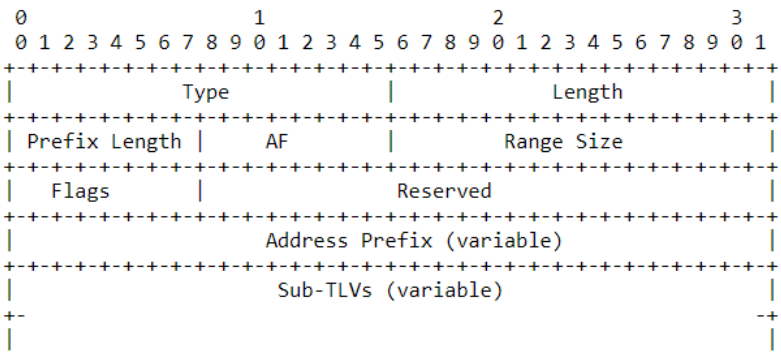

Figure 39 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – OSPF Extended Prefix Range TLV

Table 20 – OSPF Extended Prefix Range TLV `s fields

Field	Description
Type	Always is set to 2;
Length	The length of the TLV, which can be variable;
Prefix Length	Specifies the length of the address prefix;
Address Family	Always is set to 0. 0 means IPv4 address family;
Range Size	This parameter specifies number of network prefixes, which are advertised in this TLV;
Flags	This field occupies 8 bits. However, only the first one bit is used. It determines, whether this TLV is an interarea announcement or not. This bit can be only set by ABR devices;
Reserved	This field must be ignored by network devices;
Address Prefix	Here an address prefix is stored. The address prefix specified in this field must be the first address prefix in advertised range;
Sub-TLVs	Scope of sub-TLVs, which are allocated for each prefix advertised in a range. Format of such sub-TLVs is described further.

Now I would like to provide two examples of OSPF Extended Prefix Range TLV `s field's usage. If originator wants to advertise prefixes:

- 172.16.0.1/32;
- 172.16.0.2/32;
- 172.16.0.3/32;
- 172.16.0.3/34

, then Address Prefix field will contain "172.16.0.1", Prefix Length field will be set to "32" and Range Size field will contain "4".

If originator wants to advertise prefixes:

- 172.16.0.0/30;
- 172.16.0.4/30;
- 172.16.0.8/30;
- 172.16.0.16/30;
- 172.16.0.20/30

, then Address Prefix field will contain "172.16.0.0", Prefix Length field will be set to "30" and Range Size field will contain "5".

2.2.1.3.1 Prefix-SID Sub-TLV

Sub-TLV`s field in OSPF Extended Prefix Range TLV can contain multiple objects, depending on how many prefixes are being advertised, Such objects are called Prefix-SID Sub-TLVs and Adj-SID Sub-TLV. Format of the Prefix-SID Sub-TLV is provided on figure 40.

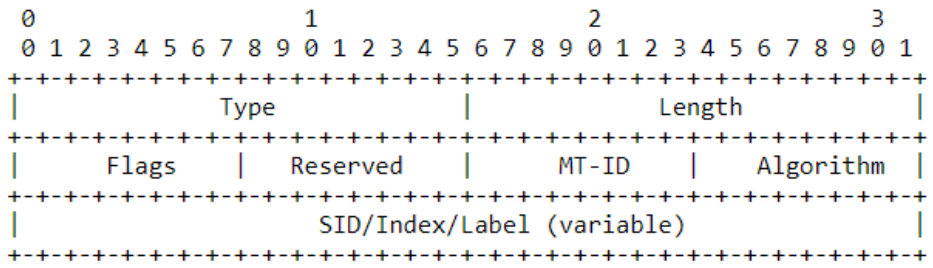

Figure 40 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – Prefix-SID Sub-TLVs

Table 21 – Prefix-SID Sub-TLV`s fields

Field	Description
Type	Always is set to 2;
Length	The length of the Prefix-SID Sub-TLV depends on flags and varies from 28 to 32 bits;
Flags	This field occupies 8 bits, however only 1,2,3,4,5th bits are in use: 1st bit – NP-flag, which instructs penultimate hop not to implement Penultimate Hop Popping procedure, if flag is set; 2nd bit – M-flag, which indicates, whether SID was announced by Segment Routing mapping server; 3rd bit – E-flag, which instructs upstream SR device to replace current Prefix-SID with the Explicit NULL label, if flag is set; 4th bit – V-flag, which indicates, whether the absolute value or index is carried in this TLV. Index is an offset in the label space, defined by the router; 5th bit – L-flag, which indicates, that SID has a local value, if flag is set; All other bits should be set to 0 and ignored;
Reserved	All 8 bits should be set to 0 and ignored;
MT-ID	Multi-Topology bit;
Algorithm	Either Strict SPF or SPF algorithm should be set there;
SID/Index/Label	The value of this field depends on V-flag and L-flag: 1) If V=0 and L=0, then this field contains a global value index; 2) If V=1 and L=1, then this field contains a local value label;

The following are the main conditions for using flags:

- If ABR device creates advertisements for intra-area or inter-area prefixes, which are not directly connected to this ABR, then NP and E flags must be set;
- IF ASBR device creates advertisements for redistributed prefixes, then NP flag must be set and E flag must not be set;
- NP, E and M flags, advertised by next-hop device, should be taken into account by any router, when it initiates a process of label calculation;
- Next hop router must pop the Prefix-SID, if NP flag was set by a downstream neighbor;
- IF NP and E flags are not set by the originator, then any upstream router must keep Prefix-SID on top of a stack;
- Of both NP and E flags are set, then any upstream router must put Explicit NULL label instead of Prefix-SID;
- If M flag is set, then router must ignore E and NP flags;

2.2.1.3.2 Adj-SID Sub-TLV

As it has already been said in previous chapters, Adj-SIDs represent links to neighbors in Segment Routing. Its usage in OSPF Extended Prefix Range TLV is optional, and multiple such Adj-SIDs can be put in one TLV. In addition, it is important to remember, that Segment Routing device can set more than one Adj-SID to one adjacency and the same Adj-SID for different adjacencies.

Adj-SID can be created and advertised for any adjacency on point-to-point links if its state is 2-Way and higher (ExStart, Exchange, Loading and Full). So, if at any moment state of the adjacency becomes Init (lower than 2-Way), then corresponding Adj-SID must be removed from the topology. The format of Adj-SID Sub-TLV is provided on figure 41:

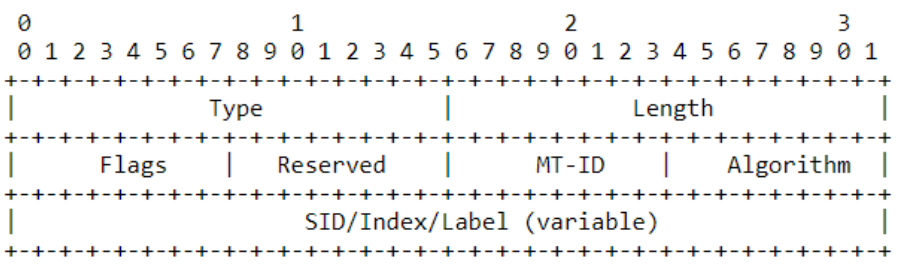

Figure 41 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – Adj-SID Sub-TLVs

Table 22 – Adj-SID Sub-TLV`s fields

Field	Description
Type	Always is set to 2;
Length	The length of the Adj-SID Sub-TLV depends on flags and varies from 28 to 32 bits;
Flags	This field occupies 8 bits, however only 0,1,2,3,4th bits are in use: 0st bit – B-flag, which indicates, that specific adjacency supports some types of protection, like Fast Reroute; 1nd bit – V-flag, which indicates, whether the absolute value or index is carried in this TLV. Index is an offset in the label space, defined by the router; 2rd bit – L-flag, which indicates, that SID has a local value, if flag is set; 3th bit – G-flag, which indicates, that Adj-SID is related to a multiple adjacencies; 4th bit – P-flag, which indicates, Adj-SID is a persistent value, which will not change after restarts of the device; All other bits should be set to 0 and ignored;
Reserved	All 8 bits should be set to 0 and ignored;
MT-ID	Multi-Topology bit;
Weight	This value is used for load balancing, if different paths are available;
SID/Index/Label	The value of this field depends on V-flag and L-flag: 1) If V=0 and L=0, then this field contains a global value index; 2) If V=1 and L=1, then this field contains a local value label; All other combinations are invalid and should be ignored.

In case of broadcast and multi-access topologies in OSPF, the Designated Router should be elected. Only this router has a Full adjacency state with any other device in the topology. All non-DR devices should have 2-Way adjacency state with each other in this model. In case of OSPF in Segment Routing, key logic remains the same, and each router advertises Adj-SIDs to the DR, using Adj-SID Sub-TLV. However, these routers can optionally advertise SIDs for adjacencies with non-DR devices. For these purposes, LAN Adj-SID Sub-TLVs are used. Figure 39 illustrates LAN Adj-SID Sub-TLV`s format:

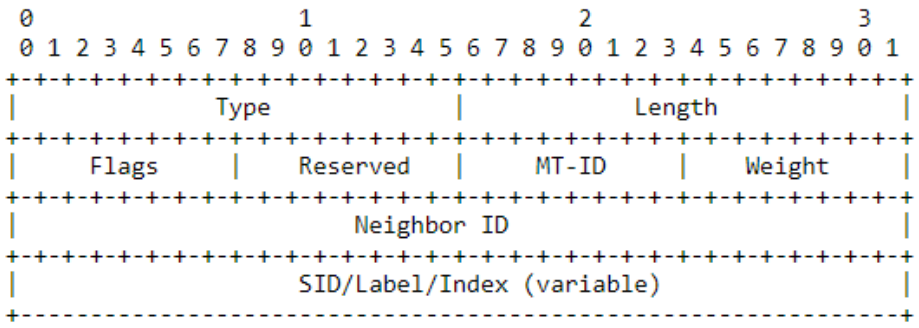

Figure 42 (source: https://tools.ietf.org/html/draft-ietf-ospf-segment-routing-extensions-27) – SRMS Preference TLV

Table 23 – SRMS Preference TLV`s fields

Field	Description
Type	Always is set to 3;
Length	Depends on V flag, can be 88 or 96 bits;
Flags	Same flags as described in table 22;
Reserved	Always is set to 0. If there is another value, TLV must be ignored.
MT-ID	Multi-Topology bit;
Weight	This value is used for load balancing;
Neighbor ID	Router ID (RID) of the neighbor;
SID/Index/Label	The value of this field depends on V-flag and L-flag: 3) If V=0 and L=0, then this field contains a global value index; 4) If V=1 and L=1, then this field contains a local value label;

2.2.1.4 OSPF Segment Routing in Intra-area topologies

There are two possible situations, which can occur in OSPF area in terms of Segment Routing technology:

- Propagation of the prefixes by Segment Routing capable routers;
- Propagation of the prefixes by devices, which do not support Segment Routing.

The first case is the easiest one. Segment Routing capable router can advertise SIDs for any prefixes, which are reachable from this device. For this purpose Prefix-SIDs are used.

The second case is a little bit more complicated. Here the Segment Routing Mapping Servers technology should be used. As it has already been said in chapter 2.2.3.3, prefixes, connected to non-capable Segment Routing devices, should be advertised in OSPF Extended Prefix Range TLV by SR Mapping Server. In addition, it is important to keep in mind, that different Mapping Servers may advertise same prefixes to a Segment Routing domain. In this case, all the Mapping Servers must be synchronized and same prefixes must be announced by all the Mapping Servers.

After all the Segment Routing devices in OSPF area have the complete topology, Area-scoped OSPF Extended Prefix Range TLVs are created on ABR routers and are advertised to other areas. ABR creates such TLVs basing on rules, described in chapter 2.2.3.3.

2.2.1.5 OSPF Segment Routing in Inter-area topologies

Very often one segment Routing domain is not limited to one OSPF zone. Vice versa, practically all OSPF topologies are divided into many zones due to certain reasons. In such situations, OSPF should be able to advertise Prefix-SIDs between areas.

Like in standard OSPFv2, OSPF in collaboration with Segment Routing can perform routing in intra-area topologies. The main difference here is that in case of Segment Routing, ABR transfers not only Type 3 Summary LSA to all connected areas, but also an OSPF Extended Prefix Opaque LSA. This LSA must contain the following parameters:

- IA flag should be set in OSPF Extended prefix TLV;
- OSPF Extended Prefix Opaque LSA`s scope has to be set to “local scope” in options field;
- Prefix-SID Sub-TLV has to be set.

The logic by which the Prefix-SID Sub-TLV is set is defined below:

- Area Border Router should check the best path to the subnet with advertised prefix in an originating area and determine, if the advertising router belongs to best path;
- If this advertising router belongs to the best path, then exactly this Prefix-SID will be used in further advertisements to other areas;
- If no Prefix-SIDs were received from the router, which belongs to the best path, then ABR will use Prefix-SID for a subnet, advertised by another router.

2.2.1.6 External routes in OSPF Segment Routing topology

The last possible option in exchanging of the routing information is a route redistribution from external autonomous systems. In this Case Autonomous System Border Router floods in OSPF domain Type 5 LSA and OSPF Extended Prefix Opaque LSA with the following characteristics:

- Route type should be set to External;
- Flooding scope should be set to Autonomous-system-wide scope;
- Prefix-SID Sub-TLV has to be set.

2.3 Segment Routing Traffic Engineering

In previous chapters about Segment Routing, mostly the common principles of the work of this technology were discussed. Now I would like to pay more attention to the Segment Routing Traffic Engineering technology.

The key idea of Segment Routing Traffic Engineering is that an upstream node can direct traffic to the desired path using specially defined policies. These policies are called Segment Routing Policies. SR Policy is associated with paths, where each path is a one segment list or a set of segment lists, which definitely initialize the whole route to the destination node inside the Segment Routing domain. Despite the fact that SR Policy can be associated with several paths, only one “candidate” path must be selected for traffic transmission. Each such path has a preference. The default value is 100.Candidate path is associated with SR Policy by Binding SID. If there is a situation, when several Segment Routing Policies decide to choose the same candidate path, then each SR Policy must be associated with its own unique Binding SID. In other words, any BSID must be associated with one path and vice versa.

Segment Routing Policy becomes active when the valid Candidate path is found. Any path becomes invalid, if it does not have a valid segment list. There are four possible conditions, when segment list can be considered to be invalid:

- Path`s segment list is empty;
- Headend device cannot resolve the first element of the segment list. For example, it has no association between first SID and any outgoing interface.
- The last element of the segment list is not a Prefix SID, which was advertised by the endpoint node;
- The last element of the segment list is not an Adj SID, which was advertised by any node, which has a link to endpoint device.

Here comes the following conclusion:

- Segment path is invalid, if all of its segment lists are invalid;
- Segment Routing Policy is invalid, if all its paths are invalid.

This path can be either explicit or dynamic. In case of explicit paths, as it has already been mentioned in this project, fixed list of segments is used. If the dynamic path is in use, headend router receives specific requirements and constraints for the traffic flow and afterwards it calculates the best path to the endpoint node. Dynamic path is considered a better choice if network is not stable, because it can dynamically adopt to the topology changes.

As SR Policies are received, network device must add it in its Forwarding Information Base (FIB). Each Segment Routing Policy can be uniquely identified by three parameters:

- Headend device, where SR Policy is implemented;
- Endpoint device. It can be specified by an IP address. 0.0.0.0 and ::00 addresses can also be used;
- The color. Color is a numeric identifier.

Here I would like to provide an example, which illustrates the SR Policy record in a network device:

Segment Routing Policy 1

Candidate Paths

Path-preference 100

BSID1

Weight 10, Segment List : 10000;10001

Weight 20, Segment List : 10002;10003

It is important to mention, that if a candidate path contains several segment lists, special parameter Weight should be used in terms of load balancing. Traffic load-balancing in this case is calculated with the formula:

$$Flow\ \%\ per\ segment\ list = \frac{w}{Sw},$$

, Where "w" is a value of the weight parameter and "Sw" is a sum of all the weight parameters in the candidate path. In the example provided above, the flow percentage of the first segment list is 33,3%.

There are four most popular ways of how headend device can obtain SR Policies:

- BGP;
- Local settings via Command Line Interface;
- PCEP;
- Netconf.

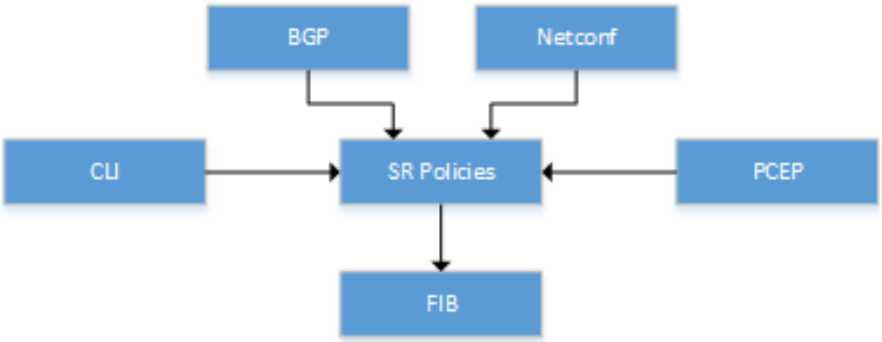

Figure 43 – SR Policies' sources

As SR Policies are received, network device must add it in its Forwarding Information Base (FIB). Each Segment Routing Policy can be uniquely identified by three parameters:

- Headend device, where SR Policy is implemented;
- Endpoint device. It can be specified by an IP address. 0.0.0.0 and ::00 addresses can also be used;
- The color. Color is a numeric identifier.

In order to compute and maintain instructions in the SR-Policy, network devices have to be able to collect updated information about the Segment Routing domain. This information is kept

in Segment Routing Traffic Engineering Database (SRTE DB). In SRTE DB, the following information can be found:

- IGP information, including IGP metrics and topology;
- Traffic-engineering information, including different TE attributes;
- Segment Routing information, including SIDs, label scopes.

All this information can be collected in three well-known ways:

- BGP-LS;
- IGP;
- Netconf.

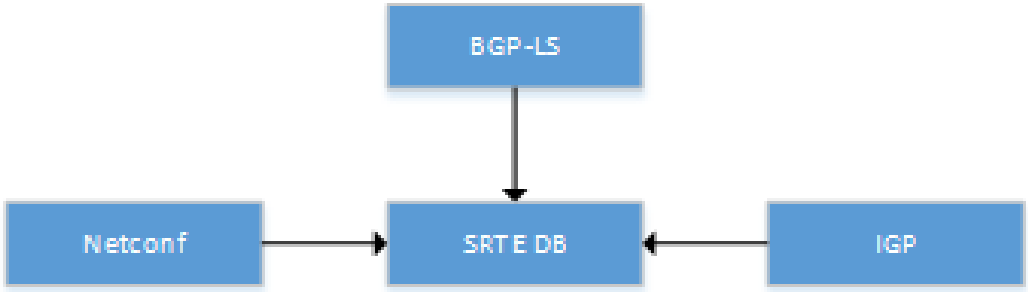

Figure 44 – SRTE Database sources

Figure 44 shows Segment Routing device`s possible opportunities for gathering topology information inside a single domain. However, Segment Routing provides us with an ability to perform inter-domain routing and SRTE DB is a multi-domain capable database. In this case, SR routers can learn information about remote domain only via BGP-LS.

2.3.1 Segment Routing Traffic Engineering Tunnels

In order to provide a required service for the traffic flows, Segment Routing Traffic Engineering technology uses a tunneling concept. The key prerequisite for the SR-TE tunnel is an IGP established relationship between involved devices.

There are three opportunities for a networking engineer to establish SR-TE tunnel:

- Manual configuration on the headend device. This option is good only for small networks. Also manual configuration does not support bandwidth reservation;
- CSPF path computation on the headend device. This method supports bandwidth reservation, however it works only in a single domain topology;

- Usage of special controller, which collects information about the topology, about application requirements and then establishes tunnels.

The third method is the most complicated one and now I would like to discuss it on example.

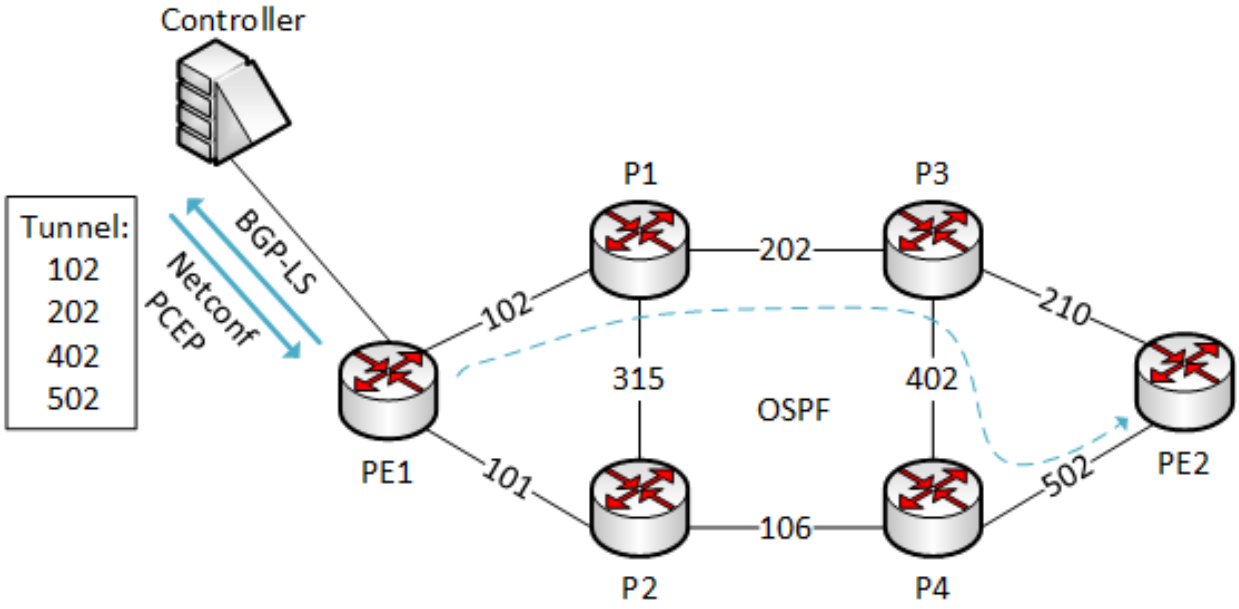

Figure 45 – SR-TE with controller example

Figure 45 illustrates an example of the SR-TE topology with controller. Here the following actions are performed:

- OSPF collects information about the topology and establishes connections;
- All the information, collected by OSPF is reported to the controller by BGP-LS protocol;
- PCEP (Path Computation Element Protocol) calculates the best path;
- Tunnel attributes are delivered to PE1 by Netconf;
- Label allocation information is delivered to PE1 by PCEP;
- Tunnel is created from PE1 to PE2.

2.3.1.1 Reliability

Segment Routing Traffic Engineering introduces new loop-free technology for its tunnels. It is called Topology-Independent-Loop-Free Alternate (TI-LFA). Unlike any other loop protection scheme, TI-LFA uses explicit paths as backups.

Now I would like to discuss TI-LFA`s principles of work on example, provided in figure 46.

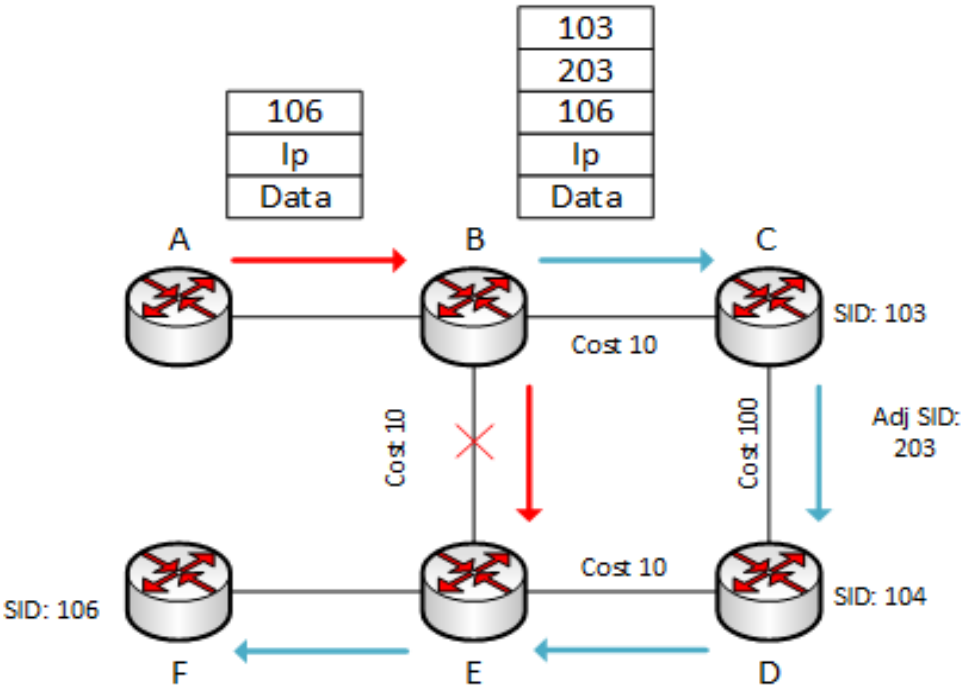

Figure 46 – TI-LFA

In this topology, the best path between routers A and F is A-B-E-F. At a certain point in time link between B and E fails due to some reasons. In order to restore the connection between routers A and F, the following actions are performed:

- Router B immediately launches its TI-LFA algorithm;
- TI-LFA algorithm enables explicit path from router B to router F;
- Router B pushes additional routing information on a label stack, accordingly to the explicit path`s instructions (103;203);
- Packet is forwarded to its destination point through the explicit part.

Another feature that allows to enhance tunnel`s reliability is SR-TE Hot Standby. Hot standby is a reserve path, which does not have any common nodes with the primary one. When something happens with an active path, Hot Standby immediately takes its role. After failed primary path recovers, the headend device switches all the traffic back to it.

The last and most important feature in this chapter is Traffic Engineering Fast Reroute (TE FRR). It functions in the similar way as SR-TE Hot Standby does. However, there is one significant difference. FRR is a local protection technique and can be executed on any node in the topology. FRR process requires time for IGP convergence and CSPF recalculations, that is why in case FRR is executed, involved services will be unreachable for some time. The main purpose of FRR is a minimization of traffic loss.

2.4 Summary

In conclusion of the second part of the capstone project, I would like to summarize the analysis of Segment Routing and Segment Routing Traffic Engineering technologies. In addition, I would like to provide a comparison between MPLS-TE and SR-TE operational mechanisms.

First, I would like to say, that Segment Routing uses only IGP protocol as a control plane and does not require Label Distribution Protocol and RSVP-TE technology in its performance. This approach significantly decreases the load on the router`s processors, in case of big topologies, and number of messages, transmitted via the links.

The second important parameter to be compared is a load of the control plane. As it has already been said, all path calculations and label manipulation processes are performed only on an ingress node in case of Segment Routing. That is the main reason why intermediate and egress nodes are not required to perform any actions with paths or labels. This approach significantly decreases the overhead in the topology. The number of connection states, maintained by the ingress device can be calculated by the following formula:

$$States = Number\ of\ nodes + Number\ of\ connections$$

Intermediate devices in MPLS-TE topology, in contrast, are required to maintain tunnels and this adds a lot of additional load to the links. The number of connection states, maintained in overall by all device in a Full Mesh for MPLS-TE topology can be calculated by the following formula:

$$States = Number\ of\ nodes^2$$

The third important characteristic of the Segment Routing technology is a standardized and smooth compatibility with Software Defined Networks.

However, every technology has its drawbacks and Segment Routing is not an exception. Some of its limitations are provided below:

- Not all equipment supports this technology. In Cisco, for example, only XR and XE iOS support this functionality;
- Cost of the equipment, that supports Segment Routing;
- Limited number of services, which can be implemented with a Segment Routing;
- Poor performance in topologies with an IPv6 addressing scheme:
    - No support for load balancing;
    - Fragmentation is not supported, only 10 labels can be in stack;

- TE FRR is not supported.

3 Implementation of MPLS-TE and SR-TE

The third part of the capstone project is devoted to the implementation of MPLS-TE and Segment Routing-TE technologies. There will be demonstrated three different topologies:

- Topology with MPLS-TE;
- Topology with Segment Routing-TE;
- Topology with Segment Routing-TE and Opendaylight controller.

Besides the configuration of the Traffic Engineering tunnels in all of these topologies, analysis of the following parameters will be done:

- Link`s load:
  - Number of control plane messages on a link;
  - Size of data and control packets on a link;
- Tunnel`s convergence and re-route time in case the primary link goes down.

3.1 Emulator

As it has already been said in previous chapter, a limited number of devices support Segment Routing technology. Due to the lack of required hardware, all topologies were built and configured in Graphical Network Simulator 3 (GNS3), which allows creating a networking environment on any laptop.

Despite the word “Simulator” in the name of this software, in fact, GNS3 is an emulator because it provides engineers an opportunity to create a model of any computer and launch the original operating system. All major device components are emulated, including processor, memory, and input/output devices. In case of Cisco devices, which were used in this part, GNS3 creates a model of router and launches Cisco IOS inside of it.

Below are provided some reasons why GNS3 was chosen for the practical part:

- Full functionality of emulated devices. When launching the Cisco router, we will have access to almost all the functions that work on a real router;
- The ability to build heterogeneous networks. It means that we can assemble a circuit where there will be not only Cisco devices but also Juniper, Mikrotik, CheckPoint, etc.;
- Adding full-fledged workstations and servers to the network;
- GNS3 is freely available and has no restrictions on use.

However, GNS3 also has some drawbacks:

- Lack of ability to emulate switches. In our case it is not a significant problem, because only routers are in use;
- Very high requirements for system resources. Every emulated Cisco router requires 3-4 cores and 4GB of RAM for full performance;
- There is no available that can reproduce 100 percent of the actual behavior of the device.

3.2 Installation of an emulated environment

The first step in an environmental setup is a choice of host Operating System. In this capstone, I have chosen Ubuntu 18.04 due to the following reasons:

- Ubuntu provides an ability to avoid usage of GNS3-VM, which is a required component for Windows machines. Therefore, there is smaller overhead on the system;
- Cisco IOS is considered to perform more stable in Linux environment.

In order to implement the first two types of topologies (MPLS-TE and RSVP-TE), a number of actions had to be performed. The first one is an installation of GNS3 on Ubuntu 18.04. In order to complete this task, I had to perform the following actions:

- Issue command “sudo add-apt-repository ppa:gns3/ppa” in order to add repository;
- Issue command “sudo apt-get update” in order to update system package list;
- Issue command “sudo apt-get install gns3-gui” in order to install Graphical Network Simulator 3.
- During the installation, process system asked if all users could use packets` s capturing option and run GNS3. Answer “Yes” was put.

The next step is an addition of Cisco router image to the GNS environment. In this capstone, Cisco IOS “iosxrv-demo-6.0.0” was chosen, because it is a free version of IOS, which supports both Segment Routing-TE and MPLS-TE technologies.

GNS3 does not have routers with such IOS in built-in solutions, so I had to choose the type of Virtual Machine, under which each of routers will run. GNS3 offers several possible options:

- Qemu VMs;
- VirtualBox VMs;
- VMware VMs;
- Dynamips.

For this part, QEMU VM was chosen for emulation of Cisco IOS “iosxrv-demo-6.0.0”. QEMU is an emulator of various devices that allow running operating systems designed for one architecture on another. In addition, QEMU is capable of emulating various peripheral devices: network cards, HDD, video cards, etc.

It works like this: instructions/binary code (for example, ARM) is converted to an intermediate platform-independent code using the TCG (Tiny Code Generator) converter, and then this

platform-independent binary code is already converted to the target instructions/code (for example, x86) (35).

ARM -> middleware -> x86

In fact, the engineer can run virtual machines on QEMU on any host, even with older processor models that do not support Intel VT-x (Intel Virtualization Technology)/AMD SVM (AMD Secure Virtual Machine). However, in this case, it works very slowly, due to the fact that the binary must be recompiled on the fly twice using TCG (TCG is Just-in-Time compiler) (22).

Before the discussion of the technology, which allows to speed up QEMU`s emulating process, I would like to refer to the processor`s architecture. Binary program codes have a different level of access to the data. This technology is called “Protection rings” and shown on the figure 47.

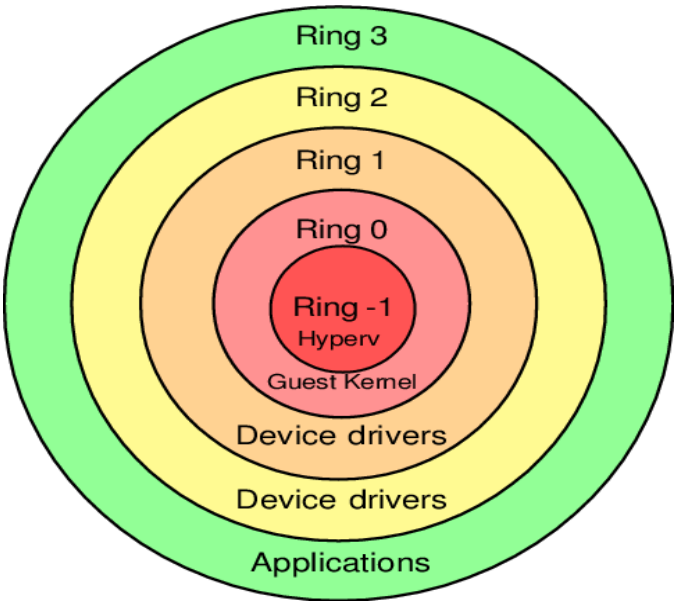

Figure 47 (source: https://geeks-world.github.io/articles/466549/index.html) – Processor`s architecture

For example, the Operating System runs on ring 0 and has full access to all the data on the machine. User applications. On the other hand, have access to a smaller scope of data and operate on the Ring 3. Before the invention of advanced virtualization technologies, like Intel VT, all virtual machines operated on Ring 1 and, in order to get access to the Ring 0 data, hypervisor had to modify each request and afterward perform it on Ring 0. Standard Qemu worked in the same way. This scheme assumes a large amount of unnecessary load and, as a result, the slow operation of the virtual machines. The solution was the invention of Ring -1, which allows hypervisors to operate directly with the hosts’ processor.

QEMU-KVM technology is a modern solution, which allows QEMU to have access to the Ring 0 and significantly increase its performance.

In order to install QEMU-KVM on my host machine, I had to issue two commands:

- "Sudo apt install qemu qemu-kvm libvirt-bin bridge-utils" – installation of QEMU-KVM;
- "Sudo usermod –aG libvirt-qemu $aleksei" – addition of my login user to the libvirt-qemu group.

3.2.1 Router`s image

As it has already been said in the previous chapter, for a capstone project Cisco IOS "iosxrv-demo-6.0.0" was chosen. Corresponding file, Cisco IOS "iosxrv-demo-6.0.0.vmdk", was downloaded from the Cisco website.

However, .vmdk file extension is not supported by QEMU-KVM technology. Conversion from .vmdk to .qcow2 format was made by command "qemu-img convert –f vmdk iosxrv-demo-6.0.0.vmdk –O qcow2 iosxrv-demo-6.0.0.qcow2". As a result, iosxrv-demo-6.0.0.qcow2 was gotten.

The next step was a creation of a Cisco router`s QEMU VM in GNS3. It was performed in several actions, explained in details below on figures 48-51.

Step 1 – Here the choice of appropriate Qemu binary and allocation of required Random Access Memory (RAM).

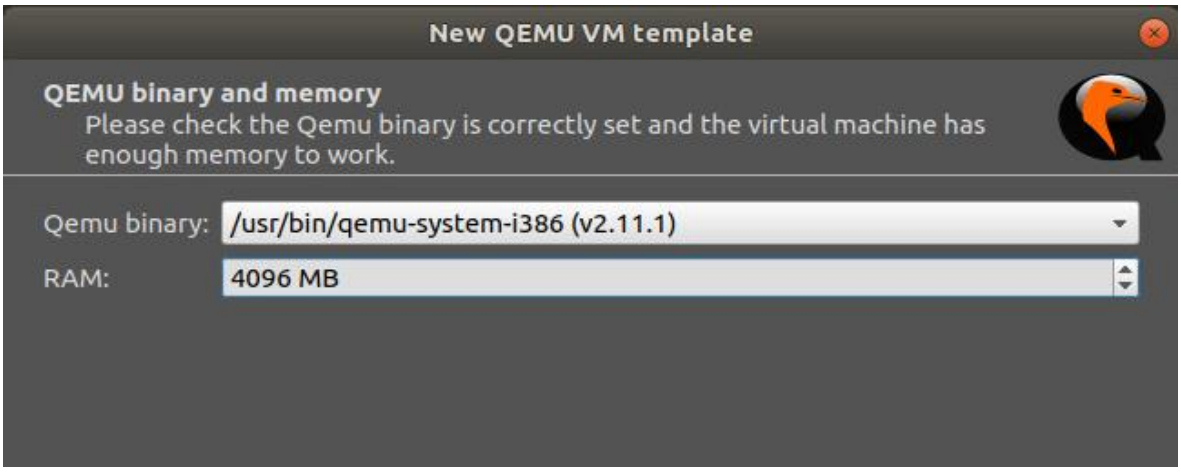

Figure 48 – Step 1

Step 2 – Choice of the disk image. Here newly created iosxrv-demo-6.0.0.qcow2 was chosen.

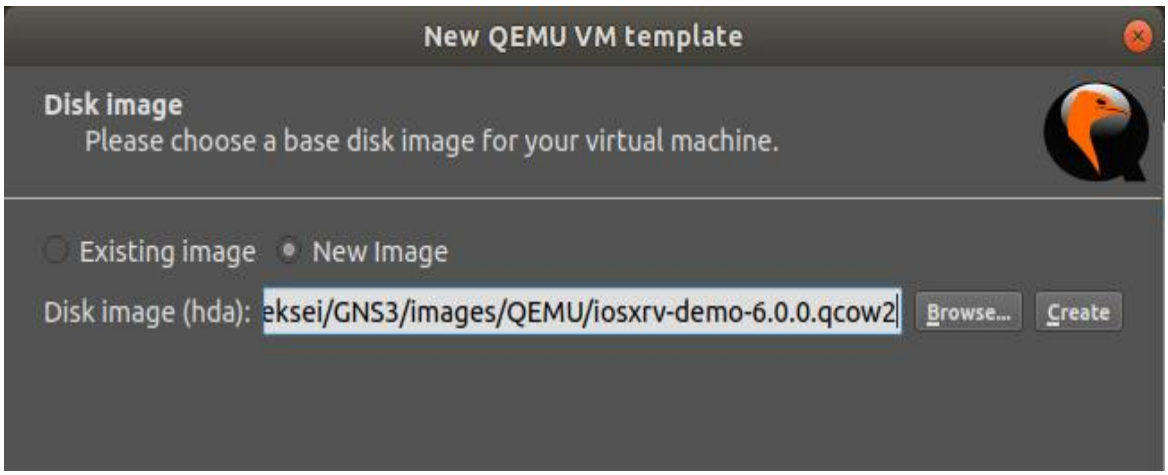

Figure 49 – Step 2

Step 3 – Here specific parameters for the selected image are configured. In our case, number of vCPUs was set to 4, telnet was chosen as a console type and category “Routers” was selected.

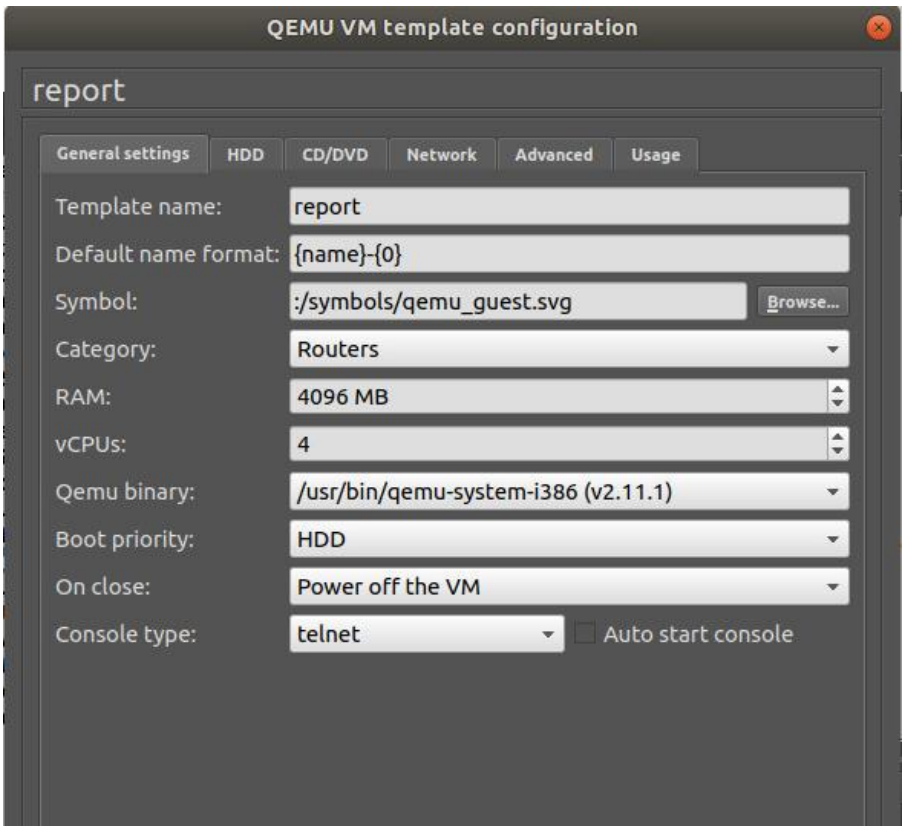

Figure 50 – Step 3

Step 4 – Setup of the required number of network ports

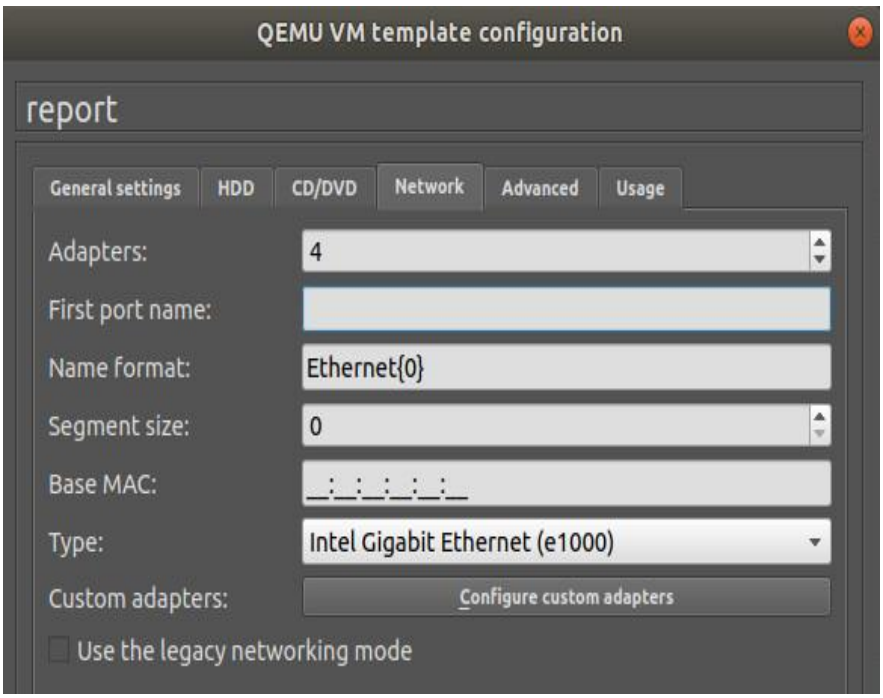

Figure 51 – Step 4

3.3 MPLS-TE topology

The topology chosen for implementation of MPLS-TE technology, is provided in figure 52.

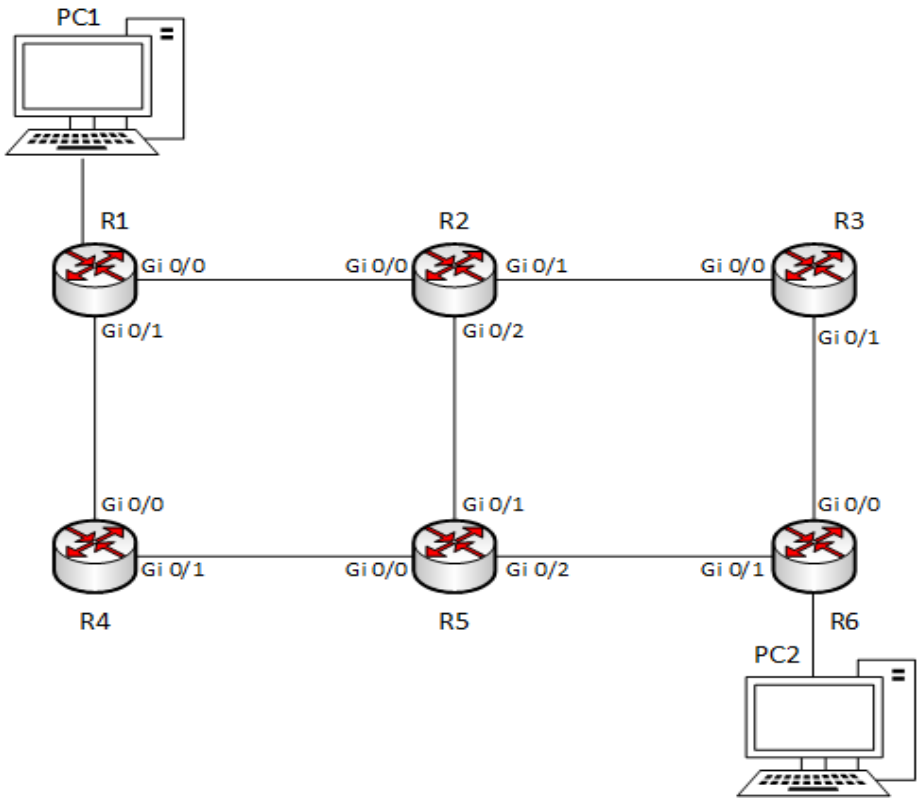

Figure 52 – MPLS-TE topology

Table 24 – IP addresses in topology

Device	Interface	IP address
PC1	To R1	172.16.1.1/24
	Loopback	100.100.100.100/32
PC2	To R6	172.16.2.1/24
	Loopback	200.200.200.200/32
R1	To PC1	172.16.1.2/24
	Loopback	10.10.10.1/32
	Gi0/0	192.168.1.1/24
	Gi0/1	192.168.3.1/24
R2	Loopback	10.10.10.2/32
	Gi0/0	192.168.1.2/24
	Gi0/1	192.168.2.1/24
	Gi0/2	192.168.5.1/24
R3	Loopback	10.10.10.3/32
	Gi0/0	192.168.2.2/24
	Gi0/1	192.168.7.1/24
R4	Loopback	10.10.10.4/32
	Gi0/0	192.168.3.2/24
	Gi0/1	192.168.4.1/24
R5	Loopback	10.10.10.5/32
	Gi0/0	192.168.4.2/24
	Gi0/1	192.168.5.2/24
	Gi0/2	192.168.6.2/24
R6	To PC2	172.16.2.2/24
	Loopback	10.10.10.6/32
	Gi0/0	192.168.7.2/24
	Gi0/1	192.168.6.1/24

All routers in this topology are MPLS-TE capable routers. MPLS-TE tunnel is configured on router R1 in this topology, while routers R2, R3, R4, R5 have the same configuration, shown in figure 53. Short command reference is provided in table 25. R6 has a static route to Customer 1 and OSPF route redistribution command.

In the process of basic topology configuration, the following actions were performed:

- Assignment of the corresponding IP addresses to all network interface on all devices;
- Starting the routing process (OSPF):
    - Assignment of the router-id;
    - Enablement of the redistribution of static routes (only on R1 and R3);
    - Area 0 configuration:
        - Allocation of the interfaces to OSPF process;
        - Enablement of MPLS-TE;
    - Assignment of the router-id to MPLS-TE process;
- Configuration of the static routes (only on R1 and R3);
- Enablement of the RSVP:
    - Allocation of the interfaces to RSVP process;
- Enablement of the MPLS:
    - Allocation of the interfaces to MPLS process.

Table 25 – MPLS-TE configuration commands

Device	**Command**	**Description**
R1	explicit-pat name *name*	Creation of a new explicit path;
	index *x* next-address strict ipv4 unicast *address*	Specification of the hops in the path;
	ipv4 unnumbered *interface*	Enablement of IP processing on explicitly unspecified interface;
	signaled-bandwidth *value*	Specification of the tunnel`s bandwidth;
	autoroute announce	Installation of the TE tunnel in the routing table;
	destination *address*	Tunnel`s destination
	path option *x* explicit name *name*	Binding of the explicit path to the tunnel;
	rsvp	Global enablement of RSVP technology;
	mpls traffic-eng	Enablement of MPLS-TE technology in global and OSPF.

```
RP/0/0/CPU0:R2#show running-config
Tue Jan 28 13:08:04.193 UTC
Building configuration...
!! IOS XR Configuration 6.0.0
!! Last configuration change at Tue Jan 28 13:07:57 2020 by aleksei
!
hostname R2
interface Loopback1
 ipv4 address 10.10.10.2 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 192.168.1.2 255.255.255.0
 shutdown
!
interface GigabitEthernet0/0/0/1
 ipv4 address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 ipv4 address 192.168.5.1 255.255.255.0
!
router ospf 1
 router-id 2.2.2.2
 area 0
  mpls traffic-eng
  interface Loopback1
   passive enable
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
 !
 mpls traffic-eng router-id Loopback1
!
rsvp
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
```

Figure 53 – R2-R6 configuration

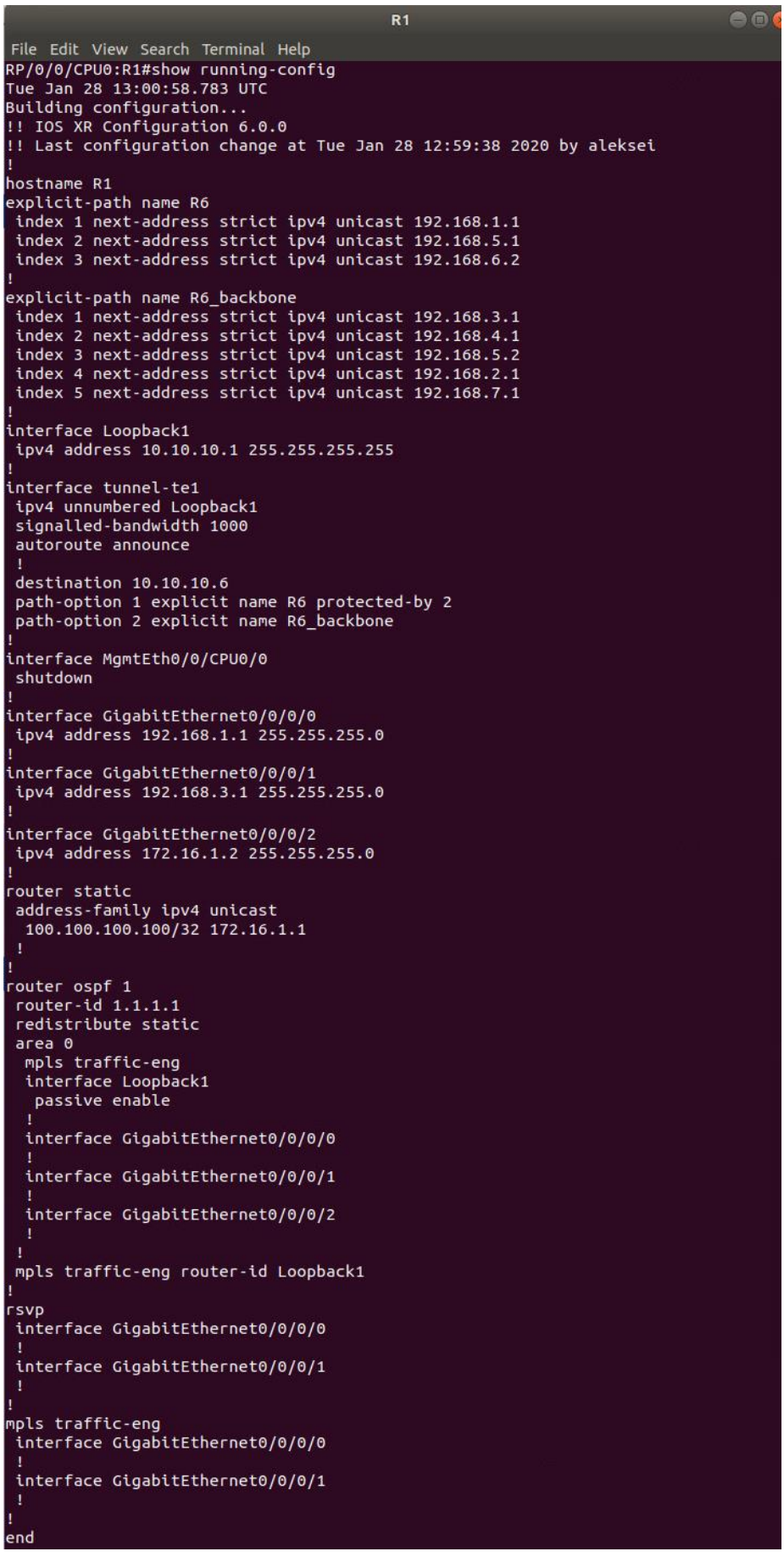

Figure 54 – R1 configuration

After the configuration of the basic functionality, on Router 1 MPLS-TE tunnel was created. It had the characteristics, described further:

- Signalled-bandwidth was enabled and was used for signaling required Bandwidth through the entire tunnel and on all devices;
- Path options were set with the two preconfigured path options:
    - First one, “R6”, was a primary one. It specified a path from router R1 to router R6 through the nodes R2 and R5 (R1-R2-R5-R6);
    - Second one, “R6_backbone”, was backup option in case “R6” path fails. It specified a path from router R1 to router R6 through the nodes R4, R5, R2, R3 (R1-R4-R5-R2-R3-R6);
- Traffic engineering tunnel was added to the routing table of the device.

On figure 50 we can see, that each Cisco router requires at least 4 GB of RAM and 4 Processors for full system operation. Since the topology was created in an emulator on my personal computer, which had limited resources, I think it is not correct to conduct different performance and time tests, because its results will not reflect the actual behavior due to the big variance in collected data.

However, it is still possible to estimate the total load on channels for all technologies and track forwarded control plane messages.

First, I would like to analyze a packet with a simple ICMP Echo Request, which was sent from Customer 1 to Customer 2. On figure 55 it can be seen, that the total size of the packet, transferred through the Traffic Engineering tunnel is 118 bytes. Multiprotocol Label Switching Header occupies 4 bytes and specifies 3 values, determined below.

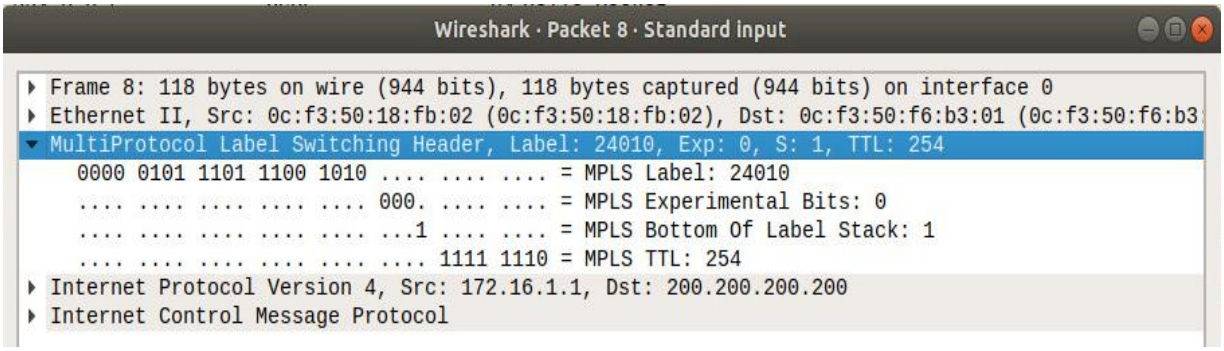

Figure 55 – ICMP Echo Request

- MPLS Label - 24010
- TTL - 254

- Bottom of the stack flag – 1, meaning that this header is the last one.

Therefore, we see that the overhead of one packet when using MPLS-TE technology is only 4 bytes. This overhead is the same in size on all the links, except the link, connected to R6, because only one label is swapped on all nodes. Packets on a link, connected to R6, will not carry MPLS header due to PHP.

Next, I would like to analyze the packets flow on the links in a stable Traffic Engineering tunnel`s condition. On figure 56 a simple collection of packets, taken with WireShark, on the link R1-R2 is provided. Here we can see two types of messages:

- OSPF hello packet;
- RSVP SREFRESH message.

It is important to notice, that both types of packets in MPLS-TE topology, as we can see, are transferred in both directions (from 192.168.1.1 and from 192.168.1.2).

No.	Time	Source	Destination	Protocol	Length	Info
271	837.682263	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
272	838.565511	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
273	847.501144	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
274	847.876029	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
275	848.968086	192.168.1.2	192.168.1.1	RSVP	54	SREFRESH Message.
276	857.381045	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
277	857.634717	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
278	862.310683	192.168.1.1	192.168.1.2	RSVP	54	SREFRESH Message.
279	866.409986	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
280	867.083546	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
281	875.530332	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
282	876.903148	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
283	885.359173	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
284	886.787611	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
285	887.319938	192.168.1.1	192.168.1.2	RSVP	54	SREFRESH Message.
286	887.405375	192.168.1.2	192.168.1.1	RSVP	54	SREFRESH Message.
287	894.812030	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
288	896.631096	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
289	904.388643	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
290	906.331321	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
291	913.436706	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
292	913.439386	192.168.1.1	192.168.1.2	RSVP	54	SREFRESH Message.
293	914.593201	192.168.1.2	192.168.1.1	RSVP	54	SREFRESH Message.
294	916.009976	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
295	923.356072	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
296	925.109519	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
297	932.905201	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
298	934.549175	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
299	942.295281	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
300	944.498208	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
301	945.285308	192.168.1.1	192.168.1.2	RSVP	54	SREFRESH Message.
302	951.603713	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
303	954.307805	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
304	958.530682	192.168.1.2	192.168.1.1	RSVP	54	SREFRESH Message.
305	961.544600	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
306	964.068545	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
307	969.273280	192.168.1.1	192.168.1.2	RSVP	54	SREFRESH Message.
308	971.543328	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet
309	973.835845	192.168.1.1	224.0.0.5	OSPF	94	Hello Packet
310	981.261728	192.168.1.2	224.0.0.5	OSPF	94	Hello Packet

Figure 56 – Stable MPLS-TE condition

The last thing I would like to analyze in MPLS-TE topology is a convergence process in case of any failure in the network occurs. In order to create such a situation, firstly, I made a shutdown of an interface on router R2 to R1. That caused a switching of the tunnel to the second path option. Then I issued a command “no shutdown” on the same interface and collected packets.

39	125.729947	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
40	127.365144	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
41	127.967263	192.168.1.1	224.0.0.5	OSPF	122 LS Update
42	127.991919	10.10.10.1	10.10.10.6	RSVP	254 PATH Message. SESSION: IPv4-LSP, Destination 10.10.10.6,
43	128.190104	192.168.1.1	224.0.0.5	OSPF	222 LS Update
44	128.379051	192.168.1.2	192.168.1.1	RSVP	162 Component Messages Dissected
45	128.868769	192.168.1.1	192.168.1.2	RSVP	62 Component Messages Dissected
46	130.006392	192.168.1.2	224.0.0.5	OSPF	98 LS Acknowledge
47	135.508193	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
48	136.554698	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
49	144.738602	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
50	146.543964	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
51	154.616484	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
52	155.400111	192.168.1.2	192.168.1.1	RSVP	54 SREFRESH Message.
53	155.696411	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
54	164.435449	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
55	164.763346	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet

Figure 57 – MPLS-TE convergence process

“No shutdown” command was issued at the time moment 125 sec of the monitored session. After that we can see update messages from IGP OSPF and RSVP. In case of IGP OSPF, typical LS update and LS Acknowledge messages are observed.

Now I would like to take a closer look at RSVP convergence process. On figure 57 different types of RSVP messages can be observed:

- Path message – each node in the topology transmits such a message along the path in order to create a path state on other routers. This helps a particular device to understand its downstream and upstream neighbors;
- Resv message – these messages create and maintain the reservation state. Such a message can be seen in a 44th packet. It is sent from downstream to upstream (from 192.168.1.2 to 192.168.1.1);
- ResvConfirm message – confirmation on the Resv message. Such a message can be seen in a 45th packet.

The total RSVP overhead while the convergence process is 478 bytes for one link of the Traffic Engineering tunnel. Exactly the same volume of overhead was observed on other links of the tunnel.

3.4 Segment Routing topology

The topology, chosen for an implementation of Segment Routing Traffic Engineering is exactly the same, as in case of MPLS-TE (Figure 52). Addressing scheme is also the same as in case of MPLS-TE (Table 25).

All routers in this topology are SR-TE capable routers. SR-TE tunnel is configured on router R1 in this topology, while routers R2, R3, R4, R5 have the same configuration, shown in figure 54. Short command reference is provided in table 26. R6 has a static route to Customer 1 and OSPF route redistribution command.

In the process of basic topology configuration, same actions, as in chapter 3.3:

Table 26 – SR-TE configuration commands

Device	Command	Description
R1	explicit-pat name *name*	Creation of a new explicit path;
	index *x* next-label *label*	Specification of the hops in the path;
	ipv4 unnumbered *interface*	Enablement of IP processing on explicitly unspecified interface;
	signaled-bandwidth *value*	Specification of the tunnel`s bandwidth;
	autoroute announce	Installation of the TE tunnel in the routing table;
	destination *address*	Tunnel`s destination
	path option *x* explicit name *name* segment-routing	Binding of the explicit path to the tunnel;
	mpls traffic-eng	Enablement of MPLS-TE technology in global and OSPF;
	segment-routing mpls	Enablement of SR-TE technology in OSPF global mode and area mode;
	segment-routing forwarding mpls	Enablement of SR forwarding in OSPF area;
	prefix-sid absolute *label*	Specification of the interface`s SID;

```
RP/0/0/CPU0:ios#show running-config
Thu Jan 30 15:18:45.563 UTC
Building configuration...
!! IOS XR Configuration 6.0.0
!! Last configuration change at Thu Jan 30 15:18:36 2020 by aleksei
!
explicit-path name toSR4
 index 1 next-label 17002
 index 2 next-label 17005
 index 3 next-label 17004
!
interface Loopback1
 ipv4 address 10.10.10.5 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 192.168.4.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
 ipv4 address 192.168.5.2 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 ipv4 address 192.168.6.2 255.255.255.0
!
router static
 address-family ipv4 unicast
  10.10.10.4/32 tunnel-te1
 !
!
router ospf 1
 router-id 5.5.5.5
 segment-routing mpls
 area 0
  segment-routing forwarding mpls
  mpls traffic-eng
  segment-routing mpls
  interface Loopback1
   prefix-sid absolute 17005
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
 !
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
```

Figure 58 – R2-R6 configuration

```
RP/0/0/CPU0:R1#show running-config
Thu Jan 30 15:12:21.679 UTC
Building configuration...
!! IOS XR Configuration 6.0.0
!! Last configuration change at Thu Jan 30 15:11:44 2020 by alek
!
hostname R1
explicit-path name to_SR6
 index 10 next-label 17002
 index 20 next-label 17005
 index 30 next-label 17006
!
explicit-path name to_SR6_backbone
 index 1 next-label 17004
 index 2 next-label 17005
 index 3 next-label 17006
!
interface Loopback1
 ipv4 address 10.10.10.1 255.255.255.255
!
interface tunnel-te1
 ipv4 unnumbered Loopback1
 signalled-bandwidth 100000
 autoroute announce
 !
 destination 10.10.10.6
 fast-reroute protect node bandwidth
 path-option 1 explicit name to_SR6 segment-routing
 path-option 2 explicit name to_SR6_backbone segment-routing
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/0/0/1
 ipv4 address 192.168.3.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 ipv4 address 172.16.1.2 255.255.255.0
!
router static
 address-family ipv4 unicast
  100.100.100.100/32 172.16.1.1
 !
!
router ospf 1
 router-id 1.1.1.1
 segment-routing mpls
 segment-routing prefix-sid-map advertise-local
 redistribute static
 area 0
  segment-routing forwarding mpls
  mpls traffic-eng
  segment-routing mpls
  interface Loopback1
   prefix-sid absolute 17001
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
 !
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
!
segment-routing
!
end
```

Figure 59 – R1 configuration

In the case of Segment Routing-TE I would like to have a look at the same parameters, which were analyzed in the previous chapter.

First, I would like to pay attention to a packet with a simple ICMP Echo Request, which was sent from Customer 1 to Customer 2. On figure 60 it can be seen that the total size of the packet, transferred through the Traffic Engineering tunnel is 122 bytes, which is 4 bytes more, that in case of MPLS-TE topology. This can be explained by the fact, that packets in SR-TE carry the whole label stack, which is defined in originating device. The Packet, which is shown in figure 56, was captured on a link between R1 and R2 and contains two labels in a stack. It is obvious, that if topology was larger and number of hops was increased, then overhead for each packet would be even bigger.

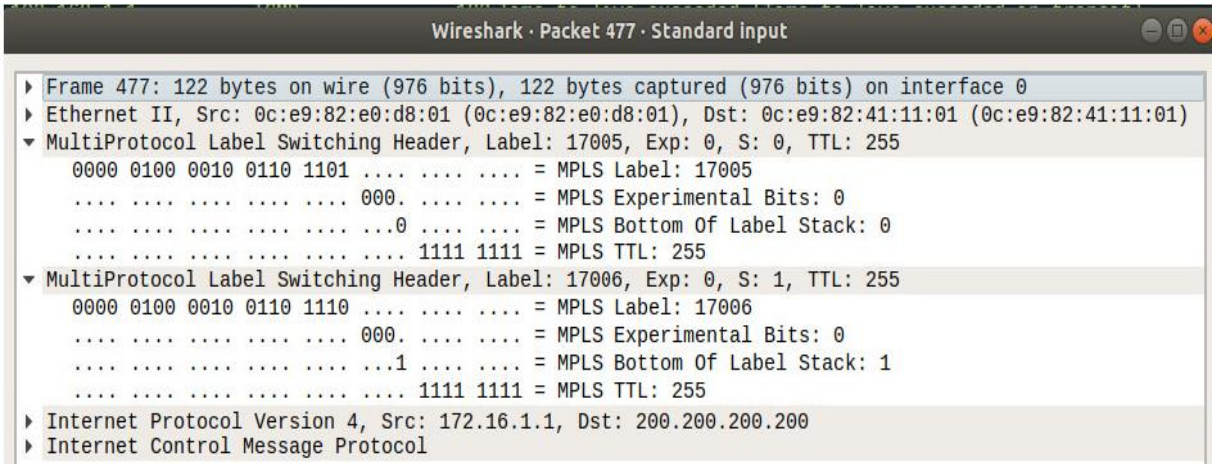

Figure 60 – ICMP Echo Request in SR-TE topology

- MPLS Labels – 17005 (SID of the R5), 17006 (SID of the R6);
- TTL – 255 in each header;
- Bottom of the stack flag – 1, meaning that this header is the last one. It is set only in the second MPLS header.

Next, I would like to analyze the packets flow on the links in a stable Traffic Engineering tunnel`s condition. On figure 61 a simple collection of packets, taken with WireShark, on the link R1-R2 is provided. Here we can see only one type of messages:

- OSPF hello packet;

Here we can notice, that there is now RSVP messages at all like it was in case of MPLS-TE technology. It is explained by the fact, that IGP protocol in SR-TE is a control plane for IPV4 and IPV6 packets, so there is no need for other protocols.

1	0.000000	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
2	3.041936	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
3	9.405773	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
4	12.921703	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
5	19.355626	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
6	22.762727	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
7	28.504667	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
8	31.950707	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
9	38.163831	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
10	41.060016	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
11	47.943633	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
12	50.229550	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
13	57.843802	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
14	60.037897	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
15	66.893973	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
16	69.167073	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
17	76.351472	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
18	78.587817	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
19	86.231224	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
20	88.306096	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet
21	95.970423	192.168.1.2	224.0.0.5	OSPF	94 Hello Packet
22	97.345182	192.168.1.1	224.0.0.5	OSPF	94 Hello Packet

Figure 61 – Stable SR-TE condition

The last thing I would like to analyze in SR-TE topology is a convergence process in case of any failure in the network occurs. In order to create such a situation, I made a shutdown of an interface on router R2 to R1. That caused a switching of the tunnel to the second path option. Results of this convergence process are shown in figure 62.

179	548.182971	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet
180	553.571479	192.168.3.1	224.0.0.5	OSPF	94 Hello Packet
181	557.351738	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet
182	562.671103	192.168.3.1	224.0.0.5	OSPF	94 Hello Packet
183	567.221111	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet
184	571.396328	192.168.3.1	224.0.0.5	OSPF	110 LS Update
185	571.497364	192.168.3.1	224.0.0.5	OSPF	122 LS Update
186	571.663753	192.168.3.1	224.0.0.5	OSPF	222 LS Update
187	572.669540	192.168.3.1	224.0.0.5	OSPF	94 Hello Packet
188	573.420537	192.168.3.2	224.0.0.5	OSPF	118 LS Acknowledge
189	576.780621	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet
190	581.990333	192.168.3.1	224.0.0.5	OSPF	94 Hello Packet
191	586.359484	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet
192	591.468989	192.168.3.1	224.0.0.5	OSPF	94 Hello Packet
193	595.438862	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet
194	600.971220	192.168.3.1	224.0.0.5	OSPF	94 Hello Packet
195	605.199839	192.168.3.2	224.0.0.5	OSPF	94 Hello Packet

Figure 62 – SR-TE convergence process

“Shutdown” command was issued at the time moment 182 sec of the monitored session. After that, we can see update messages only from IGP OSPF. Here LS update and LS Acknowledge messages are observed. All four OSPF messages, related to the convergence process, weight 578 bytes. It is more, than the total weight of OSPF messages, shown in figure 57 (442 bytes). OSPF Messages are bigger in this topology due to newly introduced OSPFv2 Extended Link Opaque LSA (LSA Type 10), which carries SR-TE related information (ADJ-SID, TLV`s types and other)

and is shown on figure 63. The size of such LSA directly depends on the number of the nodes in the path. The larger the topology, the more information is transmitted. However, as we have already seen, not only OSPF process converges in case of MPLS-TE, but also RSVP process. These two processes occupied 920 byres in total.

Therefore, we can see, that the convergence process occupies less Bandwidth in the case of SR-TE (578 bytes instead of 920).

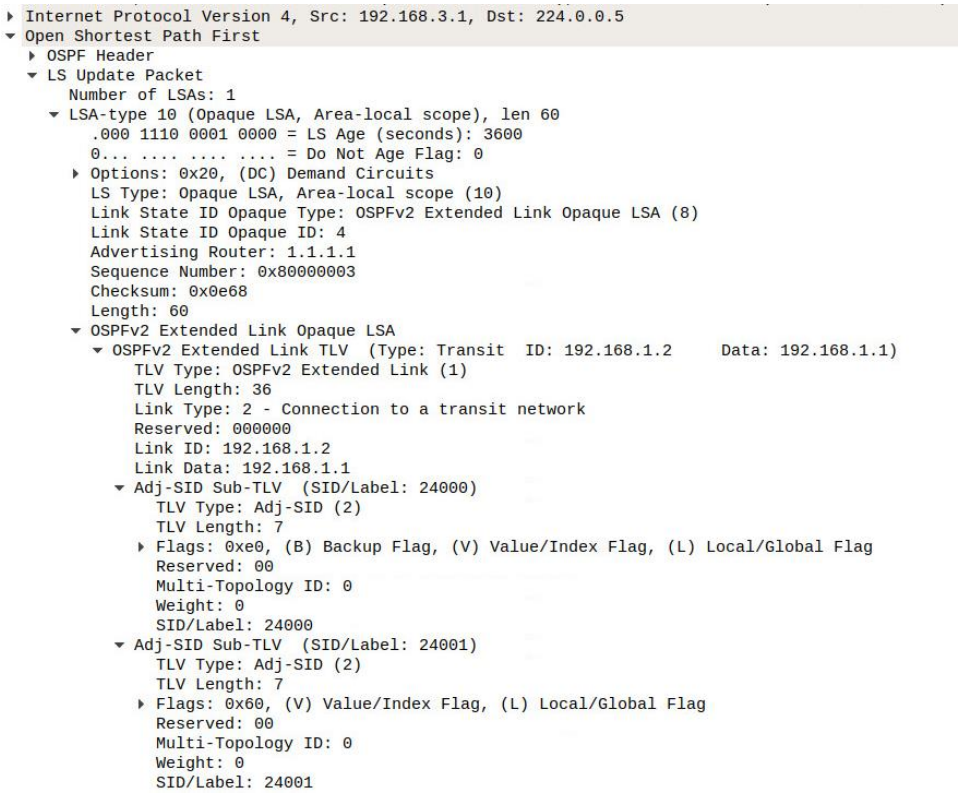

Figure 63 – OSPF LSA-type 10

3.5 Segment Routing topology with Controller

The last topology built in this capstone project is dedicated to the Segment Routing technology`s collaboration with a controller.

Nowadays Software-Defined Networking is becoming more and more popular. It can be explained by the fact that programmable networks are more flexible and responsive to the needs of companies and users. Software-Defined Networks need one more component in its topology. It is called “Controller”. In my work, I have chosen controller, developed by an Opendaylight (ODL) community due to the following reasons:

- Open source code;
- Stable and standardized operation in topologies with segment routing;

- Multi-protocol and multi-vendor support.

ODL provides network services for all network devices in multi-vendor environment. Microservice architecture allows network engineers to manage applications and protocols and provide interaction between customers and providers. Opendaylight controller combines open source code, open standards and API, which makes programming of the network more understandable and adaptable. Figure 64 illustrates ODL`s architecture.

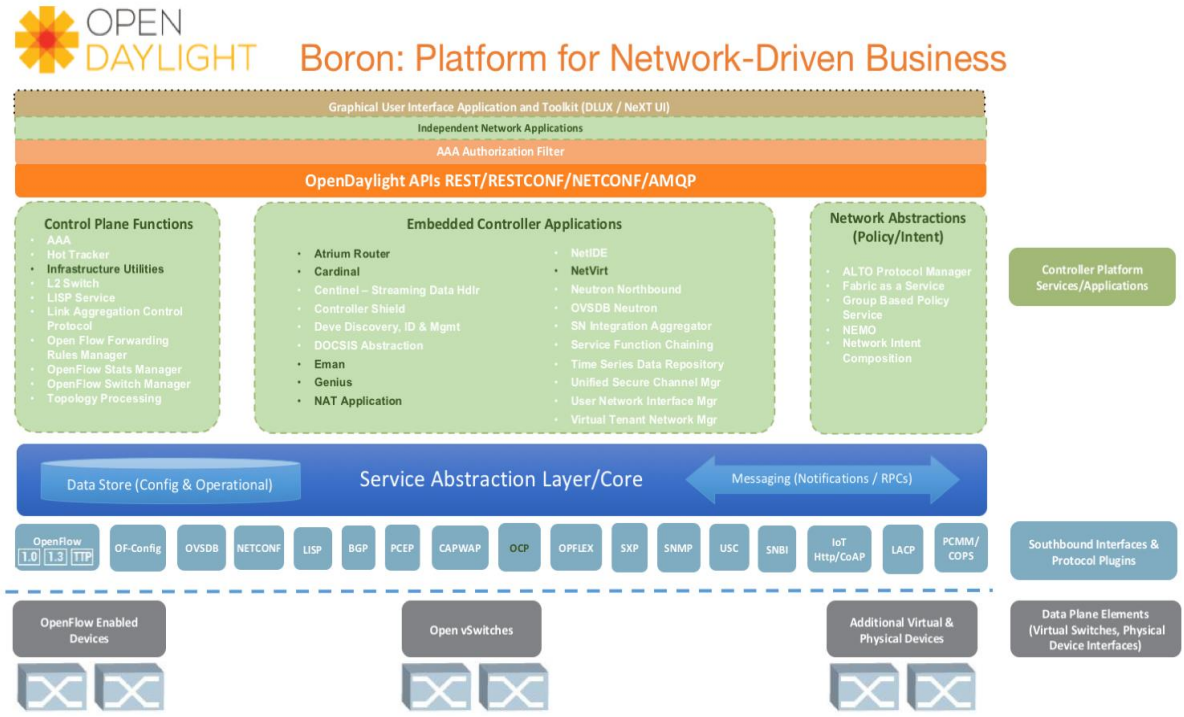

Figure 64 – Opendaylight architecture

There are three main parts in Opendaylight architecture:

- Southbound APIs – These interfaces and plugins are used to communicate with network devices. It provides a network architecture control function and is able to make dynamic changes in device`s configuration. Different protocols can be enabled as a separate plugin, BGP-LS and PCEP for example. Each plugin maintains its own connection state and performs corresponding changes.
- Control plane – This part is responsible of configuration and managing the devices over southbound interfaces. In other words, Control plane instructs networking devices how to handle packets. Below are listed the main functions of the Control plane:
    - Topology maintenance;
    - Instantiation and selection of routes, network flow management;
    - Failover mechanisms.

- Northbound APIs – This element provides a communication between ODL`s control plane and user`s applications. Here the technology called REST API is used. REST API uses HTTP basic methods for its purposes.

3.6.1 Controller setup

While the Opendaylight setup process, the following steps were performed:

- Download of the Opendaylight Boron release source code from the www.opendaylight.org;
- Creation of the directory for the binary "*sudo mkdir /usr/local/karaf*";
- Unpacking of the source code to the newly created folder;
- Installation of the Opendaylight controller "*sudo-update alternatives –install /usr/bin/karaf karaf /usr/local/karaf/distribution-karaf-0.5.4-Boron-SR4/bin/karaf*";
- Installation of additional features, needed for Segment Routing:
  - *Feaure:install odl-bgpcep-bgp-all;*
  - *Feaure:install odl-bgpcep-pcep-all;*
  - *Feaure:install odl-restconf-noauth;*
  - *Feaure:install odl-dlux-all*;
- Configuration of the IP address (1192.168.122.1) of the controller "*sudo nano /etc/network/interfaces*".

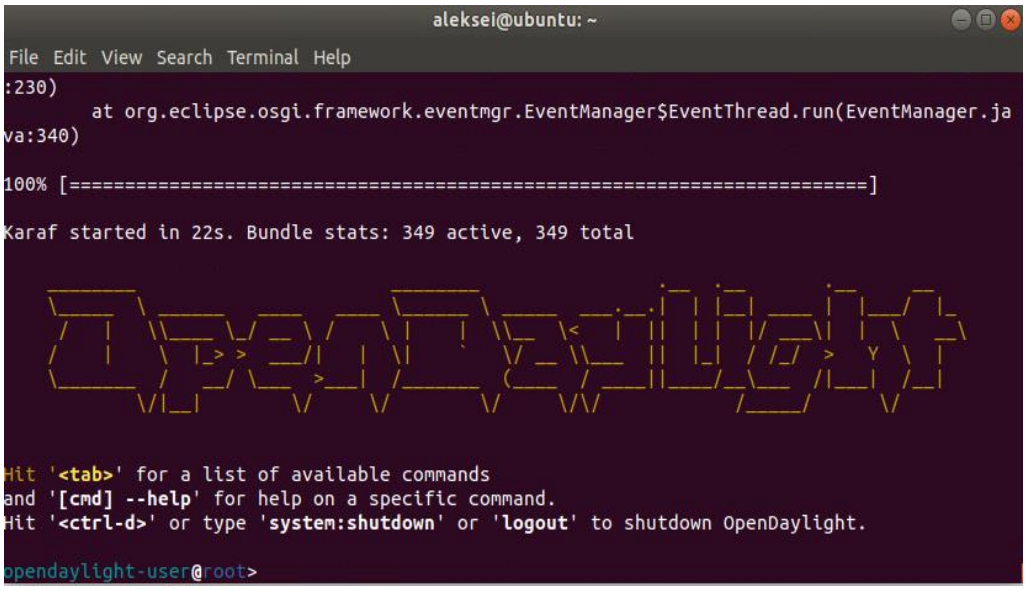

Figure 65 – Opendaylight interface

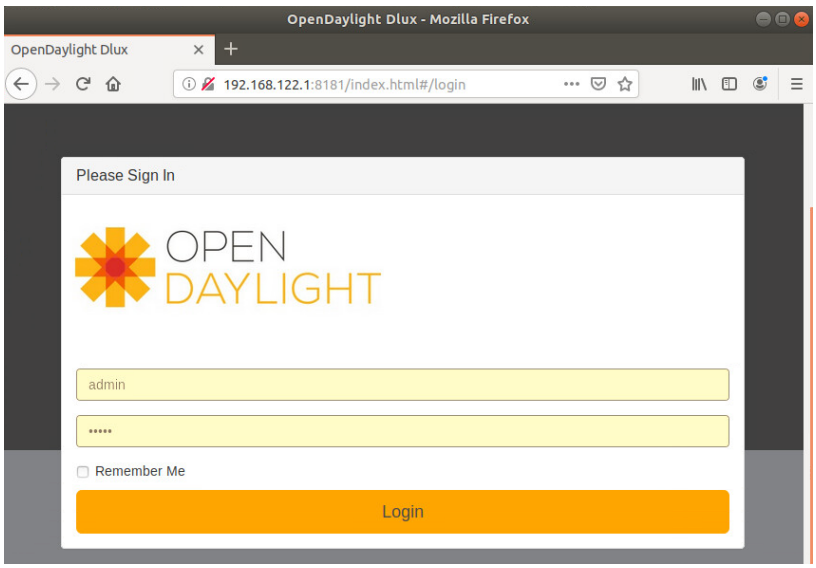

Figure 66 – Opendaylight web interface

3.6.2 Topology setup

Figure 67 illustrates the topology that was built for this part of the capstone. It is similar to the previous cases, except the presence of ODL controller. All six routers are Segment Routing capable devices. Router 1 is a connection point between topology and controller.

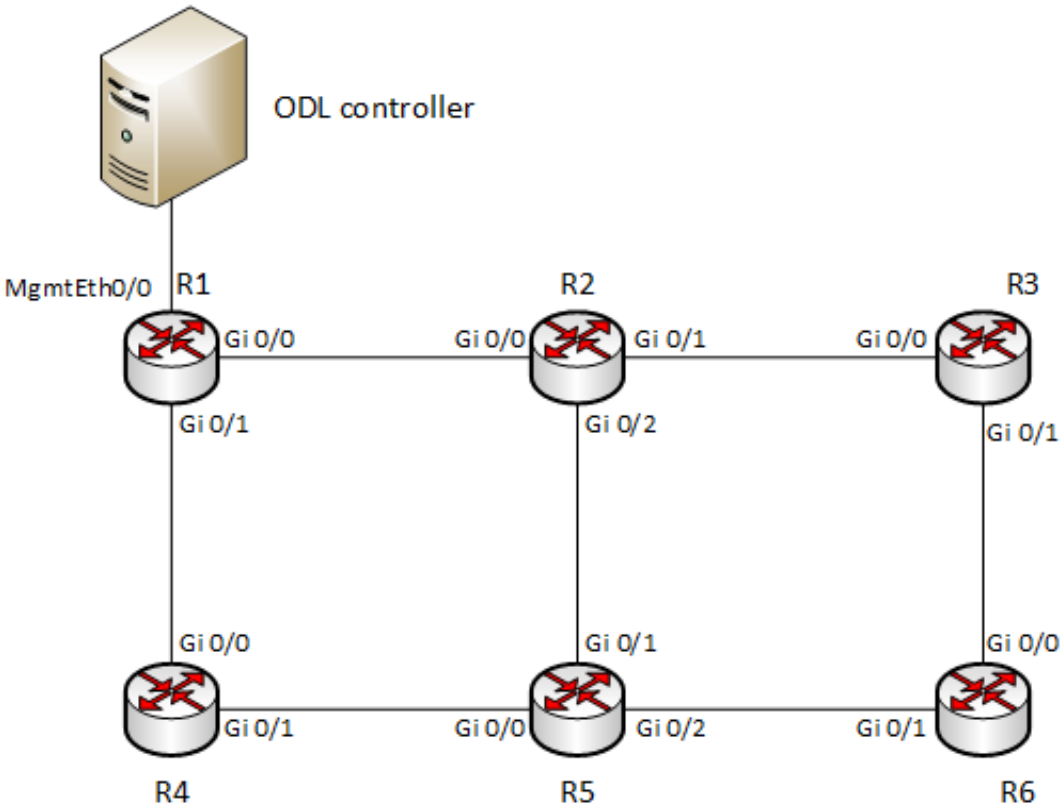

Figure 67 – SR with ODL topology

Table 27 – IP addresses in topology

Device	Interface	IP address
R1	Loopback	10.10.10.1/32
	MgmtEth0/0	192.168.122.5/24
Controller	To R1	192.168.122.1/24

In order to advertise topology information from the router to the controller, there should be configured a BGP-LS adjacency. BGP-LS is an extension to the original BGP protocol, which provides network device with an ability to report network topology, collected by OSPF, to the controller.

BGP-LS requires configuration on both sides. Firstly, I configured it on ODL controller by changing *41-bgp-example.xml* file. It was required to set correct peer`s address (192.168.122.5) and determine the type of BGP adjacency (interior BGP). Implemented changes are shown on figure 68.

```
xmlns:prefix="urn:opendaylight:params:xml:ns:yang:controller:bgp:rib:impl">prefix:bgp-peer</type>
                    <name>example-bgp-peer</name>
                    <host>192.168.122.5</host>
                    <holdtimer>180</holdtimer>
                    <retrytimer>10</retrytimer>
                    <peer-role>ibgp</peer-role>
                    <rib>
                        <type
xmlns:prefix="urn:opendaylight:params:xml:ns:yang:controller:bgp:rib:impl">prefix:rib-impl</type>
                    <name>example-bgp-rib</name>
                    <rib-id>example-bgp-rib</rib-id>
                    <local-as>1</local-as>
                    <bgp-rib-id>192.168.122.1</bgp-rib-id>
                    <!-- if cluster-id is not present, it's value is the same as bgp-id -->
                    <!-- <cluster-id>192.0.2.3</cluster-id> -->
                    <local-table>
                        <type
```

Figure 68 – *41-bgp-example.xml* file

Next, I configured BGP-LS process on Router 1:

- Enabled BGP process in global configuration mode;
- Specified BGP router-id;
- Activated required address families;
- Configured neighbor`s properties:
    - Specified remote AS;
    - Chose the correct update interface;
    - Specified link-state address family.

```
RP/0/0/CPU0:ios#show running-config
Fri Jan 31 11:29:00.145 UTC
Building configuration...
!! IOS XR Configuration 6.0.0
!! Last configuration change at Fri Jan 31 11:18:25 2020 by aleksei
!
cdp
ipv4 unnumbered mpls traffic-eng Loopback1
interface Loopback1
 ipv4 address 10.10.10.1 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 ipv4 address 192.168.122.5 255.255.255.0
!
interface GigabitEthernet0/0/0/0
 ipv4 address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet0/0/0/1
 ipv4 address 192.168.3.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 shutdown
!
router ospf 1
 distribute bgp-ls instance-id 2
 router-id 1.1.1.1
 segment-routing mpls
 redistribute connected
 address-family ipv4 unicast
 area 0
  segment-routing forwarding mpls
  mpls traffic-eng
  segment-routing mpls
  interface Loopback1
   prefix-sid absolute 17001
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
 !
 mpls traffic-eng router-id Loopback1
!
router bgp 1
 bgp router-id 10.10.10.1
 address-family ipv4 unicast
 !
 address-family vpnv4 unicast
 !
 address-family link-state link-state
 !
 neighbor 192.168.122.1
  remote-as 1
  update-source MgmtEth0/0/CPU0/0
  address-family link-state link-state
   route-reflector-client
  !
 !
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 pce
  peer source ipv4 192.168.122.5
  peer ipv4 192.168.122.1
  !
  segment-routing
  stateful-client
   instantiation
   delegation
  !
  speaker-entity-id 10.10.10.1
 !
 auto-tunnel pcc
  tunnel-id min 10 max 50
 !
 reoptimize timers delay installation 0
!
end
```

Figure 69 – R1 configuration

The next thing to configure is a protocol, which is used for sending Segment Routing Traffic Engineering tunnels information to the Router 1 from ODL controller. For this purpose path Computation Element Protocol (PCEP) was chosen.

PCEP defines a set of messages and objects used to manage PCEP sessions and to request and send paths for multi-domain traffic engineered LSPs (TE LSPs) (23). Path Computation element, ODL controller on our case, provides Path Computation Client, Router 1, with an information about tunnels. PCC initiates session and keeps it alive by sending “keepalive” messages.

PCEP transmits an information about tunnels in Explicit Route Objects, which consists of a number of subobjects. These subobjects can contain different types of information:

- Node`s ip addresses;
- SIDs;
- MPLS label`s.

PCC, in its turn, translates any of these formats in MPLS label`s format, accordingly to the rules, described below:

- If PCC gets a set of SIDs, then it converts these values to the MPLS labels. It is important that label`s values should be unique within the router and must represent only one SID;
- If PCC gets a set of IP addresses, it converts each of addresses to the SID value, accordingly to the segment routing table, which can be found by issuing command “show mpls traffic-eng segment routing summary”. Afterwards node proceeds SIDs values as described in the first point.
- If PCC gets an MPLS labels, it should not convert it to any other format. However, it should understand the value of the corresponding SIDs. In order to perform this task, router checks its SRLB and SRGB tables and specified offsets.

Subobjects, which are transmitted in ERO object, are called SR-EROs. When building the MPLS label stack from ERO, a PCC MUST assume that SR-ERO subobjects are organized as a last-in-first-out stack. The first subobject relative to the beginning of ERO contains the information about the topmost label. The last subobject contains information about the bottommost label (25). Both PCC and PCE must be capable of resolving these subobjects. On order to check this compatibility, PCC and PCE devices exchange specific types of messages with a STATEFUL-PCE-CAPABILITY TLV.

Formats of the ERO object and SR-ERO subobject are shown on figure 70 and 71 correspondingly

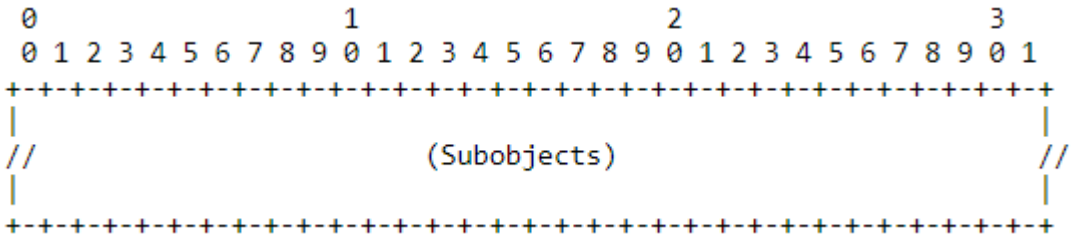

Figure 70 (source: https://tools.ietf.org/html/rfc8664#section-5.2.2) – Explicit Route Object

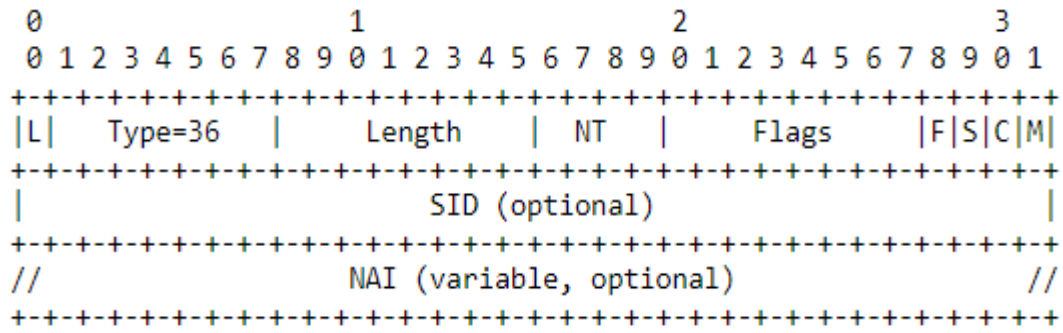

Figure 71 (source: https://tools.ietf.org/html/rfc8664#section-5.2.2) – SR-ERO

Table 28 – Adj-SID Sub-TLV`s fields

Field	Description
L	Flag, which indicates, weather subobject is “loose” or not. If it is set to 1, then PCC can replace SID value;
Type	Value = 36;
Length	Length of the subobject in bytes;
NT	Type of address, if it set to 0, then there is no address in subobject;
Flags	This field represents additional information about SIDs
SID	Segment Identifier’s value
NAI	Optional value. IPv4 or IPv6 address can be inserted there, which is associated with SID.

In each subobject at least one SID or IP address should be included. Also both of this values may be present

Required configurations can be done under "*mpls traffic-eng*" mode. Such configuration is shown on figure 69 and short command`s description is provided below in table 29:

Table 29 – PCE configuration

Command	**Description**
peer source ipv4 *address*	Specification of the router`s interface, connected to controller;
peer ipv4 *address*	Specification of the controller`s address;
segment-routing	Enablement of SR-TE technology;
stateful-client	Specification of the interaction mode;
instantiation	This command instructs router to accept new path`s and tunnels from ODL controller;
delegation	This command restricts any changes in path`s configuration to be done on router;
speaker-entity-id *address*	Specification of the node, which created tunnels and paths;
tunnel-id min 10 max 50	Specification of the tunnel-id range;
reoptimize timers delay instantiation *x*	Specification of the time interval, after which new tunnel should be signaled.

The result of mentioned configuration is shown in figure 70, where we can see an established adjacency.

```
RP/0/0/CPU0:ios#show mpls traffic-eng pce peer
Fri Jan 31 12:28:01.933 UTC
        Address   Precedence        State         Learned From
--------------- ------------ ------------ --------------------
  192.168.122.1          255           Up        Static config
RP/0/0/CPU0:ios#
```

Figure 70 – PCE peer status

The basic configuration of all other routers is shown in figure 71:

```
RP/0/0/CPU0:ios#show running-config
Fri Jan 31 11:39:21.013 UTC
Building configuration...
!! IOS XR Configuration 6.0.0
!! Last configuration change at Fri Jan 24 14:56:01 2020 by aleksei
!
ipv4 unnumbered mpls traffic-eng Loopback1
interface Loopback1
 ipv4 address 10.10.10.2 255.255.255.255
!
interface MgmtEth0/0/CPU0/0
 shutdown
!
interface GigabitEthernet0/0/0/0
 ipv4 address 192.168.1.2 255.255.255.0
!
interface GigabitEthernet0/0/0/1
 ipv4 address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet0/0/0/2
 ipv4 address 192.168.5.1 255.255.255.0
!
router ospf 1
 router-id 2.2.2.2
 segment-routing mpls
 area 0
  segment-routing forwarding mpls
  mpls traffic-eng
  segment-routing mpls
  interface Loopback1
   prefix-sid absolute 17002
  !
  interface GigabitEthernet0/0/0/0
  !
  interface GigabitEthernet0/0/0/1
  !
  interface GigabitEthernet0/0/0/2
  !
 !
 mpls traffic-eng router-id Loopback1
!
mpls traffic-eng
 interface GigabitEthernet0/0/0/0
 !
 interface GigabitEthernet0/0/0/1
 !
 interface GigabitEthernet0/0/0/2
 !
!
```

Figure 71 – R2-R6 configuration

3.6.3 SR-TE tunnel setup

As it has already been said in chapter 3.6, ODL controller has an ability to interact with user`s application through Northbound interface. User`s application can be used for sending

configurations via HTTP to network devices. In my capstone project I sent corresponding requests through the Postman application.

In order to send an information about new tunnel to the Router 1, I used "*add-lsp"* request and sent it to the "*http://192.168.122.1/restconf/operations/network-topology-pcep*" controller`s address. All the requests were sent by the POST method, because in this case server save all the data, which is enclosed in the body of the request. The following request parameters were set:

- Content-type – application/xml;
- Accept – application/xml;
- Authorization – Basic Authorization, admin/admin.

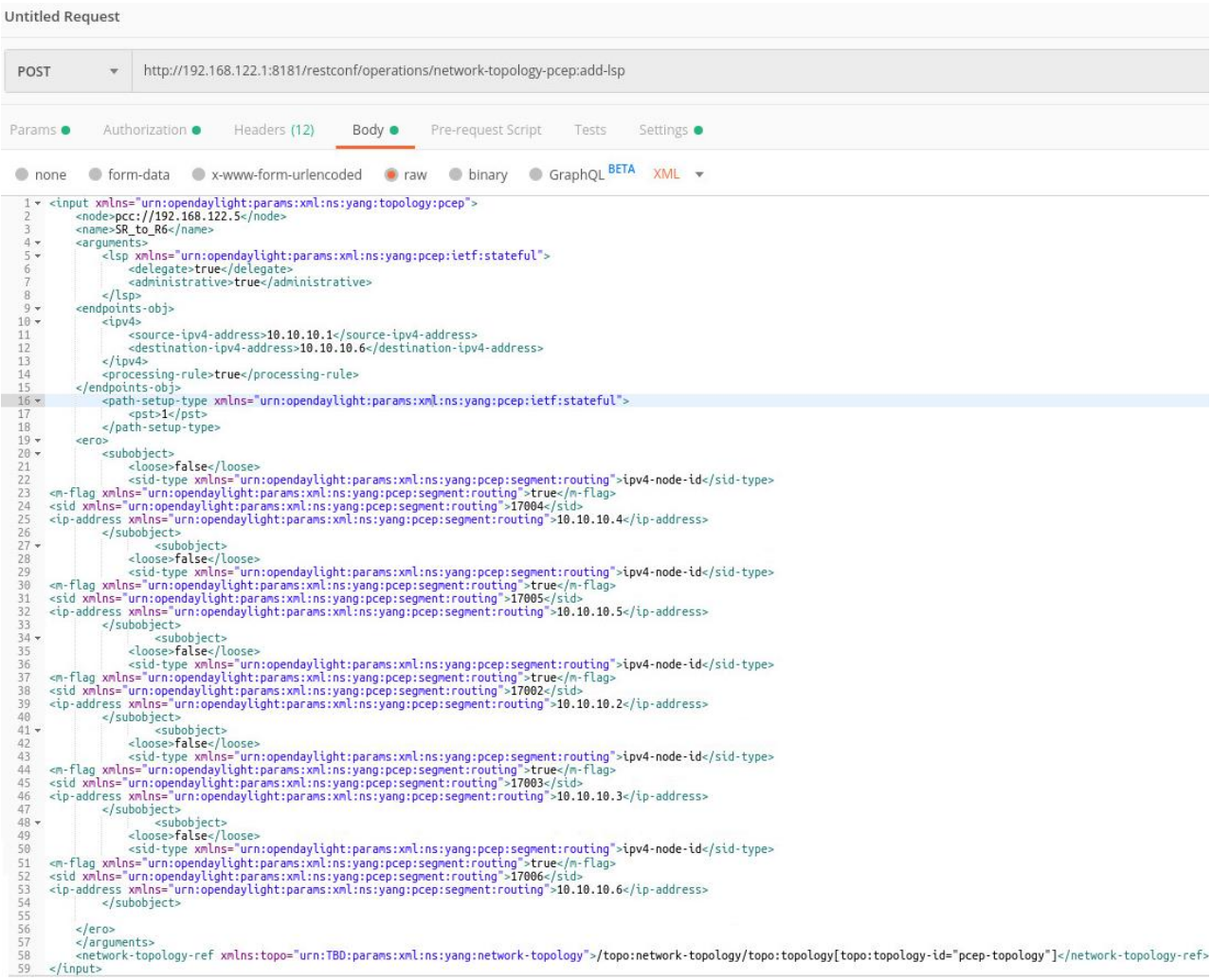

Figure 72 – Add LSP request

Figure 72 illustrates request, which establishes SR-TE tunnel from router R1 to R6 via nodes R4, R5, R2, R3 and R3.

- Line 2 specifies the ip address of the router;

- Lines 11 and 12 determine the sender and receiver nodes in this tunnel correspondingly;
- Value “1” in line 17 determines affiliation of this path to the Segment Routing technology
- Value “true” in line 6 means, that the control over this tunnel is delegated to the ODL controller;
- Values in lines 23, 30, 37, 44 and 51 indicate that in this tunnel SID labels will be used as MPLS labels;
- Values in lines 24, 31, 38, 45 and 52 determine the next hops in the path.

All this information is transmitted to the Router 1 in one PCEP packet. It is shown on figure 73. I would like to pay attention here at an Explicit Route Object, which specifies all the nodes in the path with SID line. Therefore there can be seen a flag, which instructs network devices to treat SID values as an MPLS labels.

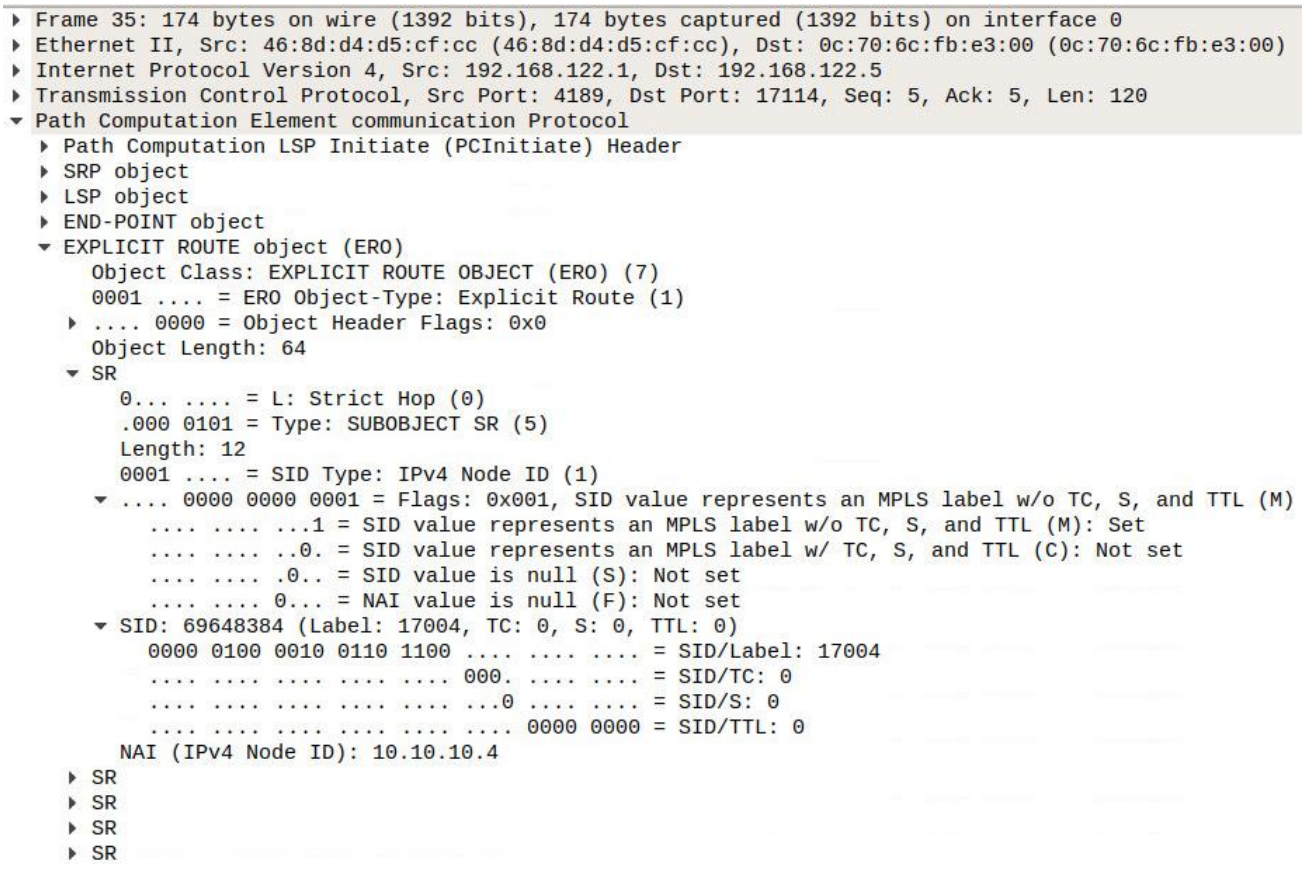

Figure 73 – PCEP packet

On figures 74 and 75 we can see an output of the commands “*show mpls traffic-eng tunnels segment-routing*” and “*show mpls traffic-eng tunnels segment-routing brief*”. First figure contains more detailed information about TE tunnel. Here I would like to pay attention to PCC section, where the originator of the tunnel (192.168.122.1, ODL controller`s address) can be found. The

second interesting section for us is a discription of Segment-Routing Path, where all nodes and its SIDs are mentioned.

```
RP/0/0/CPU0:ios#show mpls traffic-eng tunnels segment-routing
Fri Jan 31 11:25:21.870 UTC


Name: tunnel-te13  Destination: 10.10.10.6  Ifhandle:0x880 (auto-tunnel pcc)
  Signalled-Name: SR1_to_R6
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    path option 10, (Segment-Routing) type explicit (autopcc_te13) (Basis for Setup)
      Protected-by PO index: none
    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Fri Jan 31 11:09:13 2020 (00:16:09 ago)
  Config Parameters:
    Bandwidth:        0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (default)
    Path Selection:
      Tiebreaker: Min-fill (default)
      Protection: any (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (default)
    Forwarding-Adjacency: disabled
    Loadshare:          0 equal loadshares
    Auto-bw: disabled
    Path Protection: Not Enabled
    BFD Fast Detection: Disabled
    Reoptimization after affinity failure: Enabled
    SRLG discovery: Disabled
  Auto PCC:
    Symbolic name: SR1_to_R6
    PCEP ID: 14
    Delegated to: 192.168.122.1
    Created by: 192.168.122.1
  PCE Delegation:
    Symbolic name: SR1_to_R6
    PCEP ID: 14
    Delegated to: 192.168.122.1
  History:
    Tunnel has been up for: 00:16:09 (since Fri Jan 31 11:09:13 UTC 2020)
    Current LSP:
      Uptime: 00:16:08 (since Fri Jan 31 11:09:14 UTC 2020)
    Prior LSP:
      ID: 2 Path Option: 10
      Removal Trigger: reoptimization completed

  Segment-Routing Path Info (PCE controlled)
    Segment0[Node]: 10.10.10.4, Label: 17004
    Segment1[Node]: 10.10.10.5, Label: 17005
    Segment2[Node]: 10.10.10.2, Label: 17002
    Segment3[Node]: 10.10.10.3, Label: 17003
    Segment4[Node]: 10.10.10.6, Label: 17006
```

Figure 74 – Tunnel`s information on Router 1

```
RP/0/0/CPU0:ios#show mpls traffic-eng tunnels segment-routing brief
Fri Jan 31 11:27:27.312 UTC

                  TUNNEL NAME         DESTINATION      STATUS  STATE
                 >tunnel-te13          10.10.10.6          up  up
Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Figure 75 – Tunnel`s brief information on Router 1

On figure 75, the brief information about the tunnel can be found. Here I would like to point out ">" sigh, which means that this Tunnel was created outside the Router 1.

If there is a need to remove any tunnel, "*remove-lsp"* request can be used. It should be sent to the "*http://192.168.122.1/restconf/operations/network-topology-pcep*" controller`s address. The only information needed for this request is the Router`s address and the name of tunnel. Request parameters are the same as in case of "*add-lsp*".

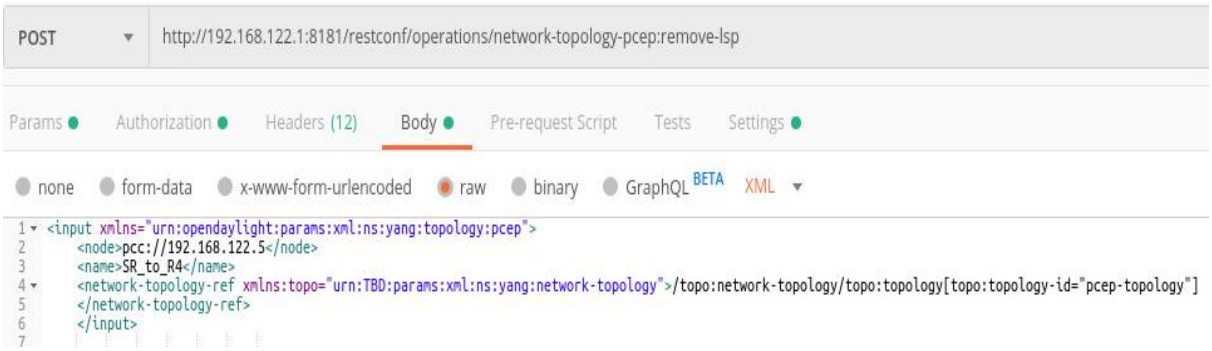

Figure 76 – "*remove-lsp*" request

```
RP/0/0/CPU0:ios(config)#do traceroute 10.10.10.6
Fri Jan 31 11:18:39.258 UTC

Type escape sequence to abort.
Tracing the route to 10.10.10.6

 1  192.168.3.2 [MPLS: Labels 17005/17002/17003/17006 Exp 0] 259 msec  109 msec
 59 msec
 2  192.168.4.2 [MPLS: Labels 17002/17003/17006 Exp 0] 69 msec  59 msec  59 msec

 3  192.168.5.1 [MPLS: Labels 17003/17006 Exp 0] 59 msec  79 msec  89 msec
 4  192.168.2.2 [MPLS: Label 17006 Exp 0] 119 msec  59 msec  79 msec
 5  192.168.7.2 59 msec  *  59 msec
RP/0/0/CPU0:ios(config)#
```

Figure 77 – Traceroute from R1 to R6

On the output of the traceroute from R1 to R6 SID`s usage process is shown. On the first hop, the full label stack can be observed. Then it is decremented by one value until the destination device.

Figure 78 shows the results of packet capturing process on the link between R1 and R4. Here we can see an ICMP echo request. Labels, used for transmission of this packets, are same, as shown on figure 77.

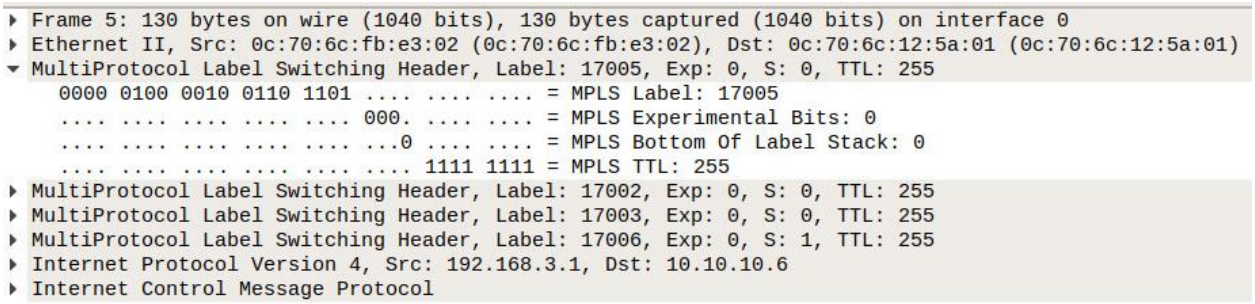

Figure 78 – Ping from R1 to R6

Unfortunately, due to the fact that for this topology free version of the router`s IOS was selected, only limited functionality was implemented, and many functions, which had been mentioned in the theoretical part, were not shown.

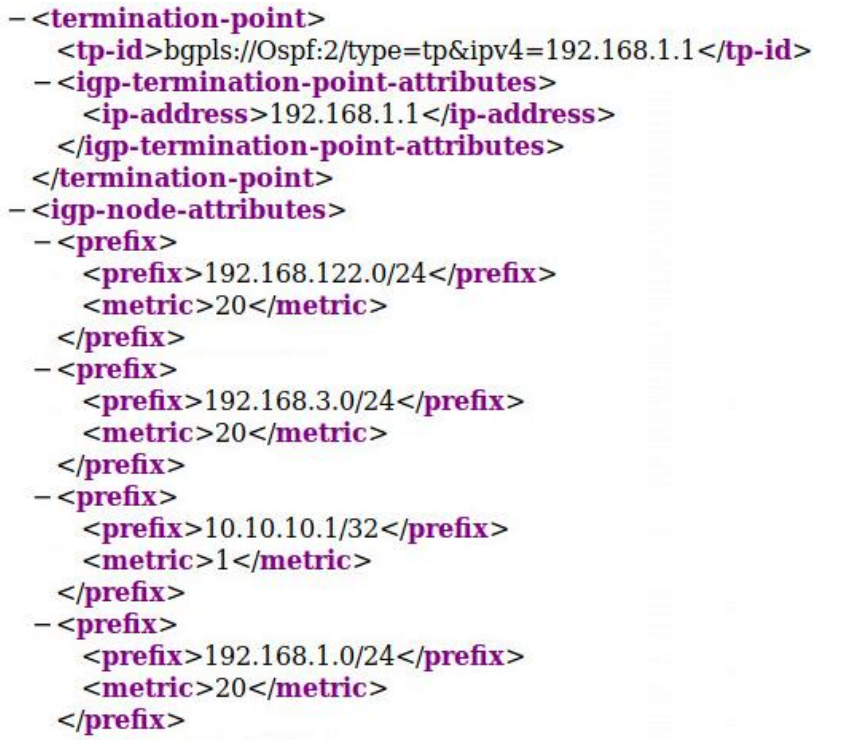

Figure 79 – Router`s attributes

Besides the limitations provided by the Cisco IOS XR, Opendaylight controller also has some interesting functions, which are not able to be implemented in case of Segment Routing. One of such limitations is a topology visualization in a web interface. Unfortunately, Opendaylight does

not support this function in case of BGP-LS. However, controller still has all the information about the topology. It can be reached on page "*192.168.122.1:8181/restconf/operational/network-topology:network-topology/*". Part of the content of this page, characterizing Router`s 1 IGP attributes is shown on figure 79.

4 Conclusion

In the final chapter of my Capstone Project, I would like to summarize the information, analyzed in previous parts, and draw some conclusions about the comparison of MPLS-TE and Segment Routing-TE.

Initially, in chapters 1 and 2, the theoretical aspects of these technologies were discussed, including the following topics:

- Use cases;
- Protocols, which are used as a control plane;
- Extensions of control plane protocols;
- Protocol`s message formats;
- Parameters and features, which can be configured in order to improve technology`s performance

Table 30 shows the results of the comparison of MPLS-TE and SR-TE.

Table 30 – MPLS-TE and SR-TE comparison

Parameter	**MPLS-TE**	**Segment Routing**
Control Plane protocols	RSVP-TE, IGP	IGP
Label Distribution	Each tunnel requires its own label allocation, so with an increase in the number of tunnels, the number of allocated labels is also increased.	In Segment Routing, labels are allocated not to tunnels, but to nodes, prefixes or adjacencies. Each tunnel does not require its own label block.
Path Control	In case of changes in the path occur, it should be delivered to each node and these nodes have to perform its own path`s control processes.	In the case of path`s changes, only ingress node performs recalculation processes.
Scalability	In MPLS-TE every tunnel`s status has to be monitored on each node. Any changes in the tunnel`s environment have to	SR-TE does not require maintenance procedures for the tunnel, because its information is transmitted in packets with data.

	be processed on each device in the path. Poor scalability.	So, it has better scalability than MPLS-TE.
Parameter	**MPLS-TE**	**Segment Routing**
Evolution to SDN	Evolution to SDN requires the implementation of the changes in the whole network.	Evolution to SDN in SR-TE needs configuration changes only on the ingress node because Segment Routing is based on a "source routing" concept

In the third part of the capstone project, I implemented three different topologies in Graphical Network Simulator-3 in order to check in practice theoretical aspects, which have been learned before. During the simulation of RSVP-TE, SR-TE and Opendaylight SR-TE I observed the following behavior:

- In the case of RSVP-TE, analysis of the traffic on links between routers showed the presence of two protocol`s control plane messages:
  - OSPF;
  - RSPV;
- In the case of SR-TE with ODL controller and without it, only OSPF protocol`s control plane messages were found on the same links. As a result, the difference in 216 bytes for 20 seconds interval (1316 bytes for SR-TE and 1532 for MPLS-TE topologies) was observed due to four transmitted RSVP Hello messages.
- Experiments with the topology convergence process also revealed the benefits of the usage of SR-TE. In the case of MPLS-TE, again RSVP and OSPF control plane messages were exchanged. In the case of SR-TE, only OSPF Hello messages were observed. It is important to say, that in such situation an additional overhead in these OSPF messages was observed, because of OSPFv2 Extended Link Opaque LSA (LSA Type 10), which carries SR-TE related information (ADJ-SID, TLV`s types and other). However, despite this fact, overhead in SR-TE is still smaller per convergence process (578 bytes instead of 978).

Analyzes, discussed above, confirm the advantages of Segment Routing-TE over MPLS-TE in terms of link and bandwidth occupation during the work process.

Despite the advantages of Segment Routing. Described above, there are some limitations, which make the complete transition of telecommunication companies to this technology impossible now:

- SR-TE is supported on a limited number of networking devices;
- Devices, which support SR-TE, mostly cost more, than devices with MPLS-TE support;
- Many features, which are supported in MPLS-TE (bandwidth reservation, FRR, etc.) are not yet supported on most of SR-TE devices;
- SR-TE testing has only recently been started in provider`s networks;
- Poorly standardized work in IPv6 topologies.

In conclusion, I would like to say, that due to the facts I have analyzed in this work, in my opinion, Segment Routing is a very perspective technology, which is able to simplify networks and satisfy daily increasing customer requirements.

List of literature

1) https://search.library.ualberta.ca/catalog/8660771 Traffic engineering with MPLS

2) https://en.wikipedia.org/wiki/Multiprotocol_Label_Switching

3)https://www.juniper.net/documentation/en_US/junose15.1/topics/concept/mpls-label-distribution-protocols.html

4) https://www.networkworld.com/article/2350577/understanding-mpls-label-stacking.html

5) IETF RFC 3031

6) Quality of Service Control in High-Speed Networks H. Jonathan Chao, Xiaolei Guo

7) https://www.iana.org/assignments/mpls-label-values/mpls-label-values.xhtml

8) https://www.juniper.net/documentation/en_US/junos/topics/concept/mpls-special-labels.html - special mpls labels

9) IETF RFC 6790

10) IETF RFC 3032

11) IETF RFC 3063

12) https://search.library.ualberta.ca/catalog/8660294

13) https://search.library.ualberta.ca/catalog/4258007 MPLS [electronic resource] : next steps Security

14) IETF RFC 2205

15) Quality of Service in a Cisco Networking Environment - Gilbert Held

16) TCP/IP Tutorial and Technical Overview - T.Britt

17) https://support.huawei.com/enterprise/en/doc/EDOC1100055048/bf2d383e/sr-te

18) https://www.cisco.com/c/en/us/td/docs/routers/ncs6000/software/segment-routing/configuration/guide/b-segment-routing-cg-ncs6k/b-segment-routing-cg-ncs6k_chapter_0111.pdf

19) https://tools.ietf.org/html/draft-filsfils-spring-segment-routing-policy-01#ref-I-D.previdi-idr-segment-routing-te-policy

20) https://support.huawei.com/enterprise/en/doc/EDOC1100092117

21) https://www.fgts.ru/collection/opendaylight

22)https://www.theseus.fi/bitstream/handle/10024/106477/SDN_MAKSIM_SISOV.pdf?sequence=1

23) https://support.huawei.com/enterprise/en/doc/EDOC1100055120/32984202/bgp-ls 24) https://www.juniper.net/documentation/en_US/junos/topics/concept/mpls-pcep-overview.html

25) IETF RFC 8664

26) IETF RFC 8281

27) https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/seg_routing/configuration/xe-3s/segrt-xe-3s-book/intro-seg-routing.html

28) https://www.ietfjournal.org/segment-routing-cutting-through-the-hype-and-finding-the-ietfs-innovative-nugget-of-gold/

29) https://support.huawei.com/enterprise/en/doc/EDOC1100028536?section =j003&topicName =segment-routing-mpls-overview

30) IETF RFC 7752

31) IETF RFC 8665

32) Mpls confiugration on cisco ios - Umesh Lakshman, Lancy Lobo, Cisco press

33) Real-world designs of converged mpls networks - Jean-Phillipe Vasseur, Jim Guichard, Francois Le Faucheur, Cisco Press

34) Practical BGP - White, Russ, Boston: Addison-Wesley

35) https://geeks-world.github.io/articles/466549/index.html