A Traffic Engineering Approach to Differentiated Multicast Services over MPLS Networks

By

Toni Barabas

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
School of Information Technology and Engineering
University of Ottawa
January 2012

Abstract

Currently, a viable solution to provide multicast provision over a multiprotocol label switch with traffic engineering (MPLS-TE) domain is unavailable because of the missing link able to couple multicast traffic distribution with an MPLS-TE enabled network. This is due to the limited or less research investigation that was done in this area. Most of the investigation methods tackle the problem individually such as deploying internet protocol (IP) multicast in a plain network or MPLS domain but without considering a combination of both technologies that is aware of differentiated services requirements.

This thesis presents an alternative solution for the multicast differentiated services provision problem over a MPLS-TE enabled network. The approach is exposed and analyzed through a practical solution that was developed within a network simulation environment.

The research presented in this thesis orchestrates the already available technologies offered by the multicast protocols suite and differentiated services (DiffServ) aware MPLS-TE that allows applying separately the constraint-based routing and admission control to different classes of services. The novelty and solution of this thesis relies on using MPLS constraint-based routing concepts (e.g.traffic trunks) in order to solve TE issues revealed during multicast traffic distribution.

Acknowledgments

I am heartily thankful to my supervisor, Dr. Dan Ionescu, whose encouragement, guidance and patience from the inception to the final chapter enabled me to develop an understanding of the subject.

I also wish to extend my appreciation for the supportive advice offered by my advisors and friends Stejarel Veres and Laurentiu Checiu.

I would like to thank my family for their support and understanding.

List of figures

List of tables

Glossary of terms

MPLS	Multiprotocol label switch protocol
LFIB	Label Forwarding Information Base
NHLFE	Next Hop Label Forwarding Entry
PIM SM	Protocol Independent Multicast Protocol Sparse Mode
PIM DM	Protocol Independent Multicast Protocol Dense Mode
SJP	Spanning join protocol
QoSMIC	QoS sensitive multicast internet protocol
QMRP	QoS-aware multicast protocol
DSCP	Differentiated Services Code Point
RED	Random Early Detection
RTT	Round-Trip Time
OSI	Open Systems Interconnection model
CBR	Constraint based routing
TE	Traffic Engineering
DS-TE	DiffServ aware Traffic Engineering
FSM	Finite State Machine

1 Problem definition

IP multicast packet transmission undergoes several mapping transformations when the packet flow traverses Quality of Services (QoS) based networks such as Traffic Engineering (TE) MPLS networks followed or in combination with plain IP networks. The way in which these networks map and distribute IP multicast traffic at the entrance into an MPLS-TE in which a signaling protocol such as RSVP-TE or CR-LDP is enabled raises a series of issues which will be addressed and solved in this thesis. It should be noted that the QoS are defined by service providers through one of the available IP QoS models.

1.1 Introduction

Internet Service Providers (ISP) are interested in the addition of multicast traffic distribution streams capabilities to their traffic engineered MPLS network. This would provide superior IP multicast transmission which would be enhanced with QoS awareness. This QoS is required by multimedia provision on-demands businesses.

Distribution of the IP multicast packets relies on the dynamic and volatile nature of the multicast distribution tree within a plain IP network. Packets with associated labels use a medium based on the MPLS label switch paths (LSP) to relay them within a MPLS TE domain.

Routing multicast traffic within any constraint-based system, combined with QoS support, exposes issues. These issues arise from the fact that the multicast distribution trees must be mapped towards the entrance of the LSPs deployed within an MPLS TE domain enabled with an IP QoS model.

1.2 Necessities for deploying multicast routing and TE

Multicast routing is the process derived of a suite of protocols to couple the one–to-one unicast packets delivery operation and the one-to-all broadcast operation.
There are several ways to accomplish this delivery operation which is based on the communication type established between a computer or host and the number of recipients' nodes from a computer network:

- *unicast* communications, in this case data is sent to one particular recipient;
- *broadcast* communications, where data is sent to all hosts/computers of a given network;
- *multicast* communications, whose recipients represents a group of hosts/computers that have expressed interest in receiving data;
- *anycast* communications, aims to send data between a single sender and the nearest several receivers in a group.

In a multicast domain environment, *group* represents a set of entities (machines, procedures, application entities, users) that takes part in the same communication. It will be used to refer to a group of receivers, senders or both.

From a larger perspective, packets of data inside an IP networks are exchanged by establishing either a point-to-point (aka P2P) (one sender and one recipient), a point-to-multipoint (aka P2MP) (one sender and *n* recipients) or a multipoint-to-multipoint (known as MP2MP) (*m* senders and *n* recipients) type of communication.

IP traffic is generated by transmitting packets within an IP based network between senders and receivers through one of the previously mentioned type of communication. This traffic must be coordinated to ensure safe packet delivery to the designated destinations. ISP must also apply QoS policies to the traffic and network resources in order to avoid congestion or transmission failures. This ensures that a specific level of agreement with the client's receiver is met.

ISPs are continuously searching for methods to enhance their existent network capacity, to improve IP traffic transmission, and to provide more value to their services by adding QoS which would address more client requirements

Firstly, ISPs are searching for a way to add more logical improvements or TE application solutions. One of the proposed solutions would be to use TE or constraint-based routing. This method is in contrast to the plain IP routing which does not take into account constraints that are applied to the network resources. This is due to the fact that with a plain IP routing the information is available only at the individual resource level and it is not distributed throughout the network.

In IP based networks there are two main directions to ensure TE but MPLS TE is more widely adopted. This solution is desired since the other approaches attempt to adjust the metrics of the Interior Gateway Routing Protocol (IGP). These methods have proved themselves to be less efficient for a systematic network-wide TE [2] since they may only improve the efficiency on some branches of the network while leaving other branches mostly unused.

The most relevant advantage to using a MPLS – TE solution is based upon the basic feature of the MPLS. This feature separates the data plane (also known as the forwarding plane) from the control plane, allowing the routing decisions to be made by taking into consideration, for example, the available link's bandwidth.

Prior to the development of the MPLS, there was a strong dependency between the routing and forwarding mechanisms. The new capability to assign addresses offered by the IPv6 caused significant changes to be made to the routing area. These changes accommodated the aggregation process of addresses and routing information [6]. Since the address prefixes may be different lengths, the forwarding algorithm was modified. . Therefore, most of the forwarding algorithms are implemented into the router's fabric, or

deployed as software firmware which cannot be updated without temporarily stopping their functionality.

The architectural evolvement of the MPLS decomposed the network layer routing into control and forwarding modules. This allows further development of new or improved control modules for label switching process which could improve routing without altering the forwarding algorithm.

The second aspect, which could result in cost savings from the ISP's perspective, is to improve traffic transmissions by attempting to distribute IP multicast traffic within this MPLS TE domain. Thus, multiple copies of the same source content could be distributed over an enhanced MPLS-TE network.

From the multicast perspective the label concept is important because once a packet has been assigned a FEC in a MPLS forwarding paradigm, there is no further analyses of the header at the subsequent routers from the MPLS domain. All the forwarding is driven by labels [12]. Furthermore, a label switching device (more specific in our case an MPLS router) allows a FEC, which is represented by a label, to be associated with a source/destination address pair [6]. This association grants an identity to a packet entering into an ingress router and allows the identity of the packet to travel along with the packet. The identity transfer between nodes was not possible with the conventional forwarding. This packet's identity knowledge is taken further by an MPLS device. Also, by maintaining a label switching table on a per interface basis, it can associate packet's dependency to a specific multicast distribution tree.

The third feature valuable to ISP services which enhance the IP traffic with QoS plus making the MPLS-TE network aware by one of the IP QoS model.
MPLS-TE may be combined with QoS support (i.e extensions to resource reservation protocol RSVP) in order to provide a guaranteed bandwidth LSP or an LSP with reserved

resources [6]. Although the resources along an LSP are reserved with the aid of a signaling protocol, the plain MPLS – TE remains unaware of the IP QoS model. Within this document the MPLS-TE is made aware of a Diff-Serv IP QoS model which results in the MPLS-TE enhanced domain to be named as a DS-TE MPLS domain.

This default IP QoS unawareness of the MPLS-TE offers a better flexibility towards one of the applicable IP QoS model but it makes it more difficult to adjust the traffic for each one of the IP QoS model. It is also difficult to map the incoming multicast IP traffic at the entrance within this MPLS-TE domain.

It should be mentioned that the IP QoS enablement is added as an extra feature to an IP multicast distributing environment. By the volatile and dynamic nature of the multicast distribution tree, transitional states within the multicast enabled routers are created. Alternatively, with MPLS-TE, or any constraint-based routing system, constrained shortest path first protocol (CSPF) must be activated. The CSPF must collaborate with one of the signaling protocols (i.e resource reservation protocol (RSVP) and constraint-based routing label protocol (CR-LDP)) that has to be extended as in [24]. Extensions of these protocols are required in order for them to be compatible with the multicast subscription/cancellation procedures dictated by the behavior of the terminal branches of the multicast distribution trees.

In conclusion MPLS adds support to the IP QoS model rather than extending it [6]. Therefore, for this reason MPLS cannot be considered an end-to-end protocol or an entire substitution candidate for a complete IP QoS provision.

Even with these partial constraints, MPLS offers several novel capabilities which are relevant to ISPs adopting this technology. The IP Qos services is more efficiently deployed and extended over a wider and heterogeneous platform e.g. ATM LSRs due to the addition of MPLS.

1.3 Motivations

With an increase in streaming demand by multiple clients, a multicast over an MPLS environment deployment is beneficial from the following perspectives:

- *Cost reduction by delivering same content to multiple receivers*. Currently a large numbers of users can watch the same video feed hosted on servers due to the existence of numerous unicast links established between those servers and each individual client's hosts. For applications requiring many-to-many connectivity such as gaming, gambling or entertainment, multiple point-to-point meshes of links are demanded to be in place [3]. As a result, the reduction of these physical hardware links becomes a vital concern for most ISPs. As a first step towards IP multicasting, they reduce the network load by transmitting only the minimum of packet copies from the source towards different recipients [4] and decentralize the recipient management. Therefore, the burden of traffic is moved towards the routers, since multicast uses the routers, not servers, to replicate packets [3].
- *Looping* is a side effect of adopting the multicast protocol because of its underlying unicast protocol. Each time a multicast packet goes around a loop, copies of the packet may be emitted if branches exist in the loop [5]. Therefore these copies lead to a large amount of unwanted traffic until the loop is broken. Within a MPLS environment, loop detection is a configurable option and handled within the LDP protocol [5]. Loop prevention has been developed later as part of the MPLS standardization [6]. It is based on the *colored threads* concept.
- *Performance* is one of the most important features of the MPLS over multicast and should be considered. A more logical solution is required for current network cost reductions since the network infrastructure is present and modifying it at this time might reveal additional costs. More enterprises deploying IP multicast networks [49] and ISPs are researching this type of solutions for at least two reasons. Firstly, it would allow them to aggregate more traffic over the same network. Secondly, the application of traffic engineering techniques could allow them to temporarily host new clients until future network deployment is

developed.

- *Enable IP multicast on ATM-LSRs* [6] by using the label switching forwarding mechanism offered by MPLS. Through this service, the cumbersome methods of mappings from IP to ATM multicast will be avoided.

In order to address these requirements, several IETF working groups from routing area are continuously involved in the proposition of drafts documents which will be published later as RFC standards [7].

It should be mentioned that the following active working groups may have an interest in this research:

- OSPF working group develops and documents extensions and bug fixes to the OSPF protocol. They also document OSPF usage scenarios [8].
- PIM working group was focused on the standardization of the Protocol Independent Multicast and recently is chartered to standardize extensions to the RFC 4601 which will become PIMv2 Sparse Mode [9].
- The MPLS working group charter is responsible for standardizing a base technology, which includes procedures and protocols, for the distribution of labels between routers and encapsulation. This group can be found on the IETF's Web site [10] .

The multicast protocol is the basis for this research and was developed within the historical Network Work Group.

1.4 Contributions and organization of the thesis

This thesis proposes an alternative deployment approach for the multicast traffic distribution over an MPLS underlying network capable of delivering DS-TE services.

The current thesis advances the following research contributions:

- Introduces a new combination of the PIM-SSM multicast suite protocol that could be used to enable multicast packets distribution within DS-TE MPLS
- Proposes an algorithm capable of deploying the DS-TE MPLS configuration and forwarding the multicast traffic through this MPLS domain
- Designs and simulates a testbed network to support the presented concepts by configuring the multicast, QoS, and MPLS modules within a network simulator environment (OPNET).

Chapter 2 is a literature review of the current multicast protocols, of the IP QoS model services provision within a multicast and MPLS networks. It analyzes the current informational RFC draft proposal of the IP multicast within a MPLS environment.

Chapter 3 presents the high level architecture components for the proposed deployment of a multicast within a Diffserv-aware TE MPLS network environment.

Chapter 4 presents, through FSM charts, an algorithm and architecture of the solution devised to deploy multicast traffic over a DS-TE MPLS domain. This solution is supported by a testbed network design capable to test, including protocols and algorithms, the multicast packets distribution within a Diffserv-aware TE MPLS network.

Chapter 5 presents the results acquired during a simulated video multicast broadcast traffic within the deployed testbed network. The traffic is simulated within a OPNET simulation environment.

Chapter 6 draws the conclusions and presents possible future work with regards to the current approach for multicast distribution within a DiffServ aware TE MPLS.

2 Existing Multicast over MPLS approaches

A review of current developments, research papers and studies pertaining to the current multicast protocols are presented in the following sections of this chapter. An in depth analysis of multicast routing protocols, MPLS and IP QoS models are also presented below. As a general conclusion several suggestions and framework were proposed by academia or commercial providers to combine multicast distribution within MPLS-TE. These propositions did not take into consideration the IP QoS model, and as a result none of the suggestions are currently recognized as a general deployable solution.

2.1 Introduction

Many different approaches have been studied by researchers and commercial providers to acquire a reliable solution to provide QoS for IP multicast within a MPLS domain. The complexity of the problem arises from several taxonomy raised from both multicast and MPLS.

2.2 Multicast components

Figure 1-1, illustrated below, is a general view of several terms commonly used in an IP multicast network. This figure will facilitate the identification of the multicast protocol elements that will be discussed throughout the remainder of the document.

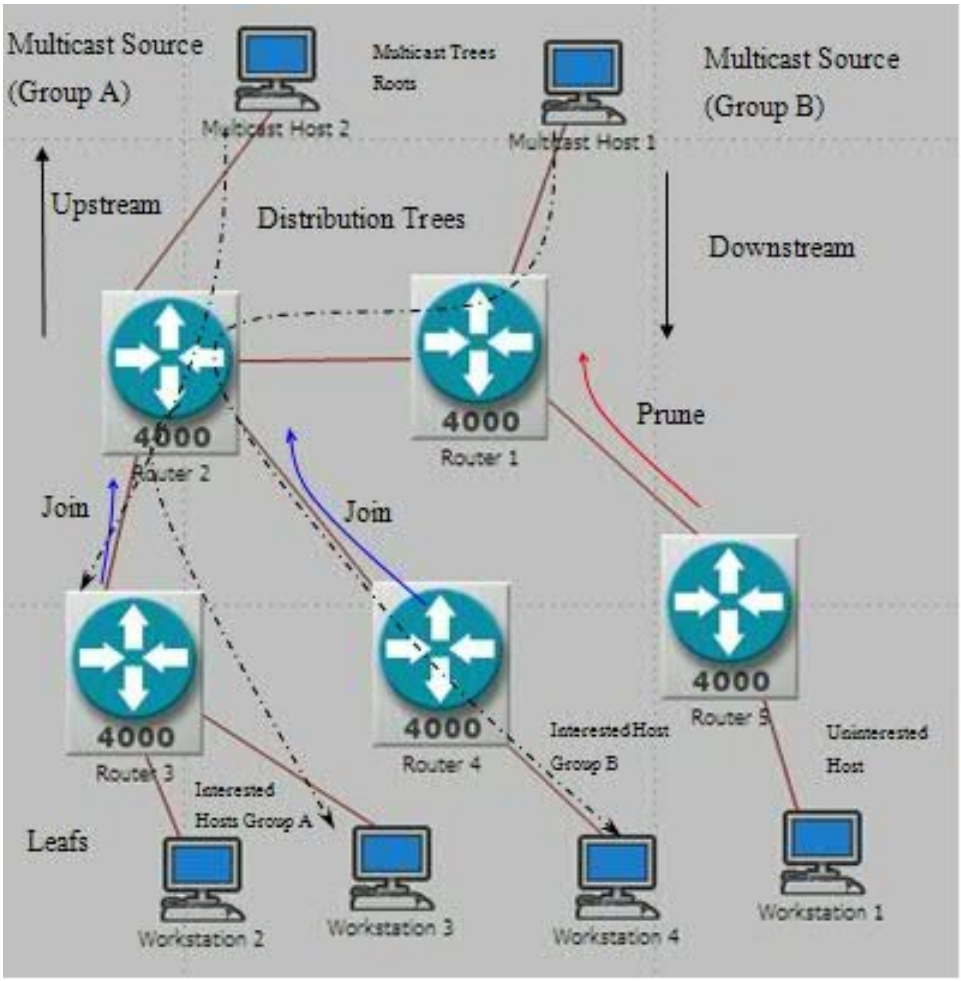


Figure 2-1: Examples of multicast terminology showing multicast distribution trees and Join and Prune messages sent using IGMP protocol from receivers towards multicast enabled routers

Multicast-capable routers, routers that take part in a multicast operation, must be enabled for this operation. These routers in the IP multicast network use a *multicast protocol* to build a *distribution tree*. These trees distribute packets from the *multicast hosting servers* towards the receivers represented through workstations 1 to 4.

The interface on the router leading towards the receiver is the *downstream interface.* This

interface is also commonly referred to in circulation as an *outgoing* or *outbound* interface. The interface on the router leading towards the source is called the *upstream interface,* but is also commonly referred to as the *incoming* or *inbound i*nterface.

Each host, or a possible subnetwork containing several hosts, that are interested in being a receiver of the sending multicast group is a *leaf* on the *distribution tree.* When a new leaf (host or subnetwork) signals its interest in listening to a multicast transmission, a JOIN message is sent *upstream* towards the router hosting the distribution tree and a new *branch* is built.

When a branch contains uninterested receivers (hosts) in a multicast transmission, the branch is *pruned* from the distribution tree and no multicast packets are sent out on that particular outgoing interface.

A *Host Group* contains all the hosts belonging to a multicast session, usually originating from the same multicast source. A host may be a member of more than one group at a time. In an IP multicast network, traffic is delivered to the multicast groups based on an IP multicast (or group) address.

RPF checks is an important procedure executed over a multicast packet by every router participating in a multicast distribution. When a multicast packet arrives at the entrance of a multicast router this procedure interprets the source address of the multicast source from the packet as a destination address of a unicast packet. The destination address is used to query the unicast forwarding table to determine the outgoing interface. It is theorized that the packet passes the RPF checks if the outgoing interface is identical with the incoming interface through which the multicast packet has arrived. Consequently, the multicast packet must arrive on the same interface that leads to the shortest path toward the source of a multicast traffic. If the packet does not arrive on the shortest path then the packets that fail this RPF checks are automatically dropped by the multicast router.

2.2.1 Source/Shared Trees

During the multicast session different states are established inside the routers which are dependent of the status of the incoming or outgoing interfaces. A multicast router in a forwarding state is defined as *turned on* for a specific group content. In this forwarding state copies of the multicast group packets received on the incoming interface, specific for that group, are sent by the outgoing interfaces listed in the router's outgoing interface list (OIL).

The multicast forwarding state in a router is determined by the types of distribution trees build within the network. There are two main categories of trees: the source trees which are known to be noted as (S,G) and the shared trees noted as (*,G).
In (S,G) the S notation refers to the unicast IP address of the source for the multicast traffic, while the G refers to the multicast group IP address for which S is the source. The (*,G) notation refers to the situation where the source of a multicast group is unknown due to multiple available sources. Figure 2-2 is a visual representation of a Source Tree and a Shared Tree.

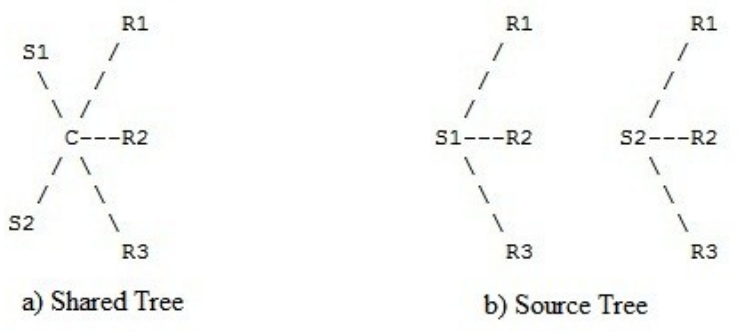


Figure 2-2: a) Shared tree one tree per (S)ource and per multicast (G)roup **b) Source tree** one tree per multicast group [5]

Given the existence of these two types of trees, the number of possible states to be stored in the router resulted in the construction of a couple of different trees. These resultant trees are frequently revealed in literature, primarily due to the tendency for the reduction

in transmission delay or the optimization trials with regards to the involved resources.

2.2.1.1 Shortest Path Tree (SPT)

The intention of a shortest path tree is to construct a tree where the transmission delays are minimal. Each leaf of the tree is connected towards the source by using the shortest path defined by the underlying unicast routing.

2.2.1.2 Centered tree (CBT)

A centered tree is a shared tree which is required to have a delegated and known Rendezvous Point (RP) or center node (in this case a router). The construction of the tree is built similarly as the SPT with the exception that the leaves are connected following the shortest path toward the RP point by again using the underlying unicast routing.

2.2.1.3 Steiner tree

The Steiner Tree is another type of shared tree that establishes an optimal solution to interconnect a given number of N nodes (e.g. routers) members of a group through a minimum cost of used resources.
The construction of the Steiner tree is a common problem in the graph theory as a NP-complete problem. Currently there is no multicast routing protocol able to maintain such an optimal tree [5].

2.2.2 Evaluation of the multicast trees

The following conclusion can be noted based on the characteristics of the trees and on the simulation results from the Internet world. Firstly, Shortest Path Trees are frequently used because of their short delays and is the simplest to compute. Secondly, Steiner trees are mostly present theoretically and currently none of the multicast protocols implements this

type of tree due to the computation complexity. Finally, the CBT trees appear to be an intermediate solution between the aforementioned trees. This type of tree has good delays if the RP is well positioned and the use of resources is more reduced compared to the SPT trees where groups have multiple sources. Thus it is concluded that a network may support higher bandwidth multicast groups if SPT trees are used instead of center based trees [11].

2.2.3 Multicasting routing protocols

The multicast routing resumes at the process of building multicast forwarding tables where visually the multicast distribution trees are determined.

2.2.3.1 DVMRP

DVMRP [13] is a dense-mode protocol that builds a source based (S,G) distribution tree through a flood –and –prune technique or implicit join method. This method delivers traffic throughout the tree and then determines the location of any uninterested receivers.

2.2.3.2 Multicast Open Shortest Path First

MOSPF is a multicast extension of the OSPF protocol but only for dense mode. It builds a source based distribution tree (S,G) through an explicit JOIN message. As a result routers do not have to flood their entire domain with multicast traffic from every source [3]

2.2.3.3 PIM Dense Mode

PIM-DM is independent of the unicast underlying protocol. The same limitations as a DVMRP protocol are present. These include periodic broadcast (requires to rebuild the tree periodically), reliability (its integrated unicast protocol is a “distance vector”

protocol that has convergence problems), and additional cost (it operates his own unicast "distance vector" type which is different of the underlying unicast protocol) [4].

2.2.3.4 PIM Sparse Mode

PIM-SM allows a router to use any unicast routing protocol. An administrator sets a router as an RP for the initial source of a multicast group during the configuration of the routers. The receivers signal their interest to become part of a multicast group the routers and the routers send an explicit JOIN messages upstream to their neighbours. This results in the construction of shared distribution trees (*,G) from receivers to RP.

2.2.3.5 Core-Based Trees

A CBT shares all the characteristics of the PIM-SM but discovers the sources more efficiently. It should be noted that this type of tree does not have a large success outside of academia discussions or experimental RFC 2201 [3].

2.2.4 Conclusion of Multicast Protocols

Five multicast protocols were presented in the above sections. These five protocols are the most commonly used protocols or are partially mentioned in academia. Two of the protocols, specifically the PIM and CBT, usually determine a single path between the tree and a new member. The largest disadvantage of the existing multicast routing protocols is that they are not scalable enough for large scale networks and most importantly, they do not guarantee any QoS. The issue of QoS has been revealed due to the fact that most of these protocols are based on or integrate an older version of an unicast routing protocol that is not capable of QoS. In order to provide a solution which has a minimal end-to-end delay, limited jitter, and efficient use of bandwidth, an important condition for deploying multimedia applications, several new multicasts were suggested. These newly suggested

multicasts include spanning join protocol (SJP) [14], QoS sensitive multicast Internet protocol (QoSMIC) [15] and QoS-aware multicast routing protocol (QMRP) [16]. These protocols are known to be multiple path routing protocols which determine several paths towards the multicast tree. This results in a higher probability of discovering a path in the multicast tree that meets the QoS requirement [4]. This higher probability produces a higher network overload that is introduced in order to accommodate the multipath resource reservation.

In order to overcome network overload and in the same time to ensure a "perfect resource utilization and QoS guarantees" a series of theories were proposed in [52-56]. These studies proposes and analyses through several simulations the "Recursive Fair Stochastic Matrix Decomposition" algorithm that would be capable to compute a "near-perfect transmission schedules" for each IP router involved into a multicast traffic distribution with QoS awareness enabled. The proposed method suggests the use of two traffic shapers. The first shaper is placed at the entrance in the network called "Application-Specific Token Bucker Traffic Shapers" [56], to limit the "burstiness of the incoming network traffic" [56]. The second shaper known as "Application-Specific Playback Queues" [56] is placed at the destinations with the role to remove the "residual network jitter" [56]. According to the same studies the regenerated traffic at the destinations has "essentially zero delay jitter and essentially-perfect QoS". In [57] the mathematical apparatus presented in [52-56] is taken further and proposes the inclusion of an "FPGA-based scheduler Lookup Table" into each router participating in "Future Internet" [57] backbone network. With this enhancement the router promises to provide an "essentially-perfect transmission schedule for all its QoS-enabled traffic flows" [57].

Despite of the promising results and benefits these studies suggest minor but still existent hardware modification to the router participating in the multicast traffic distribution. The required adjustments seem to be minor but will have to be applied to all routers within a multicast network. This will represent a disadvantage for the already wired and

functioning routers. On the other hand ISPs might be reluctant with enabling multicast traffic within backbone network because of the *looping* effect mentioned in section 1.3.

2.3 MPLS components

The following sections present the principal components of label switching technologies as they were originally defined in their correspondent RFC documents written to support the development of this research.

Multi-Protocol Label Switching (MPLS) is a multi-layer switching technology that uses labels to determine the method in which packets are forwarded through an MPLS network.

Forwarding packets refer to the common operation that both switches and routers perform on packets of a connectionless network. This operation entails receiving packets at the input, analyzing the content of the packets' header and determining the appropriate output to be transmitted.

Forwarding Equivalence Class (FEC) is a group of IP packets which are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment)[12].
This FEC class definition allows IP packets that have different content in their network layer header to be grouped under the same class of FEC. A relevant example of packets that could be grouped under a common FEC class are a set of unicast packets with the same destination network layer address.

Label is a physically contiguous identifier used to identify a FEC, usually of local significance, [12] which has a short and fixed length. Label values reflect a single data link layer subnet and do not globally affect other values.

The forwarding algorithm executed by a label switching device is called *label swapp*ing. This is due to the fact that it replaces the existing label from a coming packet with a new value before it forwards it on to the next hop.

A router which supports MPLS is known as a "Label Switching Router" or LSR [12]. In addition to its label swapping forwarding ability, it also runs standard IP control protocols (e.g. routing protocols), as with other regular routers, in order to determine where to forward packets.

Label Switch Path is defined as the path through one or more LSRs at one level of the hierarchy followed by packets in a particular FEC [12].

The MPLS edge node is a node that connects an MPLS domain with a node outside of the domain, either because it does not run MPLS, and/or because it is in a different domain. Note that if an LSR has a neighboring host which is not running MPLS, that LSR is an MPLS edge node. [12]. If the traffic enters into an MPLS edge node then the node is called an *ingress node,* while a node where traffic is leaving is considered an *egress node.*

2.4 Multicast and MPLS

The original RFC document [12] that describes the MPLS architecture, left the use of MPLS for multicast for further study. It supplied only general statements about the techniques applicable but, mentioned two important operations that an LSR has to accomplish with the regards to multicast distribution:

a) In the case where a LSR is in the path of a distribution tree, it binds a label to that tree giving the tree an identity.
b) Distributes that bind to its parents on the multicast tree. If the node is on the LAN and has siblings, then the same binding has to be published to its siblings. This

way the parent is able to use a single label when it multicast to all children on the LAN [12].

Starting from the basic premises outlined in the following sections, two categories of approaches frequently encountered in literature in correlation to multicast deployment association with a MPLS environment will be analyzed.

2.4.1 Multicast distribution tree established within MPLS networks

These two requirements impose that to support multicast, one should observe that an LSR is able to select a particular multicast distribution tree based on the following criteria. Firstly, the label carried in the packet (packet has an identity within MPLS domain) and secondly, the interface on which the packet was received [6]. These items would further require a LSR to maintain a table on a per interface basis and the MPLS control component to include the following procedures:

i. Two LSRs connected to a common subnetwork may not bind the same label on that subnetwork to different multicast distribution trees.
ii. LSRs that are connected to a common subnetwork, and are part of a common multicast tree, have to agree among themselves on a common label. This label will be used by all LSRs when sending or receiving packets associated with that tree on the interfaces connected to that subnetwork.

In relation to the premises outlined above, the authors in [6] proposed the following solutions to deploy multicast within a tag switching environment[1].
Generally, the first condition is automatically satisfied on a point-to-point subnetwork. In order to satisfy the first condition, when TSRs (or LSRs) are connected to a multi-access network (e.g. Ethernet), the Tag Switching control component of a TSR defines

[1] The authors preferred to use Tag Switching vs Label Switching, TSR vs LSR, TFIB vs LFIB in order to point out other possible name used by other vendors to refer to the same label switching concept.

procedures by which the TSR partitions a set of labels. These labels are multicasted into a disjoint subsets, where each TSR receives its own subset of labels.
This is accomplished by what is commonly referred to as a Multicast Routing Messages [5] exchange technique. It should be noted that this mechanism can only be used by IP multicast routing protocols which use explicit signaling: e.g. JOIN Messages (this is the case for PIM-SM or CBT protocol). Furthermore, the TSR advertise initially through PIM HELLO messages, a range of labels that the TSR wants to use for its local bindings. At the time when a TSR connected to a multi-access subnetwork is booting up, it checks that the range it wants to use is disjointed from the ranges used by other TSRs connected to the same subnetwork. In the situation where the TSR receives back a PIM Hello message originated from another TSR announcing that this subset of labels overlaps an already existent subset then one of the TSR must get another range of labels. Maintaining a per-interface basis table (TFIB) for a multicast allows a TSR to handle one range of labels for one of its interfaces that is completely independent from the range of labels on another interface(s). A similar idea was proposed as well in the informational draft [5] under the Multi-Access Networks section. Under this section the authors of the draft gave an indication of *who would have the priority to choose the range of labels*. If LSRs allocate a label simultaneously, the LSR with the highest IP address would maintain the label, while the other LSRs withdraw the label [5].

In order to meet the second requirement above, Tag Switching in [6] defines procedures by which TSRs connected to a common subnetwork, and are part of a common multicast distribution tree, have to elect a TSR that is capable to create a local binding for the FEC associated with that tree. Furthermore, [6] defines distribution procedures for this binding information among the other TSRs.

A TSR operates as a regular router where a PIM JOIN message must be sent towards the root of that tree in order to join to a multicast distribution tree. Therefore, it creates a local binding for that tree by procuring a label from the pool of labels associated to the

interface which will be used for sending out the message. It also creates an entry in its TFIB with the incoming label set to the label taken from the pool of labels.
The TSR now includes this label in the PIM JOIN message and sends it toward the root of the tree. When a TSR which is part of a multicast distribution tree receives the PIM JOIN message from a downstream TSR (direction it is given with respect to the root of the tree), it updates the entry in its TFIB that corresponds to the tree with the label carried by the message and the interface on which the message was received.
This way the TSR populates its TFIB entry with an outgoing label obtained from the received PIM JOIN message and the outgoing interface through which the message was received.

In case of multi-access subnetworks PIM JOIN message are multicasted [6]. Thus, all TSRs connected to a multi-access subnetwork receive all the PIM JOIN messages that are transmitted by any of these TSRs over the interfaces connected to the subnetwork.
The first TSR that decides to join a particular multicast distribution tree is automatically elected to create the binding between a label and the tree. All other TSRs are informed periodically about this binding from the PIM JOIN message that is sent by this first TSR toward the root of the tree and towards all other TSRs that are connected to the subnetwork. Furthermore, any TSR that decides to join the multicast distribution tree later will be aware of the label bound to the tree on that subnetwork. This TSR will then create an entry in its TFIB where the incoming label is set to the label received from the PIM JOIN message.
When a TSR wants to check if there is already a binding established between a particular multicast tree and another TSR connected to the same subnetwork, it can send a PIM JOIN message with a label equal to “0”. If there is any binding then the already existent binder TSR will reply back again through PIM JOIN with the existent binding label and the joining TSR will start using the existent label.

It should be noted that since the TSR maintains a separate TFIB used for multicast on a

per-interface basis, the TSR can apply the above described procedure on a per-interface basis, without requiring coordination between each interfaces' procedures.

It should be noted that the nature of the multicast group formation was not mentioned in the procedure described above. It was previously stated that for the dense mode type multicast protocol there is no JOIN messages. In this case the PIM dense-mode procedures must be adjusted such that once the TSR has created a multicast route for the dense-mode group then, PIM JOIN messages are immediately sent to the same dense-mode groups. This results in a common distribution label mechanism, which could be used for both sparse and dense groups.

The authors of [17] mentioned a few of the features outlined in the previous methods. These same features were later confirmed in [5] under an informational framework for IP multicast within MPLS networks. Looking from a MPLS perspective at the procedures, the authors in [17, 6, 5] proposed the following findings:

- The creation of the LSP multicast stream was triggered by a request driven method, which intercepts the sending or receiving of control messages (in this case multicast routing messages were used)
- The binding between a label and particular multicast tree is created as a result of receiving label binding information from a downstream LSR
- By being built on top of PIM the label advertisement can be piggy-backed on the existing control messages instead of sending two separate messages. This would be the case when by using multicast routing messages. More specific protocols, such as PIM-SM and CBT, which have explicit Join messages could be used to carry the label mappings.

The carried underlying multicast protocol does results in some disadvantages such as the following:

- In dense-mode protocol there are no controls messages on which the label advertisement can be piggy-packed.

- [17] Proposes the addition of periodic messages to the dense-mode protocols for the purpose of label advertisement. Therefore an extension of the multicasting routing protocol is required and hence becomes less desirable solution if label advertisement needs to be supported for multiple multicast protocols [5].

2.4.2 Multicast distribution tree established at the edge of MPLS networks

The main characteristic of this approach consists of enabling the multicast protocols on a provider's edge router (PE), which is connected to the customer's multicast enabled routers (CE). Therefore multicast protocols are no longer required to be enabled within MPLS domain. Distribution of the multicast packets within the MPLS domain are realized by activating the MPLS TE function that allows establishment of point-to-point or point-to-multipoint LSPs. This MPLS-TE function activation adds restoration time improvements in case of a transmission failure or a possible network resource switching.

2.4.2.1 Multicast distribution using P2P LSPs within MPLS TE

Based on the above conditions several works have been developed [18, 19 and 20] where the idea of implementing an *Edge Routers Multicasting (ERM)* protocol was proposed. Conceptually ERM converts a multicast flow into multiple quasi unicast flows at the network layer [20]. In order to implement an ERM two approaches are recommended:

- According to the authors, the first option would require a slight modification of the existing multicast routing protocols.
- Heuristic use of the Steiner trees routing algorithm (similar work developed in [21])

The first option extends the traditional PIM-SM or CBT by imposing two conditions:

- Select edge routers as the core or the RP of the multicast tree.
- Sub-trees are allowed to join only at the edge routers.

For dense mode type protocols in order to support ERM, the recommended process is to change the behavior from 'flood and prune' to 'flood and acknowledge' in this way the

upstream peer routers are informed about any active members on their outgoing interfaces. Multicast states (*,G) or (S,G) are maintained by edge routers which records their next downstream edge routers in addition to outgoing interfaces.

Source address	Group address Outgoing	Interfaces	Upstream router (next hop)
192.133.76.1	234.12.144.2	1	193.71.23.1
		2	193.25.22.3
192.133.77.2	242.12.144.4	2	193.45.11.2

Table 2.4-1: Example of multicast routing table at an Edge LSRs

By looking at Table 2.4-1 it is evident that the two outgoing interfaces of the multicast state (192.133.76.1, 234.12.144.2) are towards the next downstream edge peers with 193.71.23.1 and 193.25.22.3.

Given the multicast routing table a multicast flow can be mapped onto multiple LSPs afterwards, simply based on the downstream destination address of an LER. Considering Table 2.4-1, a multicast flow from 192.133.76.1 to 234.12.144.2 is mapped onto two different unicast LSPs for which the next edge routers' destinations are set to 193.71.23.1 and 193.25.22.3.
Multicast packets which arrive at the edge routers (LER) are forwarded in the ERM protocol by duplicating packets based on their routing table and then assigned corresponding MPLS labels.

Once labels are generated by the LER they can be forwarded along the pre-established LSPs between the core LSRs or towards the next downstream LERs. Forwarding decisions are made by the core LSRs and is based solely on the incoming labels. Therefore, the core LSR is not required to distinguish between multicast or unicast flows.

The second approach to implement a ERM protocol is based on building Steiner heuristic trees [20] and extends the ERM's routing procedures previously introduced. As a result this second approach is referred to by authors in [20] as ERM2.

The ERM2 protocol has the following characteristics:

- It constructs a source based multicast tree. This type of tree has the advantage over the core based tree of creating unique states, (S,G) type, since the sources' addresses can be chosen freely.
- Keeps the specific Join message. 'Flood and prune' behaviour of dense protocol is avoided for two reasons. Firstly, due to the fact that the density of a dense multicast group can be considered more like a sparse group compared to the scale of the Internet, and secondly the flow assignment and forwarding algorithm can be applied only after the distribution tree is constructed which by default increases the delay for delivery of the first packet.
- Introduces the notion of a centralized node *Multicast Manager* (MM). The role of this node is described as being different than the RP in PIM or core node in CBT. MM is designed to function more like a DNS server rather than being the root of a distribution tree. It maintains the group membership management in MPLS domain by keeping a record of current active on-tree LERs and returns this list to a new receiver.

The joining procedure of ERM2 is illustrated by the Figure 2-3 [20]

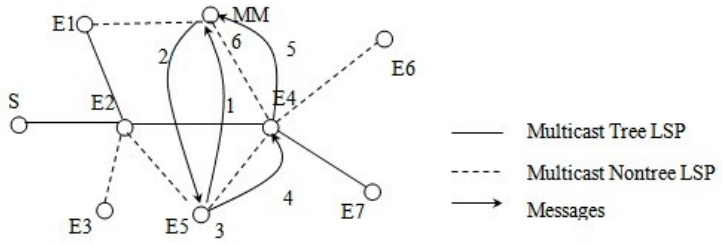


Figure 2-3 Join example of ERM2

Edge routers E1, E2, E4, and E7 are on the tree of multicast group G. The full lines represent the LSPs susceptible to forward the multicast flows while the dotted lines represent the LSPs that do not carry multicast streams.

The numbers in Figure 2-3 represent the following steps:

1. Edge router E5 sends a query message to MM.
2. An ANSWER message is returned by MM containing the list of candidates (S, E1, E2, E4 and E7) to E5.
3. E5 picks the best candidate based on its own routing table or resource availability. In this case it chooses E4.
4. E5 sends a JOIN message to E4 which creates an outgoing entry for state (S,G).
5. If the previous step is successful E4 informs MM through an ADD message that E5 is now active.
6. MM inserts E5 in the list of active members for the multicast group G.

In the case where an edge node leaves the multicast tree, it sends a SUBSTRACT message to the MM to update the member list, and a PRUNE message to its upstream peer.

The presented approach defines an ERM protocol that converts the design of a point-to-multipoint LSP setup to a multiple point-to-point LSP thus preparing the multicast traffic for aggregation. The multicast trees branch only at the edge routers and further use the MPLS tunnels set up by the core routers. Simulation results of the study show that the proposed ERM2 with heuristic Steiner trees was more efficient than the approach which relies on the extension of the PIM routing protocol.

The only current limitation is the reduction of the multicast traffic distribution to single point-to-point LSPs and eventually the ERM procedures deployments at the edge routers. According to the study, this ERM procedures deployment could be incrementally accomplished.

2.4.2.2 Multicast distribution using P2MP LSPs within MPLS TE

Much of the works in these areas are based on the capacity of building P2MP LSPs within the MPLS domain. The advantages of using these LSPs resides in the possibility of deploying quality constraints associated with trees that are naturally deployed by IP multicast transmission mode, while taking into account the resources made available in MPLS network.

The ability of adding quality constraints-based routing through MPLS alternately requires developing a better setup mechanism for LSP that would be able to provide explicit routing.

Currently there are two directions that help in the LSP setup mechanism which generally extends two popular signaling protocols and as a result creates the RSVP-TE and CR-LDP protocols. Both types of protocols have advantages and disadvantages which were analyzed or criticized from different angles such as: signaling mechanism or QoS involved models.

Summarized conclusions of these factors are outlined by the following:

- *Signaling mechanism* perspective favors the RSVP approach primarily because there is no connection requirement before label distribution can occur. It is thought that RSVP has a "lightweight adjacencies" [6] therefore new neighbor relationships can be faster established.
- *QoS models* can push ahead some relevant advantages of one protocol over the other. RSVP uses the Int-Serv QoS model, which was developed over several years, clearly with the resource allocation over heterogeneous network as usually can be found in IP networks. The service models and QoS parameters defined for Int-Serv are intended and a considerable work was involved to be supportable on all link layers [6]. CR-LDP Qos model defines a relative large numbers of

parameters similar to those used on ATM or Frame Relays, which are still unclearly specified how to be used in order to deliver this QoS model.

It is evident that there is an obvious advantage to the RSVP protocol over the CR-LDP which was designed to address the IP heterogeneous networks requirements.
This protocol was elected as the base protocol for support of multicast distribution within a MPLS TE environment by the authors in [23]. Their proposal became a standard and extended the RSVP-TE protocol in order to deploy P2MP LSPs.

Currently the plain RSVP protocol, enhanced with the Label Object, was able to establish the MPLS forwarding state (or the LSP) only along the path described by plain IP routing. It is known that this type of path computation is driven by the destination-based forwarding mechanism. Therefore, the PATH message in this IP plain environment is forwarded based on the forwarding table constructed by protocols such as RIP, OSPF, IS-IS or BGP and the destination address existent in the IP packet.
As a result the advantages of the path taken by an LSP already influenced by the path that is drawn by the PATH message could have been questioned. It was apparent that the ability of explicit routing or the possibility to "steer" the LSP path based on different constraints was lacking.

In order to overcome the aforementioned steering issue, the RSVP extension was adapted with a new addition of a new object, the Explicit Route Object (ERO) as per [24]. This object is carried in the PATH message and contains an ordered sequence of *abstract nodes* specified by an explicit route that the message has to follow. Thus the packet containing this enhanced PATH message with ERO will help the router to forward the packet based on the content of this ERO object rather than by referencing the IP destination address contained in the header of the packet.
The ERO is structured as a sequence of <type, length, value> triples, where each triples depicts an abstract node.

These abstract nodes identify a group of one or more routers and at the time of writing this document there are three types of abstract nodes defined: IPv4 address prefix, IPv6 address prefix, and Autonomous System number [6].

A SECONDARY_EXPLICIT_ROUTE (SERO) was later introduced in [23] and is defined in the same way as the ERO object who's intent is to be able to encode the path of each subsequent S2L sub-LSP.

Another important upgrade of the RSVP-TE protocol consisted of finding a way to optimize the number of control messages needed to set up a P2MP LSP. The premise of the optimization was to use a minimal number of Path messages in order to signal multiple S2L sub-LSPs. This optimization would avoid the potential repetition of path information for the parts of S2L sub-LSPs that share hops.

The document [23] describes a solution to encode the S2L sub-LSP explicit routes using compression in the following way:

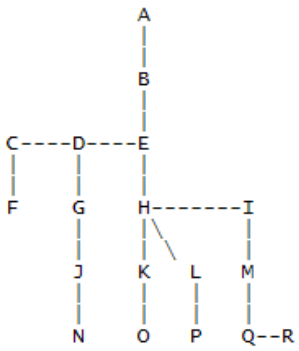


Figure 2-4 Explicit Route Compression

Figure 2-4 illustrates a P2MP LSP where A is an ingress router and six egress LSRs: (F, N, O, P, Q and R). All six S2L P2MP LSPs can be signaled by a single PATH messages by assuming that the first S2L LSP from ingress A to egress F is the first LSP and the rest S2L are the subsequent S2L LSPs. Bellow there is a way for the ingress LSR A to encode the S2L sub-LSP explicit routes using compression and the encoding objects mentioned

previously:

S2L sub-LSP-F: ERO = {B, E, D, C, F}, <S2L_SUB_LSP> object-F

S2L sub-LSP-N: SERO = {D, G, J, N}, <S2L_SUB_LSP> object-N

S2L sub-LSP-O: SERO = {E, H, K, O}, <S2L_SUB_LSP> object-O

S2L sub-LSP-P: SERO = {H, L, P}, <S2L_SUB_LSP> object-P

S2L sub-LSP-Q: SERO = {H, I, M, Q}, <S2L_SUB_LSP> object-Q

S2L sub-LSP-R: SERO = {Q, R}, <S2L_SUB_LSP> object-R

After the PATH message is received from LSR B, the LSR E sends the following PATH message to LSR D encoded using the same mechanism as the following:

S2L sub-LSP-F: ERO = {D, C, F}, <S2L_SUB_LSP> object-F

S2L sub-LSP-N: SERO = {D, G, J, N}, <S2L_SUB_LSP> object-N

LSR E sends a PATH message to LSR H in the same manner and is encoded as:

S2L sub-LSP-F: ERO = {D, C, F}, <S2L_SUB_LSP> object-F

S2L sub-LSP-N: SERO = {D, G, J, N}, <S2L_SUB_LSP> object-N

When it is required to signal a single S2L LSP branch the PATH message from LSR H to LSR K is encoded as:

S2L sub-LSP-O: ERO = {K, O}, <S2L_SUB_LSP> object-O

while from LSR H to LSR L it is encoded as

S2L sub-LSP-P: ERO = {L, P}, <S2L_SUB_LSP> object-P

The described mechanism is devised to compress the PATH message size and the time for processing these messages is also reduced at the routers which results in a quicker LSPs setup.

Assuming the MPLS domain supports enablement of these P2MP LSPs the authors in [23, 22] went further and presented an approach for deploying IP multicast using these P2MP paths.

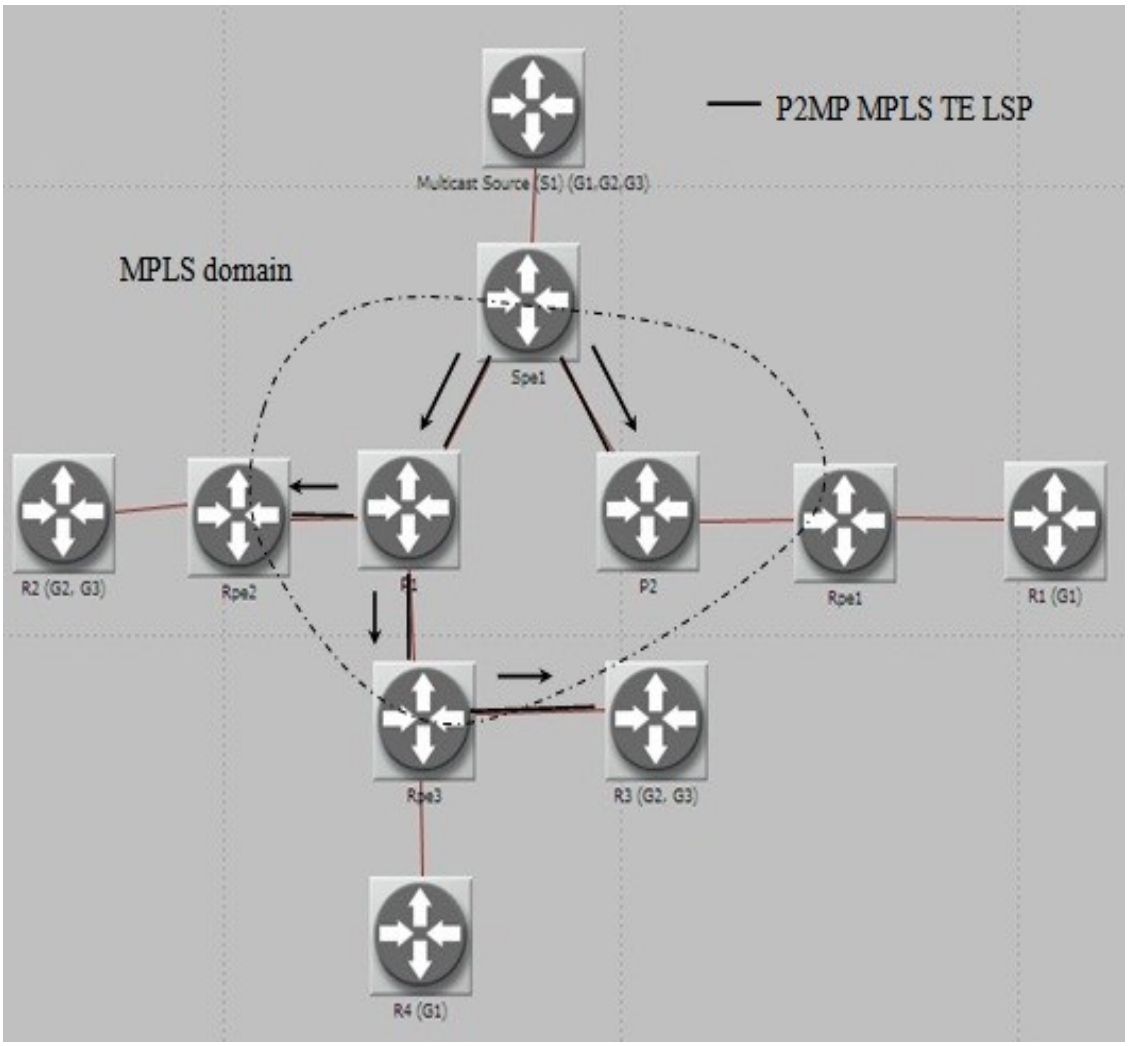


Figure2-5 MPLS domain with P2MP LSP enabled on the provider P1, P2 core routers with root starting from Spe1.

In the following section the principal components and methods participating within MPLS-TE to enable multicast transmission with P2MP LSPs enabled within core routers as described in [22] are analyzed.

Considering Figure 2-5 the following roles are defined:

S1 is a multicast source and is the only available and transmitting source for multicast group G1, G2, G3.

Spe1 is the PE router that receives traffic from a multicast source.

Rpe {1,2,3} are PE edge routers participating in multicast routing with S1, R1, R2, R3 and R4.

R{1, 2, 3, 4} are multicast receivers. R1 listens to G1, R2 listens to (G2, G3), R3 listens to (G2, G3), and R4 listens to G1.

The presented scenario assumes the following conditions:

- The core routers are BGP free, which is required by the unicast protocol for the RPF check condition required by PIM-SM.
- RSVP-TE is used by the core routers as a required condition [23] to enable the P2MP LSPs.
- PIM-SM control traffic originates outside of the provider domain. Problem definition assumes the network core is free of PIM-SM control messages.
- Multicast application [P2MP-MPLS-TE-REQ] requires enabling TE in the core to facilitate multicast traffic.

The [22] describes the following operations:

i. PIM-SM control state exchange between PEs.
ii. MPLS P2MP LSP endpoints discovery ie **Rpe** are discovered by **Spe**.
iii. Mapping IP Multicast Traffic at the Spe to a P2MP LSP
iv. RPF interface determination at the RPE

i. PIM-SM control state exchange between PEs

PE routers participating in multicast traffic have to exchange conversation PIM-SM control messages with other remote PE or CE routers. This PIM-SM routing exchange is accomplished by the following [PIM-SM-REMOTE] sequences.

When Rpe receives the PIM-SM Join/Prune message for a particular multicast source (S), Rpe determines its adjacent neighbor Spe advertising source S by using (BGP next-hop).

Rpe initiates a called “remote” PIM-SM adjacency with Spe. At this point Rpe sends Join/Prune message to the Spe.

ii. P2MP LSP endpoints discovery

Previous sequences [PIM-SM-REMOTE] allowed the Spe to get signaled from Rpe with a Join/Prune messages request. The receipt of a Join request causes the Spe to treat the Rpe as a P2MP LSP leaf. Spe associates the Join message with a P2MP LSP. Spe initiates the setup of the P2MP LSP with an ending point represented by Rpe. The following cases are possible:

If the P2MP LSP is not existent then a new P2MP LSP is installed. The P2MP LSP is existent but without adding Rpe as an existent leaf . Spe then adds Rpe as a new leaf.

If the P2MP LSP is existent and Rpe is already a P2MP LSP leaf, then Spe leaves the existing P2MP LSP unchanged.

By looking at the Figure 2-5 the above procedures can be sequenced as outlined below:

- Initially there is no P2MP LSP present. If R4 sends a PIM-SM Join for (S1,G1) to Rpe3, this Rpe3 determines its adjacent neighbor Spe1 as the advertiser (next-hop) for S1. Spe1 treats this received Join message from Rpe3 as a leaf join of (S1,G1) to a newly initiated P2MP LSP1 with Rpe3 as its destination.
- Another Join request for (S1,G1) towards Rpe1, but from R1, will follow the [PIM-SM-REMOTE] procedure that determines Spe1 as the adjacent neighbor (next-hop) advertising S1 as its source. Spe1 takes over the Join request from Rpe1 and maps (S1,G1) to P2MP LSP1. Therefore, a new branch to P2MP LSP1 is added with a Rpe1 destination.

iii. Mapping IP Multicast Traffic to a P2MP LSP

When Spe receives a Join message for a (S,G) or (*,G) it associates it with a P2MP LSP. Therefore, Spe creates a multicast forwarding entry for source S and group G with the P2MP LSP as an outgoing interface which allows an IP multicast traffic to be sent to the P2MP LSP.

The selection of '*the outgoing P2MP LSP*' for a particular (S,G) becomes a local decision problem and the authors from [23] propose a few options but are open to further development by future vendors.

iv. RPF interface on a Rpe

This procedure is primarily about the condition in which the Rpe associates a P2MP LSP as the RPF interface for a given (S,G) pair entry that was propagated to a Spe. This P2MP LSP has to be identical with the P2MP LSP used by Spe to send IP multicast traffic for that particular (S,G). Because the selection of P2MP LSP is Spe driven, this implies that Spe needs to communicate all the (S,G) entries mapped onto that LSP, to all Rpes that are part of that P2MP LSP.

In order to facilitate these associations the authors in [22] introduced a new PIM-SM message, the *Join Acknowledge Message*. This new PIM-SM message also introduces the notion of *PIM-SM Route Attributes*. These attributes can be used to associate common properties of the Group Sets in the Join Ack Message.
With the aid of the Join Ack message, the Spe is able to convey to Rpe the P2MP LSP identifier associated with the (S,G) entries encoded as a Route Attribute.

Once the Rpe receives the P2MP LSP association (identifier) for a particular (S,G) entry, it uses that P2MP LSP as the RPF interface for the (S,G) entry [22].

In order to distribute the multicast over the P2MP LSP within a MPLS-TE environment the presented approach relies on the following:

- P2MP LSPs whose settings are based on the signaling properties offered by the extensions of RSVP-TE protocol must be enabled. The MPLS routers involved in setting up these LSPs will need to process the RSVP_PATH and RSVP_RESV messages.
- Introduction of a Join Acknowledge Message along with PIM-SM Route Attribute that mitigate the issue of RPF interface determination at the receiver PE node. This condition being a priori checking condition as mentioned in section 2.2 of this thesis. With this new Join message carrying the P2MP LSP attribute (identifier) the receiver at the PE node is able to receive Group Sets - P2MP LSP association and ultimately the IP multicast distribution can be initiated.

This approach is a promising solution to the implementation of the distribution of IP multicast as it was advanced to the level of standard [23]. This approach does have the disadvantage of enabling P2MP LSPs and modifying the PIM-SM protocol.

A disadvantage of setting the P2MP LSPs thus far is the use of the resource reservation process utilized by the RSVP-TE protocol. This reservation process leads to the maintenance of a significant number of states that grows in relation with the dynamic of subscription/withdrawal process implied within a multicast group.

2.4.2.3 Conclusion of multicast distribution within MPLS TE

The IP multicast transmission mode relies on a dynamic establishment and maintenance of the multicast distribution trees that are computed based on the information inherited from the activation of an IGP routing protocol. These trees are deployed over a network of resources consisting of hosts and adjacent routers that communicate with each other through the Internet Group Membership Protocol (IGMP) communication protocol in

order to establish multicast group memberships.

The challenge of mapping these multicast trees to MPLS-TE LSPs reveals a few issues. Most of these issues are raised by any constraint-based routing system and must be solved prior to enabling multicast transmissions within a MPLS-TE enabled network:

A. Computation and establishment of LSP

The multicast tree suggests the establishment of a point-to-multipoint LSP. Although the MPLS-TE, as a constraint-based routing system, is able to enable a constrained shortest path first (CSPF) protocol, it has to collaborate with an LSP setup mechanism that provides explicit route support. The resource reservation protocol (RSVP) and constraint-based routing label (CR-LDP) protocol are two options for this LSP setup mechanism component. These signaling protocols have to be extended as proposed in [24], in order to be compatible with the dynamics of multicast subscription/cancellation procedures. These procedures have a characteristic behavior for the terminal branches of the multicast distribution trees.

B. Routing methods for multicast traffic within P2MP LSP

The application of only MPLS TE functions to establish point-to-multipoint (P2MP) LSPs to forward the multicast traffic does not represent a sufficient condition to enable the routers to replicate packets and to make the routing decisions. Forwarding over a multicast distribution tree whose structure is dynamic and is influenced by the deployed multicast protocol (e.g protocol independent multicast sparse mode PIM-SM) creates transitional states within a multicast enabled router. One of these transition states appears in a PIM-SM environment, when a shared distribution tree, for example the rendezvous point tree (RPT - whose root is a specific router in the network), switches towards a shortest path tree (SPT- whose root is the source of an established multicast group). As a result a multicast enabled router participating in the establishment and maintenance of a P2MP LSPs may be placed in several situations as mentioned in [5] where it has to

maintain two states – the state (*,G) for which the router is supposed to forward traffic along the LSP, and the recently created state (S, G) (appeared after the changing from RPT to SPT). When the router is in the state (S,G) the router will not have the ability to associate a new outgoing label in order to forward the multicast traffic on a proper and existing point-to-point (P2P) LSP.

Given these challenges, several cases have been studied in [5] or advanced as a solution within IETF working groups as in [22]. In both cases the proposed scenarios were approached with a "divide and conquer" method by treating routing and QoS separately or by not considering them as was the case in [22]. In actuality there is a strong relationship between the two problems, and this relationship cannot be ignored otherwise the established P2MP LSP structures stay rigid and unaware of the imposed QoS applied to the traffic and the network resources.

2.5 Quality of Service

In previous sections QoS was only mentioned as a reference. In the following section the QoS definition and service meaning within an IP network to support the SLA requirement of the applications it is designed to support will be addressed.

Taking apart the constructing words, the following definitions can be given:
Service in the context of IP networking is a description of the overall treatment of a customer's traffic across a particular domain [2].
Quality in IP networking refers to the underlying requirements for an application that can be expressed in terms of SLA metrics for IP service performance such as: delay, jitter, packet loss, throughput, service availability, and per flow sequence preservation [2].

Considering the aforementioned definitions as a whole, QoS defines an optimization problem that tries to maximize an application's SLA requirements while minimizing the delivery cost of the application's service.

When the issue that QoS is trying to resolve in a given IP network is know, the toolset of the QoS should be reviewed and the method in which its consisting components can be engaged within the MPLS domain should be determined.

2.5.1 QoS components

QoS uses a range of functions (e.g. classification, scheduling, policing, shaping) within a IP QoS model architecture (consisting of Integrated Service and Differentiated Services) in order to ensure that a network service meets the SLA requirements for the supported applications.

The mechanisms used to engineer the QoS in a network can be divided into data plane and control plane mechanisms that are available in network devices such as routers. A short introduction of these planes is presented bellow as they appear in [2]. A detailed description of each individual entity is beyond the scope of this thesis but will be presented within a minimum description that is required in order to develop the research of this document.

Therefore according to [2] the following components can be found within these planes

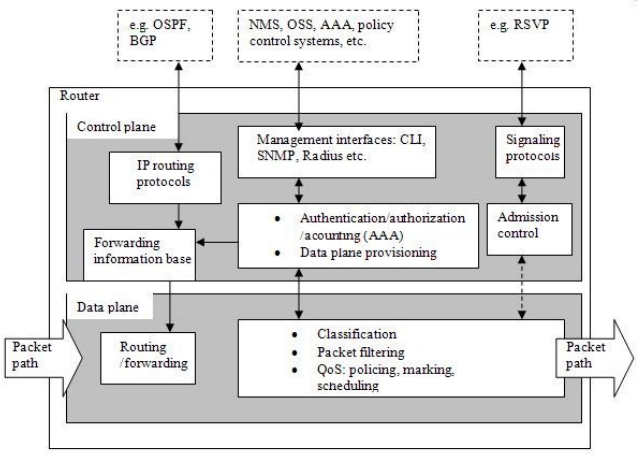


Figure 2-6 Control plane and data plane QOS functions [2]

2.5.1.1 Data plane

Functions belonging to this plane have a direct impact on the forwarding behavior of the packets. These functions can be summarized as the following, based on the actions that they apply to the incoming traffic streams:

a) Classification is defined as the process of identifying flows of packets and grouping individual traffic flows into aggregated streams. These streams can be later subduing them to different actions such as packet filtering. The relevance of this classification function helps to understand the following:

- Integrated Services IP QoS model architecture uses a signaling protocol (RSVP) that was mentioned previously in chapter 2.4.2.2 in order to set up the per flow[2] classifier.
- Differentiated Services IP QoS model architecture uses a *simple* classification that is stored directly in the DSCP field of the IP packet header that has been specifically designed for QoS classification. Therefore, the classification is based upon the matching aggregates of traffic that are marked through this DSCP field. Different types of traffic can be defined as: A traffic stream which is an aggregation of flows that are based on common classification criteria (e.g. all VOIP traffic stream originating from a VOIP gateway), and traffic class that is an aggregation of individual traffic flow or streams, for the purpose of applying a common action to the constituent flows or streams [2].

b) IP packet's marking, or *coloring,* is actually the process that happens before the *Classification*. It marks the values of the fields designated for QoS classification which are stored in the DSCP field of the IP packet header or EXP field of the MPLS packet headers.

---

[2] An IPv4 flow is defined as a 5-tupe of source, destination IP addresses, source, destination TCP/UDP port and the transport protocol (e.g. TCP or UDP).

c) Policing is a mechanism that ensures a traffic stream does not exceed a defined maximum rate.

The policing mechanism can be visualized as a token bucket with the elements shown in Figure 2-7. A simple token bucket policer is defined through a *Burst* (expressed in bytes), which represents the maximum bucket depth, and a *Rate* (expressed in bps) at which the bytes-sized tokens are added to the bucket.

There are different types of policers implemented depending on the method in which the tokens are added to the bucket. Tokens can be added at a rate R every time the policer finishes processing a packet or at regular intervals by filling up with a maximum numbers of tokens (expressed by Burst) that can be inside the bucket.

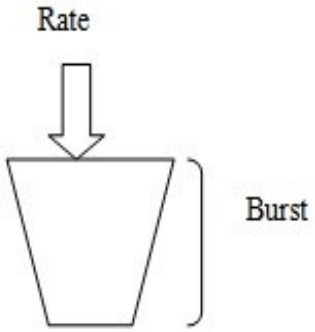


Figure 2-7 Token Bucket

The action that a policer executes on a traffic streams depends on whether the packet from the arrival stream *conforms* or *exceeds* the bucket definition. If there are fewer tokens in the bucket then the number of bytes of an arrival packet then it is said that the packet has *exceeded* the bucket's definition.

The actions of a policer are not reduced only to transmit the traffic stream or drop it, based on the packet's conformance (conformed or exceeded), but are also responsible for the marking of the traffic. This way a node receiving a marked traffic can take actions based on this marking and act according to the specification defined in SLAs.

A few important characteristics must be noted with a policer. It does not delay traffic

because it does not store packets in the bucket and it cannot re-order or prioritize traffic the same as a scheduler. [2].

Different types of policers were specified based on the marking techniques applied and their behavior with regards to the incoming traffic stream such as “single rate three color marker” in [25], or “A Two Rate Three Color Marker” in [26].

d) *Queuing and Scheduling* are important functions of the QoS that mediates traffic departures at the router level in order to reduce congestion. Compared to the Policer function an IP packet scheduler has the ability to prioritize and group traffic’s packets into entities called queues.
A scheduler has the ability to store temporary packets into a buffer. This allows them to schedule the departure of packets and to shuffle packets departures according to rules that are derived from constraints of rates and priorities [2].

The two most common queuing algorithms, first-in-first-out (FIFO) and fair queuing (FQ), are mentioned in the following sections, though several variations have been proposed to improve deficiency of the algorithms.

The premise of FIFO queuing is reflected through its name: The first packet that arrives at a router is the first packet to be transmitted. The following situation can be observed during the arrival of the packet at the router according to Figure 2-8.
When an arrival packet attempts to lineup at the entrance of a full queue (there is no empty buffer space left) then the packet is dropped. This case is called *tail drop,* since packets that arrives at the tail end of the FIFO will be discarded.

The approach that simply drops the packets in case the buffer is full, is the simplest method for implementing a FIFO scheduling type of scheduler.

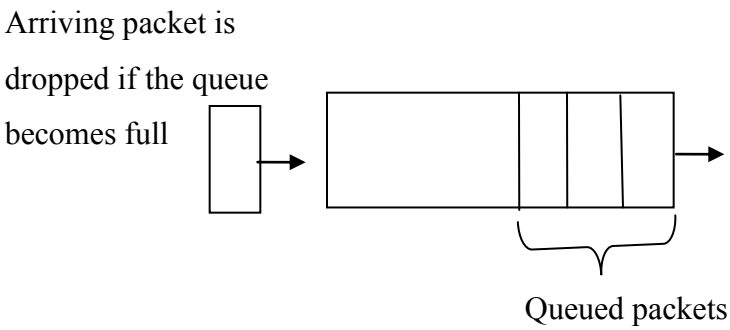


Figure 2-8 Tail drop happens of an arriving packet at a full FIFO queue.

Different complex algorithms for "drops policy" that attempt to decide when the packets are to be dropped by using different signaling techniques at the level of the router (e.g. DECbit or RED congestion schemes) or by measuring the tendency of the packets' increase in queues (e.g. by measuring RTT) are available.

A simple variation of the FIFO queuing was created by adding queuing priority. The main purpose of this feature allows routers to implement multiple FIFO queues, one for each priority class. This way the router checks and sends packets from the non-empty queues that have the highest priority queues. The addition of this feature has an obvious disadvantage. This feature causes the higher priority queues to starve out lower priority queues.

The second queuing algorithm, FQ, attempts to resolve the difficulties present in the FIFO queuing approach. Firstly, the FIFO queuing approach does not differentiate the traffic based on the source, therefore the packets could not be grouped according to their flow.
This presents a disadvantage because it could not check how successful the congestion control system is through its controlling actions.

Conversely, most of the congestion control system is addressed at the source level and

due to the inability of separating packets that belong to a specific application flow, could cause flooding of the router with packets originating from only one type of application.
Flooding would be worse if that specific application uses UDP protocol (which is common for IP telephony application) as a transport layer.The congestion control mechanism is based on TCP and does realize that a congestion is occurring through the network.

FQ was proposed in order to overcome the issues mentioned above. This algorithm maintains a separate queue for each flow that is currently serviced by the router.
In the case when one queue from *n* queues is empty or it does not have packets to be serviced then the available bandwidth will be shared among the other *n-1* remaining queues.

The order that the packets are processed from the queues by the scheduler's algorithm generated variations of the FQ algorithm.
All variations of the FQ take into account the arrival rate of the packets and define processing order rules for the available packets in the queue based on their associated rated services. These rated services (or weighted services) are able to delay packets queuing, while the ability to assign weights allows it to service queues in a relatively differentiated manner.

The Weighted Round Robin (WRR) is the simplest example of such a weighted bandwidth scheduling algorithm [2].
The manner in which the WRR works can be described by considering the following example from Figure 2-9. This example considers 3 weighted queues A, B, and C with weights of 1, 3 and 4. In each round a WRR scheduler will service 1 packet from queue A, 3 packets from queue B and 4 packets from queue C.
When all queues are permanently filled with same size packets and the capacity of the outgoing link is for example 54Mbps, then the link bandwidth will be shared between

queues according to the formula:

$$A_i = \frac{Wi}{\sum_{i=1}^{n} Wi} \times B \qquad \textbf{(1)}$$

Where A= bandwidth allocation for queue i, B = is the available bandwidth and W_i is the weight associated to the queue.

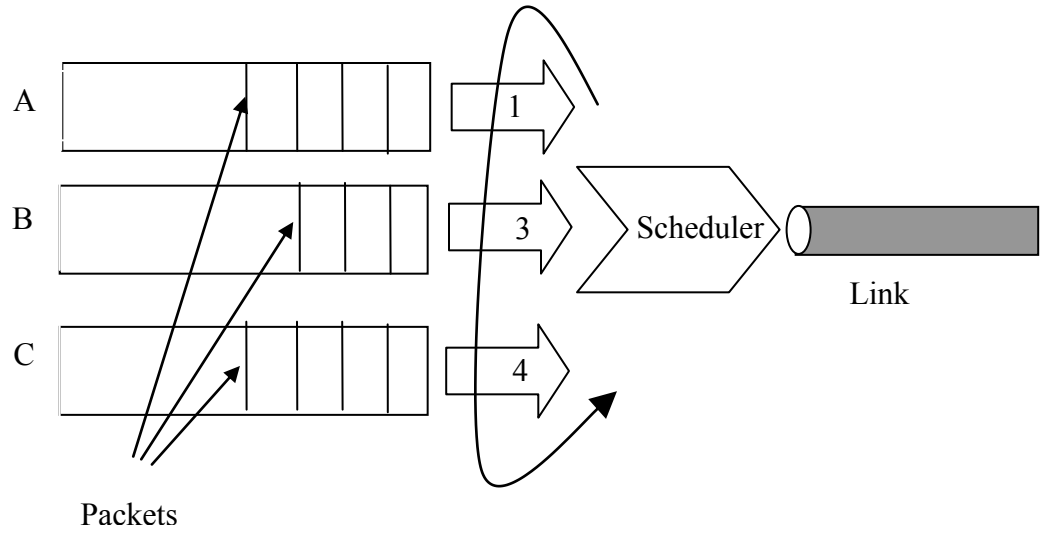


Figure 2-9 Packets arriving in queues are serviced by Scheduler based on their associated weights

According to formula (1) the following allocations will be obtained: $A_A = 1/(1+3+4) \times 54 = 6.75$ Mbps (for queue A), $A_B = 3/(1+3+4) \times 54 =$ 20.25 Mbps (for queue B), and $A_C = 4/(1+3+4) \times 54 =$ 27 Mbps (for queue C).

In terms of percentage, the given weights could be translated to 12.5 % bandwidth allocation for queue A, 37.5 % to queue B and 50% to queue C.

For the case where one queue becomes inactive (for example queue A) then its allocation is distributed between the other active queues, B and C in this case. Therefore the new allocation for these queues becomes $A_B = 20.25 + 6.75 \times (3/(3+4)) = 23.14$ Mbps,

$A_C = 27 + 6.75 \times (4/(3+4)) = 30.85$ Mbps.

In general the size of the packets is not the same. Therefore they are grouped based on their size in the queues. For this case and with the consideration that in queues A, B and C there are buffered packets of 300, 1500, 1024 bytes size, then before applying the previous formula (1), the weighted services have to be normalized and converted to integer values.

The integer values of the normalized weights become Integer (1/300) = Integer (3.3×10^{-3}) = 33 for queue A, Integer (3/1500) = Integer (2×10^{-3}) =20 for queue B, and Integer (4/1024) = Integer (3.9×10^{-3}) =39 for queue C.

With these new normalized weights the allocation for the queues reveals more fairness

for queue A: $A_A = 54 \times (33 \times 300 / (33 \times 300 + 20 \times 1500 + 39 \times 1024)) = 6.69$ Mbps,

for queue B: $A_B = 54 \times (20 \times 1500 / (33 \times 300 + 20 \times 1500 + 39 \times 1024)) = 20.29$ Mbps,

for queue C: $A_C = 54 \times (39 \times 1024 / (33 \times 300 + 20 \times 1500 + 39 \times 1024)) = 27$ Mbps

WRR enables fairness between queues as long as the average size of the packet is known. Regardless of whether the size of the packet is known, these values are continually changing which causes the scheduler to become un-calibrated, and lose the fairness between queues servicing.

Another popular scheduling queuing discipline is the Weight Fairness Queuing (WFQ). The algorithm behind this queuing establishes the allocated bandwidth based on the weights, as well as the moment of packets departures.

Using the same example as mentioned above, three queues: A, B, and C that are processing packets of 300, 1500, 1024 bytes, a more clear description is given. Queues A, B, and C have the following associated weights 1, 3, 4. In WFQ's case the bandwidth

share of a queue is inversely proportional to that queue's weight [2]. In order to maintain a bandwidth share of 1:3:4, the example above requires the association of a 4:3:1 weight to queues A, B and C respectively.

The WFQ algorithm computes the servicing time (or sequence number) using the following formula

$$SequenceNr = RoundNr + SizeofPacket \times W_i \quad (2)$$

where *SequenceNr* is servicing time, and *RoundNr* is round number. Initially *RoundNr* = 0, *SizeofPacket* is the size of the packet, W_i is the weight of the queue *i,* and the following rule applies. This rule specifies that the scheduler will service first the packet that has the lower *SequenceNr* and the *SequenceNr* of that packet becomes the next RoundNr.

Considering a sequence of packets that arrive in queues as A1, C1, B1, B2, and C2, then the followings servicing times will be determined. The initial RoundNr is equal to "0",

$SequenceNr_{A1} = 0 + 300 \times 4 = 1200$, $SequenceNr_{C1} = 0 + 1024 \times 1 = 1024$, $SequenceNr_{B1} = 0 + 1500 \times 3 = 4500$, $SequenceNr_{B2} = 4500 + 1500 \times 3 = 9000$, $SequenceNr_{C2} = 1024 + 1024 \times 1 = 2048$

The following packet transmission sequence: C1,A1,C2,B1 and B2 is the result of the aforementioned example when considering the rule for servicing packets, starting with the lowest servicing time.

The most commonly used queuing disciplines that are used in conjunction with other QoS functions within the framework of an IP QoS model architecture were briefly discussed in this section. Also presented through examples in this section were different queuing disciplines. Their main features that ensure a better fairness of packets transmission contained in queues were outlined. This increased fairness can be translated into a better bandwidth allocation.

For this purpose WFQ from FQ proves itself as a preferred scheduler compared to WRR which is only able to ensure fairness under the condition that the packet sizes are equal.

FQ reaches a better fairness among the queues compared to the simple FIFO queuing discipline due to the FIFO's inability to service queues with lower priority or as in some cases ignores traffic streams because of the incapacity to recognize the traffic's nature.

2.5.1.2 Control Plane

According to [2] the control plane QoS mechanisms deals with the admission control and resource reservation which eventually could be used to set up the data plane QoS functions.

Admission control is a checking process to determine if there are enough network resources in order to meet a request for QoS from an application side, and in the case of a shortage to deny it.

Functions of the QoS control plane are used in combination with other routing protocols e.g. BGP, OSPF that maintain and advertise link states information between available routers in order to accomplish the control functionality of this plane.
An important function of this QoS control plane that presents interest for this research is the signalling function which in practice is primarily the RSVP protocol.

RSVP is found in both IP QoS model context Int-Serv and Diff-Serv. The over usage is justified by the fact that RSVP is the protocol that allows applications to signal QoS requirements to the network [6] and the network is capable of sending back a successful resource reservation or a failure answer.

The following sections present a short description of the IP QoS models in order to

understand which approaches were proposed to meet QoS applications' requirement traversing networks.

2.5.2 IP QoS models

This section will parse the main features of the existing IP QoS models and indicate which features will help to enable QoS support within a MPLS domain.

2.5.2.1 Int-Serv model

The main goal of the Integrated Services working group of the IETF was to develop an architecture capable of ensuring an end-to-end QoS guarantee to those applications which may require them. Examples of this requirement are teleconferencing applications. [28] proposed the inclusion of two sorts of services aimed towards real-time traffic applications: *guaranteed service* and *predictive service*. Proposals of these service models arose from the conclusion that applications could be classified as *intolerant* or *tolerant* applications.

In order to avoid the possibility of late packets, for a given distribution of packets delays, the condition dictated by an intolerant application is to have a fixed offset delay larger than the absolute maximum delay [28].
In contrast to this type of intolerant application sits the tolerant application that does not need to set this fixed offset delay (or known as "reliable upper bound delay") greater than the absolute maximum delay due to the fact that late packets are tolerable in this type of applications.

In combination with these service models, the Int-Serv group is defined as the method in which the RSVP protocol could make use of these service models.

The Int-Serv architecture's description started from the ability to handle a data structure,

called flow specification or *flow spec* which was previously defined in [29].
The flow spec defines the special services concluded from the network as guarantees.
For this purposes the flow spec must describe the method in which flow's traffic will be injected into the network through a Tspec (Traffic specification).
Conversely, the flow spec must be able to describe the application's expectation to be obtained in terms of QoS, expressed through Rspec (Request specification).

In order to ensure an end-to-end QoS guarantee, the Int-Serv architecture requires that each network node (resource) perform those QoS functions specific to data plane mentioned in section 2.5.1.1. (classifying, packet marking, policing, queuing and scheduling).

2.5.2.2 Diff-Serv model

The Int-Serv is known as a *fine-grained* IP QoS model for which the resources are allocated to each individual flow. It was recognized that a *coarse-grained* model would be more beneficial in order to avoid scalability issues raised by Int-Serv.

DiffServ resolves scalability issues by applying QoS functions and maintenance through a per-customer treatment at the edge of the network. These edges of the network are considered to be the edges of a Diff-Serv enabled domain. Therefore regardless of the applications grow within the network, the scalability requirement is established at the entrance of a Diff-Serv domain.

For this purpose, the Diff-Serv QoS model divides the traffic into a number of classes and allocates resources on a per-class basis.

An incoming traffic at the ingress entrance of a Diff-Serv enabled domain is firstly divided into classes of traffic or "behavior aggregates", and then these aggregates are

compared for conformance against agreed profiles known as Traffic Conditioning Agreements (TCA). Functions of QoS mentioned in previous sections (e.g. policing) will ensure that the traversing traffic will conform to the TCA condition, and the non-conformant traffic will be delayed by applying a departures scheme (e.g queuing disciplines), re-marked or simply dropped.

It is important to observe that the traffic's division into classes of traffic occurs at the level of the node (router) where packets are already stored or marked to hold information about how to be treated according to a specified QoS level.

This information is marked directly in the packet known as Differentiated Services Code Point (DSCP) and is carried inside the 6-bit Differentiated Services field of the IP header, which initially was part of the TypeOfService (ToS) byte. DSCP signals the level of treatment that must be applied to the packet to the router. This signal identifies the per-hop-behavior (PHB). The PHB definition can be established locally specific to a network or it could be used as a standard definition. This standard PHB includes:

- The default and most commonly known is the "best-effort" class for which all packets are treated equally from a QoS perspective. This results in no special treatment.
- Express forwarding (EF). Packets from this type of class are forwarded with a minimal delay and receiver experiences low loss. In general a scheduler attempting to ensure an EF PHB could use a WRR or WFQ queuing scheme as mentioned in 2.5.1.1, but in this case the worst delay obtained will depend on the algorithm implementation as well as the number of queues dedicated for EF traffic. The condition to ensure an EF PHB QoS level regardless of the queuing scheme will require the arrival rate of packets to be lower than the servicing rate.

  EF PHB might support the applications classes (such as VoIP) where low-delay, low-jitter, low-loss and assured bandwidth is required.

- Assured forwarding (AF). According to [30] a set of AF PHBs are defined with the following characteristics: each PHB in the set is typically represented through an AFij codepoint notation. The AF PHB group "provides forwarding" for IP packets from N independent AF possible classes (1<= i <=N). Each packet having assigned a drop precedence j (1<= j <= M).

  By way of explanation, a queue from N number of possible AF classes for which a drop value of j is selected for a given IP packet. The recommended number for general use is 12 AF PHBs, representing N=4 AF classes of M=3 drop preference levels in each class.

  The same document [30] describes an important condition for queue selection that ultimately influences MPLS support of Diff-Serv. A Diff-Serv enabled node must forward packets from one AF class independently from packets in another AF class. Therefore, the packets cannot aggregate two or more AF classes together.

  This condition is translatable into the following: packets belonging to an application that are differentiated only through the dropping value preference (e.g. AF_{11}, AF_{12}, AF_{13}) must go into the same common queue belonging to AF_1 class.

These standard definitions of PHBs are based on a recommended DSCP, but this value inscription could be created or modified at the node level based on a network operator's decision. Therefore, the treatment applied to the traffic may be based on a standard PHB definition but, the input selector that is given by DSCP belongs to the node's local decision. Practically, each router applies a mapping between DSCP and a standard PHB class.

In conclusion and in support of the decision to select Diff-Serv over Int-Serv as the underlying IP QoS model for an MPLS enabled network, the following arguments should be mentioned:

- DiffServ allows the traffic to be marked and classified at the router level. These provisions are configured manually or by a network management system (NMS), rather than being signaled by a network signaling protocol such as RSVP.
- Continuing the previous idea, a traffic classification decision could be pushed to the network's edges. This results in the avoidance of the scalability problem presented by Int-Serv.
- Take advantage of the DSCP value established at one hop, preferably at the entrance edge of a Diff-Serv domain. Later apply different QoS levels of treatment at the successive downstream nodes will help support a DiffServ within a MPLS domain.

2.6 Enabling QoS support with Diff-Serv within MPLS

With the knowledge of the DiffServ features, on which the MPLS network is going to be built, it is important to investigate the connecting element existent in MPLS that will make an MPLS network aware of the QoS requirement needed to be granted for different traffic classes.

It is important to note that packets travelling an MPLS domain are forwarded based on the label and that it is the only element taken into account by an LSR during the forwarding process without looking further into a packet's header. Therefore, the existent DSCP information used by the Diff-Serv mechanism is not sufficient to influence the LSR to apply an according PHB treatment for the packet.

In order to determine the connection between DiffServ support and MPLS, two different elements were determined to become expandable to two different solutions. These are the experimental field known as EXP and the label of the packet itself. Each of the next solutions has its own characteristic and applicability to the domain as is discussed below.

2.6.1 PHB inferred from EXP

The first method according to [31] in order to provide Diff-Serv support relies on the MPLS' shim header that includes an experimental (Exp) field of 3 bits long as per the figure.

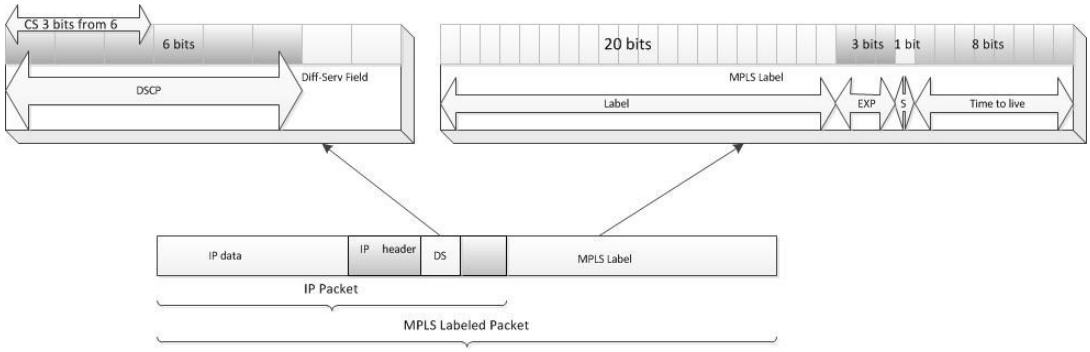


Figure 2-10 MPLS Labeled packet with EXP field populated from CodePoint selector (CS)

With this reduced field number it is possible to define only eight PHBs levels of treatment compared to 2^6 = 64 PHBs offered by DSCP. This is possible with the condition that no more than eight levels of treatment need to be defined. The LSR will use the EXP bits of the packet and map them to the associated PHBs. This behaviour is similar to a regular router enabled for Diff-Serv, which uses the DSCP number, extracted from the IP header of a packet and maps it to a pre-established PHB.

The action that an LSR applies to an incoming labelled packet marked for QoS awareness through an EXP field will result in locating that packet into a queue where the packet receives a specific treatment, initially assigned for that EXP field number. For example, the LSR that receives two labelled packets with EXP 000 and 001 respectively may place packets marked with 000 into a queue where the packets receives a treatment associated to an AF21 PHB type of QoS level. It may then place the rest of packets, marked as 001, into a queue that will receive a treatment associated to an EF PHB type of QoS level.

An LSR which sets up an LSP based on this method is thought to establish an *E-LSP* type of paths due to the fact that its PHBs is inferred from the EXP field.

2.6.2 PHB inferred from Label

When more than eight distinct PHB treatments need to be applied, the next available element to solve this problem is the label of the packet. The applied PHB treatment is based on the label information. This method to setup up LSP paths is referred to as *L-LSP* type (or label inferred type of LSP setup).

It is possible to observe that the label caries the information on which the L-LSP will be configured. Therefore, the mechanism that distributes the labels will have to be enhanced in order utilize this information. For this purpose and according to [31], a new term called PHB *scheduling class* (or PSC) has been defined. This PSC groups packets with different PHBs and ensures that they are not miss-ordered during their transmission. MPLS implies that the packets with different PHB, but under the same PSC, must travel on a common LSP. A similar condition was imposed in Section 2.5.2.2 where packets with special imposed restriction are placed in a common queuing discipline.

This situation appears in MPLS when one instance of a PHB group (e.g the case of AF type of PHB) imposes ordering constraints on packets carrying different PHBs (e.g., AF21, AF22 and AF23 packets). As a result of this constrain condition imposes that packets of a common PHB scheduling class PSC (that is all packets AF_{2x}, for example) to travel on a common LSP.

According to the above rules a packet label will define only a general PHB class (e.g AF_{2x}, AF_{3x} or AF_{1x}). In order to define a full PHB description for a packet, details for drop precedence is also required. To achieve this [31] proposes the use of a “shim” header and the EXP field can be used to store this information.

2.6.3 Conclusion of the PHB methods

Taking into consideration the reduction of the EXP field from the MPLS's shim header and the ability to use labels to convey PHB two different methods for propagating Diff-Serv information within MPLS networks are created:

The first method is defined as a single E-LSP and which can carry packets requiring up to eight different PHBs.

The second method is defined as a scheme where an L-LSP (label inferred paths) carries a single PSC (e.g. AF or just EF). Therefore, the remaining information of the PHBs containing the drop precedence is defined inside the EXP field.

The following conclusions can be drawn to aid in determining whether E-LSP or L-LSP is the best solution to convey labels within a MPLS-TE with Diff-Serv awareness:

- by default L-LSP is the sole choice over ATM. This is due to the fact that the MPLS shim header does not existent and the Exp therefore cannot be used.
- L-LSP can support a larger number of PHBs than eight. Therefore, different paths for different PHBs can be engineered since L-LSP can define one LSP for each PHB. It should be noted that L-LSPs more commonly used within ATM LSRs that are not widely deployed according to [2]. In conclusion neither are possible associated deployments of L-LSPs.
- E-LSP offers more PHBs on a single E-LSP thus it reduces the number of labels
- The E-LSP model has many common similarities with the standard Diff-Serv model [1] and therefore is a good candidate to be engaged in forwarding traffic shaped with Diff-Serv QoS.

3 High Level Solution's Components

Previously reviewed papers essential to outline the problem definition formulate a solution for the multicast distribution over MPLS-TE with Diff-Serv aware domain.
Therefore, this chapter presents the logical concepts used to solve the multicast distribution over DS-TE problem. The architecture of this solution is also developed in this chapter through high level solution components and its basic behavior to visualize the supporting elements.

3.1 Introduction

The following section identifies the problem definition of the TE which is similar to the problem definition of the IP multicast distribution problem over a Diff-Serv aware MPLS-TE network.

The purpose of illustrating the similarity between these two problem definitions allows a solution to the multicast distribution problem by applying similar concepts that are applicable in MPLS constraint-based routing (or TE) problem to be determined.

The problem definition that TE is trying to solve depends on the following requirements:

a) It is looking to establish routes that are optimal with respect to a certain scalar metric

b) It has to take into account the available bandwidth on individual links [1] that will be used for traffic distribution.

The similar challenges that the multicast traffic distribution reveals within a MPLS domain are clearly evident.

The following sections present the main functionality of an MPLS TE, they outline the underlining interactions, and they explain the reasons for adding Diff-Serv features to the model.

3.2 MPLS TE functionality

The primary advantages of introducing MPLS were outlined in Chapter 2. If the conventional IP network was able to forward traffic based solely on the destination address included in the packet's header, the MPLS network made a step forward in the routing domain through its structural division of the forwarding plane from data plane. It improved traffic engineering through the addition of the ability to construct TE paths, which are computed upon imposed constraints and the ability to deploy explicit routing. Explicit routing is an important feature that can be used to divert traffic if the required bandwidth resource is insufficient or congestion occurs on the targeted paths.

The collaboration between components can be seen by reviewing the following minimal MPLS enabled configuration. The primary network example defined to prove the effectiveness of an MPLS enabled network is the traditional *fishnet* type of network. This example will be referenced to explain the MPLS TE behavior during of an LSP setup (or "TE tunnel setup").

Figure 3-1 is an example where all the nodes have MPLS enabled. Each connecting link has a capacity, for example 2 Gbps, with the exception of the links entering the ingress node (from LSR1 to LSR2) or exiting the egress node (from LSR5 to LSR6) which both have a capacity of 3 Gbps. A single area is defined and each link is configured with an equal cost, for example "1". With the assumption that an LSP1 of 1 Gbps defined from LSR1 to LSR6 already exists, the obvious path will pass through LSR 7. This is due to the fact that this is the shortest path computed based on the cost.

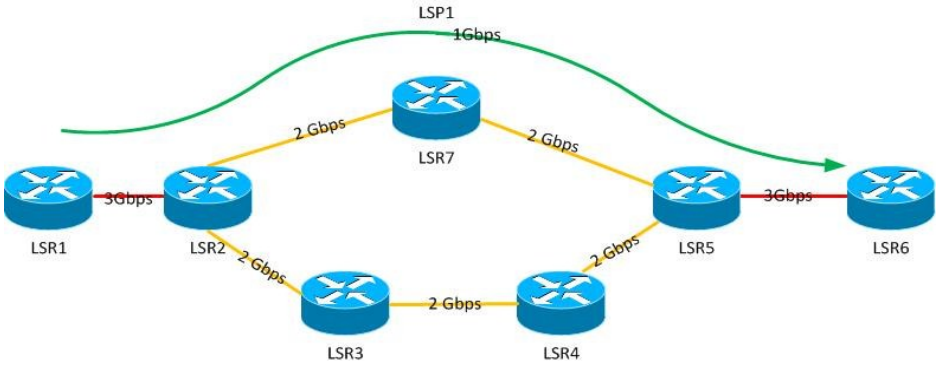


Figure 3-1 Fishnet type of network enabled with MPLS

The presented example describes the sequence of events that will appear during a new LSP2 setup, which has a bandwidth of 2Gbps. This example emphasizes the issues that will remain unsolved if the MPLS-TE domain is not enhanced with QoS support.

1) *Distribution of resources and policies (constraints).* In the first phase a traffic engineering extension of one of the link state routing protocol (e.g. OSPF [32] or IS-IS [33]) starts publishing the available resources along with the administrative policy constraint information.

2) *Constraint-based path calculation.* Once the resources information and administrative policies are advertised the next step is to compute the optimal path required to setup the second LSP. It should be noted that, an LSP path can be either specified statically (or manually established by a network operator) or dynamically.

The dynamical method subsequentially offers another two options. The first option is where the LSP path can be computed by an offline centralized function ("tunnel server" as in [2]) dedicated for this purpose. This will be responsible for the maintenance of the LSP paths and proposes an explicit head-end path. The second option allows the LSP to be calculated online in a distributed way by each TE LSP sources (known as tunnel "head-ends"[2]). Regardless of which dynamical method is used, the constraint-based path calculation is computed by a constrain-based shortest path first algorithm (CSPF). This algorithm will determine the path on which the LSP should be configured. The

algorithm takes into account the available resources (e.g. links' bandwidth), the constraints (required LSP's bandwidth) and the lowest cost path available. Non-conformant paths are automatically removed from the possible topology. The result of this algorithm calculation is an explicit route object (ERO) as was mentioned in Section 2.4.2.2. This ERO defines the hop-by-hop path that an LSP should follow.

In the second required LSP path, the CSPF algorithm concludes that only the lower path (LSR1 ->LSR2->LSR3->LSR4->LSR5->LSR6) can be used. The use of this lower path is due to the fact that it is the only path with sufficient bandwidth capable to satisfy the LSP's setup requirement as presented in the aforementioned example. As a result, the output values of the CSPF algorithm contain the IP addresses that build the path which will be followed during of the LSP setup.

3) *Signaling and reservation for LSP setup (tunnel setup).* Once the path prescription is acknowledge in the form of an ERO, a list containing node's IP addresses, the signaling protocol most likely to be used will be either RSVP-TE [34] or CR-LDP [35]. With the assumption that the RSVP-TE is used in the presented example, this signaling protocol will start signaling resource reservations. For this purpose the RSVP-TE signaling protocol uses the PATH message carrying ERO and the LSP bandwidth requirement. The PATH message will travel from the head end to the tail end of an LSP and is sent along the specifically routed path using the ERO. Each router, which receives the PATH message, will check for resource capability reservation on the outgoing interface. This is done to ensure that future traffic can be forwarded towards the next node.

There is the possibility that this operation may override the results previously obtained by the CSPF algorithm but it is necessary to assure that the path with the required bandwidth is still available and in place.

There are a variety of different situations that could arise with this method such as a link failure. Another possible situation could be the instantiation of another LSP path after the

current LSP path has started to be deployed. In case an unsatisfactory outbound interface is detected at one of the node, the adjacent upstream node will signal by returning a PathErr message to the head-end of the LSP. This PathErr message states that the admission control decision was unsuccessful.

A successful admission control decision [2] at a node is concluded by further forwarding the PATH message defined by the ERO until eventually reaches the end-tail of the LSP path. It should be noted that MPLS-TE supports the concept of pre-emption [2] that might appear during of the LSP signaling process. This results in a lower priority LSP path which can be pre-emptied in case an LSP with a higher priority is required to be deployed.

If the PATH message is successfully received by the tail-end router (and where the LSP path ends) the receiving node will reply back with a RESV message. This message follows the signaled ERO's path backwards towards the head-end router of the LSP (or where the LSP path starts).

The reverse journey of the RESV message, also known as the reservation phase, has to be successful again at each hop. A node (hop) that successfully received the RESV message confirms the LSP reservation or confirms tunnel reservation [2]. Next, from this successful confirmed node, the RESV message travels upstream (towards the head-end of the LSP) accompanied by a value of the label created by the same transmitting router. This router later expects to receive the same label from an adjacent upstream neighbor located on the LSP when the traffic starts coming from that router.

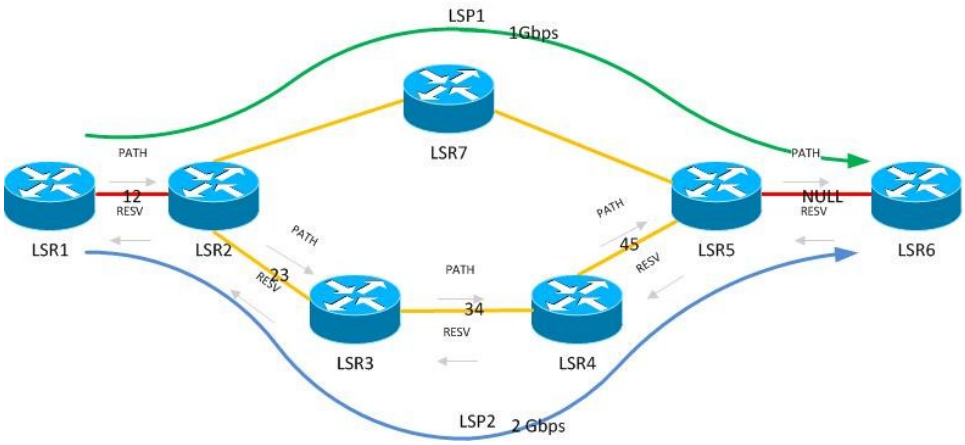


Figure 3-2 MPLS enabled network showing the RSVP messages (PATH, RESV) transmitted during of the LSP 2 setup. First PATH message travels downstream from the head-end of the LSP2, then RESV message travels upstream from the tail-end of the LSP2.

Figure 3-2 presents the reverse trip of the RESV message, which starts, from the tail-end of the LSP2. The same Figure also shows example values for the labels as they are created and advertised together along the LSP path during the RESV message's journey.

The label's binding direction, which establishes the methods' name for label binding, is called *downstream-assigned* [12] label with a label binding distribution from *downstream-to-upstream* [12].

4) *Associating traffic on the established LSP path.* With the confirmed LSP setup configuration enabled, a head-end router is almost ready to start placing traffic on the LSP's pipe. Before this process is capable of transmission on the LSP's pipe, the head-end router must map the incoming traffic and associate its outbound traffic to be placed on the LSP path. This is necessary to ensure that the head-end router will continue to use the default IGP protocol which in turn will try to detect the shortest path (from LSR1towards LSR 6 through LSR7).

The mapping process at the head-end router can be fulfilled through static routing [2]. This is accomplished by defining a static route that maps the incoming traffic onto the path used by LSP in order to reach the traffic's destination address. Another possibility could be to dynamically define IP routes in order to forward traffic over the LSP paths, although this is dependent on the vendor implementation of the router.

Once the traffic mapping has been established at the head-end router, the traffic can start flooding the network. Although this is only through the links which were signaled and reserved for usage by the LSP's path.

In the presented context the LSR1 router will map the incoming traffic onto the specifically static routed LSPs or dynamically defined LSPs deployed through the MPLS-TE capable network. Specific to this case, traffic travels from LSR1 towards LSR2 with a label value, for example 12, bound to it. LSR2 *swaps* the label of the incoming traffic to a label value of 23 and forwards the traffic further to the LSR5 router in a similar fashion. This node is called penultimate hop popping (PHP) and its function is to *pop* the label before entering into the last hop LSR6. When there is a stack of labels, the PHP pops it to the outermost label, and based on the nature of the packet, decides where to keep forwarding the packet.

5) *Maintaining the path of LSP.* The available network resources indicate that the nature of the network is volatile enough. As a result of this knowledge an established LSP has to be continuously maintained through asynchronous RSVP PATH/RESV messages. An LSP may be voluntarily torn down by a head-end router through a PathTear message.

Another example of a maintenance message is the PathErr message which was previously mentioned. This message is sent towards the head-end LSP's node by an upstream adjacent node when a downstream node/link fails. When this occurs the head-end tries to find a new path around the failed node. The same message is also utilized when an LSP is pre-empted to allow a higher priority LSP to be instantiated.

Based on the sequence of events presented in the previous network example, the followings can be concluded in general for an MPLS-TE's functionality:

- TE or constraint-based routing allows an MPLS to signal and reserve LSP's path instantiation over the non-shortest path in order to achieve a higher throughput with more efficiency.
- It should be noted that there is a high level of solicited operations demanded for the head-end LSP. These operations map the incoming traffic with the outbounding interfaces according to the computation established by the CSPF protocol's algorithm.

3.3 Diff-Serv awareness within MPLS-TE

Taking into consideration the information that has been previously present with respect to the operations of the MPLS TE, it was observed that the TE helps the head-end router (originator of an LSP path) to become conscious about the path selection on which the traffic will be placed. The division of incoming traffic on a per service basis has yet to be resolved for a simple MPLS-TE. This division of incoming traffic would allow different treatments according to the SLA specification to be applied.

The addition of QoS support is required to accomplish this traffic differentiation. Without enabling any QoS support, the TE computes the LSP paths that are aggregated from and across all incoming traffic without knowledge of the type of treatment, which should be individually applied. The result of this traffic aggregation and the establishment of an LSP path on a link, which meets the bandwidth requirement, will result in multiple traffic which belonging to different service classes to be driven onto the same LSP path.

QoS awareness must be enabled by involving an IP Qos model framework mentioned in the previous Chapter to satisfy the problem discussed above and to avoid different traffic

classes to be intermixed on the same LSP. This document investigates the use of Diff-Serv over Int-Serv in particular due to the fact that a method to separate traffic on a per-class of service basis is required.

According to [2], the enablement of the Diff-Serv awareness within MPLS-TE (known simply also as DS-TE) extends TE capabilities such as compute constraint-based paths, specific routing definition and admission control enablement to be applicable over different type of traffic that are divided into classes of services.

With the combination of these two technologies, Diff-Serv and MPLS-TE, the link's overall bandwidth availability can be seen as an aggregation of bandwidth sub-pools where the available bandwidth is defined within the boundary of TE's bandwidth constraint. The practical side of this Diff-Serv extension consists of the ability to associate a bandwidth sub-pool for a traffic class in such a manner that allows the application of constraint-based routing and admission control for the LSP carrying that specific traffic class.

When this document was written, there were known two models capable to apply sub-pool bandwidth constraints to different classes of traffic:

- *Maximum Allocation Bandwidth (MAM)* Constraints Model for Diffserv-aware MPLS Traffic Engineering. According to [35] this model sets a maximum number of Bandwidth Constraints (MaxBC) which is restricted to "8" which is equal to the maximum number of class types. It also defines the rules for the sub-pools bandwidth usage that can be allocated for each class of traffic independently. This model specifies that the aggregated sum of each individual maximum allowed sub-pool's bandwidth allocated for individual classes of traffic cannot exceed the maximum reservable bandwidth, which is limited by the link's capacity. Figure 3-3 is a visual representation of this situation.

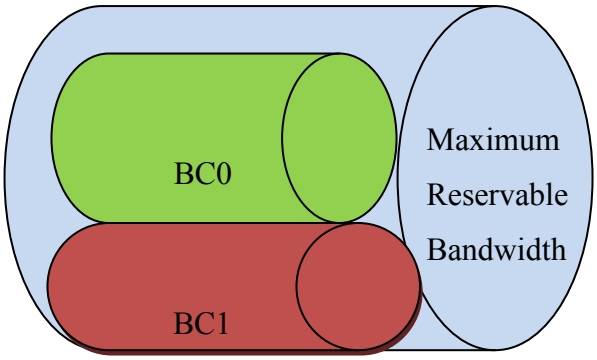


Figure 3-3 MAM model example. Presents two classes of traffic with bandwidth BC_0 and BC_1 cannot exceed the maximum reservable bandwidth.

Figure 3-3 presents two types of traffic classes traveling on link with a capacity L_C. Both, traffic have a bandwidth constraints set to BC_0 and BC_1.

Maximum reservable bandwidth (MRB) = $L_C = BC_0 + BC_1$

- *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering (RDM)*. In [36] this model similarly defines the maximum number of bandwidth constraints (MaxBC) as the MAM model. It is equal to the maximum number of class types that is set to "8". For this model, the bandwidth of the sub-pools allocated for the classes of traffic are hierarchically defined. The model considers an aggregated constraint associated to a global pool bandwidth that is limited by the link's capacity. This aggregated constraint embeds a number of sub-constrains definable for each sub-pools bandwidth. Figure 3-4 is a visual representation where CT0 is a sub-pool of constraint CT1, while CT1 is a sub-pool of constraint CT2.

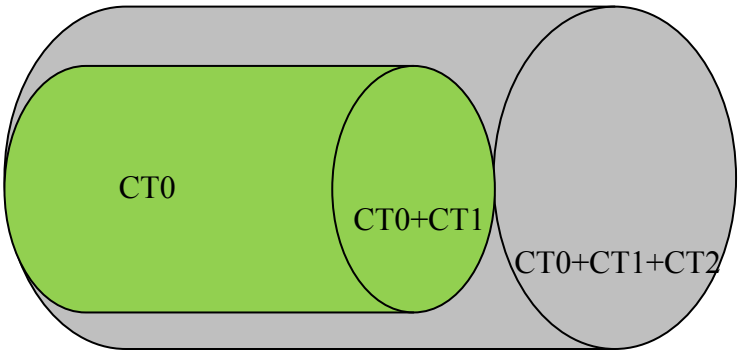


Figure 3-4 RDM model example. Constraint 0 is a sub-pool of constraint 1, constraint 1 is a sub-pool of constraint 2.

Each model is justified for a particular case that is influenced by the bandwidth requirement of the LSP or by the pre-emption requirement of different classes of traffic. For example, RDM is more suitable for the situation where traffic is established over an LSP (e.g with a traffic constraint CT0) with a lower priority and can take advantage of an available bandwidth of an LSP with a higher priority because of a non-existent traffic (e.g CT1).

Alternatively, the MAM model is preferable when the allocation of well delimited bandwidths for different traffic classes is required.

The inclusion of one of the model in any future DS-TE deployment is necessary according to [34]. “DS-TE technical solution MUST specify at least one Bandwidth Constraints Model” but “MAY specify multiple Bandwidth Constraints Models” is also true. Without a measurable expression of the Diff-Serv, the QoS support could not be imposed and restricted for different classes of traffic on per service basis. Therefore a network operator could not realize the performance of the network, and to what degree can fulfill the clients SLAs requirements.

3.4 Traffic trunks

An important concept that is introduced with DS-TE, and on which both Diff-Serv bandwidth models (MAM, RDM) rely on to define their model definition, are the traffic trunks as they were first defined in [38] and later re-referenced in [34, 37]. The importance of this traffic trunk, and its properties, are reiterated in this section due to the fact that its capability helps converge to the solution of the traffic engineering problem.

According to [38] a "traffic trunk is an aggregation of traffic flows of the same class which are placed inside a Label Switched Path". Thus a traffic trunk can be seen as an abstract representation of a collection of individual TCP, or UDP flows, known as "microflows," that share two common properties [6]. These properties are: all microflows are forwarded along the same common path and all microflows within a trunk share the same Class of Service`.

With the given traffic trunk's definition, it is important to observe that there is a clear separation between LSP path and a traffic trunk. A traffic trunk utilizes the LSP and is characterized by several properties within TE's context as they are mentioned in [37]. For the interest of this document only the following traffic trunk attributes and properties will be presented as outlined below:

- Without any Diff-Serv awareness a traffic trunk within a bare MPLS-TE context considers the current Internet as a single class of service model. A traffic trunk is able to aggregate all of the traffic between an ingress LSR and egress LSR.

  It was illustrated in Section 3.3 how this property was relaxed and how the addition of Diff-Serv awareness was the subject of MAM [35] and RDM [36] definition models. In this way sub-pools of bandwidth were created and associated for separate type of classes of traffic.

- Traffic trunks are considered as routable objects within an abstract representation context.
- A traffic trunk, which is a separate entity from an LSP path, has the ability to pass through an LSP and move onto another path.

A whole set of operations that are significant to TE, with the exception of two, are defined in [37]. These allow a large amount of capabilities for traffic trunks to be possible. These capabilities include the option of modifying traffic trunk's attributes and rerouting a traffic trunk's route. The last operation, the traffic trunk's rerouting, offers the choice of being controlled by an administrative action of a network operator or to be controlled automatically by the underlying protocols.

The reason for introducing the concepts of these traffic trunks, such as routable routes within MPLS-TE domain, is because it switches individual flows traveling this domain to trunks. This switch is done to map traffic trunks onto the physical network's topology through LSP paths. Through the capability of this mapping function the TE problem is approached in a way that results in more reliable network operations with better performance.

The advantage of using traffic trunks vs microflows is supported by the fact that it decouples the amount of forwarding states and the control of traffic required to establish and maintain these states from the amount of the traffic transported by a service provider. This allows the volume of the traffic pushed through an MPLS-TE domain to be independent from the number of trunks that can be decided based on the provider's topology requirements.

To summarize, the concept of traffic trunks reduces the problem of TE to other subproblems. Traffic is distributed by the service provider as a collection of traffic trunks with specific bandwidth requirements. A second subproblem that has to deal with the routes of these trunks within an ISP is a result of this.

3.5 Coupling IP multicast traffic over an Diff-Serv aware MPLS-TE

With the information provided above a checklist of the available components existent in the project infrastructure must be determined. Also, the solution to the question "What else it is required for a DS-TE to be ready to welcome a multicast traffic flooding?" for a traffic that should be differentiated (and treated accordingly based on the nature of the traffic) " must be devised.

A network operator will be allowed to put together a network infrastructure that is capable of accepting multicast traffic, classify it based on the traffic classes and forward this traffic over DS-TE domain with a high performance network operation can be done with the knowledge of the components outlined in the previous Sections.
This means the wiring of the network is ready to establish the connections, although the missing part persists in the logical side of the coupling multicast traffic to this powerful network.

Although the multicast traffic was analyzed from different angles of the available multicast protocols, none of them were able to prove to be the best candidate to distribute multicast traffic over a MPLS domain, especially over a Diff-Serv aware MPLS-TE domain.
Section 2.2.3, 2.2.4 in Chapter 2 outlined the main issues and challenges faced by these different types of multicast protocols. The volatile nature of this multicast distribution tree created a difficult landscape for a flooding traffic to be coupled over an ordered network such as is provided in the MPLS-TE domain.

Besides this difficult nature of multicast traffic distribution, more impediments arouse with the requirement of QoS support.

Following a review of the IP QoS models, it was determined that Diff-Serv promises to

be the most advantageous for an LSP path through an MPLS-TE. For the enlargement of the network infrastructure, Diff-Serv offers high scalability due to its tendency to push traffic classification operation to the edge of the network, particularly to an ingress LER node. Preferably this Diff-Serv aware ingress LER node should be placed at the entrance of each Diff-Serv aware MPLS-TE domain. Through this chained up Diff-Serv, with MPLS enabled routers, it would be possible to enable a full QoS support along several MPLS domains.

It should also be noted that during the presentation of MPLS-TE functionality, the ingress LER was the principal decision maker router, through which the CSPF protocol decided the right, but not necessarily the shortest LSP path. This path would have an optimal bandwidth requirement that would ensure the forwarding of traffic with a given requirement.
The large sequence of operations that had to manage traffic differentiation, LSP path signaling and reservation that this single ingress LER router undergoes has been presented.

Emphasizing the high number of operations performed by this ingress LER router has a variety of reasons. This router is capable of setup an LSP because it is the head-end of the LSP, which means path selections are made at the level of this router through the use of the CSPF protocol.

Alternatively, with the amount of information already available at this router it is important to look for a combination of multicast protocol suite that could benefit all of this information. Also it must start mapping the multicast traffic onto the physical network's topology by using the traffic trunk concepts of TE in order to flood traffic over on an LSP passing through a Diff-Serv aware MPLS-TE domain.

With the knowledge of these requirements imposed by the network's topology, that

primarily can be adjusted through traffic trunk's attributes, the multicast protocol that will lead to a solution of the multicast distribution over an DS-TE domain must be more controllable. This means that it should be able to influence multicast distribution based on the requirement information propagated from the network's available resources and topology available at the ingress LER node level.

It is preferred that the ingress LER that is in charge of the traffic classification, establishment of traffic trunks, signaling and reservation of the LSP paths, and traffic enablement through traffic trunks onto the established LSP be aware of the existence of the multicast traffic source originator. The ingress LER will then be able to engage multicast traffic flood and shape it according to the service requirement. In this document this was accomplished with Diff-Serv.

In order to obtain all of the aforementioned requirements, this document proposes the use of the following multicast protocol: Source Specific Multicast (SSM) protocol. This protocol derives from the PIM family.

The following section provides a detailed description of the SSM protocol. It will also outline its main features that will to be used to offer that missing coupling link element required by the DS-TE network to start flooding multicast traffic.

3.5.1 Overview of PIM-SSM protocol

The PIM Source Specific Multicast (SSM) belongs to the PIM source family, This source family's subset methods are used in combination with the IGMP v3 to deliver multicast packets to a receiver. PIM SSM delivers multicast packets only from the source address that is specified and requested by the receiver. This request is accomplished by allowing the receiver to specify the source address to listen to by using the IGMPv3. For a more in

depth presentation of IGMPv3, an interested reader should refer to the document [39].

Firstly, a key function that causes a multicast traffic to happen according to [40] is the source discovery of the network itself. The router participating in multicast group distribution must be able to determine the locations of all sources whenever there are interested listeners. This is not an issue for the dense mode multicast protocol since the sources are in charge by default with the traffic flooding, hence the source detection function is an unnecessary step. The sparse mode type is different due to the fact that it is necessary to determine the root of a multicast source from which a multicast group will start to receive traffic. Initially, a dedicated RP router was designated to overtake this traffic through itself towards the receiver and eventually to switch over to the shortest path tree (SPT) from source to receiver. From this perspective, the SSM distribution type uses only a subset of PIM functions. Through this an SPT is created from source to receiver without the help of a dedicated RP node. This reduces the overhead complexity for establishing an RP node.

It is relevant to emphasize the sequences of events through which a specified multicast source successfully delivers packets to the listening receivers. A better understanding of how this type of multicast traffic distribution works over a DS-TE domain will help to complete the network design proposed to solve the problem definition of this research.

All routers involved in a SSM multicast distribution must have PIM–SM enabled on all of their interfaces and IGMPv3 specification enabled at the receiver's level.

Taking the similar fishnet network presented in the previous section, and assuming that there are a couple of receivers (hosts) directly connected to LSR 6, Figure 3-5 is a demonstration of the following sequence of messages:

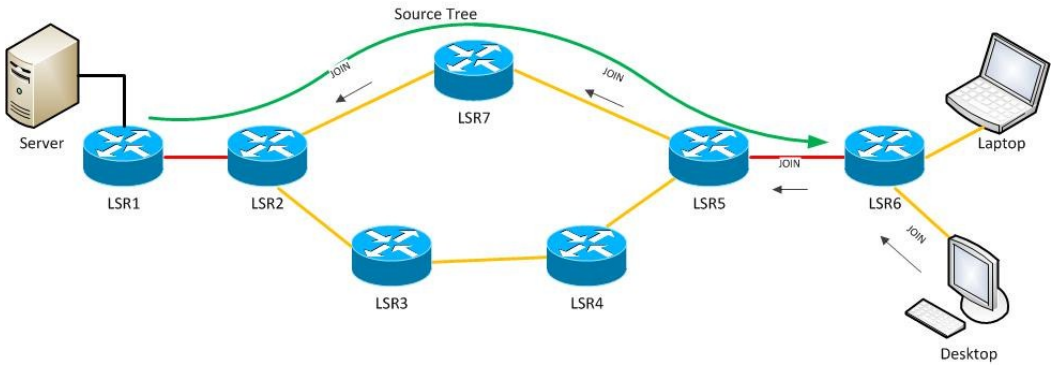


Figure 3-5 PIM-SSM example. A successful JOIN message sent from Desktop up to the last hop LSR2 connected to the source LSR1 is followed by a SPT *source tree.*

1) The receiver host (in this case the Desktop computer) sends a JOIN wish message from the group G to the directly connected router. In terms of SSM, a receiver host subscribes to a source's channel described through an (S,G) pair. This Join message is propagated from router to router (hop by hop) until it reaches the multicast source. In this way a SPT path is built but without the need of an RP node point.
2) The last hop router connected to the source, and receiving the (S,G) Join message, initiates a *source tree* construction starting from this router.
3) (S,G) *state* is turned on, or otherwise said multicast traffic can be delivered from source towards the requiring host receiver.

Based on these basic operations, the one-to-many type SSM model brings a couple of benefits over the traditional PIM-SM:

- PIM SSM builds SPTs rooted immediately at the source, for in SSM the router closest to the interested receiver host is informed of the unicast IP address of the source for the multicast traffic [41].
- PIM-SSM is simpler than PIM-SM. This is due to the nature of the source rooted

tree. This rooted tree does not require the establishment of an administrative RP router which is responsible for knowing all multicast sources. Hence, there is no need for shared trees and the complexity of the network is reduced.

- Receivers are less exposed to various security attacks such as denial-of-service attacks (DoS). This is due to the fact that the receivers are the nodes that choose actively their source of multicast [42].
- Receivers establishes a listening channel described through a pair of unique (S,G). This eliminates the uniqueness of the multicast address group within a domain. In other words the same multicast group can be reused offering the option, for example, for a receiver from one group to subscribe to receive feeds from different sources.

With all of these features, PIM-SSM proves itself to be a viable choice from many perspectives. This is not only from the DS-TE point of view but for example from ensuring secureness of a multicast transmission through a network. At this stage, the most advantageous appears to eliminate the difficulty of choosing an RP node. Previously, it was a cumbersome operation to find an optimal place for that node in regards to the shared path's construction by the multicast distribution path, and from which later to switch to a shortest path distribution from source towards receiver. The necessity to choose an RP node continues to remain a required step within IPv6 enabled MPLS networks as it is shown in [50]. Nevertheless, IPv6 networks support natively multicast.

From the MPLS perspective, one should recall based on the conclusions made in Section 2.4.2.3, that the transitional changing state from (*,G) to (S,G) for a router is practically eliminated since the operation is no longer a required operations performed by the RP node. Hence, the dilemma of determining the method to create and distribute new labels for the outgoing interface belonging to a recently transitioning router in order to forward multicast traffic is also eliminated.

3.6 Solution's architecture and behavior

With this last missing coupling link capable *to turn on* a multicast traffic distribution over a Diff-Serv aware MPLS-TE domain, the next topic of discussion is the method in which all of these elements will be utilized in their individual role within an MPLS network to solve the problems exposed in this document.

The first part of this section presents a general overview of the proposed architecture and later it presents the interaction between the main modules.

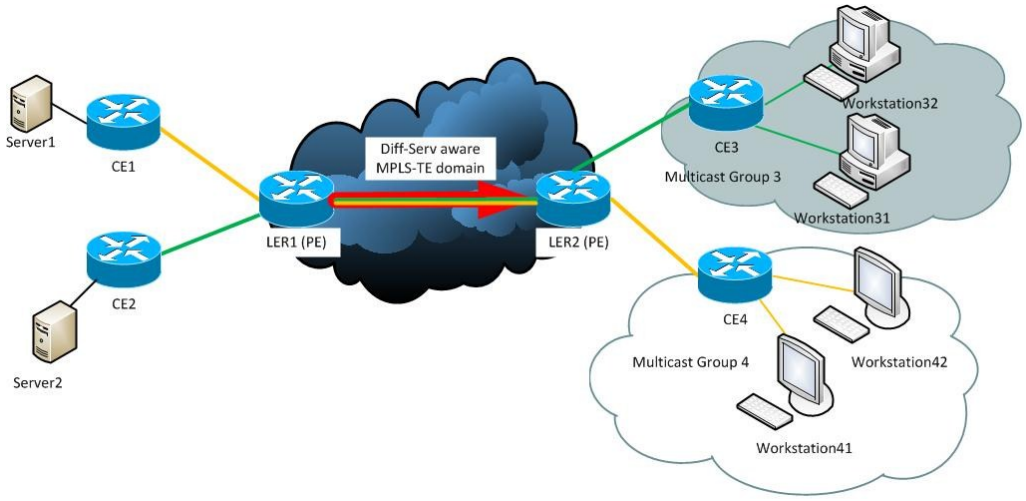


Figure 3-6 PIM-SSM multicast traffic distribution within Diff-Serv aware MPLS-TE domain.

Figure 3-6 presents a hypothetical and simplified view of a typical network architecture that has enabled Diff-Serv awareness. This Diff-Serv awareness starts from the multicast source up until the last desktop receivers and includes over the MPLS-TE domain.

For simplicity in Figure 3-6, one could recognize that the enabled and ready for SSM multicast type of transmission multicast servers are shown as customer edge routers (CE1, CE2). They are located inside of the same autonomous system (AS). However, the SSM multicast protocol is not limited to the intra-domain usage, it could be enabled for

inter-domain multicast transmission as per [43]. This would allow clients (receivers) from different autonomous systems to subscribe to a source located in a separate AS. This type of network deployment will demand the use of a multiprotocol BGP (MBGP) to maintain IP multicast connectivity over multiple ASs. For the concepts description simplicity, all further explanations will assume that the network is deployed intra-domain within the same AS.

Alternatively, clients subscribe on a per-channel basis to the available multicast traffic sources (Server1, Server2) and for which their services is provided according to the QoS configurations enabled at the customer edger router (CE3, CE4). The QoS provided by the terminal CE routers follows the conformance regulated by the SLA agreements agreed between the clients and network provider.

The distribution of multicast traffic enhanced with QoS awareness could be extended and split at a later hop, and inside of the customer domain as shown in Figure 3-7

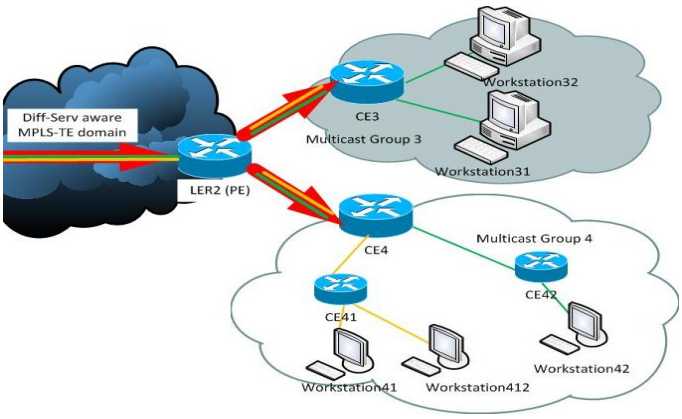


Figure 3-7 Extending aggregated traffic trunks over new hop to forward further multiple Diff-Serv multicast traffic

When this occurs, both multicast traffic propagated by (Server1, 2) could be offered inside a particular customer's domain (e.g. inside of the domain where CE4 belongs). Hence, the variety of the service provisions is no longer restricted only at the edge of one customer router. This translates into receivers connected to CE4, are able to subscribe to

both multicast traffic channels offered by Servers 1, 2.

The following discussion will focus on the ingress LER1 and the egress LER2 provider's edge (PE) routers. These routers will have a major impact on how the multicast traffic will be driven over the MPLS-TE domain and ultimately distributed towards the CE routers.

3.6.1 Applying QoS policies

Referring to the problem definition of this research, one could recall that one of the requested items was to enable QoS support within the proposed future DS-TE domain. Therefore, this section will discuss tthe possibilities for deploying QoS policies within a DS-TE enabled network.

Section 2.5.1 enumerated the data planes tools as they are used within an IP QoS model; classification, marking, policing, queuing and scheduling. Throughout this section the method and location where they will be applied in practice will be discussed. Usually, the location where they are applied will determine the type of QoS policy required and it defines the proper combination of the tools. Typically there are edge policies and core policies relative to the edge node which apply QoS policing at the entrance within a QoS aware domain.

With the Diff-Serv IP QoS model selected to be deployed in this document's case due to its advantages previously presented the following policies can be identified in the Figure.

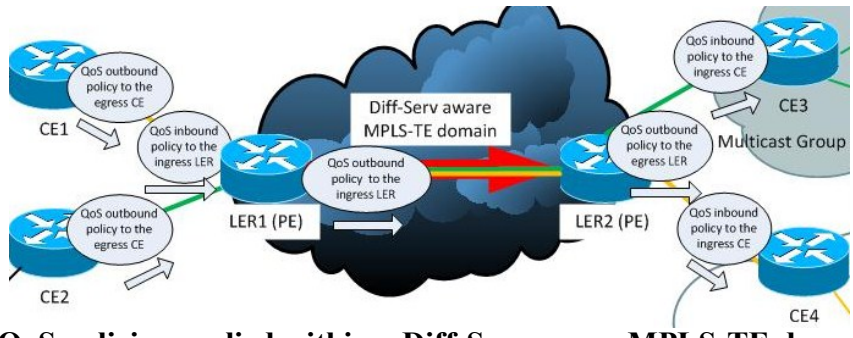


Figure 3-8 QoS policies applied within a Diff-Serv aware MPLS-TE domain

QoS applied inbound on the ingress LER1, QoS applied on the outbound interfaces of the ingress LER1, QoS applied on the outbound interfaces of egress LER2. It could be observed that there are typically no QoS policies applied on the inbound interfaces of the core routers (i.e. located inside of an MPLS domain). The main reason for this is that it is not typical to re-apply a new marking (DSCP) and classification within a MPLS domain.

Taken separately for each QoS policy applied, the following QoS functions could be identified according to Figure 3-9 for the QoS applied inbound to the ingress LER1

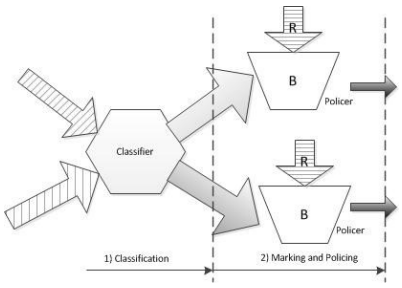


Figure 3-9 QoS applied inbound to the ingress LER.

As per the Fig. 3-9 on the ingress router interfaces:

1. A *classifier* component of the QoS data plane welcomes the incoming multicast traffic. This operation classifies the packets and assigns them based on their traffic class.
2. Marking and policing are applied to their classes.

The DSCP and the MPLS EXP field are the principal fields used for IP and MPLS packet marking and classification. The procedure follows one of the methods mentioned in Sections 2.6.1 or 2.6.2 but is primarily the method mentioned in Section 2.6.1. This method is the PHB inferred from an EXP field. It should be noted that the basic mechanism of Diff-Serv for IP multicast is similar to the IP unicast with regards to the IP header's content. The DSCP value of the replicated multicast packet is the same as for the incoming packet of the same multicast group. Hence, the applied PHB for the replicated multicast packet is the same as for the incoming one.

The policing operation on the inbound interface of an ingress router is supported only per-class in order to perform conditioning. It might be accomplished according to one of the method presented in Section 2.5.1.1 by using single rate three color marker or two rate three color marker.

A QoS inbound policy applied on the inbound ingress LER1 *marks (stamps)* the packets and *classifies* customer's traffic into a limited number of traffic classes known as *"behavior aggregates"* as previously mentioned in Section 2.5.2.2.

The second type of policy is the QoS outbound policy applied on the ingress LER1. Based on the Figure 3-10 this consists from the following QoS components:

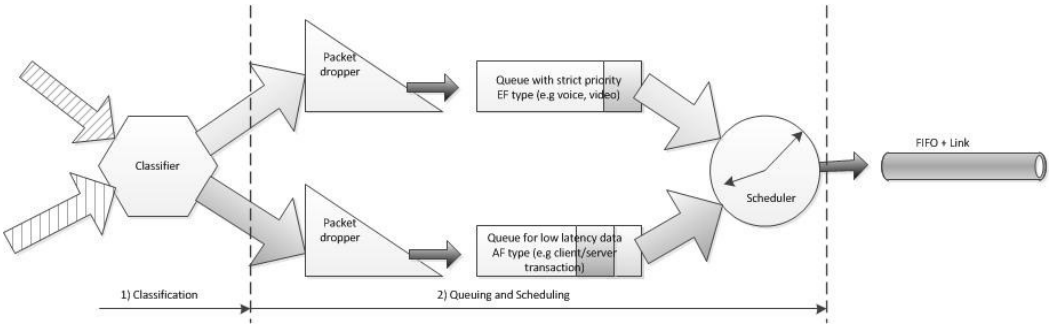


Figure 3-10 QoS applied on the outbound policy to the ingress LER1 or egress LER2 router

A typical QoS implementation used on the outbound interfaces of the LER1 router will keep the classifier functionality as described previously on the inbound case to LER1. It should be noted that it will not use policer functionality applied to weighted bandwidth queues (see 2.5.1.1). The policer functionality is replaced by the queuing mechanism which may drop packets when packets arriving at the entrance of the queues have a higher rate and start to build up queues with a larger depth than the capacity of the queues (see Section 2.5.1.1). Later, the scheduler's functions has to be involved in order to shuffle and schedule packets departures in order to keep conformance according to the imposed Diff-Serv restriction. Packets aligned at the exit of the scheduler, according to

the scheduler's decision, are previously placed onto a FIFO queue to be physically placed on the connection's link. Its role is to provide pre-buffering for packets before going onto the link's hardware to maximize the link's throughput. A similar type of QoS outbound policy is applied to the egress LER 2 as shown in Figure 3-10.

The other type of a typical policy implementation that could be found at the outbound interfaces to the egress CE routers level is a combination of the QoS applied inbound to the ingress LER1 as in Figure 3-9 and QoS applied on the outbound policy to the ingress or egress LER of an MPLS as in Figure 3-10.The QoS applied policy on the outbound policy to the egress or ingress LER of an MPLS domain router is enhanced by adding some policers that are applied to the weighted bandwidth queues as in Figure 3-11.

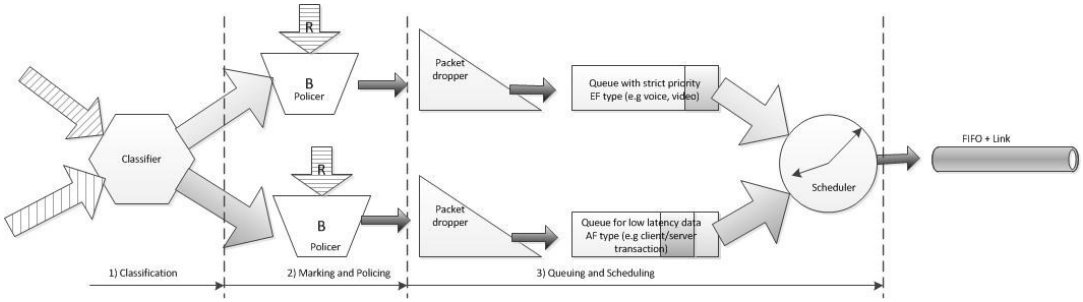


Figure 3-11 QoS applied on the outbound policy to the egress CE router

This policing operation would be necessary as part of the traffic conditioning. This results in the incoming encountered non-conformant traffic being policed at the beginning without implying other round-trip mechanism developed by tail drop or random early detection procedures.

3.6.2 Mapping traffic

Once the rollup plan of the QoS policies applied across the proposed architecture is known, the sequence of events appearing within this DS-TE domain is presented.

The three main elements that contribute and mold the foundation for this proposed architecture are: the PIM-SSM protocol, traffic trunks and LSP within MPLS Diff-Serv domain.

Based on the overview of the PIM-SSM protocol in Section 3.5.1, the role of this multicast distribution method, within this actual proposed architecture, is reinitiate the sense of controllability of a created multicast source distribution tree. It is no longer required to search and discover potential multicast sources. The request now comes directly from the receivers (listeners) belonging to a multicast group (G). These receivers are expressing their wish to listen (subscribe) to a channel broadcast by a multicast source (S), which, in a multicast scenario, is expressed through a pair of (S,G). With the creation of controllability of the source tree distribution's, an operator or an automated system is now able to decide the place of the source tree. This gives the ability to control the path of the multicast distribution tree. In parallel, the multicast protocol deployment adds the ability to establish a common FEC for unicast IP packets with different contents. Using multicast's feature this FEC will group all IP unicast packets under the same destination network layer addresses which is the desired multicast's group address.

In conclusion, the combined benefit of using the PIM-SSM results from having the ability to assign a common FEC for the packets and the ability to place them accordingly onto a desired, specific routable and controllable path. In order to reach this goal, a couple of transformations must be applied to the multicast traffic's packets. Furthermore, conditions must be met in MPLS-TE with Diff-Serv awareness case, because the required path is an LSP path which has to take into account network's resources.

Traffic trunk, which was previously introduced in Section 3.4, is the second important element of the proposed architecture. The outline of the packets transformation process into traffic trunks is important because trunks are laid onto on the LSPs paths.

As previously mentioned, the support of QoS within a network must have the following condition for traffic treatment differentiation: the ability to mark packets with codepoints (i.e. DSCP). Later, packets with same DSCP were grouped in "behavior aggregate" for which an external behavior treatment called PHB was applicable at the node (router) level according to [44]. The result of this PHB treatment developed into a collection of packets for which their DSCP mapped to the same PHB called traffic aggregate as per [46]. Traffic aggregate is a more generalized definition for behavior aggregate. . It is from this location that traffic aggregates are manipulated further according to [34] in order to be mapped into traffic trunks.

Otherwise said, before being able to manipulate traffic trunks, packets are subdued to a transformation. This transformation added a DSCP codepoint to the packets for traffic differentiation purposes, grouped them into behavior aggregates, mapped them under a specific PHB and finally grouped them under traffic aggregates. Figure 3-12 illustrates the steps of the transformation through which the packets are grouped and finally classified into traffic trunks at the router level.

According to [34] a network can service multiple traffic aggregates in one of the two ways:

- The first option, maps traffic from all traffic aggregates of a given FEC onto a traffic trunk containing all the traffic aggregates corresponding to all PHBs of a given per hop scheduling class (PSC). This traffic trunk places all traffic aggregates onto a single common LSP and applies a single common set of constraints.
- The second option splits traffic aggregates of a given FEC into multiple traffic trunks. Each traffic trunk contains all the traffic aggregates corresponding to all PHBs of a given PSC. This option splits traffic based on "classes of services" [34] into traffic trunks which could be transported over separate LSP that may follow different paths through the network. DS-TE can enforce the traffic engineering

constraints on these split traffic trunks. These constraints limit traffic aggregates transporting a particular traffic aggregate corresponding to a PHB of a given PSC to a specific percentage of a link capacity. It should be noted that this feature was the subject for defining Diff-Serv bandwidth models.

An important aspect that has to be considered after the creation of traffic trunks is the resultant mapping process onto the LSP defined by the following:

- Traffic trunks with a single exit point which share common internal path can be merged to form a single sink tree as per the definition given in [38].

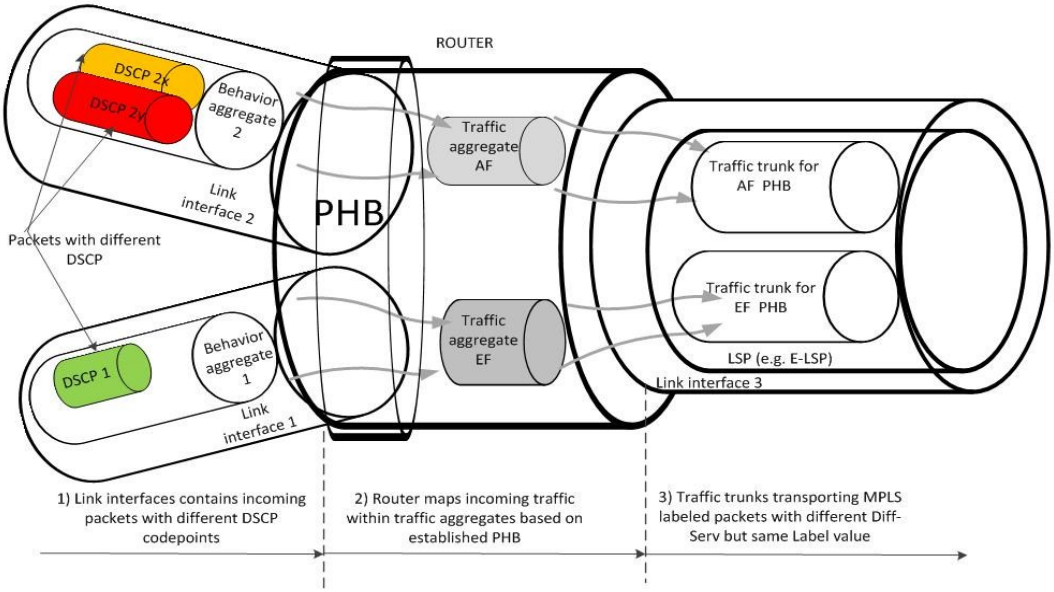


Figure 3-12 Visualization of the transformation process for the multicast incoming packets marked with different DSCP for QoS (e.g. Diff-Serv) support, that in the end are grouped under different traffic trunks.

The result of this conclusion is important because the introduction of the traffic trunks definition allowed the concept of labeled packets from separate trunks, each of them transporting different traffic classes, to share the same label on a shared LSP path. This shared LSP path ends at a common exit router point as per Figure 3-12.

Following a better understanding of the traffic trunks' behavior and characteristics, it is

evident that the proposed architecture could make use of a traditional Diff-Serv MPLS model. The MPLS model involved to develop this proposed architecture is a traditional MPLS Pipe Model, or a variation such as the MPLS Short Pipe Model, both of them being well defined within the document [31].

The third element used in the proposed architecture and its surrounding concepts, or the LSP path, was previously mentioned within this document. In the context of a multicast traffic that was enhanced with QoS (specifically with Diff-Serv type), and which is required to be forwarded further according to Figure 3-12, the LSP embeds traffic trunks delivering multicast labeled packets that were classified into "traffic classes". When compared to the case without traffic trunks, where packets were delivered as labeled packets without any QoS support, the difference is more evident. The QoS support is added through codepoints into the IP header of the packets. Delivery was simpler since all of the traffic was just aggregated into a general traffic aggregate that ended into a single traffic trunk. As was previously mentioned, it was also unnecessary to understand the method in which to setup extra steps (i.e Diff-Serv setup, or traffic engineering settings) because most of the packets were considered and treated as part of the best effort type of traffic, the default type of service.

3.6.3 Solution architecture

The following section presents the main components of a multicast traffic distribution over a Diff-Serv aware within a MPLS-TE domain architecture which utilizes the components that were outlined in the previous section. Only the relevant interaction between those components that have direct effect related to this current research will be present.

It should be noted prior to the presentation of the proposed architecture that the view of the solution's architecture is placed at the entrance ingress LER into a Diff-Serv aware MPLS-TE domain. The selection of this point was concluded by the observation of

multiple, but required, operations that are performed by the ingress LER at the entrance of the MPLS domain. The majority of the functions executed by a generic ingress router are standardized for all routers. A few examples of functions would be the routing functionality or the label forwarding operation which is specific to a MPLS enabled router. The burden of the operations is brought over to the router by a few methods:

1) by the Diff-Serv classifier that is in charge of packet coloring
2) by the CSPF algorithm required to define the shortest path under the influence of the network resource's constraints
3) by the extended RSVP-TE signaling protocol that is in charge with the resource reservation path for the purpose of LSP's path establishment
4) by the newly added member, which is the PIM-SSM that is delegated with the multicast traffic distribution.

Therefore, all these major functions are summed up within an UML components diagram according to Figure 3-13.

As per Figure 3-13, every node router can be conceived as containing an *IP module* that performs a "controller role" which is UML terminology. This *IP module* uses different functions that are declared as interfaces (e.g. Diff-Serv, PIM, TE).
The applied functions from the interfaces are not visible unless the components are represented as specialized classes. These specialized classes use the <<interface>> stereotype notation. The appealed functions are implemented as regular methods belonging to a class (e.g. setDSCP(), PIM-SSM(), PIM-SM(), setTrafficTrunk()).

Based on these observations, *IP Module* uses *Diff-Serv* <<interface>> of the *IP packet*'s module to setup the codepoint DSCP value inside of an IP packet. With the help of the *PIM* <<interface>>, the *IP module* configures the multicast group destination address within a IP header's packet.

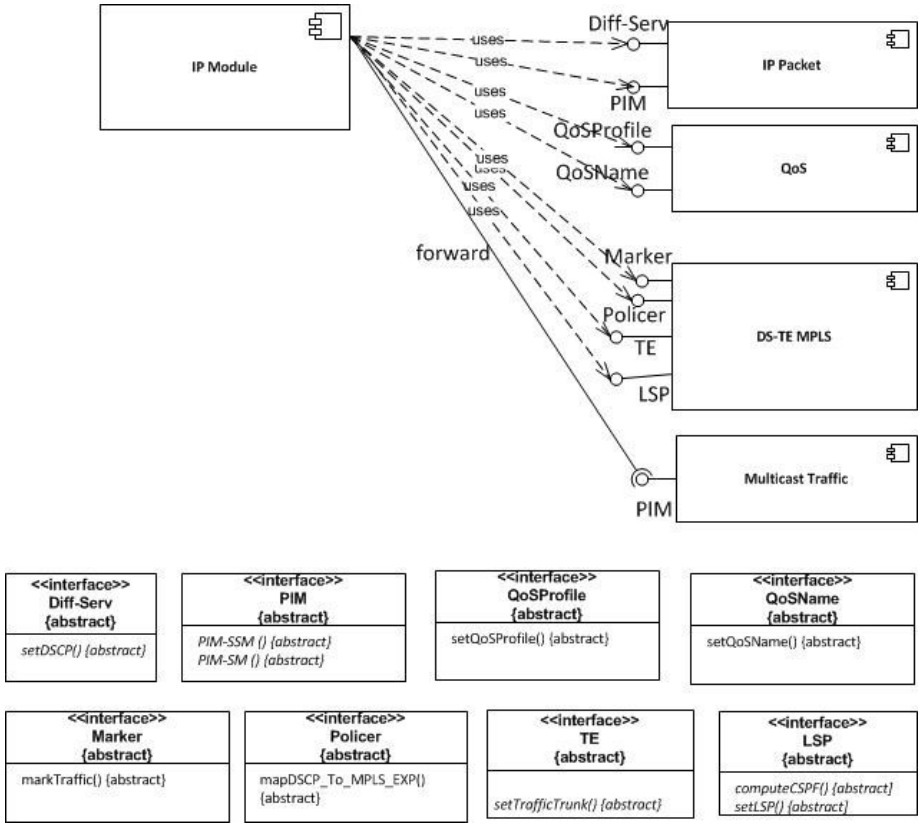


Figure 3-13 Architecture for IP Multicast with PIM-SSM within Diff-Serv aware MPLS-TE presented with an UML components diagram

The role of this *QoS* module is to define the queuing profiles that are going to be applied on the interfaces. For this purpose the *QoSProfile* <<interface>> could apply, for example, a WFQ profile or an RSVP type of profile. The preferred profile used within this document is a WFQ profile. The selection of this profile is based on the advantage of WFQ queuing discipline introduced in Section 2.5.1.1 d). Within a queuing discipline, the criteria (or "scheme") that indicates how the packets are placed in a queue can be selected by using the *QoSName* <<interface>> definition. Thus far four standard schemes are proposed which are based on: Type of Service (ToS), Protocol, Port and DSCP. It was concluded that the best choice to enforce a Diff-Serv QoS and propagate this imposed QoS within DS-TE MPLS domain, is the DSCP based on scheme.

Following this the *IP module* uses the DS-TE MPLS's component functions represented by *Marker, Policer, TE* and *LSP* interfaces. As the name suggests, the *Marker* <<interface>> definition marks the already colored IP packets, based on the DSCP values, into classes of traffics.
Going forward, these classes of traffic are handed over to the *Policer*'s <<interface>> method which starts mapping the IP's DSCP towards the MPLS' EXP value. These mappings define the "polices" established according to the SLA's contract, which later are applied on the inbound and outbound interfaces of the router.

Technically the *TE* (traffic engineering) module provides the ability to setup traffic trunks by calling an instance of the setTrafficTrunk class. It is evident that this setTrafficTrunk class has to be defined in such a way that the methods responsible for the traffic trunks configuration should be aware of the Diff-Serv support requirements that were established earlier. The other important interface provided by the MPLS-TE component, and which the *IP module* uses for the purpose of laying down the path for traffic trunks, is the *LSP* <<interface>>. The *LSP* <<interface>> declares two methods that have to be further implemented; the computeCSPF and setLSP. As per their suggested names, the computeCSPF performs the shortest path computation, which has to be aware of the constraints imposed by the network resources. The second setLSP class instance performs the LSP path's configuration using one of the signaling protocols, RSVP-TE or CR-LDP.

Once the multicast traffic is established, which is usually defined at the source router, the *IP module* will be able to forward the multicast traffic. For this purpose the *Multicast Traffic* component provides the *PIM* <<interface>> which does not need to define the same method signature as the one used for IP packet. This implementation of the *Multicast Traffic* component is more extended and it is the one that is presented in Figure 3-13. This is because in UML, the interface implementation depends on the component's provision requirement. The *PIM* defines two multicast protocols and the protocol

discussed in this document is the PIM-SSM. The function of the PIM-SSM protocol will be to configure IGMP v3 on the router link's interfaces, which is not included in this diagram, in order to support this type of multicast distribution traffic.
Most likely, the majority of the described components will appear at the ingress LER router. A regular enabled MPLS router will not be responsible for the traffic trunks setup or with the initial LSP signaling or paths establishment. Therefore, it is not necessary for this type of router to define them. The presented architecture handles this type of optional structural case without difficulties since the proposed UML representation suggests interfaces implementations. By default, the advantage of using these types of UML elements offers the flexibility to define optionally required structures. For example, a core MPLS router will not need to define the setTrafficTrunks or setLSP even though the specialized classes, (noted as <<interfaces>>) TE or LSP, defines them.

Thus far this section revealed the important components used to build the architecture which will be advantageous to a Diff-Serv aware MPLS-TE domain. This architecture would allow the domain to support multicast traffic distribution. At this point the architecture has been briefly outlined to clarify the participating components and their concepts. The following Chapter will focus on the architecture's keys element interactions; again the attention will be given to the components that directly influence this research's solution implementation.

3.6.4 Solutions' behavior

The behavior of the consisting blocks of the architecture will be outlined in the following Section. A well-suited methodology for this purpose is to have it presented through a set of classical finite state machines (FSM).

In order to support the presentation of the architecture's component behavior, the next FSM diagram was developed and simulated within a Matlab environment, utilizing the Stateflow module.

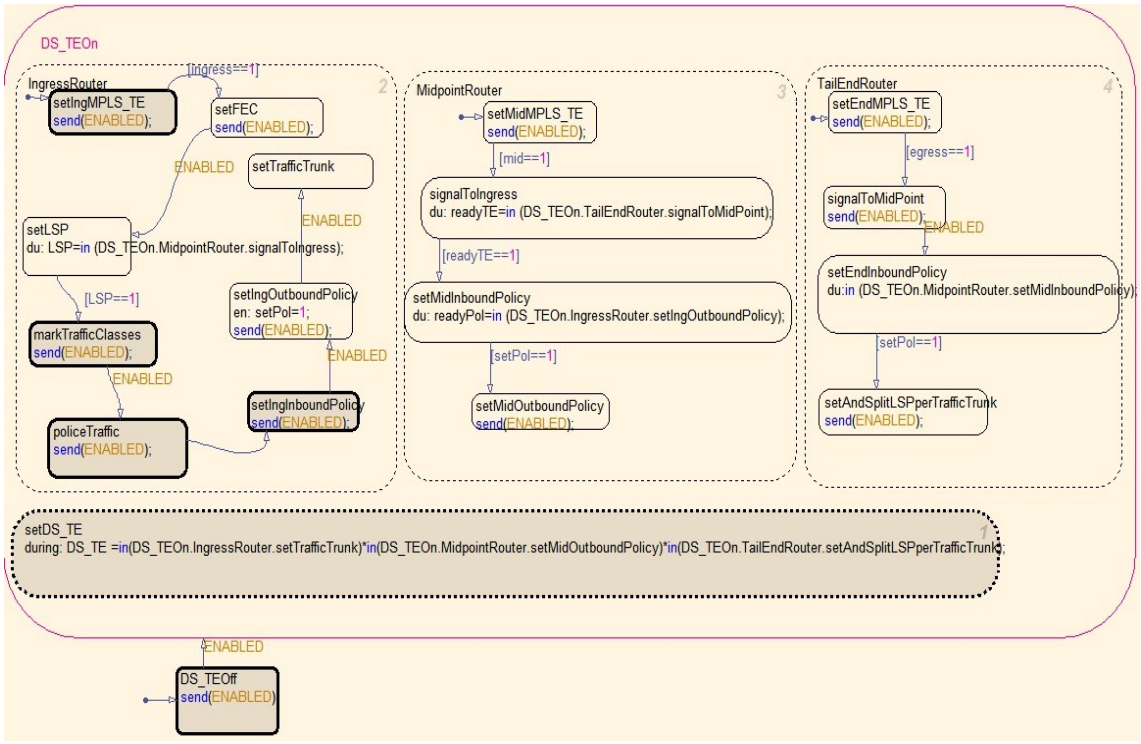


Figure 3-14 FSM chart representing the routers' states during DS-TE MPLS configuration

Figure 3-14 is a visual representation of a FSM and captures the main states required to be turned "on" on the network nodes (routers). This turned "on" state allows multicast traffic distribution over Diff-Serv aware MPLS-TE to be accomplished.

The two main states can be distinguished for a Diff-Serv aware MPLS-TE domain: *DS_TE Off* state which denotes the turned off state and *DS_TE On* state that configures the DS-TE within an MPLS domain.

Once the setup command is triggered for a DS-TE configuration over a minimum set but with the required routers consisting of Ingress, Midpoint, TailEndRouter, the following phases can be distinguished:

1) In parallel, the Ingress, Midpoint and TailEndRouter will start to enable MPLS-

TE functionalities within each of the *set<Ing,Mid,End>MPLS_TE* states. This step enables one of the IGP protocol IS-IS or OSPF, enables MPLS module within routers, and enables one of the extended signaling protocol, RSVP-TE or CR-LDP.

2) The *IngressRouter* will set the FEC within *setFEC* state. This step assigns a destination address for the incoming multicast traffic towards a pre-established Multicast Group.
3) The *IngressRouter* will move to the next *setLSP* state that initiates the LSP setup. It is evident that this is a process signaled among the routers involved with the LSP establishment to find and assure the required network resources. Hence, within this state the *IngressRouter* will wait until the *MidpointRouter* and the last *TailEndRouter* signal back the network resources readiness. Therefore, the LSP path can be established and it is ready to be used.
4) Once the LSP setup is ready (this is signaled through an [LSP==1] event condition) the *IngressRouter* moves to the next state *markTrafficClasses*. This state marks classes of traffic according to the SLA's contract.
5) Next the *IngressRouter* sets the traffic policy inside the *policeTraffic* state.
6) Within the next two states, *setIngInboundPolicy* and *setIngOutboundPolicy,* the *IngressRouter* applies the inbound and outbound policies on the router's interfaces. Inside the *setIngOutboundPolicy* state the policy application status is flagged as finished through an output data *setPol*. This output signal is expected by the *Midpointrouter* inside the *setMidInboundPolicy* waiting state and by the *TailEndRouter* within the *setEndInboundPolicy* waiting state. The reason for implementing these waiting states is the application of identical inbound/outbound policies on the *MidpointRouter* interfaces and inbound policies on the TailEndRouter interfaces. Therefore, these policies are identical with the previously defined outbound policy applied on the MPLS cloud facing *IngressRouter's* interfaces.
7) *IngressRouter* enters the *setTrafficTrunk* state which mainly deals with

configuring the traffic trunks along the signaled and established LSP path.

8) The *TailEndRouter* router enters into the *setAndSplitLSPperTrafficTrunk* state which reverses the mapping, policing and sets traffic trunk operations applied at the *IngressRouter*. It also separates traffic coming from the MPLS cloud and redirects traffic towards receivers based on the SLA's contract.

The status of the states that currently run within the routers is constantly monitored within a supervising state *setDS_TE*. This state will flag the finish of the DS-TE configuration setup.

The following Chapter presents the method in which this MPLS DS-TE configuration process could be integrated to develop the functionality of a possible Control Server . This Control Server could automatically deploy MPLS DS-TE configuration over an MPLS cloud and in the end start forwarding multicast traffic distribution.

4 Detailed Solution's Implementation

The previous chapter referred to the structural components of the solution's architecture. This section presents, in detail, the method in which the key components are combined, using several concepts, into a final solution to the multicast problem.

4.1 Introduction

This chapter will focus on the design of a possible multicast MPLS control server that is introduced with the intention to better describe the occurrence of states required to deploy and distribute successfully multicast traffic over an MPLS cloud with DS-TE MPLS configuration.

Initially, the algorithm of the solution for enabling DS-TE MPLS configuration multicast used by either a human operator or by an automated multicast control server was introduced.

Next the algorithm is sketched through a set of FSM diagrams that were developed in a Matlab environment to prove the concepts through simulation for all possible occurrences of states.

Finally, the whole concept is proof-tested with the help of a testbed network that is modeled within the OPNET simulation environment. OPNET is an advanced commercial network simulator that offers high fidelity mathematical models network components and in particular the MPLS modules.

It should be indicated that the entire discussion is focused only on the usage of intra-domain multicast routing protocols. The support of inter-domain is out of the scope of this document. It is not excluded but rather proposed as a future possible implementation.

4.2 Algorithm for PIM-SSM multicast traffic distribution over DS-TE MPLS environment

Using Figure 4-1 as reference, the following assumptions are considered to be true:

- The ISP authority, which offers multicast capability over DS-TE MPLS domain, associates the QoS services offering with its exiting egress routers. Each egress router E_1, E_2, E_3, … is capable of delivering all of the available service offerings O_1, O_2, O_3,…
- There is a ControlServer block, preferably a dedicated type of server who's function might also be accomplished by a human operator. This ControlServer associates the listening receivers to pre-established multicast groups based on the receivers' subscription to the available service offerings.
- Multicast sources S_1, S_2, S_3, …that are proposed to be made publicly available for receivers, are listed by the ControlServer within a distribution list.
- The previously mentioned multicast sources S_1, S_2, S_3, … are made available at the entrances within MPLS cloud into multiple ingress routers. For example S_1, S_2, S_3, … enters into ingress routers I_1, I_2, … I_n (n = maximum available ingress routers within DS-TE MPLS configured domain).

Taking into consideration these assumptions, the following algorithm is proposed as a solution to the PIM-SSM multicast distribution over the DS-TE MPLS configured domain. For a better visualization of the proposed algorithm please follow Figure 4-1:

1) Receivers subscribe to the multicast sources S_1, S_2, S_3, …
2) ControlServer creates multicast group associations based on the available ingress routers. For example, the ingress router I_1, I_2, I_3, … associates groups G_1, G_2, G_3,
3) ControlServer initiates a DS-TE MPLS configuration setup command as described in section 3.6.4 (see Fig. 3-14), between I_1, M_1, E_1. As a result, an LSP

path is signaled and a traffic trunk is configured with the head end at the entering ingress router I_1 and tail end exiting at the egress router EI_1 .

4) At this stage the multicast traffic can start to flow. If the capacity of the bandwidth is at a maximum level and all interfaces of the egress router EI_1 or ingress router I_1 are fully wired for multicast traffic distribution then the situation is signaled to the ControlServer and one of the sequences of operations is taken. The next steps, from 4a) to 4e), are followed as long as the pre-established contract for service offerings are not diminished. If the service offerings are diminished the process continues with step 5.

   a) In case the egress router EI_1 reaches the maximum capacity then a new egress E_2 router is inserted.

   b) ControlServer initiates a new DS-TE MPLS configuration setup command between I_1, M_1, E_2. The newly added E_2 will be able to deliver the pre-established service offerings O_1, O_2, O_3,…

   c) More available egress routers are added to the midpoint M_1 (or LSR1) router, until M_1 wires all of its available outbound interfaces

   d) When M_1 wired its last available outbound interface, the ControlServer is made aware of a new midpoint router addition request.

   e) A new M_2 midpoint router is inserted between I_1 and M_1. ControlServer attaches a new egress router E_{21} and initiates a new DS-TE MPLS configuration setup command between I_1, M_2, E_{21} routers.

5) The process of inserting new midpoint routers continues until either there are no more available egress routers within the analyzed MPLS domain or the ingress router reaches the maximum bandwidth capacity on its outbound interfaces. These outbound interfaces are enabled for DS-TE MPLS configuration and are involved in the multicast traffic forwarding process.

6) If the ingress router I_1 can no longer ensure an established throughput dictated by

the pre-established contracts, then the addition of a new ingress router I_1 is required to the MPLS domain. Following this addition the steps 4a) to 4e) are repeated. The same multicast sources Mcast1, Mcast2,… are fed towards this new ingress LER.

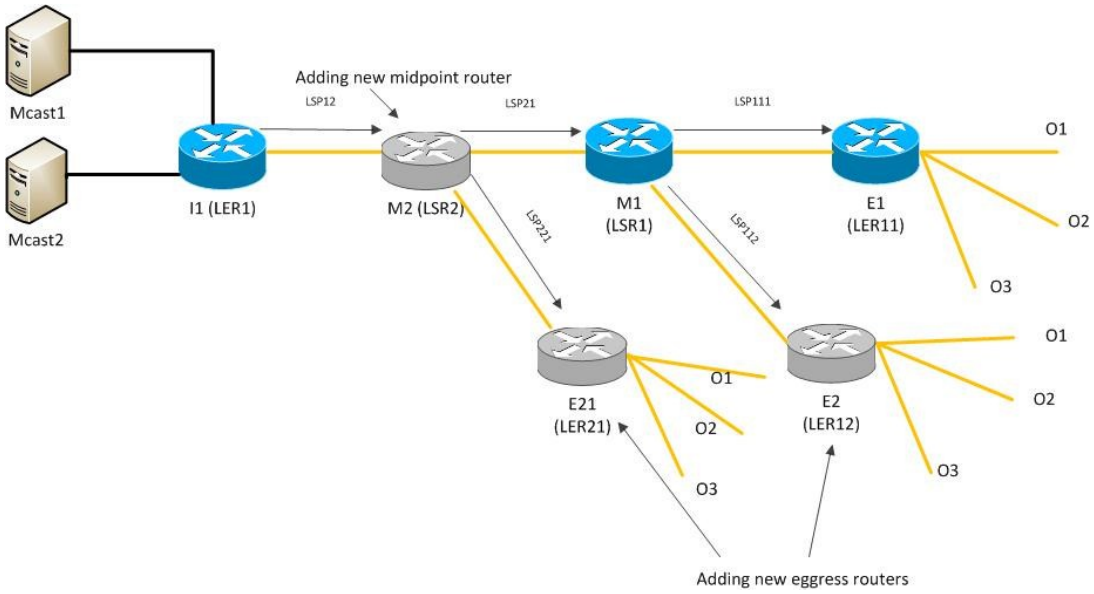


Figure 4-1 The process of adding new egress router E2, or a new midpoint router M2 with egress router E21

Reviewing the previously defined algorithm and Figure 4-1, it should be noted that the previously established LSPs (e.g. LSP111, LSP112) are no longer required to be re-configured with each new addition of multicast receivers. Therefore, the existent LSP paths towards the exit egress routers do not need to be re-signaled and re-established again. Hence, the connected and configured consumers are not affected by the process of adding new multicast receiver clients.

The following paragraphs present the connections and communication established between a ControlServer and routers from an MPLS domain to enable DS-TE MPLS configuration. The purpose of this is to distribute multicast traffic over the analyzed MPLS domain.

In order to visualize the connections between the proposed ControlServer and the elements of a DS-TE MPLS (refer to Figure 3-13) configured domain, the previously presented algorithm can be synthesized using the Matlab Simulink model shown in Figure 4-2

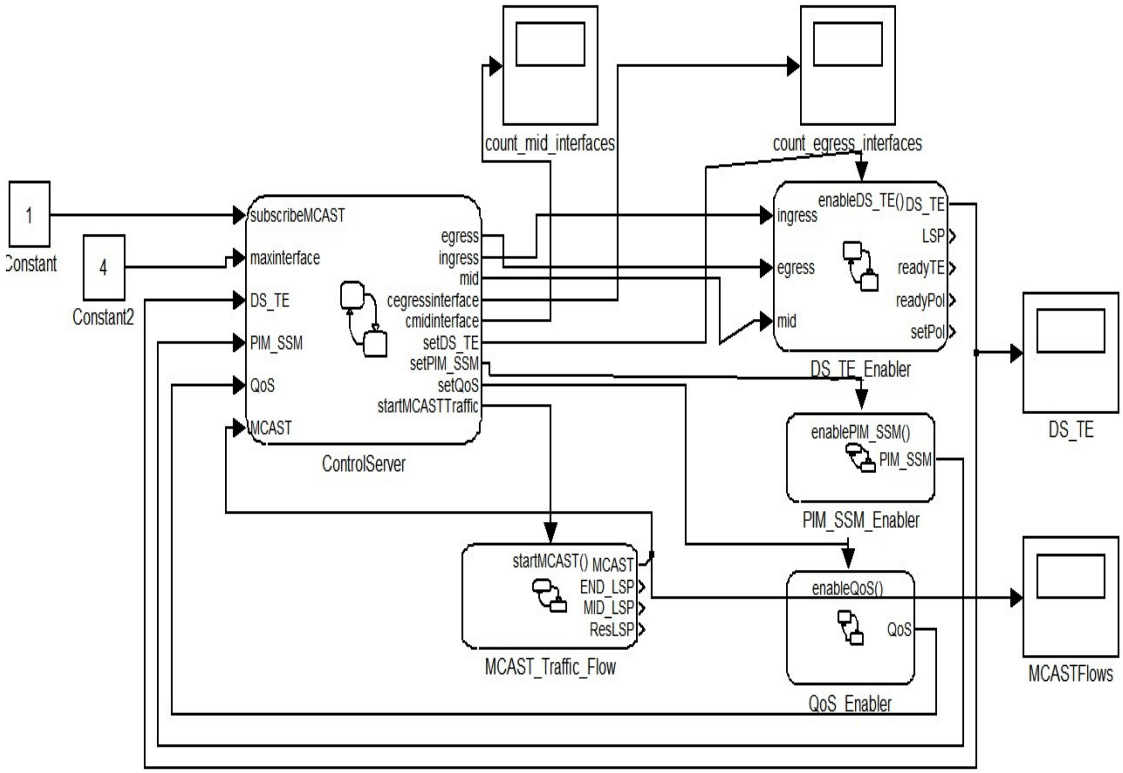


Figure 4-2 Matlab Simulink model representing the connection between a ControlServer block and elements of a DS-TE MPLS configured domain.

One could easily recognize the *DS_TE_Enabler* block that is responsible for the DS-TE MPLS configuration. Its internal functionality has been presented through an FSM diagram according to Figure 3-14. The *DS_TE_Enabler* block produces the value "1" on its output only when the MPLS domain consisting from an Ingress, Midpoint, and Egress router were enabled for DS-TE MPLS ability. *DS_TE_Enabler*'s output value is monitored through a scope block *DS_TE*.

The *ControlServer* block implements the previously described algorithm. It primarily coordinates the commands between the model's blocks in such way that the multicast

traffic can start flowing over a DS-TE MPLS domain. For a better understanding of the commands sequences, the FSM diagram shown in Figure 4-3 presents the internal functionality of the block. The *ControlServer* is triggered through a subscription to the multicast source request command (*subscribeMCAST*) received from a receiver which would like to listen (subscribe) to a channel. This channel subscription is represented through a pair of multicast sources and destination groups (S,G). For the simplicity of the model, it is assumed that all routers within an MPLS domain have exactly the same maximum number of outbound interfaces possible to enable DS-TE MPLS configuration. The ControlServer, shown in Figure 4-2, has an input that is connected through a constant that enters a value of *maxinterface* = 4.

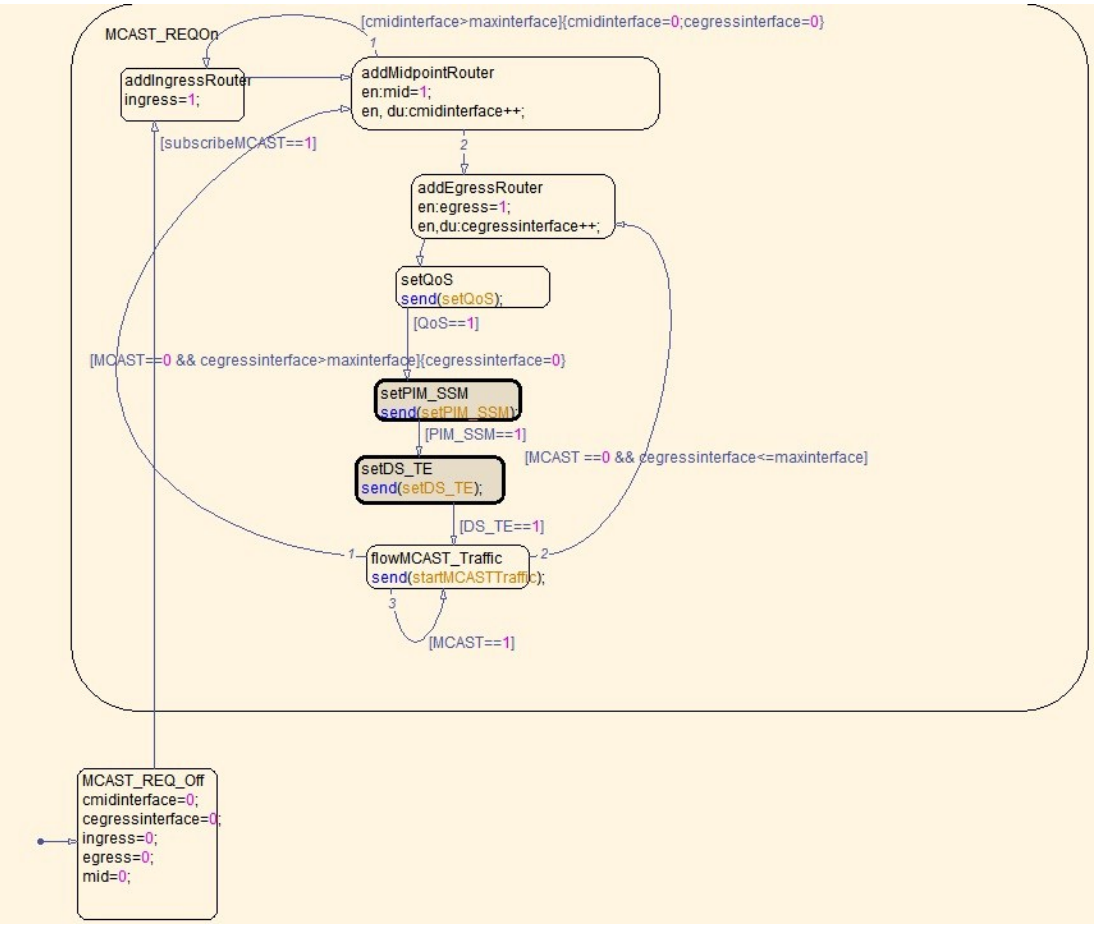


Figure 4-3 FSM chart for the ControlServer's states

Once the ControlServer is switched into the *MCAST_REQOn* state (or it is turned "On") the ControlServer will start sequentially chaining up the available Ingress, Midpoint and Egress routers from an MPLS domain. The process of adding them up to the chain is triggered by default transitions and is marked one by one in each of the following states: *addIngressRouter, addMidpointRouter, addEgressRouter*. This marking occurs by setting the entrance output variables *ingress*, *mid* and *egress* to "1". The values of these outputs are propagated to the entrance of the *DS_TE_Enabler* block and will represent a minimal condition in order to be able to setup a DS-TE MPLS configuration.

Each time the *addMidpointRouter* and *addEgressRouter* states are visited, the output variables *cmidinterface* and *cegressinterface* are incremented due to the actions executed at the entrance while in these states. These actions are described through the following *en,du:cmidinterface++* or *en,du:cegressinterface++* statements. These output variables are used as counter variables and keep the current number of used interfaces.

With the routers ready to be configured, the *ControlServer* initiates the steps required to set QoS on the routers interfaces. This process is accomplished within the *setQoS* state and is represented through the FSM diagram in Figure 4-4.

The process to enable QoS is straight forward and primarily requires it choosing a profile and selecting the manner in which to place packets in the queue.
Therefore, there are two sequentially executed states that could be named as *setTypeOfQoSProfile* and *setQosName*. Both states are executed in parallel at each node router required to be made aware of Diff-Services.

As was previously mentioned in this document the WFQ type of QoS profile was selected and the scheme used to place the packets in the queue on the routers' interfaces was determined to be DSCP.
The "watch on" state which expresses the readiness for QoS is tracked by the *setQoS*

substate. This substate waits until all the routers finish implementing the QoS required steps.

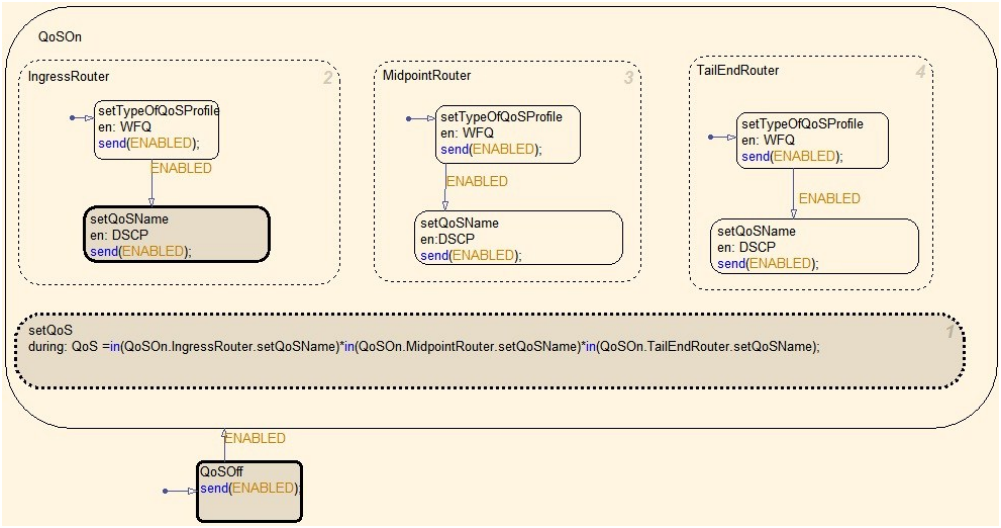


Figure 4-4 FSM chart describing QoS setting on routers preparing for DS-TE MPLS

As a result, the output variable *QoS* from *setQoS* substate acknowledges the accomplishment for setting QoS within a domain and forwards this value back to the *ControlServer* through a back loop. The *ControlServer* actions, based on this entered value when *QoS* equals "1", (as per Figure 4-3) and initiates further, through an output event *setPIM_SSM* from *setPIM*_SSM substate, the next block module *PIM_SSM_Enabler* as per Figure 4-5.

The *PIM_SSM_Enabler* block module is responsible for preparing multicast capabilities on the routers' interfaces. In order to accomplish this task, each router must enable one of the available routing protocols defined for routing IP packets towards multicast groups. Based on the arguments described in previous chapters, each router will define the PIM-SM protocol and must ensure it applies the IGMP v3 on each interface involved for multicast distribution. This enforces a PIM-SSM type of multicast traffic distribution

Due to the fact that the sequence of states to setup multicast on each router can be

implemented independently, the *setPIM_SSM* sub state checks continuously over the routers settings. The successful setup of the multicast over the nodes is signaled back to the *ControlServer* block.

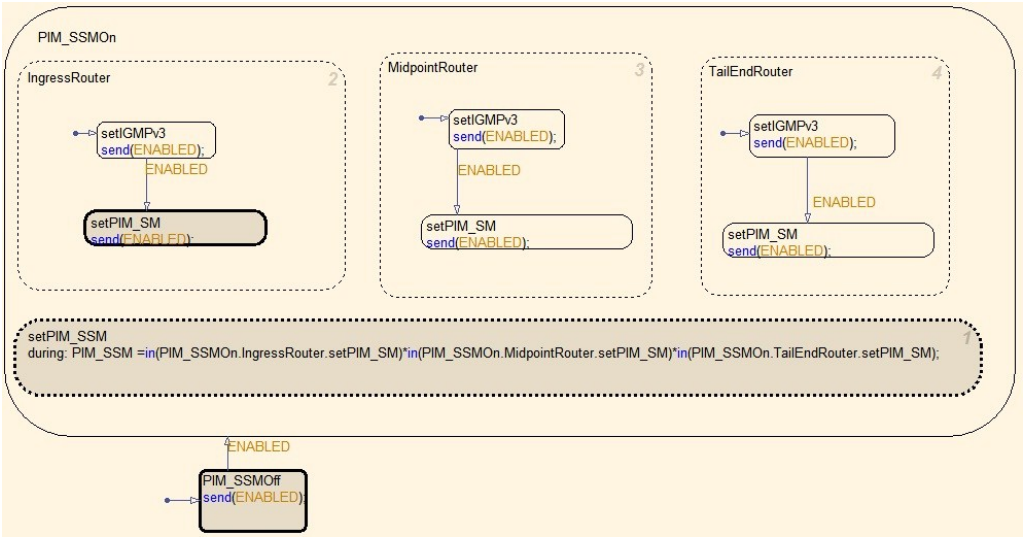


Figure 4-5 FSM chart outlining main states for configuring PIM-SSM

The *ControlServer* acknowledges the end of this operation (according to Figure 4-3) when its input variable *PIM_SSM* equals "1". This becomes the next triggering signal to start enabling DS-TE MPLS configuration within the MPLS domain. Therefore, the process is concluded with a new substate *setDS_TE* and wherein the output event *setDS_TE* triggers the presented block *DS_TE_Enabler* (Figure 3-14).

Once the DS_TE MPLS configuration is flagged back to the ControlServer as a successful process, through the input value *DS_TE* equal to "1", the *ControlServer* moves forward to the next substate *flowMCAST_Traffic*. This state describes the flowing process for multicast traffic.

Essentially in this stage (see Figure 4-6) the multicast subscription to channels is already established based on the receiver's demands and the ISP's service offerings. Multicast traffic flows towards the Ingress node (LER) from MPLS through QoS policed interfaces. The incoming multicast traffic is marked and policed within the *classifyMCASTTraffic*

state.

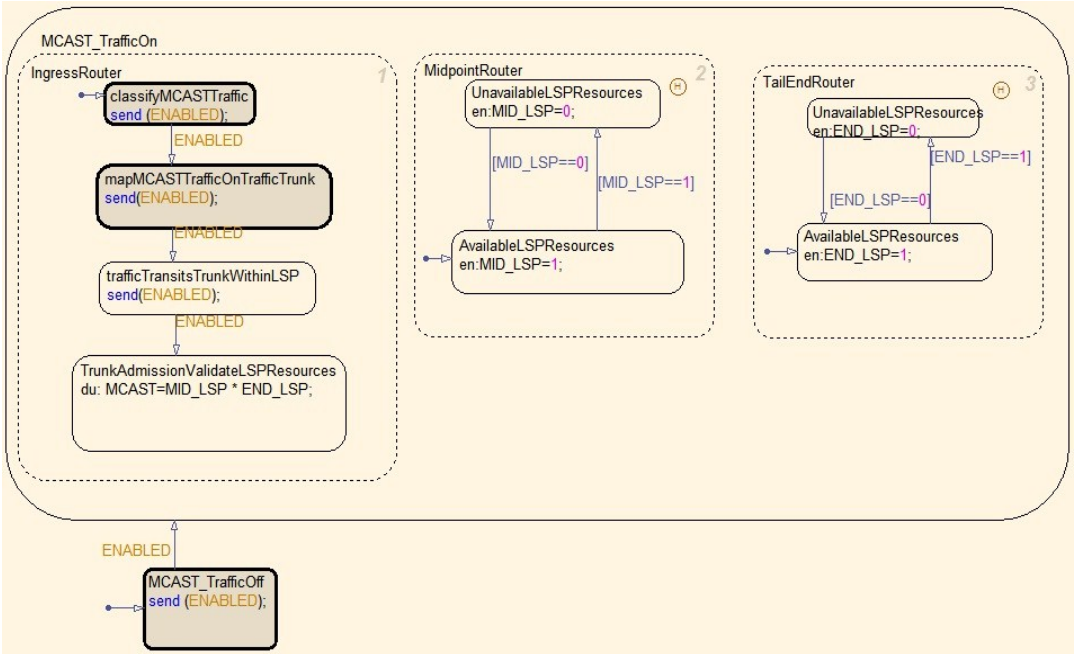


Figure 4-6 FSM representing multicast traffic flow over DS-TE MPLS domain

The Ingress router is capable of mapping and placing the packets inside the traffic trunk. This is based on the QoS definition and multicast group's destination and is accomplished within the *mapMCASTTrafficOnTrafficTrunk* state.

Within the *trafficTransitsTrunkWithinLSP* state the principal activities are developed around traffic trunks which are driven on tunnels (or LSPs). It should be recalled that LSPs could be defined either statically or dynamically with the aid of the CBR algorithm.

Next, the multicast distribution process transitions to the *TrunkAdmissionValidateLSPResources* state by default. At this moment the trunk admission control plan steps activates in order to check resources availability across an established LSP. It was seen in the previous chapters how the link admission controls use the PATH message to signal the bandwidth availability.

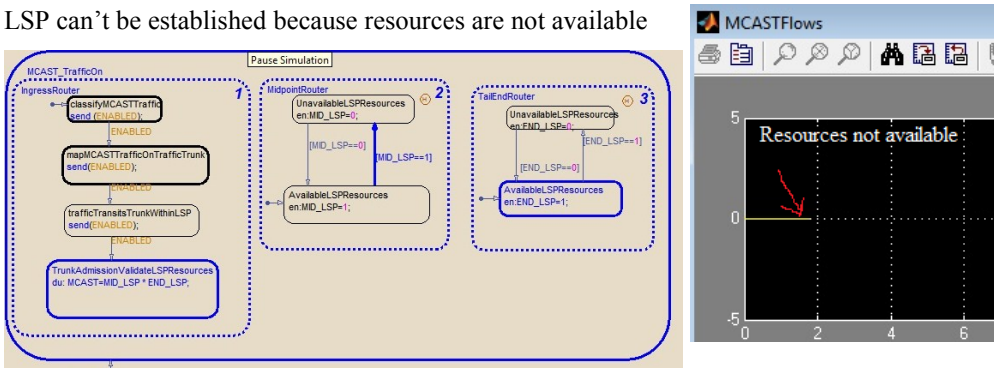
The activity within this state is modeled as a waiting state which searchs for the network

resources availability. In particular, it expects the LSP to be signaled and ready to be established on both the *MidpointRouter* and *TailEndRouter*..

In order to represent the possible cases for the available network resources, history junctions and two exclusive substates *Unavailable/AvailableLSPResources* were introduced through which both routers switch at some point in time. In actual practice these states should be triggered through external events caused by the network resources availability. They were defined this way to simplify the design of possible cases within the routers.

The following summarizes what is happening within the *TrunkAdmissionValidate LSPResources* state: Initially multicast traffic flow is unable to flow because the LSP is signaled as not established due to unavailable network resources as per Figure 4-7a).

a) LSP can't be established because resources are not available



b) LSP are established because resources are available

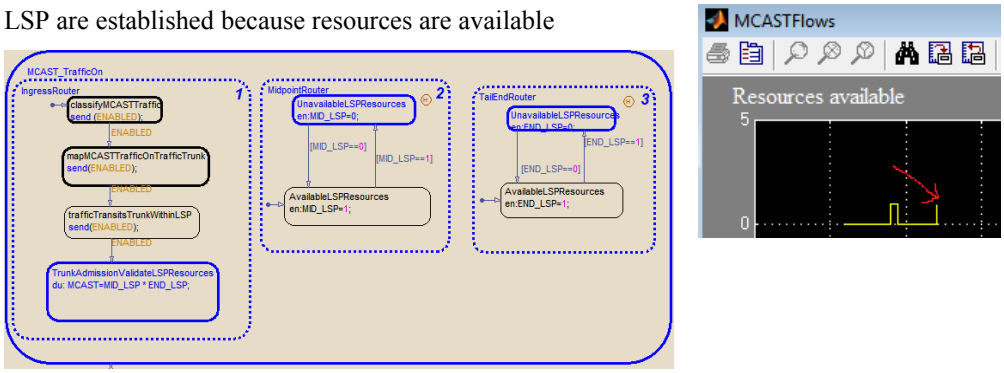


Figure 4-7 FSM simulating a) network resources unavailability and b) with resources available for LSP establishment

Routers are artificially switched from these unavailable states (*UnavailableLSPResources*) into a network with resources that are available (*AvaialbleLSPRessource*) states by setting the output transitioning conditions at the moment of entering into these activities (see *en: MID_LSP=0* and *en: END_LDSP=0*).

The incapacity of multicast traffic to flow is recorded at the ingress node from the MPLS domain inside of the *TrunkAdmissionValidateLSPResource* state within the MCAST output variable. This value is propagated back to the *ControlServer* block (see Figure 4-3), that will acknowledge it as a MCAST equals "0" condition and based on the situation will decide to:

1) Add a new egress router connected to the last midpoint router as long as there is an available interface that could connect the last midpoint router with this egress router. This is indicated when marked in a FSM with a transition of *[MCAST==0 && cegressinterface<=maxinterface]* type.

2) If there is no available interface between midpoint-egress which is marked in the FSM through a conditioned transition such as *[MCAST==0 && cegressinterface>maxinterface] {cegressinterface=0},* then simply add a new midpoint router. One should observe that this conditioned transition is followed by a reset action of the number for the available egress interfaces between midpoint and egress *{cegressinterface=0}*.

Going forward with more network resources available, the *ControlServer* starts to repeat the same steps for setting up *QoS, PIM-SSM, DS-TE* and finally to launch the multicast traffic flow process.

When the *ControlServer* reaches the *flow_MCAST_Traffic* state, and due to the existence of history junctions added to the routers' states, the process remembers the last visited states from the routers which indicate available network resources as per Figure 4-7 b). As a result, the LSP is successfully signaled and established with the head end LSP

starting at the ingress router and the tail end finishing at the egress router.
Therefore, the multicast traffic can start flowing as visualized represented by the Matlab scope in Figure 4-7 b).

The ControlServer keeps monitoring and forwarding the requested multicast traffic provided the MCAST input variable equals "1". When the network resources become unavailable, which is signaled through MCAST == 0, the ControlServer tries to escalate the situation using the previously mentioned actions. If all of the interfaces between midpoint-egress are wired and busy, then it tries to add a new ingress router. The conditioned transition towards the state addIngressRouter is marked through the *[cmidinterface>maxinterface]{cmidinterface=0;cegressinterface=0}* statement. This statement has an action condition expressed by *{cmidinterface=0;cegressinterface=0}* that resets the number of interfaces between ingress and midpoint router. It also resets the number of interfaces between the midpoint and egress router. These actions are obvious due to the fact that new routers will be added from the analyzed MPLS domain.It is also possible to reuse some old routers that have become available since some receivers are no longer listening to the multicast transmission

In conclusion, this chapter discussed the algorithm for the proposed solution of the multicast traffic distribution problem over a DS-TE MPLS domain. The functionality of a *ControlServer* through a set of FSM charts and the interaction established between this server and DS-TE MPLS architecture's elements were described in detail.

4.3 Modules interactions

Taking into consideration the known architecture previously described and the knowledge of the proposed solution, this chapter associates the identified architecture's elements with modules that are already present within OPNET.

More finely defined models which are more accurate to practical situations are desired. It must also be possible to be observed and measured these models under different traffic flows situation.

Each node (router), which is added during the modeling process, is built upon several functional modules described in OPNET through the process model.

The proposed solution makes use of the MPLS enabled router in order to engage traffic engineering functionality. Each router that is MPLS enabled also contains an IP module. This IP module embeds an IP dispatcher process that spawns the MPLS processes.

By referring to the simplified IP dispatcher state diagram in Figure 4-8, the following activities can be depicted.

In the *Initializing* state the module initializes the child process handles, prepares the packages' IP address, IP datagram, ICMP packet and important state variables. This step also collects the IP QoS attributes, creates a list of the multicast addresses that this node can handle and publishes this list through the *Registration* state. If this node is a multicast capable router, it obtains the multicast routing protocol.

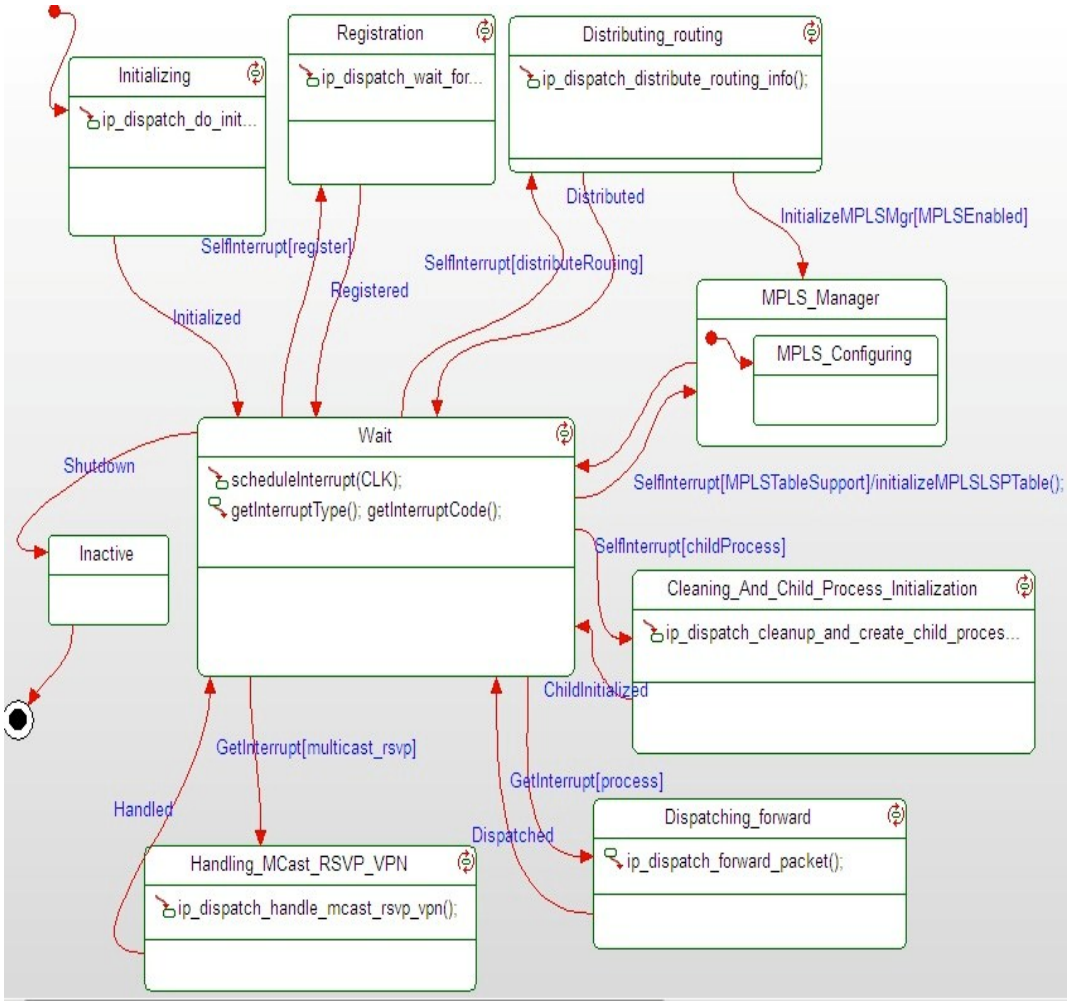


Figure 4-8 IP dispatch module

Once the initialization state is finished, the IP dispatcher module remains in the Wait state while it schedules the following states: it waits to finish all of the *Registration* process in the process registry of the IP and IPX processes and it calls an *MPLSInitialize* function belonging to the MPLS manager module (*MPLS_Manager*) to initialize the MPLS LSP table. This MPLS LSP table may have been initialized previously due to the MPLS Config object existent in the analyzed OPNET network. If this is the case then the *MPLSInitialize* returns back without continuing any other actions. If the MPLS Config is not present, but there are LSPs defined in the network, then this function will parse all of them.

The next scheduled state of the IP dispatcher module is the *Distributing routing* state. In this state the router redistributes the initial routing information among the available routing protocols in this node. This step has to be finished prior to the IP inserting these routes into the common route table itself. It parses the statically defined routing table and adds entries to the common routing table. The router initializes the route information for the available static route table. At this stage the following actions occur: the interfaces information is configured, it initializes the Internet Control Message Protocol (ICMP) message processing from this node and afterwards initiates the import of the forwarding table.

This state parses the "multicast routing protocol" model's node attribute and if the PIM-SM protocol was enabled on the node's interfaces then the router will pass the PIM-SM's process handle to the IGMP router process for each multicast enabled IP interface. Therefore, the routers' interfaces are prepared for the multicast forwarding state.

Depending on the attributes that are configured on the node, the IP dispatcher module will continue to check the RSVP setting, and if it is available, it will send a remote interrupt for the RSVP process. This must be sent after the IP PIM registration process since the RSVP process will have to be aware of the IP PIM routing table setting.

The next step that is initiated during the *Distributing routing* state relates to the proposed method to engage the multicast stream distribution with Diff-Serv characteristic within the MPLS-TE environment. This step prepares the table of label spaces for the interface and spawn process belonging to the *MPLS_Manager* module

The other available state performed by the IP dispatcher as part of the self-scheduled interruptions that should be mentioned is the cleanup operations. Memory de-allocation and child process creation on each interface are delegated by the IP dispatcher creation (*Cleaning and Child Processes Initialization*). These child processes remain busy with

the Queue management processing that might be configured with the following: FIFO, Weight Fair Quality (WFQ), Priority Queuing or Custom Queuing types.

Once the child processes for routing and queuing management are delegated and initialized, the IP dispatcher continues to listen to the possible interruption that might appear depending on the type of the packet arriving at the node.

The main task of the IP dispatcher at this point is to dispatch the packet to the right treating module and to control the task executions that are self-scheduled at the node level. The same process occurs with either a labeled packet or packets with a matching FEC description received by the IP dispatch module of an LSR.

The previously mentioned and spawned *MPLS Manager* process will take over this labeled packet, or packet with matching FEC criteria. It will forward the packet further through the MPLS domain using incoming label map (ILM) and FEC to Next Hop Label Forwarding Entry (FTN) maps. The complexity of the *MPLS Manager* function is beyond the scope of this document, but the directly affected states to the proposed solution are mentioned in the state diagram of this process model shown in Figure 4-9. According to the OPNET model description, the *MPLS Manager* represents the forwarding component and the forwarding control plane of the MPLS[9].

Referring to Figure 4-9 after the process is spawned by the IP dispatch module, the *MPLS Manager* modules enters into an initializing state that is beyond the state variables initialization. It obtains a list with the parsed FECs specifications, gets pointers to the label space table, stores information about each interface, initializes the LSPs list, initializes the EXP to PHB maps that are used for this node, creates lists with the used flows and hashs tables for used NHLFE. This state also parses the LSPs passing through this node and creates ILMs for nodes that are not ingress LER. Prior to switching into a *Wait*-ing state, it transits into a temporary initialization state *Initialize Signaling Protocol*

which invokes a self-scheduled action to check for any constraint routed dynamic signaling protocol CR-LDP. If this node is an ingress LER, it checks for any LSP starting from this node and parses the TE configurations set at this ingress LER.

.

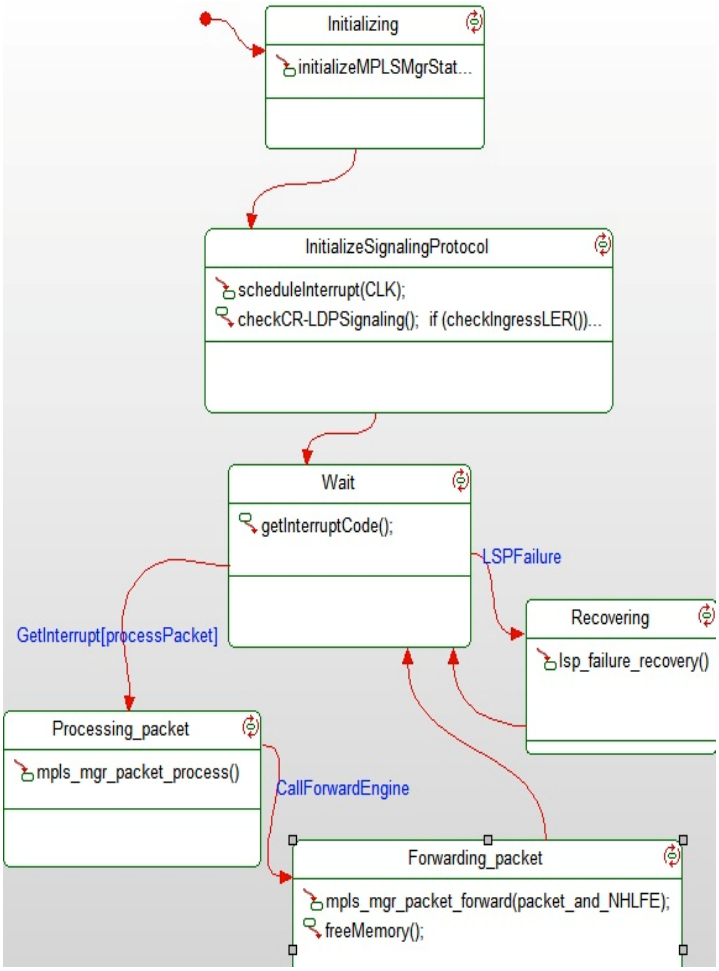


Figure 4-9 MPLS manager module

Once it is transitioned into the *Wait* state, the process model waits for interruptions that could be triggered by the IP dispatcher module. These interruptions either redirect the received packet without any other processing, or wait for the interruption triggered by any LSP failure. If this failure occurs the node must determine a different recovery LSP path.

The call from IP dispatcher's module is passed to the *mpls_mgr_packet_process* action. This checks for the validity of the packet, reads the incoming interface index and initializes the label space index for the incoming interface. It also reads the incoming stream, and if the label is not yet initialized, maps the incoming packet into a FEC. If a valid FEC is found then it starts to search for an outgoing label which will have to go out through an MPLS capable interface. Therefore, the *MPLS_Manager* module tries to map an NHLFE for the transferred FEC name and start the "do push label" operation. After the validity check of the NHLFE, it verifies that the packet confirms to the associated traffic trunk. From this traffic trunk the traffic class is also extracted, which the proposed solution will make use of.

Following this the packet and NHLFE are ready to be passed on further to the *Forwarding_packet* state. The main actions during this state will focus on forwarding the packet based on the information stored in NHLFE through the common operations allowed on the label (e.g. "swap label", "pop up label").

4.4 Network Design

This section presents a practical testbed implementation devised to show and capture results. These results support the proposed solution for multicast traffic distribution over a Diff-Serv aware MPLS-TE domain.

The proposed testbed is implemented using the OPNET network models by simulating real world components such as computer servers, workstations, CISCO routers, and connecting hardware links. A final assembly containing all of the required networks components is presented at the end of this Chapter in Figure 4-23. .

The entire topology is deployed at the level of a *Campus* size. This translates into a deployed network simulated and spread to an area between $1\,Km^2$ to to a maximum $10\,Km^2$.

4.4.1 Objects configuration

To configure the proposed network's resources, the following OPNET objects are added from the Opnet models library to the network's workspace through the *Object Palette*  :

- *Ethernet_wkstn_adv* (that will represent 3 servers that will behave as multicast sources plus 5 workstation receivers).
- *Application Config* (required to define a set of applications and their general characteristics that will be conveyed over the network) and *Profile Config* (this component defines the profile of a specific application that was previously listed in the *Application Config*).
- *QoS Parameters* is a component used for QoS definition that will be applied over the network resources.
- *MPLS Configuration* is the component that will be used to configure MPLS-TE related items (e.g. FEC, LSP, Traffic trunks, Diff-Serv QoS traffics conveyed over the associated traffic trunks that are placed on an LSP).

The connectivity between components that are placed inside the testbed's workspace is accomplished using the following link models:

- Servers and receivers workstations are attached to their distributing router using *100BaseT_int*. These are point-to-point link models associated to Fast Ethernet standard links, which in this case have a transmission speed of 100 Mbit/s.
- Routers are connected between each other through the *PPP_DS1* link model that represents the T1 joined (or "bonded") lines[3].

Equipped with these components available from the testbed's toolbox, the next step is to configure each of them in order to wire the network's infrastructure.

[3] One T1 line provides a transmission speed of around 1.5 Mbps for 24 channels each of which providing 64Kbps. In this way 2 joined T1 provides a transmission speed of 3 Mbps for 48 channels.

A. *Application* and *Profile* configuration (or services configuration)

The service provided is enabled for two multicast sources which are running a video conferencing application. The simulated video conferencing has the following characteristics: frame interarrival time is set to 10 frame/sec, the size in bytes of a single incoming or outgoing video frame is configured to 5000 bytes. The video frames are assumed to be sent or received at a constant rate.

In order to apply these requirements, the *Application Config* attributes are modified as follows:

1. *Application Definitions = Default*
2. Reopen *Application Definitions* and set *Number of rows = 17* (This adds a new row that will be set as *Enter Application Name=Video*)

The final settings of the values are presented in Figure 4-10.

For the *Profile Config* object the attributes will be set to use the newly added and configured *Video Conferencing* application. The *Video Conferencing* application is set to be repeated 10 times at a constant rate (see *Repeatability)* as shown in Figure 4-11. All of the other parameters are left at the default values provided by the Opnet.

The *QoS Parameters* configuration object is automatically added to the workspace of a modeled design by using the following *QoS Configuration* dialog:

1. Selecting the connecting link between the nodes that are under the provision of QoS. For this testbed all of the links were enabled to be aware of QoS
2. Select *Protocols ->IP -> QoS -> Configure QoS*...
3. Enable a QoS support with DiffServ IP QoS model. This means the interfaces of the nodes will be made aware of the Diff-Serv code point (DSCP). To configure this type of QoS support, the settings applied have to be the same as Figure 4-12.

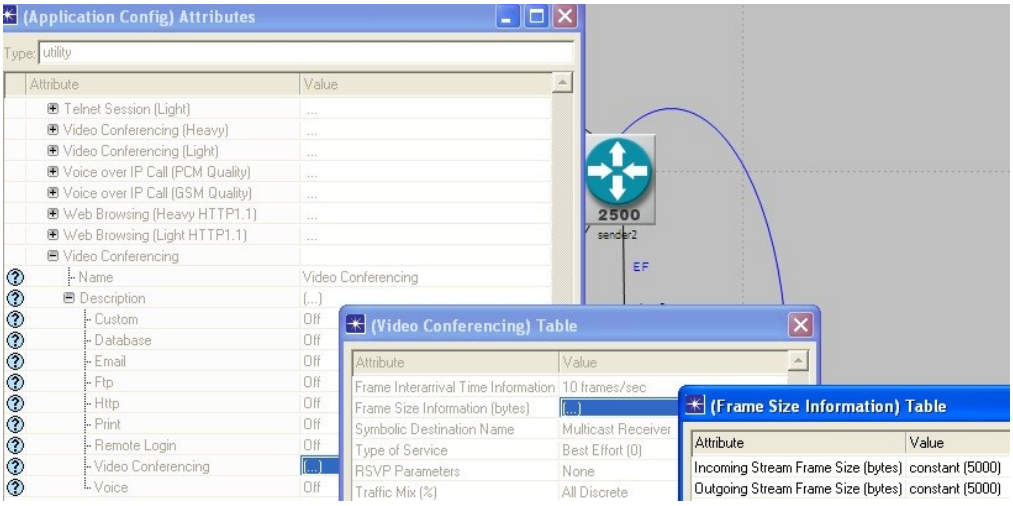
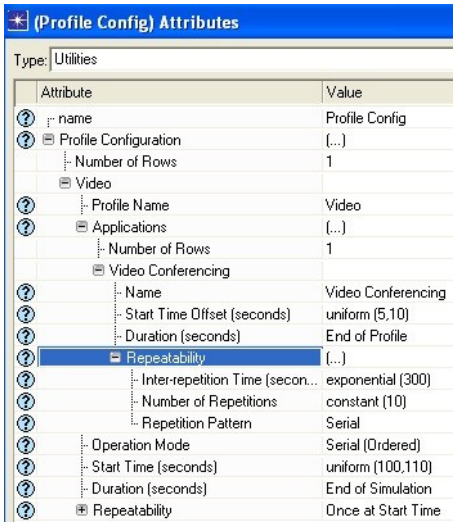


Figure 4-10 Application Config attribute



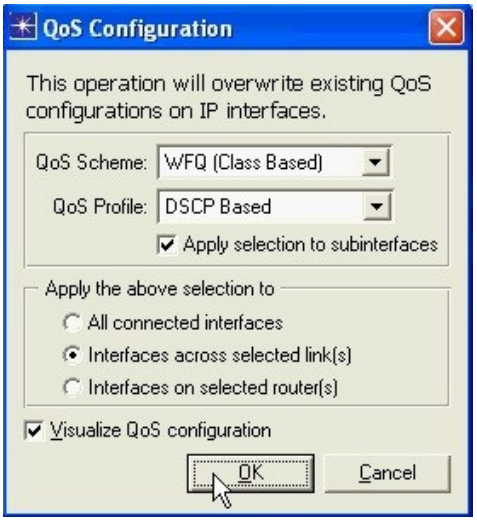
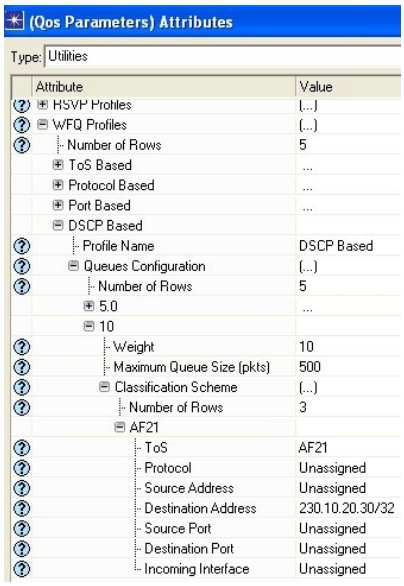


Figure 4-11 Profile Config attributes **Figure 4-12 QoS Configuration**

At this stage, a *QoS Parameters* object is placed in the testbed's workspace. All node interfaces have enabled QoS support and have applied a WFQ queuing profile scheme with the characteristics illustrated in Figure. 4-13. According to the theory presented in previous chapters, this WFQ queuing profiles is implemented in this research's case based on the DSCP values. Therefore, the following entries should be recognized as an example for the AF21 and EF PHB QoS traffic treatment levels. each of these traffic

treatment levels are defined in Figure 4-13 and Figure 4-14. It is evident that OPNET appeals to the notation used for TypeOfServices (ToS) which was superseded by the DS field containing the DSCP value according to [47].

One of the required and significant parameters that will ensure the necessary QoS treatment up to the destination's point is the *Destination Address.* This parameter is configured for a multicast destination group that has IP = 230.10.20.30/32 address for this research.



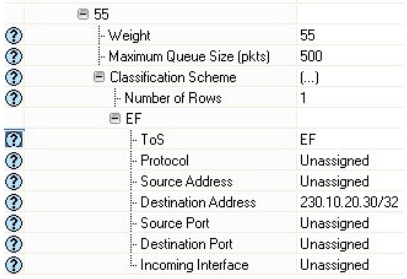


Figure 4-13 Qos Parameters for AF21 PHB **Figure 4-14 Qos Parameters for EF PHB**

The next object that has to be configured relates to the MPLS and TE configuration. This object, defined by *mpls_config_object,* has to be added specifically to the testbed's workspace as part of the MPLS configuration. Attributes of the MPLS configuration related to the EXP field remains unchanged, which OPNET by default considers as standard mappings. These standard mappings are the *Drop Precedence* and *PHB* mappings of the EXP field. Therefore, this document considers that the 3 bits of the EXP field are sufficient to map enough QoS PHBs in order to determine the PHB of the behavior aggregates (aka BA) coming into an ingress LER (see Figure 3-12 and Figure 4-15a)).

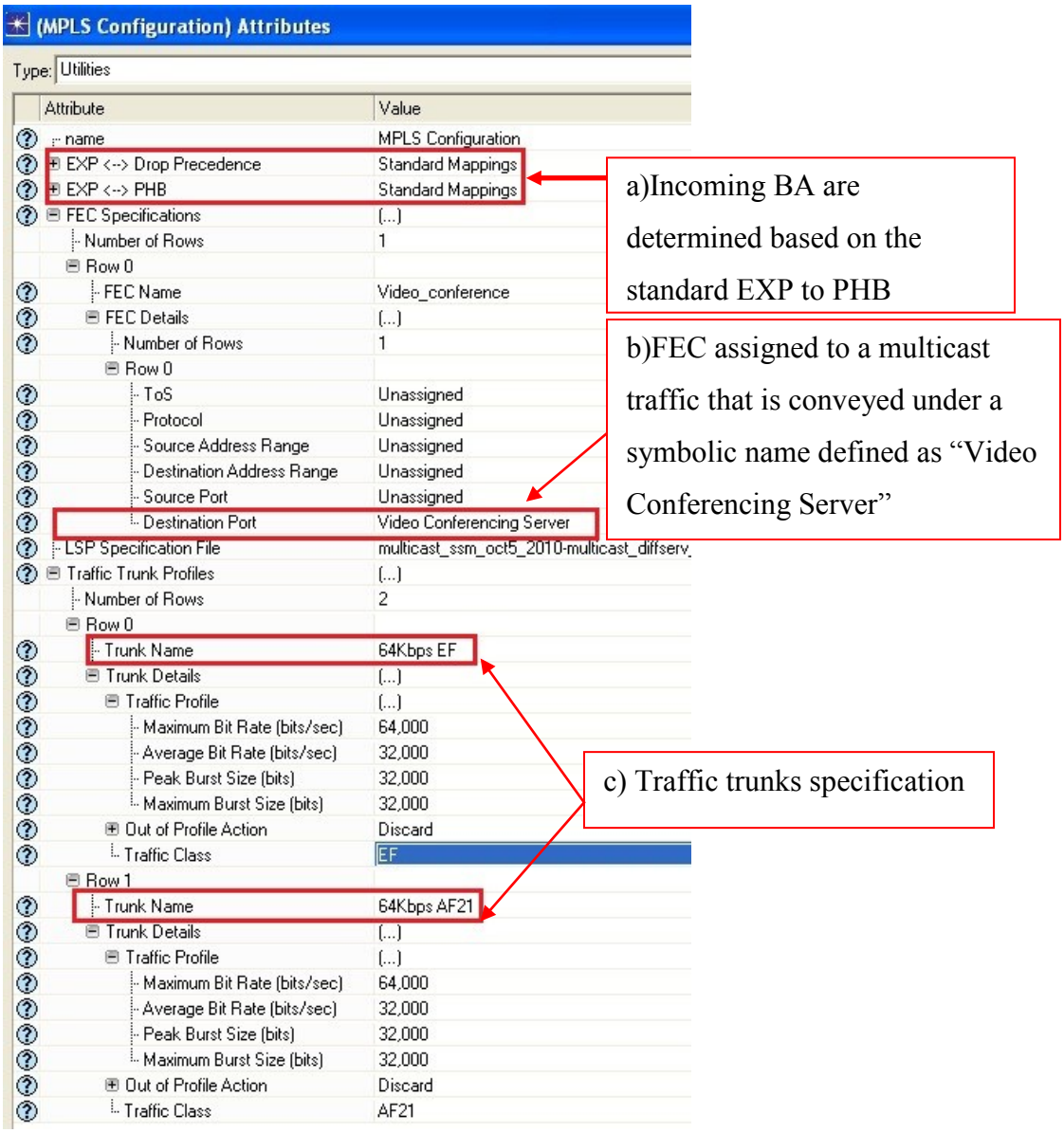


Figure 4-15 MPLS Configuration

The following two attributes of the *MPLS Configuration* object are relevant to the success of the multicast traffic distribution over the Diff-Serv aware MPLS-TE problem. These attributes are the *FEC Specifications* and *Traffic Trunk Profiles*. A TE problem can be simplified into three steps: associate a FEC for the traffic stream (see Figure 4-15 b) that is planned to be distributed (which in this case is a multicast traffic), associate this

FEC with the traffic trunks (that are specified according to the QoS PHB mappings decided to be conveyed in the network), and place these traffic trunks on LSP (which in case of this document will use an E-LSP).

Traffic trunks specifications are defined in the *MPLS Configuration* as per Figure 4-15 c). This document focuses on defining the two traffic trunks which allow the incoming LER to map the incoming BA into two traffic classes. These traffic classes are treated according to the AF21 and EF QoS PHB levels. When the incoming traffic exceeds the imposed traffic trunks limits, the applied profile will simply discard the non conforming traffic.

4.4.2 Network resources configuration

A list with the deployed components in the network is composed of the following resources:

Network Inventory.Summary

File Edit View Help

	Element	Type	Count
1	Devices	Total	15
2		Routers	7
3		Workstations	8
4			
5	Vendors	Cisco Systems	7
6			
7	Physical Links	Total	15
8		Serial	7
9		Ethernet	8
10			
11	Virtual Links	Total	4
12		MPLS Static LSPs	4
13			
14	Traffic Flows	Total	2
15		IP Multicast	2
16			
17	Other	Configuration Utilities	9

Table 2 Network inventory

Assuming that the network resources are connected as mentioned in the previous section, the purpose of the following step is to associate their logical functionality, described through the Objects configuration section, with the items themselves.

Firstly, all nodes have an assigned IP address v4 and it is preferred that the OPNET

automatically associates these addresses. With this automatic association a network operator user does not need to worry about the possible IP address changes.

A. Multicast application's service configuration

According to the problem definition, the configure of at least a source generating multicast traffic is required. For this purpose the testbed is configured with two servers that broadcast multicast video towards the same multicast group, 230.10.20.30. Each server transmits the previously configured applications with two AF21 and EF QoS Diff-Serv levels.

The application's service attributes are configured as shown in Figure 4-16.

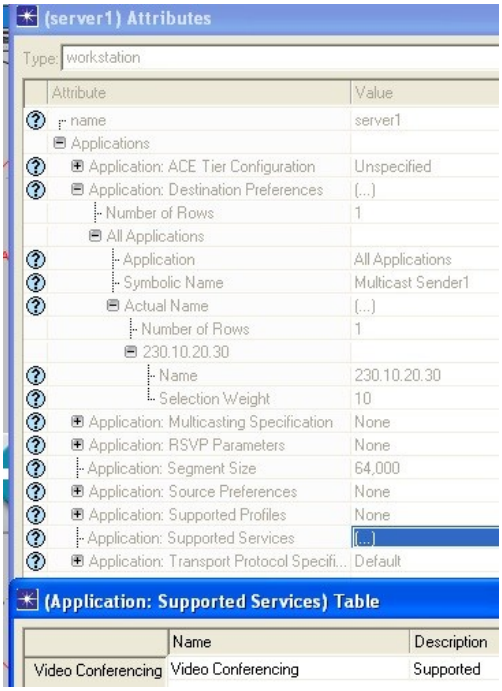


Figure 4-16 Server 1 and Server 2 Application's service configuration

The flag that allows the multicast applications to run on these servers (Server1 and Server3) is configured through the IP module of the models associated to these nodes as shown in Figure 4-17. The outgoing interface of the server will require the application of a QoS profile. For this testbed's case it is configured with a WFQ Class Based profile as

shown in Figure 4-18.

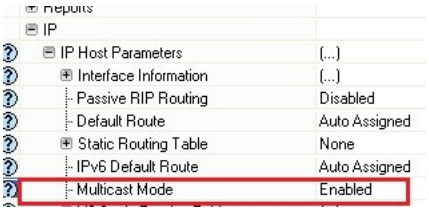


Figure 4-17 Enable application's multicast mode

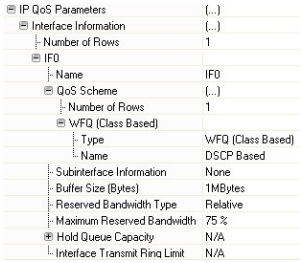


Figure 4-18 QoS applied on the server's interface

For simplicity all the interfaces involved in conveying traffic, and which are aware of the Diff-Serv QoS support regardless of the type of the node (server, hosts, and receivers), have applied a similar WFQ Queuing profile as shown in Figure 4-18.

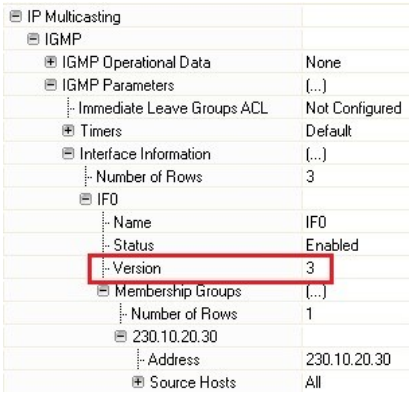
The router sender1 corresponding to the ingress LER models the entering edge router LER1 of a PE domain into a Diff-Serv aware MPLS-TE domain (or simply DS-TE) as illustrated in Figure 3-6. In the testbed's case a Cisco 7200 router is used.

B. PIM-SSM configuration

To support the multicast traffic distribution governed by the PIM-SSM protocol's rules, and according to the documentation presented in Section 3.6.3, one of the preparing condition of the transmitting interfaces is to have them capable of accepting IGMP v3 type of multicast membership group messages. For this purpose the *IP Multicasting* module of the router is configured as shown in Figure 4-19. The IGMP's version is marked as version 3 in the IGMP Parameters interface's information. Another parameter that is specified is the Membership Groups which points to the address of the multicast group.

Each interface participating in the multicast transmission demands that the routing

protocol be enabled. The generic name is PIM-SM as per Figure 4-20, but since the IGMP version was already been established as "3", then the OPNET's compiler will be aware that the multicast protocol intended to be used is PIM-SSM.



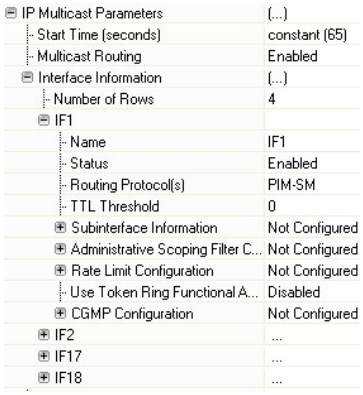


Figure 4-19 Enable router's interface for IGMP v3 **Figure 4-20 IP Multicast Parameters**

The final item to complete the interface's multicast enablement under the PIM-SSM protocol's rule is the specification of the multicast's group destination address.

According to the Internet Assigned Numbers Authority (IANA), the SSM applications and protocols have reserved the address range 232.0.0.0 through 232.255.255.255. Most of the router's OS-es allow these values to be shifted to use another addresses range. For example, according to Cisco, "Cisco IOS allows SSM configuration for an arbitrary subset of the IP multicast address range 224.0.0.0 through 232.255.255.255"[43]. Therefore, the proposed testbed uses the *Destination Address* = 230.10.20.30/32 for the multicast group address. This value is set under the *SSM Adress Range* attribute within the *IP Multicastin*g attribute group.

The next paragraph will add LSP paths and configure the *MPLS Configure* object related to these LSPs. This *MPLS Parameters* group of attributes defines the path of the LSP and maps the incoming traffic. This will be a multicast distribution type toward an LSP for this case. The current testbed implements a static specifically routed LSP, but the scenario can be modified such that it can be used with a dynamically defined LSP. In this

case, a dynamic LSP with CSPF must use a routing protocol (e.g. OSPF or IS-IS) that is based on the SPF protocol. These protocols announce the link's states and communicate it between the routers.

The purpose is to define a Diff-Serv aware MPLS-TE domain entering through an ingress LER (as LER belonging to a PE domain) and map two traffic trunks assigned for two QoS EF, AF21 traffic classes onto a common LSP static path. This static path is specifically routed and defined between *sender 1 – receiver 2* through *receiver 1*. In general, LSPs can be created based on two deployment approaches: tactical deployment and fully traffic engineered deployment. For the purpose of this research an E-LSP creation that most likely falls in the tactical deployment category will be chosen. This choice is ideal because the path is selected by an external user (or network operator). The choice of the E-LSP is based on the arguments presented in previous chapters of this document. For further details on the tactical deployment and fully traffic engineered deployment refer to the documents [2, 49].

C. Static LSPs definition

When a static LSP is used, the path is defined completely from the first router until to the last router of the domain. The procedure for defining an LSP static is as follows:

1. Select *MPLS_E-LSP_STATIC* object from the *Object Palette*
2. Select and click on the first router (e.g. sender1) and then continuing with the next adjacent router that is planned to be used on the LSP's path.
3. When the LSP path definition is finished then just right clicks on the testbed's workspace and select *Abort Path Definition.*
4. From *Protocols -> MPLS Menu,* choose *Update LSP Details* to configure label switching information on the LSP(s).

Through a similar method, two other separate LSPs are defined starting from receiver2 towards receiver 4 and receiver 5. The reason behind this implementation solely as a

demonstration to illustrate a practical example for further traffic distribution that exits a PE with a Diff-Serv aware MPLS-TE domain. The provider is able to split the provisioned traffic based on the QoS traffic classes requirements demanded to be provisioned at the edge of a customer's domain. Referring to the previously presented testbed, at the exit of the router (*receiver 2*), the provider will be able to supply two types of multicast traffic based on the two types of ensured QoS PHB treatment levels, AF21 and EF.

In order to achieve the aforementioned provision, the *MPLS Parameters* must be the same as defined in Figure 4-21. It is apparent that the two incoming traffics on *Interface 1 and 2* are mapped accordingly to the two traffic trunks, *64Kbps EF* and *64Kbps AF21*.

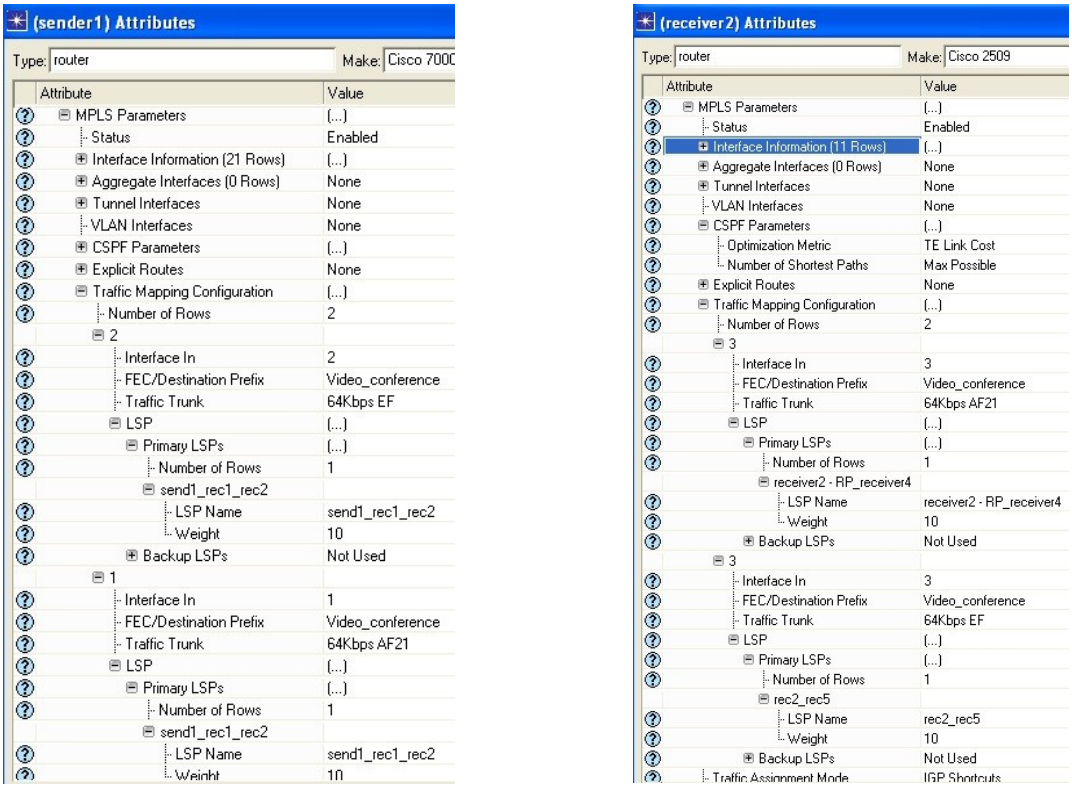


Figure 4-21 MPLS Parameters for ingress LER sender 1 and egress LER receiver 2

Both traffics are grouped under the same symbolic FEC's value ,*Video_conference*.This establishes a common destination, which is targeted towards the same multicast group. The fact that both traffic trunks are placed on a common LSP path, in this case called

send1_rec1_rec2 in the case of *sender 1* configuration, should be noted.

A reversed procedure is conducted over the traffic at the exit edge of a provider from a DS-TE domain at the level of the router *receiver 2*. The single incoming traffic available at the entrance of *Interface In 3* of the *receiver 2* is forwarded through the router towards the exits and it is split at the exit. It is split into two different traffic class provisions, AF21 and EF QoS levels, based on the traffic trunks definition.

The CE routers are now able to overtake these incoming multicast traffic streams and forward them further towards the designated multicast group. The provision is supplied and classified based on the QoS Diff-Serv requirements within the customer domain, and later within the designated multicast group up to the listening host receivers' entrance.

In order for workstation receivers to be able to listen to the channels offered through the pair of (S,G) subscriptions, it will need to be equipped as well for multicast transmission. These settings are defined in Figure 4-22

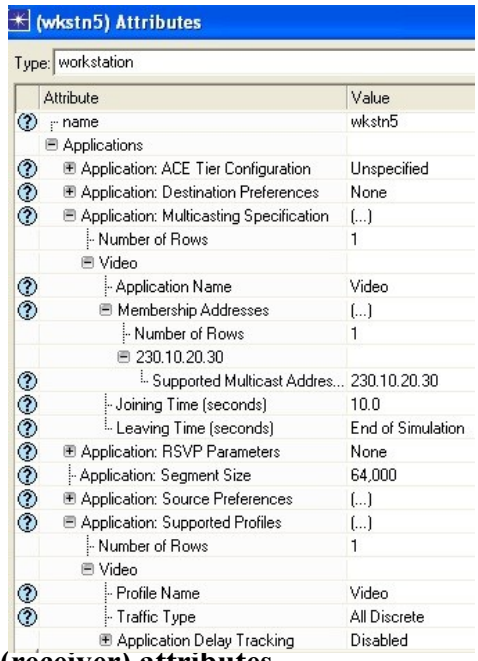


Figure 4-22 Workstations (receiver) attributes

In conclusion, all of the components described in this chapter can be synthesized into a

network testbed that is presented in Figure 4-23.

In Figure 4-23, one of the routers was designated as the rendezvous point within the customer's domain. Therefore, the RP_receiver4 – the CE LER is defined as a RP within a multicast group to successfully forward multicast traffic at the level of hosts. If this does not occur then the aggregated traffic distribution engaged by PIM-SSM multicast protocol will end at the exit PE LER router (receiver2). Alternatively, another RP router that could be selected is, for example, receiver 3 which belongs to the second multicast group identified through the 230.10.20.31 IP address.

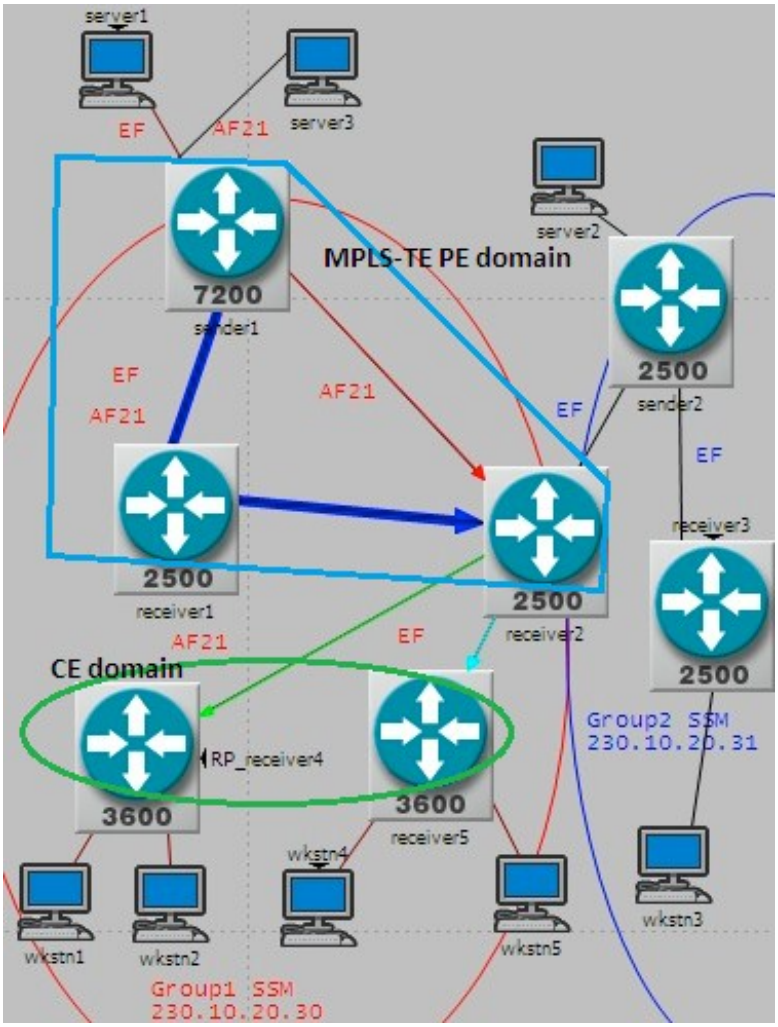


Figure 4-23 E-LSP paths (blue, green, magenta) within an IP Multicast with PIM-SSM enabled at the PE LER routers of the MPLS domain and PIM-SM enabled at the CE LER.

5 Experimental Results and Evaluations

This chapter presents the experimental results and their evaluations for several scenarios simulated for the proposed and designed testing network.

5.1 Introduction

Starting with an initial baseline configuration of the proposed and designed testbed network in Figure 4-23, this chapter will outline the different simulation scenarios that were varied in order to present the benefit, the robustness and to validate the proposed methodology for multicast traffic provision within a MPLS-TE with Diff-Serv aware domain.

Prior to discussing characteristics for specific scenarios, this section starts by configuring a set of reports and statics that must be set prior to running the simulations. These reports and statics will collect the desired results provided by the OPNET simulator..

5.2 Configuring reports and statistics

A. Configuring Reports

During the network's node configurations, each node presents a set of attributes that are grouped under the *Reports* name. Different settings of these attributes can enable or disable the node's data. This data is associated with the node's behavior which is collected for final statistics and reports.

For example,Figure 5-1 shows the settings that are applied for the workstation and routers in order for reports to be collected at the end of simulation

a)

b)

Figure 5-1Enable IP forwarding Table (a) and RIP routing table for routers (b)

B. Choosing Statistics

OPNET offers a large set of statistics, each of them targeting a specific type of analyzed domain for a designed network. The purpose of this research is to show how multicast traffics corresponding to different types of services are conveyed over the associated traffic trunks using MPLS-TE techniques. It also graphically displays the multicast path distribution. Finally, the collected results obtained for the transmitted traffic are compared at the receiver level for different type of services for which they were associated (in this document's case a QoS PHB with EF and AF21 levels).

Therefore, the set of statistics required is selected in OPNET through the following procedure:

1. Right click anywhere in the analyzed testbed's workspace
2. From the *DES* menu select *Choose Individual Statistics*
3. From the *Node Statistics* choose:
    a. *IP* and from here *Multicast Traffic Received/Sent (packets/sec)*, *Traffic Received/Sent (packets/sec)*.

    The *Multicast Traffic Received/Sent* results are collected to prove and demonstrate the number of multicast IP datagrams. that are received or sent by a node from the network across all of the IP interfaces of the node.

    The *Traffic Received/Sent* reflects the total number of broadcast, multicast and unicast IP datagrams which are received by a node from the network across all of

the IP interfaces of the node.

b. Both *TCP* and *UDP* will collect the *Traffic Received/Sent (bytes/sec, packets)* results in order to represent the total number of bytes that are forwarded (or received) towards (or received from) the application layer by the TCP/UDP layer.

These results are collected in order to emphasize the scenarios where TCP or UDP congestion occurs, and how this issue can be remediated through the use of additional LSPs.

c. *Video Conferencing*. This will collect data about the packet-end-to-end delay, traffic received/sent (in bytes/sec or packets/sec). Results have a specific classification in regards to the video conferencing application.

4. From *Link Statistics* select *point-to-point* item containing
   a. *Queuing delays* which according to OPNET represents "instantaneous measurement of packet waiting times in the transmitter channel's queue". Therefore, it is the interval of time between the moment a packet enters into the transmitter's channel and the moment the last bit of packet is transmitted.
   b. The *Throughput* of the links for sending or receiving packets
   c. The *Utilization* of the links, representing according to Opnet "the percentage of the consumption of an available channel bandwidth".
5. From *Path Statistics* select
   a. *Flows Traffic In/Out* represents the traffic sent (or received) into (or from) an LSP at the ingress (or egress) end of the tunnel. The collected results illustrate the existence and intensity of each traffic flow that is carried through a specified LSP.
   b. *LSP Traffic In/Out* describes the total traffic that is sent (or received) into (or from) an LSP at the ingress (or egress) end of the tunnel. These results

are primarily used to represent the overall value of an aggregated traffic.

5.3 Generating Traffic

An important condition that has to be fulfilled before the collection of values through the network's resource, is to instantiate a traffic within the designed network.
As discussed, the proposed testbed is developed to test multicast traffic distribution and thus a relevant traffic is planned to be engaged. Furthermore, the simulated multicast traffic is enhanced at the source to be transmitted and treated with the associated QoS PHBs corresponding to AF21 and EF QoS levels.

A series of traffic flows which are instantiated according to the scope of the simulation will be generated. Therefore, the traffic flows is going to be described initially and enabled only when a simulation scenario wishes to emphasize a particular context. Thus, the next few sections present the procedures within OPNET that generate traffic flows which will later be used in simulation scenarios.

A. *Multicast traffic provision over TCP protocol targeted towards http source/destination ports*.

Presented below are the required procedures to setup a multicast traffic that will be provided over the TCP protocol. The purpose of this traffic is to simulate a multicast content provision generated by a multicast server (server 2) when there is a request initiated at the application layer. This request is demanded by a workstation computer (wkstn 3) belonging to a multicast group 230.10.20.31 through http protocol.
Here are the main steps used to configure this type of traffic:

1. Select the transmitting server (e.g. server 2) and from the menu bar choose *Traffic -> Create Traffic Flows -> IP Multicast…*
2. In the *Create IP Multicast Traffic Flows* dialog set *Destination IP address=230.10.20.31,* in *Type of Service* choose *Differentiated Services Code*

Point (DSCP)= EF, choose *Protocol=TCP* as per Fig. 5.2

3. Click on the *Create* button to generate the required settings.

It should be noted that the traffic is described at a higher OSI's model layer. It starts at the application layer through an http access request launched by wkstn 3 from server 2. Streams of data are then packaged at the transport layer by using the TCP protocol. By choosing TCP protocol, it is desired to receive reliable and ordered streams of bytes.

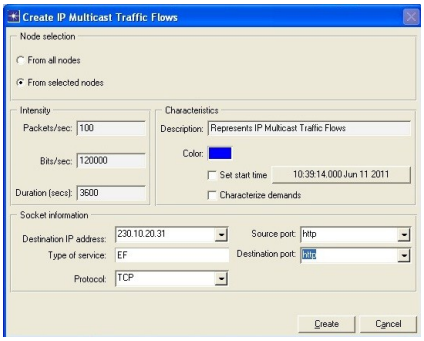


Figure 5-2 Generating multicast traffic over TCP protocol with DSCP = EF level type of service

B. *Two multicast video traffic provisioned with two QoS PHBs levels AF21 and EF by two separate multicast servers*.

The initiated multicast traffic will be transmitted from server 1 which will be setup to broadcast a video transmission with a QoS associated with a PHB of EF. Server 3 will broadcast a multicast video transmission with QoS associated with a PHB of AF21 level.

The generated traffic for this case specifies only the network layer type of protocol which in this case is the IP protocol. The upper (application and transport) layer selection is determined based on the predefined *video conference* profile.

The procedure to create these traffic flows within the testbed is presented below:

1. Select the transmitting server (e.g. server 1) and from the menu bar choose *Traffic*

-> Create Traffic Flows -> IP Multicast…

2. In the *Create IP Multicast Traffic Flows* dialog set *Destination IP address=230.10.20.30,* in *Type of Service* choose *Differentiated Services Code Point (DSCP)= EF* (or AF21 for *server3*), choose *Protocol=IP* as per Fig. 5.3 a)
3. Select the *Create* button to generate the traffic's configuration.

The same procedure is repeated for both servers 1 and 3 in order to create both an EF and AF21 type of service. The result of this procedure can be seen in Figure 5-3 b) by selecting from the menu bar *Traffic -> Open Traffic Center*.

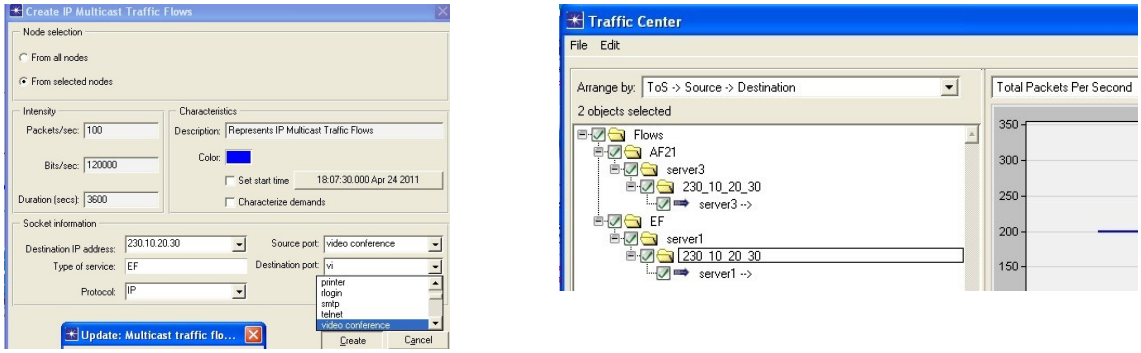


Figure 5-3a) Create multicast traffic with EF and EF21 type of services for server 1 and server 3 b) Edit traffic flows characteristics in Traffic Center

The Traffic Center allows different statistic summaries regarding the flooded traffic in the network to be displayed. It also allows the finer tuning of the traffic throughput's characteristics.

C. Increasing multicast traffic provision

This section will focus on the procedure required to change the characteristic (in particular the intensity) of one of the multicast traffics. A similar procedure can be used to change any other predefined traffic, for example a unicast type of traffic. Following the procedure for opening *Traffic Center* described at section B, the method to modify the intensity of the multicast traffic to be injected over the distributing multicast tree is outlined below.

1. Open *Traffic Center* as per Figure 5-3 b)
2. Right click on the blue arrow of the server 3. As a result the following dialog is opened as in Fig. 5-4
3. Apply an increased traffic that changes every 5 minutes of the simulated traffic generation.

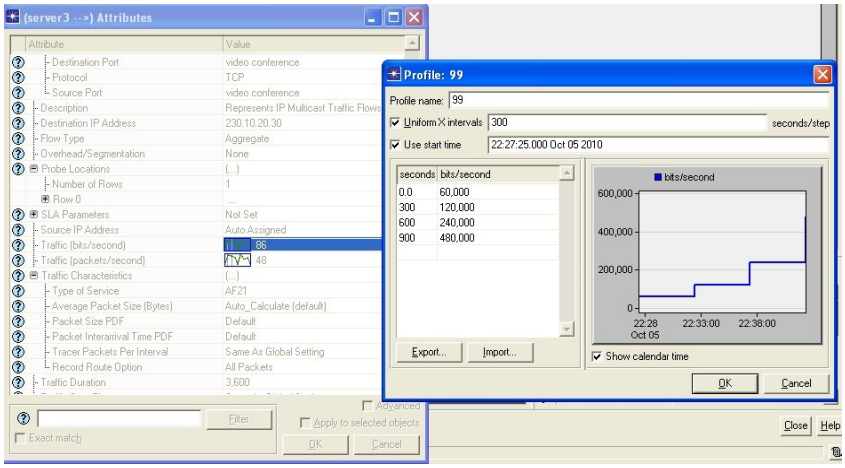


Figure 5-4 Multicast traffic's intensity is increased at every 5 minutes for server 3.

A valid reason to generate this type of traffic is to observe the TCP protocol's congestion mechanism. The effect of this traffic increase will be presented later during a simulation scenario.

D. Unicast VOIP traffic over UDP protocol between two calling parties.

Lastly, a unicast VOIP traffic which is established between two calling parties is generated. The steps required to establish this type of traffic are as follows:

1. Select server 3 and wkstn1 and then choose the menu option *Traffic -> IP Unicast…*
2. Adjust traffic settings to be similar to those presented in Figure 5-5.

Prior to collecting the results for the scenario which engages this type of traffic, the following changes have to be applied to the *Application Configuration* and *Profile*

Configuration objects.

Inside the *Application Configuration* a new entry is added for VOIP with the following attribute values

1. From *Application Definitions* select *Voice over IP Call (PCM Quality)*
2. Expand *Description* hierarchy and assign *Voice= PCM Quality Speech*

for which the *Profile Configuration* defines next fine tunings as per Figure 5-6.

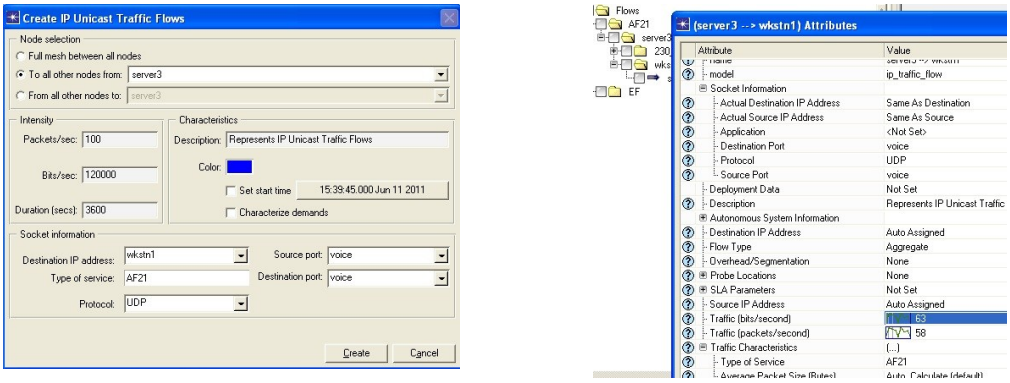


Figure 5-5 Generating and settings for a unicast VOIP traffic generated between two calling parties.

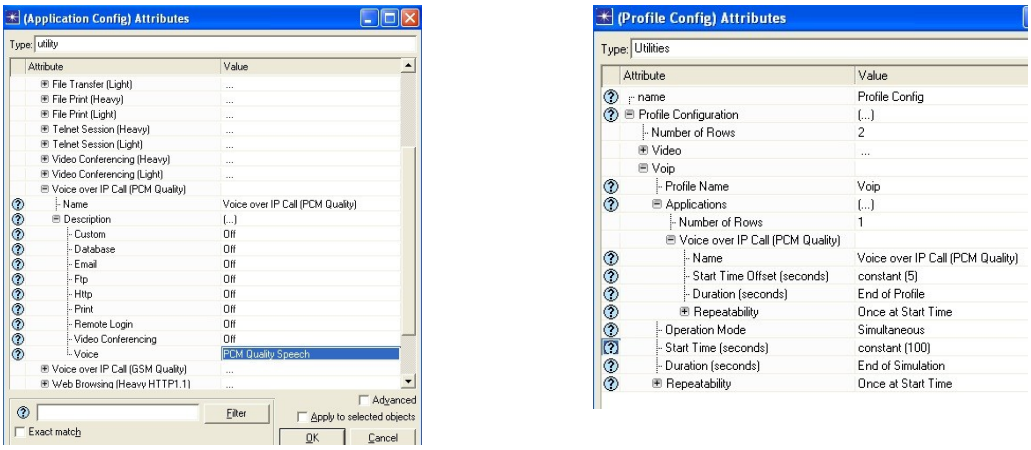


Figure 5-6 Adding support for VOIP application

5.4 Running simulation

Once the reports and the desired set of statistics are enabled, the simulation is ready to be started. It is a good practice to create a new scenario starting from the initial scenario each time a new requirement which involves an adjustments for the network nodes definition is required,. The initial scenario is also referred to as a *baseline* scenario. The procedure to duplicate a current scenario (e.g a baseline scenario) requires the following: from the menu bar select *Scenarios -> Duplicate Scenario…*, and provide a new name for the duplicated scenario. The comparison of the results obtained from the different scenarios and overlaying them in order to make a better conclusion in regards to the observed phenomena is an advantage to using scenarios.

The simulation can be started from the same menu item as mentioned above: *Scenarios -> Manage Scenarios*. Choose a specific scenario, or if a comparison between multiple scenarios is intended, then select multiple scenarios. It is important to note that the simulations usually consume a lots of computer memory and might slow down when multiple simulation are being processed.

If the network components are properly defined, and the testbed project is successfully compiled, the paths of the multicast traffic distribution can be visualized by following the procedure: Select *Protocols -> IP -> Demands ->Display Routes for Configured Demands…*

The next simulation cases will demonstrate the proposed methodology of providing IP multicast traffic with QoS Diff-Serv awareness within a MPLS-TE domain. These methodologies are proved through the collected results.

A. Providing plain PIM-SSM type of multicast traffic

The role of this simulation case is to demonstrate how a plain PIM-SSM multicast traffic distribution tree is built. In order to simulate this scenario, the testbed network is flooded with a traffic that was previously described in Section 5.3 A. Based on the collected results, OPNET displays the PIM-SSM multicast tree's paths as per Figure 5-7.

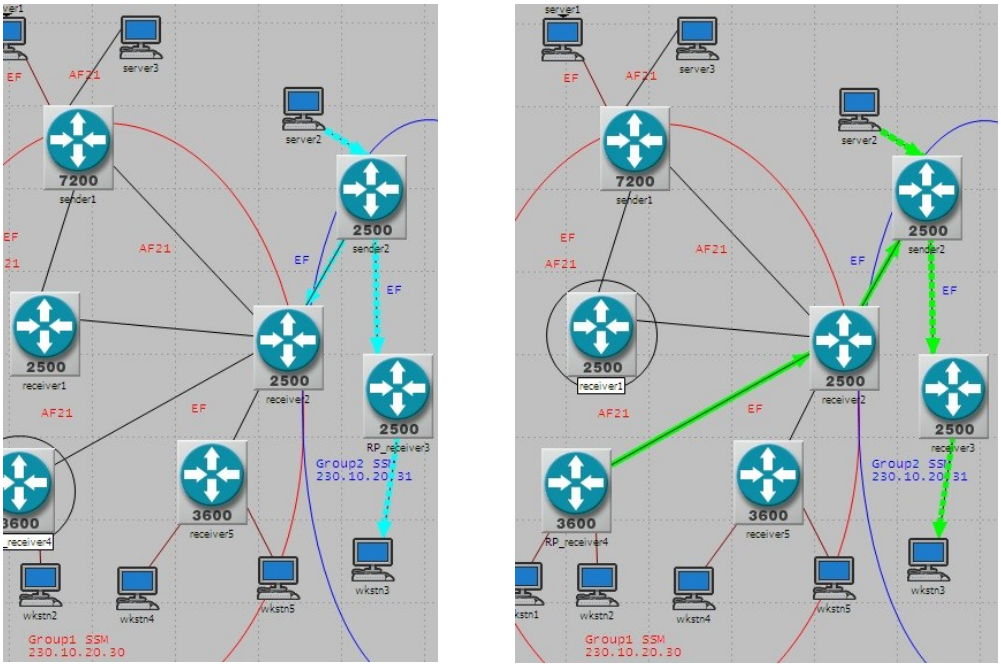


Figure 5-7 a) PIM-SSM distribution tree's path with RP enabled on router RP_receiver3 b) PIM-SSM distribution trees's path with RP enabled on router RP_receiver4

The following conclusions can be drawn from these examples:

- A PIM-SSM distribution tree is built close to the source generating the multicast traffic.
- Multicast traffic is distributed towards all the nodes branches that are subscribed on a channel (S,G) per channel basis. In both cases the wkstn 3 belonging to the multicast 230.10.20.31 is the only workstation listening to server 2's provision.

Receiver 2, located at the border of the two multicast groups, is configured to have the capability to provide traffic "channels" for workstations incoming from all servers: server1, 2 and 3.

B. Multicast traffic tree distribution paths

The purpose of this simulation is to present a comparison between the multicast traffic tree's distribution paths of two different scenarios. The first are the paths built over the network nodes which only have the PIM-SSM protocol enabled. The second are the paths built over network nodes that have the PIM-SSM protocol enabled and are actively engaged in explicit routing process of the multicast distribution traffic. This traffic is aware of the imposed QoS expressed through the Diff-Serv requirements and conveyed through "traffic trunks" that are driven on a static LSP over the testbed's network resources.

Simulation for this case floods the testbed network with both types of traffic described in Sections 5.3 A and B.

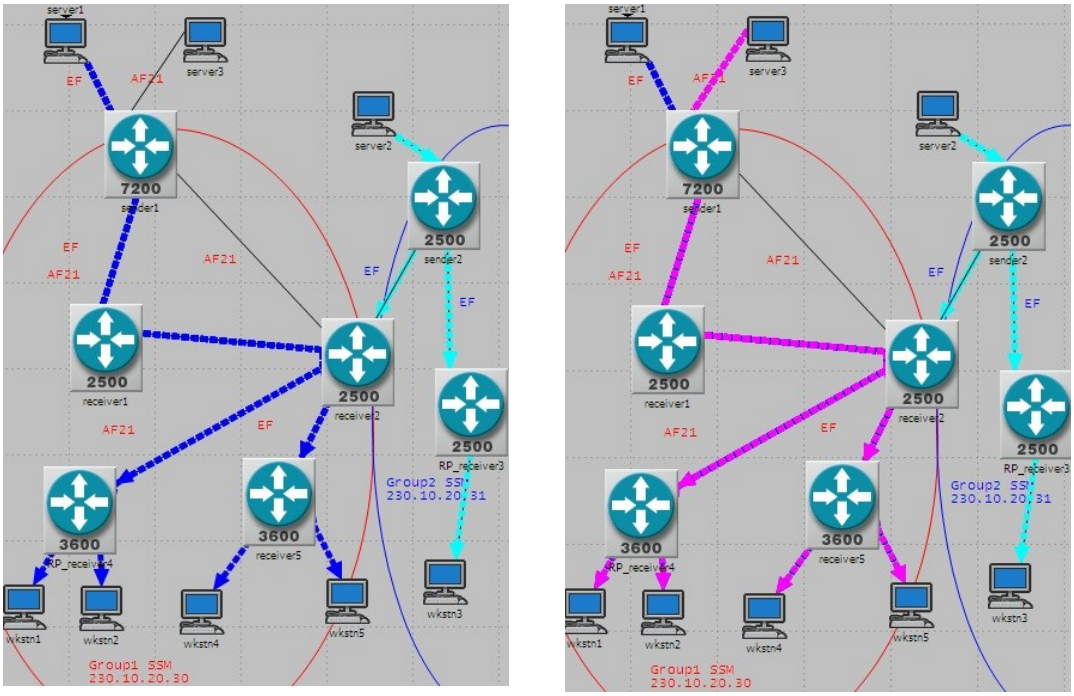


Figure 5-8 Multicast traffic distribution over network nodes enabled with PIM-SSM protocol (magenta paths) and multicast distribution path controlled at the ingress LER sender 1 and distributed over traffic trunks driven on a static LSP (blue and pink paths).

This simulation visually presents the basis of the proposed methodology to provide multicast traffic distribution over traffic trunks aggregated at the entrance of the ingress LER in an MPLS-TE domain with Diff-Serv awareness.

It shows how the construction of the multicast tree distribution path is regained by a network operator. In other words, the construction of the path is developed "under control". Therefore, the construction of the path is developed "under control". Referring to Figure 5-8 it is apparent that the construction of the multicast tree distribution's path is controllably built (see blue and pink multicast distribution paths), closer to the initiating multicast server (server 1 and 3). It is then aggregated on the traffic trunks at the ingress LER (sender1) with established QoS (expressed through Diff-Serv requirements) and is placed on an explicit routed LSP.

The plain PIM-SSM distribution tree's path is built from the requested source all along the nodes interfaces that have PIM-SSM protocol enabled. These are located on the path towards the requesting multicast group (see magenta paths). Although there is no attached, requesting listener (e.g. workstations), the requested traffic is flooded towards all the routers (in this case receiver 2) belonging to a multicast group e.g. 230.10.20.31.The traffic reaches wkstn 3 destination which is the only subscribing listener (workstation) of the multicast traffic generator server 2.

C. *Incoming unicast and multicast traffic at the ingress LER entrance of the MPLS-TE domain with Diff-Serv awareness*

The proposed methodology presents a question which relates to the behavior of a unicast traffic forwarded within this MPLS-TE domain. The question appears due to the fact that the network resources are primarily designed to handle multicast traffic. The proposed solution will now be tested in to real life situation where multicast and unicast traffic will be ubiquitous present. The proposed methodology is questioned from the robustness perspective to determine how well the unicast traffic is handled within the presence of a multicast traffic conveyed simultaneously over this established infrastructure.

A theoretical explanation arises from the fact that traffic is characterized through a tuple of source and destination IP addresses. If the unicast traffic is established through an

unicast source/destination IP addresses, then an incoming multicast traffic is also described through a unicast source address. This source address comes from the multicast traffic's source generator and the multicast's group destination address (e.g 230.10.20.30 or 230.10.20.31).

As a result, the generated unicast traffic will travel over the MPLS-TE domain under the influence of the RIP protocol that leads a path with a minimal hop numbers. This path is not necessarily the fastest for the traffic, nor is the traffic driven over a path that ensures enough network resources. This could cause congestion, although the other longer path has enough network resources.

In order to simulate this case, the proposed scenario sets up a unicast VOIP traffic established between two calling parties. The network testbed is flooded with multicast traffics established by servers 1, 2 and a unicast VOIP traffic established between server 3 and wkstn1 as described in Sections 5.3 A, B and D.

Collected results are graphically represented in Figure 5-9.

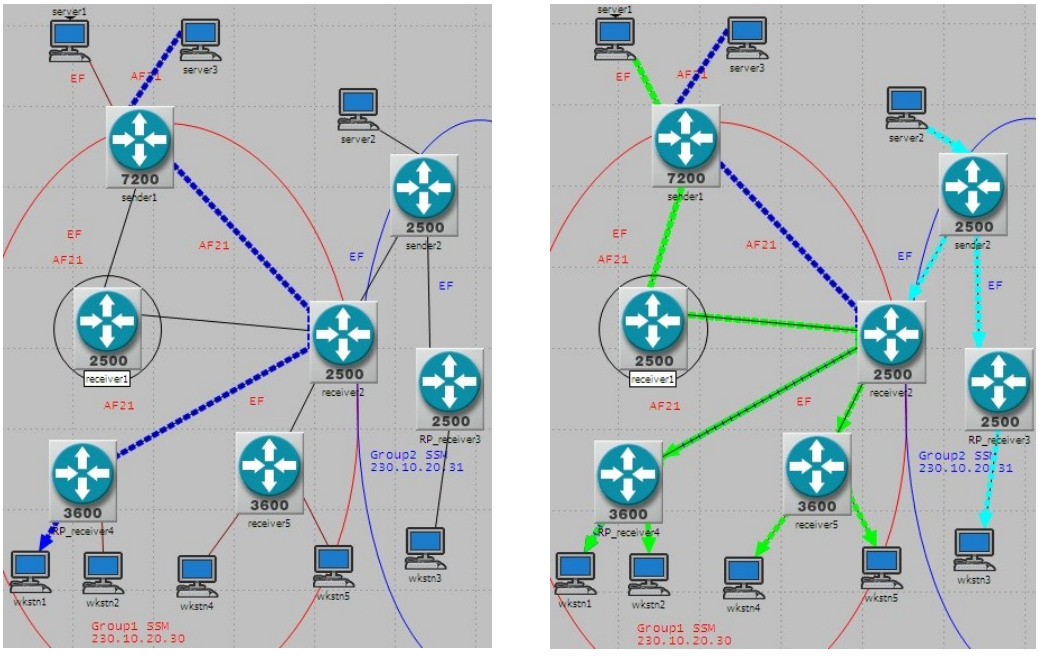


Figure 5-9 Unicast VOIP traffic between server 3 and wkstn1 (blue path) follows its regular paths governed by RIP protocol while the multicast traffic aggregated over LSP follows the explicit constrained path (green path). The multicast traffic from server 2 defined for 230.10.20.31 still follows its own distribution paths.

D. Effects and recoveries from an increasing multicast traffic

This scenario studies the effect of an increasing multicast traffic that is merged and flooded at the entrance of the ingress LER with another multicast traffic. The other multicast traffic is constant in intensity through an MPLS-TE with Diff-Serv awareness domain over a single and common LSP. For this purpose the simulation will use traffic generated based on the description presented in Sections 5.3. B and C.

A review of the involved components reveals the following: The video multicast traffic starting from server 3, and deployed through a TCP protocol towards the multicast group 230.10.20.30 through an ingress LER1, has ensured a PHB with a QoS at AF21 level. The intensity of the traffic is increased every 5 minutes as per Figure 5-4. Another video multicast starting from server 1, which is deployed through a UDP protocol towards the same multicast group 230.10.20.30 and through the same ingress LER1, has ensured a PHB with a QoS at EF level. This traffic has a constant intensity all over the simulation time.

Based on the collected results, the chart from Figure 5-10 further supports known facts about the TCP protocol. As the video multicast traffic over the TCP increases in intensity every 5 minutes, it eventually reaches the limit of the links' interface and signals to the TCP congestion control mechanism. This sends back a signal to the multicast source to reduce the generated traffic. In this scenario is it completely suppressed because of the high injected traffic. The chart from Figure 5-10 represents the traffic flows injected through the single existent static LSP at this time, which is defined from sender1 to receiver 2 through receiver1.

This type of situation is often encountered in real life by ISPs and results in negative effects on the client sides. These negative effects are primarily due to the fact that the network links are already flooded with traffic over the UDP protocol. When it is falsely

assumed that the network is capable of accepting a higher traffic demand over TCP, the traffic is suddenly chopped down at the source by the TCP congestion control mechanism. This occurs because the "pipes" are full with the other UDP type of traffic.

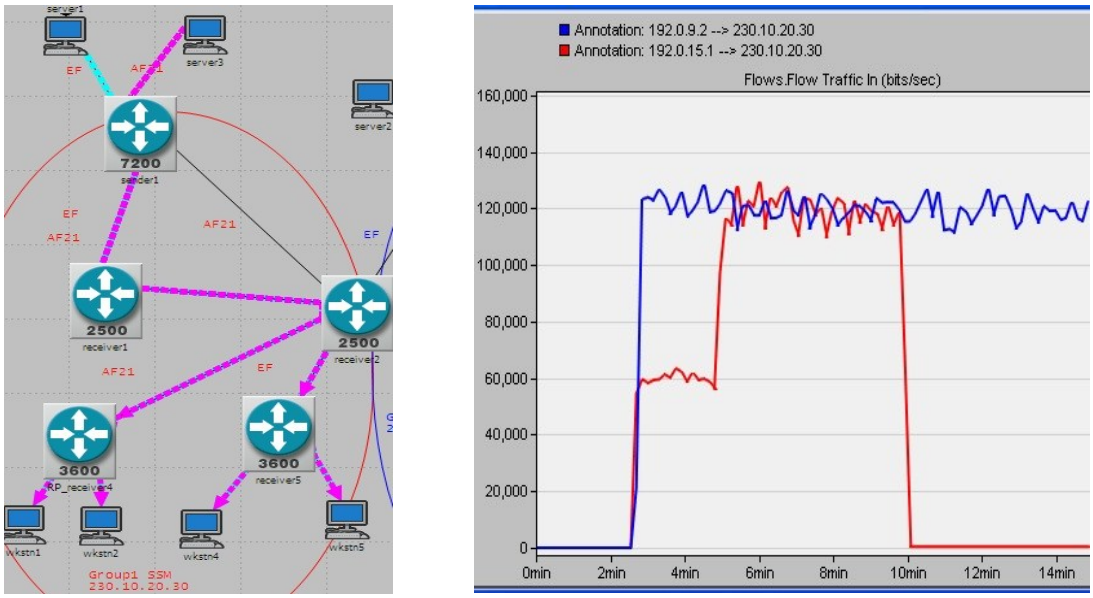


Figure 5-10 TCP Video multicast incoming from server 3(or IP address 192.0.15.1) is cut down at the source due to TCP congestion control's mechanism while the UDP multicast video from server 1 (or IP address 192.0.9.2) is forwarded further without almost any modification.

A possible solution to this problem could be implemented without expensive infrastructure changes but rather through logical TE deployments. This case in particular might be solved by defining a second LSP as shown in Figure 5-11.

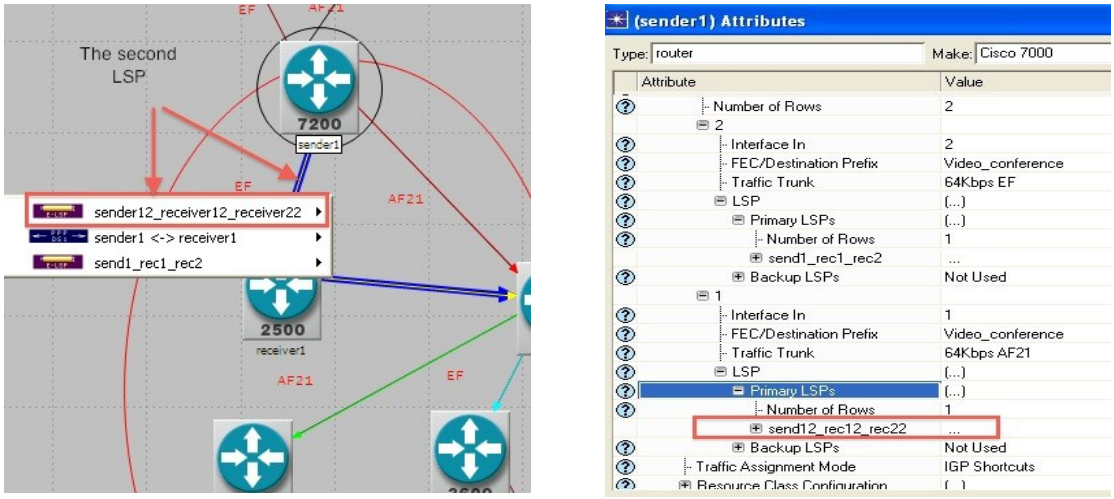


Figure 5-11 Defining a second E-LSP sender12_receiver12_receiver22 that will drive the traffic trunk (e.g. 64Kbps AF21) associated for the multicast traffic from server 3.

By defining this second E-LSP (in the simulation's case *sender12_receiver12_receiver22*) traffic flooded from the multicast server3 will be more relaxed. This way a possible congestion build-up is avoided through this logical TE implementation, which is a software setting at the router level.

The new results collected for both defined LSPs, send1_rec1_rec2 and sender12_receiver12_receiver22, can be seen in Figure 5-12.

Figure 5-12 a) Results for the multicast traffic flows collected for two E-LSPs send1_rec1_rec2 and sender12_receiver12_receiver22 b) Next network segments with single LSPs (between receiver2 and receiver5 or receiver2 and receiver4) are still under the TCP congestion' control mechanism influence.

It is evident with the introduction of the second E-LSP, sender12_receiver12_receiver22, that the multicast traffic flow is improved. This is due to the fact that the newly added LSP is capable of accommodating increase in the multicast traffic's flow. It should be noted that the traffic was only improved on that particular segment of the network. A similar intervention is required for the next following network segments, otherwise (e.g between receiver2-RP_receiver4 and receiver2_receiver5) the traffic flows will suffer from the same TCP's congestion control mechanism.

E. Comparing the quality of traffic reception within same multicast group without MPLS-TE enabled

The next two scenarios (E and F) present and conclude the observations for using DiffServ aware MPLS-TE domain compared to plain PIM-SSM type of multicast network.

This current scenario deploys a network that does not activate any MPLS functionalities at the routers levels. Links' interfaces of the nodes (routers and workstation) are enabled for DiffServ support as part of the selected IP QoS model. In addition, links' interfaces are configured to be capable to distribute PIM-SSM multicast type of traffic (see section 4.4.2. B.) Two servers (server1 and 3) generates the multicast traffic described in Section 5.3 B. The multicast traffic flowing from server 1 with IP address 192.0.9.2 is shown in Figure 5-13 a).

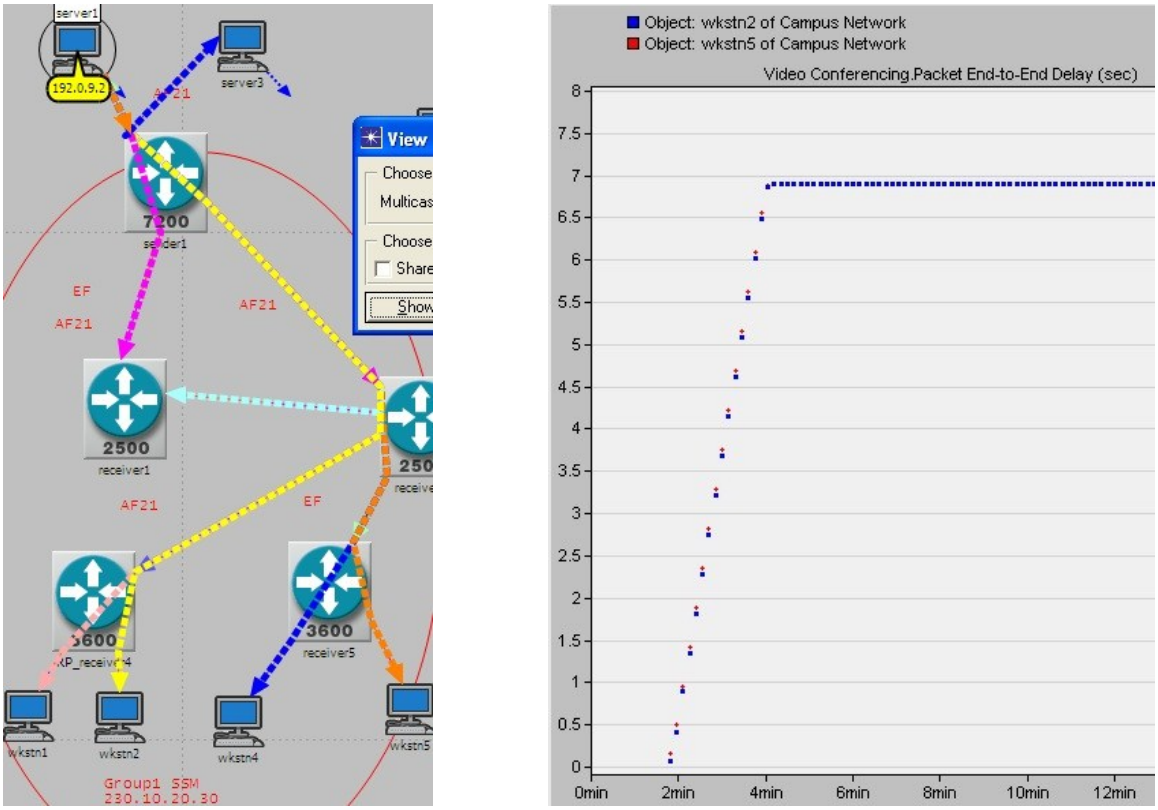


Figure 5-13 a) IP Multicast traffic flowing from server 1 over plain PIM-SSM enabled network b) Packets end-to-end delays recorded at receivers wkstn2 (blue) and wkstn5(red).

The paths of the multicast traffic originating from server 3 will take similar paths as those originating from server 1.

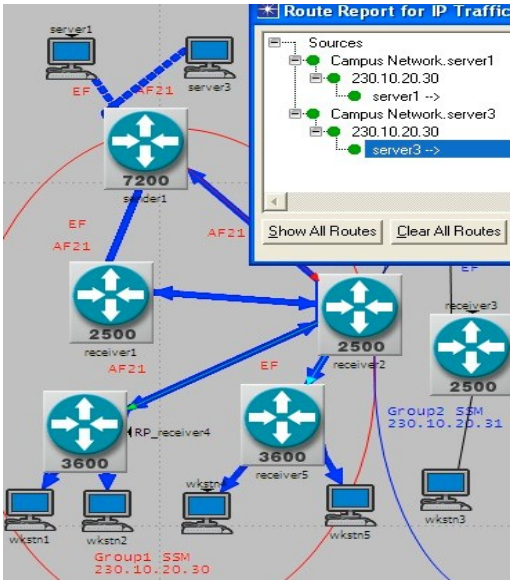
Figure 5-13 b) represents the values of the packets end-to-end delay recorded at wkstn 2 and wkstn5. These two workstation receivers have applied for two different types of QoS. Workstation 2 (Wkstn2) demanded an AF21 QoS level traffic reception while workstation 5 (wkstn5) subscribed for an EF QoS level traffic reception. Regardless of the demanded QoS multicast traffic reception, it is visible that for both receivers the time end-to-end delays for packets are almost similar.

In this scenario the received QoS traffic could not be enough emphasized in order to provide an acceptable solution to distribute multicast traffic aware of QoS.

F. Comparing the quality of traffic reception within same multicast group with MPLS-TE enabled

The generated traffic used for this scenario was presented in Section 5.3 B and the collected results are revealed in Figure 5-14 a). This Figure displays the path of the multicast traffic distribution and Figure 5-14 b) lists the paths followed by the multicast traffic. In Figure 5-14 the multicast traffic follows the routed path indicated by the traffic trunks. These traffic trunks were used to map traffic, with a differentiated type of service, onto a single shared LSP through an MPLS-TE domain.

Based on the selected statistics important results are expressed through Figure 5-15 and Figure 5-16. These figures compare the throughput values of the received transmission at the receivers (wkstn 2 and 5) over the time. This occurs to prove an expected result with regards to the two types of services, EF versus AF21 QoS.



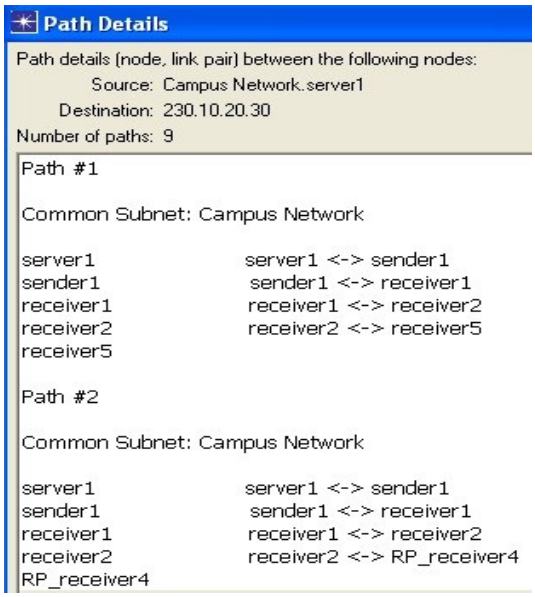


Figure 5-14 a) IP Multicast traffic flowing from source to receivers over a Diff-Serv aware MPLS-TE domain b) Multicast path details

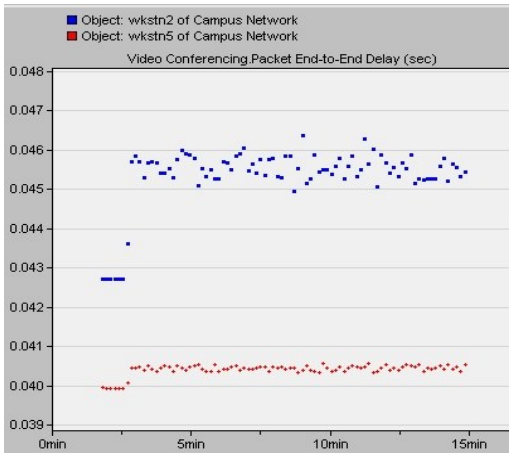


Figure 5-15 Traffic received by wkstn 5 with an EF Diff-Serv level has lower end-to-end delay times than wkstn 2 with AF21 Qos level

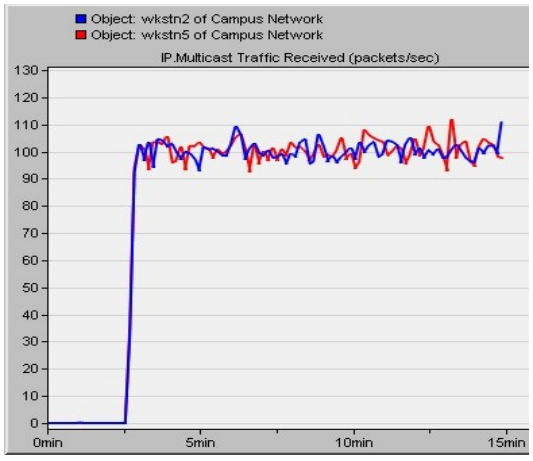


Figure 5-16 The number of packets received by wkstn 5 have a better arrival frequency rate for which was ensured an EF Diff-Serv level requirement.

Figure 5-15 illustrates the packet end-to-end delays as they are received during the simulation at the wkstn2 and wkstn5. The graphic demonstrates how the traffic received by wkstn 5, with a guaranteed Diff-Serv EF level, had a superior provision with respect to packet delays over the guaranteed Diff-Serv AF21 level traffic received by wkstn 2.

Firstly, comparing Figure 5-15 with Figure 5-13 b), one could easily observe how the packets end-to-end delay times, at wkstn2 and wkstn5 receivers, are overall improved for both type of QoS traffic levels (AF21 and EF). Secondly, the difference between the ensured QoS traffic levels (AF21 and EF) is more prominent.

The lower end-to-end delay time of the EF compared to AF21 reinforce the success of forwarding multicast traffic over the MPLS-TE domain while the demanded traffic's QoS level is preserved at reception.

Figure 5-16 illustrates how the multicast packets, received by wkstn 5, have a better arrival rate for which an EF Diff-Serv level was required compared to wkstn 2. It also shows that a Diff-Serv of AF21 level was ensured.

These graphics prove that the EF Diff-Serv level has better quality insurance, and therefore it is a better candidate for video multicast transmission. Therefore it is probable that clients will select this case if they desire a high quality video reception.

F. Extending the number of multicast receivers

The premise of this experiment is to demonstrate how the algorithm for PIM-SSM multicast traffic distribution over a DS-TE MPLS environment can be used to extend the number of multicast receivers. The case where a constantly increasing number of multicast listening clients occurs is an example of where this experiment may be used. The procedure to add multicast receivers considers the type of QoS services offered from an ISP side and then selects the multicast sources from the client side.

The algorithm briefly suggested that if the egress router reaches the maximum number of outbound interfaces capable to distribute multicast traffic, then a new midpoint must be added along with a new egress router(s).

Each of these additions is dedicated for one type of QoS service offering.

This example expands the basic network cell represented by Figure 4-23. This expansion occurs by inserting a new midpoint router, (or LSR) receiver 3, that has the same type of QoS inbound/outbound policies applied on the interfaces as the outbound policy applied on the outbound interface of the ingress router sender 1. The new network, along with the added LSPs, becomes wired as per Figure 5-17.

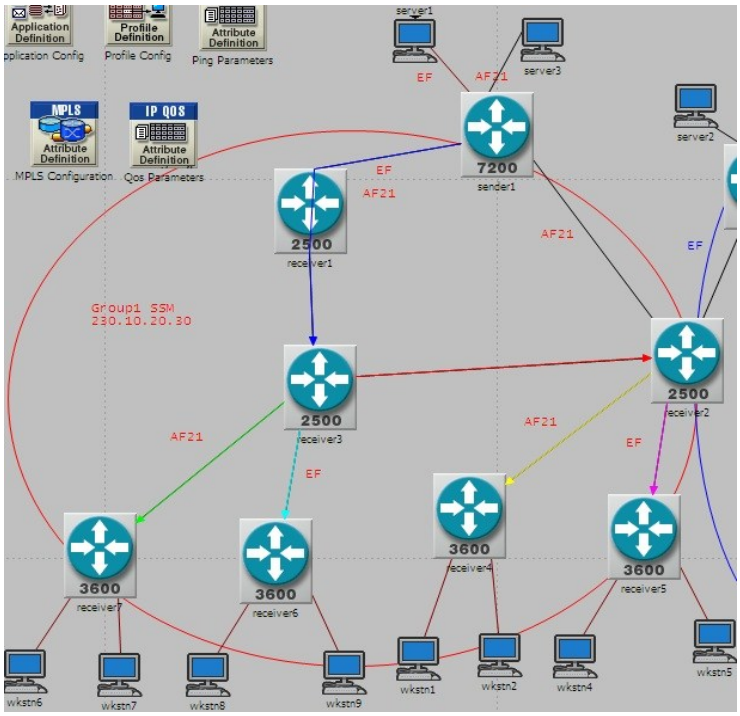


Figure 5-17 Extending multicast traffic distribution by adding new midpoint router (LSR) receiver 3 together with egress router (LER) receiver7, receiver 6

It is evident that the last group of routers, consisting of midpoint router (receiver 2) and the last two egress routers (receiver4, receiver 5), are duplicated with this new midpoint router addition. Furthermore, the last LSPs are left unbroken between receiver 2 and receiver 4 or receiver 2 and receiver 5. Therefore, a network operator does not need to revisit the previously established clients listening to the multicast traffic since the required network resource configuration is already allocated.

The task of the newly added midpoint receiver 3 will have to be enhanced to forward the incoming multicast driven on the traffic trunks over the LSP from sender1 through receiver1 to be forwarded toward receiver2 as per Fig. 5-18. Simultaneously, the same traffic will have to be split between the newly added egress router receiver7 and receiver 6 based on the QoS service offering that in this case are expressed through AF21 and EF QoS levels as per Fig. 5-19.

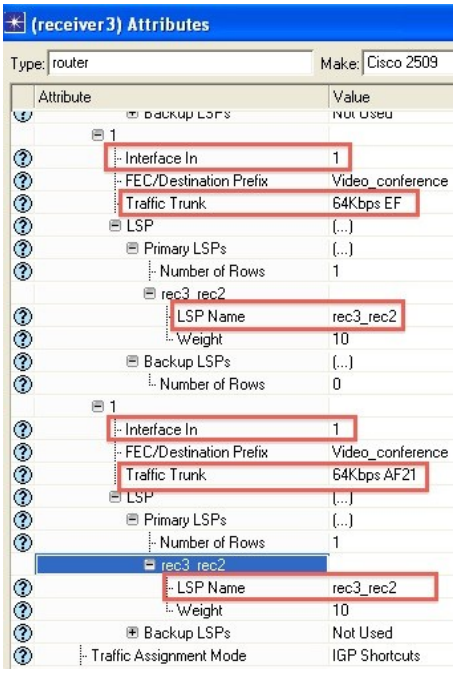


Figure 5-18Traffic mapping of the LSP from receiver 3 to receiver 2

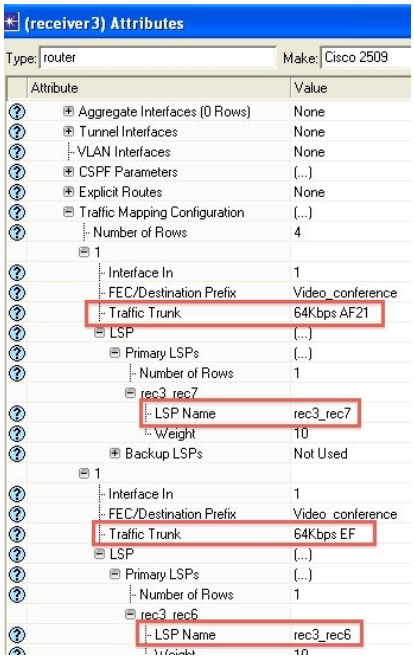


Figure 5-19 Traffic mapping of the LSPs from receiver 3 towards receiver 7 and receiver 6

The newly added group of routers from the midpoint router receiver 3 (LSR) and the other two egress routers, receiver 7 and receiver 6, will be added to the same multicast group 230.10.20.30. This assumes that the newly attached multicast receivers wkstn 6 - wkstn 8 have subscribed to listen to server 1 and server 3 multicast transmission.

A new generated simulation of the multicast traffic with the same parameters as defined in Section 5.3 *B,* but flooded over a larger network as in the Figure 5-20, will further

emphasize the expected result for the path taken by the multicast traffic. Therefore, the multicast traffic is delivered to the destination client, in this case the workstations, but based on the QoS subscription demanded by the client and offered by an ISP through an DS-TE MPLS domain.

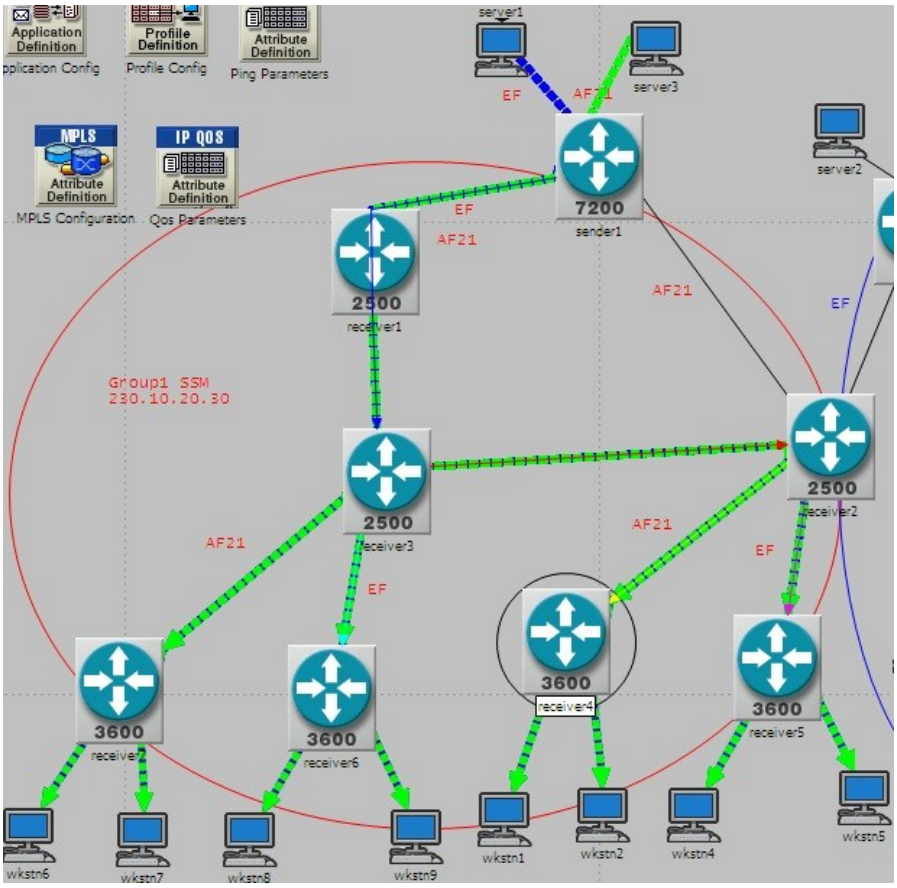


Figure 5-20 Multicast traffic flowing from two multicast sources toward workstations over DS-TE MPLS domain with newly added midpoint (LSR) receiver 3 and two egress router receiver 7, receiver 6

6 Conclusions and Future Work

Multicast traffic distributions, quality of services assurance, and MPLS with constraint based routing (or TE) are important requirements for the ISP providers who accommodate the newly rising video on demand providers. A successful traffic engineering implementation depends on a viable combination of these technologies. The proposed theories were not matured enough to be practically implemented due to the single embedded solutions. This would imply the deployment of multicast traffic distribution in a plain IP network, or multicast distribution in a MPLS domain, that does not consider the traffic of services. Therefore, there is still a vast area to be researched in relation to this.

6.1 Contributions of this Thesis

Overall, this thesis proposes a solution to the multicast traffic distribution over a Diff-Serv aware MPLS-TE domain and furthermore, it designs a practical testbed network within OPNET environment in order to support the proposed solution.

The research of this thesis gathers the following contributions that can be summarized as follows:

- Demonstrated the principal issues of the current and most utilized PIM-SM protocol. The PIM-SM's implementation is based on the arbitrary selection of an RP node by an external user network operator. Unveiled an important, but missing, *controllability* feature which is required by most of the multicast protocols. This controllability feature is constructed through multicast distribution trees.
- Modeled and identified the potential of the PIM-SSM protocol to be used as a substituting medium for the missing *controllability* feature used in the construction of multicast distribution trees.

- Converged the most appropriate and logical location in a network of where to combine this *controllability* feature of the PIM-SSM protocol with the Diff-Serv requirement imposed by the Diff-Serv tunneling models for MPLS.
- Combined the traffic trunks, and their surrounding concepts, to reveal the PIM-SSM protocol's *controllability*'s feature with a Diff-Serv aware MPLS-TE domain. Therefore, two conflicting problems can be solved:
  - The traffic multicast distribution
  - QoS support assurance through Diff-Serv IP QoS model within MPLS-TE
- Devised an algorithm able to deploy the DS-TE MPLS configuration and forward the multicast traffic within an MPLS domain.
- Designed and simulated a FSM of a control server which is capable of controlling the mapping of the IP multicast traffic over a DS-TE MPLS network.
- Implemented a testbed network within OPNET environment and simulated several multicast traffic distribution scenarios. Collected statistics were analyzed, documented and published both in this thesis and [51, 58].

6.2 Future Work

Based on the solution introduced by the research of this thesis, several new directions could be envisioned for future development work:

- The presented results collected within the designed testbed were based on a complete route definition established through a static LSP within a DS-TE domain. The presented methodology to distribute multicast traffic over DS-TE is not limited to only this type of LSP. It could be extended to use a dynamically signaled LSP, such as the CR-LDP protocol. This approach results in the addition of an extension requirement to the link state routing protocols, currently implemented through OSPF or IS-IS. This extension requirement allows the links to propagate information about their ability to accommodate the required constraints. Therefore, one of the possible research subjects could be to find a

method to efficiently and dynamically setup traffic trunks to distribute multicast traffic within a DS-TE domain.

- Another direction would be to extend the specifications of the proposed solution in order to support inter-domain multicast traffic distribution over an inter-domain DS-TE. This could be accomplished by using similar concepts that were applied for the PIM-SSM protocol over DS-TE intra-domain.

Bibliography

[1] B.S. Davie, A. Farrel, “MPLS: Next Steps”, Elsevier Science and Technology Books, 2008.

[2] J. Evans, C. Filsfils, “Deploying IP and MPLS QoS for Multiservice Networks”, Elsevier Science and Technology Books, 2007.

[3] W. Goralski, “The Illustrated Network: How TCP/IP Works in a Modern Network”, Elsevier Science and Technology Books, 2009.

[4] A. Benslimane, “Multimedia Multicast on the Internet”, ISTE. (c) 2007

[5] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, F. Ansari, "Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment", RFC 3353, August 2002.

[6] B.S. Davie, Y. Rekhter, “MPLS: Technology and Applications”, Elsevier Science and Technology Books, 2000.

[7] The Internet Engineering Task Force (IETF) http://www.ietf.org

[8] Open Shortest Path First IGP (OSPF) http://datatracker.ietf.org/wg/ospf/charter

[9] Protocol Independent Multicast (PIM) https://datatracker.ietf.org/wg/pim/charter

[10] Multiprotocol Label Switching www.ietf.org/html.charters/mpls-charters.html

[11] D. Estrin, L. Wei, “A comparison of Multicast Trees and Alrgorithms”, Infocom’94

[12] E. Rosen, A. Viswanathan, R. Callon “Multiprotocol Label Switching Architecture”, RFC 3031, January 2001.

[13] T. Pusateri, “Distance Vector Multicast Routing Protocol”, draft-ietf-idmrdvmrp-v3-as-01, Internet Draft 2004

[14] K. Carlberg, J. Crowcroft, “Building Shared Trees Using a One-to-Many Joining Mechanism”, Computer Communication Review, no. 1, p 5-11, 1997.

[15] M. Faloutsos, A. Banerjea, R. Pankaj, “QoSMIC: Quality of Service Sensitive Multicast Internet Protocol”, SIGCOMM 98, 1998.

[16] S. Chen, K. Nahrstedt, Y. Shavitt, “A QoS-Aware Multicast Routing Protocol”, IEEE INFOCOM, 2000.

[17] D. Farinacci, Y. Rekhter, E. Rosen, T. Qian, "Using PIM to Distribute MPLS Labels for Multicast Routes", Work In Progress.

[18] A. Boudani, B. Cousin, J.M. Bonnin, “MPLS Multicast Traffic Engineering”, IEEE ROC&C, 2003.

[19] C.Y. Lee, L. Andersson, K. Carlberg, B. Akyol, “Engineering Paths for Multicast Traffic using MPLS”, Internet draft, draft-leecy-multicast-te-00.txt, 1999.

[20] B. Yang, P. Mohapatra, “Edge Router Multicast with MPLS Traffic Engineering”, IEEE International Conference On Networks (ICON), 2002.

[21] R. Ravi, “Steiner Trees and Beyond: Approximation Algorithms for Network Design”, CS-93-41, http://citeseer.ist.psu.edu/ravi93steiner.html, 1993.

[22] R. Aggarwal, T. Pusateri, D. Farinacci, L. Wei, “IP Multicast with PIM-SM over a MPLS Traffic Engineered Core”, Network Working Group, Internet Draft, March 2004.

[23] R. Aggarwal, D. Papadimitrou, S. Yasukawa, “Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)”, RFC 4875, May 2007.

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

[25] J. Heinanen, Telia Finland, R. Guerin, "A Single Rate Three Color Marker“, RFC 2697. September 1999.

[26] J. Heinanen, Telia Finland, R. Guerin, "A Two Rate Three Color Marker“, RFC 2698. September 1999.

[27] L.L. Peterson, B.S. Davie, "Computer Networks: A Systems Approach, 4th Edition", Elsevier Science and Technology Books, 2007.

[28] R. Braden, D. Clark, S. Shenker, “Integrated Services in the Internet Architecture: an Overview”, RFC 1633,June 1994.

[29] C. Partrige, “A Proposed Flow Specification”, RFC 1363, September 1992.

[30] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski,“Assured Forwarding PHB Group”, RFC 2597, June 1999.

[31] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval,

J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.

[32] D. Katz, K. Kompella, D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", Proposed Standard, RFC 3630, September 2003.

[33] H. Smit, T. Li, "Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)", RFC 3784, June 2004.

[34] Le Faucheur, F. and W. Lai, "Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering", RFC 3564, July 2003.

[35] Le Faucheur, F. and W. Lai, "Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering", RFC 4125, June 2005.

[36] LeFaucher, "Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering", RFC 4127,June 2005.

[37] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J.McManus, "Requirements for Traffic Engineering over MPLS", RFC 2702, September 1999.

[38] T. Li, Y. Rekhter, "Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)", RFC 2430, October 1998.

[39] B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan," Internet Group Management Protocol, Version 3", RFC 3376, October 2002.

[40] S. Deering, "Host Extensions for IP Multicasting", RFC 1112, 1989.

[41] JunOS Software, "Multicast Protocols Configuration Guide".

[42] OPNET, "OPNET Modeler Documentation Set", Version 16.0, November 2009.

[43] Cisco, "Interdomain Multicast Solutions Using SSM", http://www.cisco.com/en/US/ docs/ios/solutions_docs/ip_multicast/Phase_2/mcst_p2.pdf, March 2001.

[44] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[45] D. Grossman, "New Terminology and Clarifications forDiffserv", RFC 3260, April 2002.

[46] K. Nichols, B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification", RFC 3086, April 2001.

[47] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[48] Cisco Systems, “Cisco Network Planning Solution Models Methodologies and Case Studies”, Software Release 11.0, 2005.

[49] L. Xiaohua, C. Kai, "The Research and Application of IP multicast in Enterprise Network", in Proceedings of the International Conference on Internet Computing and Information Services (ICICIS), 2011; pp.191-194; ISBN 978-1-4577-1561-7.

[50] S. Xu, J. Wu, "A QoS Guaranteed MPLS Multicast Scheme In IPv6 Network", in Proceedings of the 2nd International Conference on Computer Engineering and Technology (ICEET), 2010; pp. V3-232 - V3-235, ISBN 978-1-4244-6347-3.

[51] T. Barabas, D. Ionescu, S. Veres, "PIM-SSM within DiffServ-aware MPLS Traffic Engineering", IEEE SACI 2011, Timisoara, Romania, May 19-21, 2011; pp. 263-268; ISBN 978-1-4244-9108-7.

[52] T.H. Szymanski, ”A Low-Jitter Guaranteed-Rate Scheduling Algorithm for Packet-Switched IP Routers”, IEEE Trans.Comm., Nov. 2009.

[53] T.H. Szymanski, ”Method and Apparatus to Schedule Packets Through a Crossbar Switch with Delay Guarantees”, US Patent Application.

[54] T.H. Szymanski and D. Gilbert, ”Internet Multicasting of IPTV with Essentially-Zero Delay Jitter”, IEEE Trans. Broadcasting, March 2009.

[55] T.H. Szymanski and D. Gilbert, ”Provisioning Mission-Critical Telerobotic Control Systems over Internet Backbone Networks with Essentially -Perfect QoS”, IEEE JSAC, Vol. 28, No. 5., June 2010.

[56] T.H. Szymanski and D. Gilbert, ”Design of an IPTV Multicast System for Internet Backbone Networks”, Int. Journal Digital Multimedia Broadcasting, Vol. 2010, Article 169140.

[57] T.H. Szymanski, “Future Internet Video Multicasting with Essentially Perfect Resource Utilization and QoS Guarantees”, IEEE IWQoS, June 2011.

[58] T. Barabas, D. Ionescu, S. Veres, "A Traffic Engineering Algorithm for Differentiated Multicast Services over MPLS Networks", IEEE SACI 2012, May 2012.