Slovak University of Technology in Bratislava

Faculty of Informatics and Information Technologies

Bc. Ivana Hucková

OPTIMIZATION OF DATA FLOW IN SERVICE PROVIDER NETWORKS

Master's thesis

Degree Course: Computer and Communication Systems and Networks

Field of study: 9.2.4 Computer Engineering

Place of development: Institute of Computer Systems and Networks, FIIT SUT, Bratislava

Supervisor: Ing. Martin Hrubý, PhD.

2014, May

Slovak University of Technology in Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Optimalizácia toku dát v sieti poskytovateľa služby

Študijný program: Počítačové a komunikačné systémy a siete

Autor: Bc. Ivana Hucková

Vedúci práce: Ing. Martin Hrubý, PhD.

Máj 2014

V práci sú opísané základné princípy riadenia premávky spolu s klasifikáciou typov riadenia premávky na základe nimi zvoleného prístupu. V práci sa zameriavame na riadenie premávky v MPLS sieťach a otázku zabezpečenia kvality služieb, ktorá je hlavným dôvodom vývoja riadenia premávky. Analýza pokračuje podrobnou charakteristikou architektúry MPLS. Posledná časť analýzy je zameraná na algoritmy použité pri riadení premávky v MPLS sieťach. V ďalšej časti práce navrhujeme štruktúru a fungovanie online servera použiteľného na optimalizáciu riadenia premávky v MPLS sieti. Navrhnutý server bol implementovaný a jeho funkčnosť overená na navrhnutých testovacích topológiach. Pri testovaní boli generované toky premávky rôznych tried v rôznych množstvách. Vyhodnocovaná bola priepustnosť, vyťaženie tunelov v sieti, stratovosť, oneskorenie a variácia oneskorenia. Výsledky funkcionality servera boli porovnané so situáciou bez použitia servera v rovnakých testovacích podmienkach. Testovanie preukázalo zvýšenie priepustnosti pre jednotlivé triedy premávky, optimálne využitie sieťových zdrojov s ohľadom na zachovanie QoS požiadaviek na jednu z tried.

Slovak University of Technology in Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Optimization of data flow in service provider networks

Degree Course: Computer and Communication Systems and Networks

Author: Bc. Ivana Hucková

Supervisor: Ing. Martin Hrubý, PhD.

2014, May

This work describes the principals and characteristics of traffic engineering together with its classification based on the routing approaches. The work focuses on MPLS TE and the question of quality of service, since it is the main reason of deploying TE. The analysis continues with detailed description of the MPLS architecture. The last part is dedicated to MPLS TE algorithms and newly proposed approaches which were deployed lately. Later we propose the structure of an online server which can be used to optimize the traffic flow in MPLS network. The proposed server was implemented and its functionality tested on proposed topologies. Testing was performed by generating various amounts of classified traffic. The evaluated parameters were throughput, packet loss, utilization of tunnels, delay and jitter. The results of the proposed server were compared to the same testing scenarios without the use of proposed server. The experiments proved higher throughput, optimal distribution of traffic while preserving required QoS for Class1 traffic.

I would like to thank my supervisor Ing. Martin Hrubý, PhD. for help and valuable advices when writing this thesis.

Ivana Hucková

Contents

Introduction 1

1 Traffic Engineering 2

1.1 TE classifications 3
1.1.1 IP-based TE 3
1.1.2 MPLS-based TE 3
1.1.3 Online TE 4
1.1.4 Offline TE 4
1.1.5 Interdomain TE 5
1.1.6 Intradomain TE 5
1.1.7 Multicast TE 5

1.2 TE characteristics 6

1.3 Building blocks of MPLS TE 7
1.3.1 IGP extensions for TE 7
1.3.2 TE tunnel 9
1.3.3 Signaling for TE tunnels 10

1.4 Quality of Service 12
1.4.1 Classification and marking 12
1.4.2 Congestion Avoidance 13
1.4.3 Congestion Management 13
1.4.4 Policing and shaping 14
1.4.5 Measuring the quality factors 15

2 MPLS 16

2.1 History of MPLS 16

2.2 MPLS architecture 18
2.2.1 MPLS label 18
2.2.2 Label stack 18
2.2.3 Label switched router 19
2.2.4 Label switched path 19
2.2.5 Label distribution 20
2.2.6 Cisco Express Forwarding 22

3 MPLS TE Algorithms 23

3.1 Routing algorithms 24
3.1.1 Min-Hop Algorithm (MHA) 24
3.1.2 Widest-Shortest Path algorithm (WSP) 24
3.1.3 Shortest-Widest Path algorithm (SWP) 24
3.1.4 Minimum Interference Routing Algorithm (MIRA) 25
3.1.5 Dynamic Online Routing Algorithm (DORA) 25
3.1.6 Profile-Based Routing (PBR) 26

3.2 Advanced routing algorithms employment 27
3.2.1 RATES 27
3.2.2 Multiple path selection algorithm 29
3.2.3 QoS Routing algorithm with delay and bandwidth constraints 30
3.2.4 Load Balancing Algorithm Using Deviation Path 30
3.2.5 Flow distribution and flow splitting algorithm 31

3.3 Summary 33

4 Proposal 34

4.1 System requirements 35

4.2 Proposed solution 36
4.2.1 Main contribution of this work 37
4.2.2 Classification of traffic 37
4.2.3 Creation of LSPs 38
4.2.4 Measurements of quality parameters 39
4.2.5 Calculating the cost of LSP 40
4.2.6 Assigning traffic trunks to LSPs 42
4.2.7 Optimization of traffic flows 44

4.3 Proposed implementation 46
4.3.1 Communication 46
4.3.2 The measurements 48
4.3.3 The generation of traffic 50

4.4 Topologies and experiments 51

4.5 Software components 53
4.5.1 Server daemon 54
4.5.2 Network analyzer 54
4.5.3 Traffic handler 54
4.5.4 Measurement engine 55
4.5.5 Calculator 55
4.5.6 Database 55

4.6 Summary 59

5 Implementation 60

5.1 Communication 61

5.2 Network configurations 62
5.2.1 Classification of traffic 62
5.2.2 Implementation of IP SLA 62
5.2.3 Implementation of the tunnels 63

5.3 Implementation of the server 65
5.3.1 Database 65
5.3.2 Daemon 66
5.3.3 Network analyzer 66
5.3.4 Measurement engine 67

5.3.5 Traffic handler 69
5.3.6 Calculator 73

6 Experiments 74

6.1 Experiment 1 75
6.1.1 Testing scenario 75
6.1.2 Evaluation 76

6.2 Experiment 2 79
6.2.1 Testing scenario 79
6.2.2 Evaluation 80

6.3 Experiment 3 83
6.3.1 Testing scenario 83
6.3.2 Evaluation 83

6.4 Experiment 4 87
6.4.1 Testing scenario 87
6.4.2 Evaluation 87

6.5 Summary 90

7 Conclusion 91

8 Resumè 92

8.1 Experiment č. 1 95

8.2 Experiment č. 2 96

8.3 Experiment č. 3 97

8.4 Experiment č. 4 98

References 100

Appendix A 102

Appendix B 103

List of acronyms

AAL5 - ATM Adaptation Layer 5

ASBR- Autonomous System Border Router

ATM - Asynchronous Transport Mode

AToM - Any Transport over MPLS

BoS - Bottom of Stack

CBWFQ - Class-based Weighted Fair Queuing

CE - Customer Edge

CEF - Cisco Express Forwarding

COPS - Common Open Policy Service

DiffServ – Differentiated Services

DORA - Dynamic Online Routing Algorithm

DSCP - Differentiated Service Code Point

FEC - Forwarding Equivalence Class

FIB - Forwarding Information Base

FIFO - First In First Out

ICMP – Internet Control Message Protocol

IGP – Interior Gateway Protocol

IntServ – Integrated Services

ISP – Internet Service Provider

IS-IS – Intermediate System to Intermediate System

LBDP - Load Balancing algorithm using Deviation Path

LDP - Label Distribution Protocol

LFIB - Label Forwarding Information base

LLQ - Low Latency Queuing

LSA- Link-state Advertisement

LSP - Label Switched Path

LSR - Label Switched Router

MHA - Min-Hop Algorithm

MIRA - Minimum Interference Routing Algorithm

MPLS – Multiprotocol Label Switching

NSIS – Next Steps in Signaling

TE – Traffic Engineering

TLV - Type Length Values

OSPF – Open Shortest Path First

PE - Provider Edge

PHB - per hop behavior

PQ - Priority Queuing

PBR - Profile-Based Routing

QoS – Quality of service

RATES - Routing and Traffic Engineering Server

RD - Route Distinguisher

RED - Random Early Detection

RIB - Routing Information Base

RSVP – Resource Reservation Protocol

RT - Route Target

SLA – Service Level Agreement

SNMP – Simple Network Management Protocol

SWP - Shortest-Widest Path

ToS - Type of Service

TTL - Time To Live

VC - Virtual Circuit

VPN - Virtual Private Network

VRF - Virtual Routing Forwarding

WFQ - Weighted Fair Queuing

WRED- Weighted Random Early Detection

WSP - Widest-Shortest Path

List of figures

Figure 1.1 – TE example 2
Figure 1.2 - - RSVP label distribution 11
Figure 1.3 - Policing 14
Figure 1.4 - Shaping 14
Figure 2.1 - MPLS label 18
Figure 2.2 - The location of label 19
Figure 2.3 - Nested LSP 20
Figure 2.4 - LDP label distribution 21
Figure 2.5 - Attached label 21
Figure 4.1 - The main system's processes 36
Figure 4.2 - Assigning data to LSP 42
Figure 4.3 - Assigning Class1 to LSP 43
Figure 4.4 - The process of optimization 45
Figure 4.5 - The scheme of communication 47
Figure 4.6 - The scheme of SNMPv3 connection 47
Figure 4.7 – The scheme of SSH connection 48
Figure 4.8 - IP SLA configuration scheme 50
Figure 4.9 - Implemented topology 1 52
Figure 4.10 - The architecture of the server 53
Figure 4.11 - The database 58
Figure 5.1 - Network IP adressing 60
Figure 5.2 - The configuration of SNMP 61
Figure 5.3 - The configuration of SSH 61
Figure 5.4 - Example of classification on CE router 62
Figure 5.5 - The example of IP SLA configuration 63
Figure 5.6 - The example of tunnel configuration 64
Figure 5.7 - Trigger for updating act_cost on LSP 65
Figure 5.8 - Trigger for updating unused_bw of LSP 66
Figure 5.9 - Trigger for updating in_bw of class 66
Figure 5.10 - Operation of the Network Analyzer 67
Figure 5.11 - Example of policy configuration 68

Figure 5.12 - Operation of the Measurement engine 69
Figure 5.13 - Function act_cost_alarm 70
Figure 5.14 - The function in_bw_high 71
Figure 5.15 - Function find_tunnel_for_extra_traffic 72
Figure 5.16 - The function optimize 73
Figure 6.1 – Experiment 1, Input bandwidth, without TE server 76
Figure 6.2 – Experiment 1, Input bandwidth, with TE server 76
Figure 6.3 – Experiment 1, Throughput, without TE server 76
Figure 6.4 – Experiment 1, Throughput, with TE server 76
Figure 6.5 – Loss, without TE server 77
Figure 6.6 – Loss, with TE server 77
Figure 6.7 – Experiment 1, Utilization of tunnels, without TE server 77
Figure 6.8 – Experiment 1, Utilization of tunnels, without TE server 77
Figure 6.9 - Experiment 1, Delay, without TE server 77
Figure 6.10 - Experiment 1, Delay, with TE server 77
Figure 6.11 - Experiment 1, Jitter, without TE server 78
Figure 6.12 - Experiment 1, Jitter, with TE server 78
Figure 6.13 – Experiment 2, Input bandwidth, without TE server 80
Figure 6.14 - Experiment 2, Input bandwidth, with TE server 80
Figure 6.15 - Experiment 2, Throughput, without TE server 81
Figure 6.16 - Experiment 2, Throughput, with TE server 81
Figure 6.17 - Experiment 2, Loss, without TE server 81
Figure 6.18 - Experiment 2, Loss, with TE server 81
Figure 6.19 – Experiment 2, Utilization of tunnels, without TE server 81
Figure 6.20 – Experiment 2, Utilization of tunnels, with TE server 81
Figure 6.21 - Experiment 2, Delay, without TE server 82
Figure 6.22 - Experiment 2, Delay, with TE server 82
Figure 6.23 - Experiment 3, Jitter, without TE server 82
Figure 6.24 - Experiment 3, Jitter, with TE server 82
Figure 6.25 - Experiment 3, Input bandwidth, without TE server 84
Figure 6.26 - Experiment 3, Input bandwidth, with TE server 84
Figure 6.27 - Experiment 3, Throughput, without TE server 84
Figure 6.28 - Experiment 3, Throughput, with TE server 84

Figure 6.29 - Experiment 3, Loss, without TE server 85
Figure 6.30 - Experiment 3, Loss, with TE server 85
Figure 6.31 - Experiment 3, Utilization of tunnels, without TE server 85
Figure 6.32 - Experiment 3, Utilization of tunnels, with TE server 85
Figure 6.33 - Experiment 3, Delay, without TE server 85
Figure 6.34 - Experiment 3, Delay, with TE server 85
Figure 6.35 - Experiment 3, Jitter, without TE server 86
Figure 6.36 - Experiment 3, Jitter, with TE server 86
Figure 6.37 – Experiment 4, Input bandwidth, without TE server 88
Figure 6.38 - Experiment 4, Input bandwidth, with TE server 88
Figure 6.39 - Experiment 4, Throughput, without TE server 88
Figure 6.40 - Experiment 4, Throughput, with TE server 88
Figure 6.41 - Experiment 4, Loss, without TE server 88
Figure 6.42 - Experiment 4, Loss, with TE server 88
Figure 6.43 - Experiment 4, Utilization of tunnels, without TE server 89
Figure 6.44 - Experiment 4, Utilization of tunnels, with TE server 89
Figure 6.45 - Experiment 4, Delay, without TE server 89
Figure 6.46 - Experiment 4, Delay, with TE server 89
Figure 6.47 - Experiment 4, Jitter, without TE server 89
Figure 6.48 - Experiment 4, Jitter, with TE server 89
Figure 0.1 – Initial settings 103
Figure 0.2 – Connection fail 104
Figure 0.3 - Successful connection 105
Figure 0.4 - The analysis of the network 105
Figure 0.5 - Help 105
Figure 0.6 - Logging enabled 105
Figure 0.7 - The start of TE server 106
Figure 0.8 - Re-configuration due to high input rate 106
Figure 0.9 - Re-configuration due to optimization 106

List of tables

Table 1.1 – OSPF Link TLV Sub-TLVs [3] 8
Table 1.2 – IS-IS Sub-TLVs of TLV type 22 [3] 9
Table 5.1 - The mapping between IP Precedence and EXP values 38
Table 5.2 - Reference table for delay values 41
Table 5.3 - Reference table for jitter values 41
Table 5.4 - Reference table for packet loss values 41
Table 5.5 - The meaning of cost values 41
Table 4.6 - Bandwidth of LSPs 51

Introduction

The creation of Internet started as a closed network consisting of few computers in Pentagon called Arpanet in 1969. Since then, it has undergone huge development to become this great communications and information facility. Internet as we know it nowadays represents a multifunctional tool for interconnecting, communication, education, entertainment, sharing or any other action one can imagine. Thanks to the design of the protocols and underlying technologies on which it is built, Internet can expand at great rate. It is built in hierarchical layered architecture in which numbers of private and public networks are connected. At each level, individual network operators maintain peering relationships with other operators at the same level. The center of the Internet is created by "Tier-1" ISPs which provide national and international connections. These ISPs treat each other as equals. "Tier-2" ISPs are smaller and often provide regional service. Tier-2 ISPs usually pay Tier-1 ISPs for connectivity to rest of the Internet. "Tier-3" ISPs are the local providers of service directly to end users. Tier-3 ISPs are usually connected to Tier-2 ISPs and pay Tier-2 providers for Internet access.

For the end customers the architecture of the Internet is not important since they mostly focus on the Internet connection and quality of provided services. Therefore many different applications and approaches are developed to improve the performance and to provide effective resource utilization.

This work focuses on using MPLS in network providers networks with the use of different traffic engineering approaches. The traffic engineering is classified into multiple classes according to its characteristics and each class is shortly described. The TE characteristics and building blocks of MPLS TE are described in detail in following sections.

The next part of this work is dedicated to the MPLS technology, its history and architecture which is analyzed in detail. The use of MPLS VPNs is described together with its classification into layer-2 and layer-3 VPNs.

In the last part of this work various MPLS TE approaches and algorithms are analyzed. The main principles of each algorithm are described .

1 Traffic Engineering

The Internet has over the years become a multiservice network that supports many types of multimedia applications with different demands. Customer traffic often suffers from congestion due to the bottlenecks in the network which leads to degradation of service's quality. Traffic engineering as a way of efficient resource optimization is being deployed to address this problem. By balancing the traffic load distribution in the network and minimizing bandwidth consumption, traffic engineering provides the maximization of network's utilization. Simplified view of how TE works in shown in Figure 1.1.

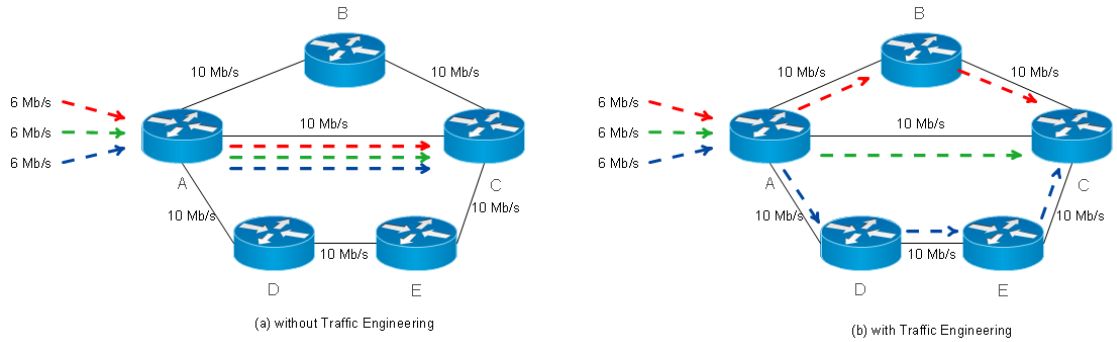

Figure 1.1 – TE example

Besides the network utilization, TE also deals with the question of quality of service (QoS). Many applications require certain QoS guarantees, such as end-to-end delay, jitter or loss probability. These requirements need to be addressed by TE mechanisms in order to provide satisfying services to customers.

1.1 TE classifications

TE routing approaches can be classified as follows [1]:

- IP-based and MPLS-based TE – from the aspect of routing enforcement mechanisms
- online and offline TE – from the aspect of availability of traffic demand or timescale of operations
- interdomain and intradomain TE – from the aspect of traffic optimization scope
- unicast and multicast TE – from the aspect of traffic type

1.1.1 IP-based TE

Conventional IP networks use IGPs such as OSPF or IS-IS to forward IP packets on the shortest cost path toward the destination. Each link has its metric and the cost of the path is the sum of the link metrics on the path.

The main problem in IP networks is that traffic aggregates on the shortest paths thereby causing congestion on these links while links on alternative paths remain underutilized. This leads to suboptimal use of network resources and affect the quality of service.

To control the traffic distribution in IP networks changing the IGP metrics on links is used [12]. The advantage of this approach is remained scalability of the network since alternative paths are still available. On the other hand, changing the metric on one link in the network may affect traffic distribution in other parts of the network. Also one link failure can cause congestion on another link, since traffic is automatically rerouted on the shortest path available. Another drawback of IP-based TE is the lack of explicit routing and uneven traffic splitting. The effectiveness of IGP metric-based TE is dependent on the network topology, traffic demand matrix and optimization goal [2].

Due to these drawbacks in IP-based TE the use of MPLS-based TE (explained in the next section) is preferred and widely used.

1.1.2 MPLS-based TE

Multiprotocol Label Switching (MPLS) TE uses the implicit MPLS characteristics to make routing decisions based on other criteria than the destination address of packet.

MPLS TE provides constraint-based path computation and explicit routing capabilities to divert traffic away from congested parts of the network.

By setting up dedicated label switch paths (LSP), MPLS TE can provide an efficient distribution of traffic. The disadvantages of this approach are the additional overhead produced by creating LSPs and also the total number of LSP in the network. In large networks this can be an issue, since the number of LSPs can become very high – when considering full mesh topology. Also the necessity of backup LSPs can be considered as a disadvantage, especially when compared to IP-based TE.

However, the main advantage of MPLS-based TE is in its capability of explicit routing and arbitrary traffic splitting, which is used to optimize the traffic flows and maximize the network's utilization. Also, MPLS-based TE achieves high robustness, since single link failure does not negatively impact traffic distribution in other parts of the network.

More details on this topic will be presented in later section of this work.

1.1.3 Online TE

The main advantage of online TE is its dynamic and rapid reaction to traffic changes. It does not require any information about actual traffic flows or future traffic demands, yet provides optimal assignment of incoming traffic onto the network. In this approach, order of traffic demands is crucial, since the traffic is assigned one by one [1]. Traffic can be rerouted when required, although this action should not involve significant amount of traffic.

Online TE may experience difficulties in handling future traffic since the traffic pattern is not known at the time of providing the optimization. Another issue is the question of self-convergence of the system without any human intervention [14].

1.1.4 Offline TE

Offline TE requires the knowledge of future traffic demands before performing the optimization. This knowledge can be gained by monitoring and measurements of the network or analyzing a service level specification [13]. It is usually performed by a server outside the network. By knowing all traffic requirements, offline TE is able to

optimally map all traffic onto the physical network. The order of demands is not important in this approach.

The disadvantage of offline TE lies in the lack of adaptive traffic manipulation, since it operates with forecasted traffic matrix. Another issue can be the difference between forecasted traffic matrix and the actual traffic pattern in the network. Also, traffic burst and link failures are not taken into account when providing offline TE.

Cooperation of online and offline TE is considered to be a good solution to overcome the disadvantages mentioned [1].

1.1.5 Interdomain TE

Interdomain TE provides the optimization of traffic flows across multiple autonomous systems (AS). This type of TE focuses on selecting AS border routers (ASBRs) as ingress/egress points for interdomain traffic in local AS. ASBRs are selected for both incoming and leaving traffic flows. Therefore, interdomain TE can be further divided into inbound TE for traffic entering the network and outbound TE for traffic leaving the network [15].

1.1.6 Intradomain TE

The main goal of intradomain TE is to optimize the path selection between the pair of ASBRs within a single domain.

Since both interdomain and intradomain TE affect the path of traffic flows, they should not be considered independently, but in cooperation [1].

1.1.7 Multicast TE

The main goal of multicast TE is to minimize the consumption of bandwidth in the network. This is also known as bandwidth conservation, where traditional routing mechanisms are not optimal solutions. There are also other TE objectives such as throughput maximization or load-balancing of traffic, that has to be fulfilled.

1.2 TE characteristics

Traffic engineering can be defined as a way of optimal routing of traffic flows across the network to achieve desired network performance. The main goal is to maximize the utilization of the network resources while providing end-to-end QoS for end users.

In traditional IP network routing is provided by choosing the least-cost path through the network. The cost of each path can differ for each IGP used, but the main problem resists. This kind of routing does not take the available bandwidth capacity of links into consideration which leads to overutilizing the best paths while underutilizing other possible, but not chosen paths. Therefore, traffic suffers from congestion and loss while the network is not optimally utilized.

One solution for this problem could be adjustment of link cost used by IGP to provide equal load-balancing. The traffic would be distributed more evenly, but not perfectly, since links have different bandwidth, which can be upgraded anytime. This change would increase the complexity of the problem, since it would require further change of IGP cost on the links. Traffic engineering is a solution for this problem and this work will focus mainly on traffic engineering used in MPLS networks.

MPLS TE provides these functions [3]:

- Avoiding overutilized and underutilized links in the network by efficient spreading of traffic
- Adapting dynamically to changes in bandwidth and attributes of TE links
- Taking into account configured bandwidth of the links
- Taking into account the attributes of the link, such as delay and jitter

Avoiding overutilized and underutilized links in the network is achieved by a TE scheme where the head end router of LSP calculates the most efficient path through the network. The head end router needs to have information about the topology and bandwidth on each link. This information with MPLS enabled on the routers allows for source-based routing instead of traditional destination-based routing.

1.3 Building blocks of MPLS TE

The building blocks of MPLS TE involve [3]:

- Link constraints (the bandwidth of each link and which TE tunnel can use it)
- Distribution of TE information (MPLS TE-enabled link-state routing protocol is required)
- A signaling protocol for TE tunnels
- A algorithm to calculate the best paths in the network
- A way to forward traffic onto the TE tunnel

1.3.1 IGP extensions for TE

Link constraints are configured on each link and advertised by link-state protocol. For the distribution, MPLS TE-enabled link-state protocols are used, such as OSPF or IS-IS. These extended IGPs need to carry this information of a link [3]:

- TE metric – represents TE value, which can differ from IGP metric of the link
- Maximum bandwidth – physical or configured bandwidth of the link
- Maximum reservable bandwidth – maximum bandwidth of the link available for TE
- Unreserved bandwidth – remainder of the link's bandwidth available for TE
- Administrative group – 32-bit field which can be used by the operator of the network

All of this information is flooded either periodically or when a change occurs.

OSPF extensions for TE

Extensions to OSPF had been made in order to provide the possibility of flooding resource information required by TE. A new class of link-state advertisements called Opaque LSAs has been created and consists of three new LSAs – type 9, 10 and 11. Opaque LSAs consist of a standard LSA header followed by a 32-bit application-specific information field [4].

The main difference among Opaque LSAs is in their flooding scope. Opaque LSA type 9 has only link-local scope, so they are not flooded beyond the local network.

Type 10 denotes an area-local scope which means their flooding is stopped by ABRs. Opaque LSA type 11 is flooded through the whole AS (as type 5 LSAs).

In the Options field of OSPF was defined a new bit – the O-bit – to indicate whether a router is capable of processing Opaque LSAs.

The Opaque LSA type 10 is used for TE since it carries one or more Type Length Values (TLV). Two kinds of TLV exist and they carry all the information that is needed by TE. The *Router Address TLV* is used to carry the router ID for TE and *Link TLV* carries a number of sub-TLVs with details about link attributes for MPLS TE. Information carried by these sub-TLVs is shown in Table 1.1 [3]. Their meaning is as follows:

- Link type – point-to-point or multi-access link
- Link ID – is set to the router ID of the neighbor. In case the link is multi-access, it is set to the interface address of the designated router
- Local/Remote interface IP address – the IP address of local/remote interface
- Traffic engineering metric – the metric used by TE
- Bandwidth parameters – expressed in B/s. The unreserved bandwidth uses the legth of 32 octets due to its expression in 4 octets for each of eight priority level. These priority levels are used by MPLS TE tunnel
- Administrative group – unspecified 32-bit field

Sub-TLV Number	**Name**	**Length (octets)**
1	Link type	1
2	Link ID	4
3	Local interface IP address	4
4	Remote interface IP address	4
5	Traffic engineering metric	4
6	Maximum bandwidth	4
7	Maximum reservable bandwidth	4
8	Unreserved bandwidth	32
9	Administrative group	4

Table 1.1 – OSPF Link TLV Sub-TLVs [3]

IS-IS extensions for TE

To enable IS-IS to carry TE information, two new IS-IS TLVs have been defined. Besides, other changes have been made, such as the extension of the link metric in these TLVs (from 63 to 2^{24}-1), sub-TLVs, and the introduction of a down bit [3].

The first new TLV is type 22, which extended the IS Reachability TLV (type 2). It describes the neighbors and the cost among them. The second TLV is TLV type 135, which has extended the IP Reachability TLVs (type 128 and 130).

TLV type 22 carries the sub-TLVs required by MPLS TE. Details on these sub-TLVs are listed in Table 1.2.

Sub-TLV Number	Name	Length (octets)
0-2	Unassigned	--
3	Administrative group	4
4-5	Unassigned	--
6	IPv4 interface address	4
7	Unassigned	--
8	IPv4 neighbor address	4
9	Maximum link bandwidth	4
10	Reservable link bandwidth	4
11	Unreserved bandwidth	32
12-17	Unassigned	--
18	TE default metric	3
19-254	Unassigned	--
255	Reserved for future expansion	--

Table 1.2 – IS-IS Sub-TLVs of TLV type 22 [3]

1.3.2 TE tunnel

The TE tunnel represents the path dedicated to the data flow routed through the network. It can be set up either explicitly or dynamically. The TE tunnel which is set up explicitly has specified every router along the path from head end router to tail end router. This can be done either by specifying the TE router ID or the link IP address of the intermediate routers. When setting up the TE tunnel dynamically, the whole path towards the tail end router is selected by the head end router. The only information needed is the destination of TE tunnel. The head end router selects the path based on information in MPLS TE database learned from OSPF or IS-IS, while it takes resources on the links into account.

In the network more than one dynamic and explicit path option can be configured if they have different preference (a number from 1 to 1000). The path option with lower preference number is the one preferred. Each tunnel has two types of priorities: setup and holding priority. The setup priority indicates the importance of the tunnel among other tunnels and the holding priority indicates the level of possible preemption by other tunnels. Important tunnels use to have low setup priority, which

means that they can preempt other tunnels, and low holding priority, which means that they cannot be preempted by other tunnels [3].

It may occur that the path which was selected for the TE tunnel is no longer the best possible path in the network. This can happen if a new link arises in the network or if parameters of another link suddenly change to a better state. The reoptimization of the TE tunnel is needed in such a situation, so that the tunnel is re-routed onto a more optimal path in the network. The reoptimization can be caused by three triggers: periodic reoptimization, event-driven reoptimization, and manual reoptimization [3].

The TE metric used to route the TE tunnel is by default equal to IGP link metric. This option can be overridden by setting the TE metric to another specific value. This way is possible to use another metrics to route TE tunnels than to route classic IP traffic.

1.3.3 Signaling for TE tunnels

For creation of TE tunnel and for hop-by-hop propagation of labels used a signaling protocol for TE tunnels is required. In the past, two signaling protocols were proposed: RSVP-TE and CR-LDP (constrained-based LDP). The IETF made a decision to further develop the RSVP-TE and to stop any development of CR-LDP. This is documented in [19]. To accomplish these requirements RVSP was enhanced so that it can signal TE tunnels across the network.

RSVP uses the RSVP PATH message and RSVP RESV message to signal the TE tunnel across the network. The RSVP PATH message is sent by the head end router to the tail end router carrying a request for an MPLS label. The tail end router responses with RSVP RESV message in case the TE tunnel can be created. RSVP RESV message contains the MPLS label that each LSR along the tunnel can use for forwarding the TE traffic. RSVP verifies whether the TE tunnel with constraints can be set up on each node. This should not be a problem since an IGP advertises this information. However, a situation may occur when another TE tunnel has reserved an amount of bandwidth on a specific path and IGP has not advertised this change yet. In this case, RSVP does not reserve the required bandwidth and the tunnel has to be routed on another path [5].

RSVP also supports explicit routing capability by using Explicit Route Object (ERO). ERO encapsulates a concatenation of hops which constitutes the explicitly routed path. Using this object, the paths taken by label-switched RSVP-MPLS flows can be pre-determined, independent of conventional IP routing. At each hop the PATH

message temporarily reserves the bandwidth required and requests a label. When the PATH message gets to the tail end router, it returns a RESV message with the selected label using the same path in opposite direction. The RESV message also confirms the reservation of resources for the links at each intermediate LSR [5].

The exact process of using the RSVP PATH and RESV messages is shown in Figure 1.2. Besides the signaling for the TE tunnel, RSVP also carries the MPLS label across the network. The PATH message carries the Label Request object from the head end router (R6) to the tail end router (R5). The tail end router (R5) assigns a label to this TE tunnel and advertises it with the Label object in RSVP RESV message to the upstream router (R2). This advertised label is the incoming label in the LFIB of the router R5. Router R2 receives the label from the router R5 and uses this label as the outgoing label in the LFIB for this TE tunnel LSP. Router R2 then assigns a label from the global label table to this LSP and advertises it to the router R1. This way is the label distributed through the network all the way to the head end router. This form of distribution (from tail end router to the head end router, hop by hop) after the request from head end router is called Downstream-on-Demand (DoD) label distribution [3].

Another object used in RSVP is the Record Route object (RRO). It is used by both the PATH and RESV message to record the IP addresses of routers that the TE tunnel traverses. Also, the label used at each hop can be recorded into RRO.

There are three possible uses of RRO in RSVP. The RRO can be used as a loop detection mechanism to discover L3 rooting loops, RRO collects actual detailed path information hop by hop about RSVP sessions and RRO could be, with minor changes, used as input to the Explicit Route object (ERO).

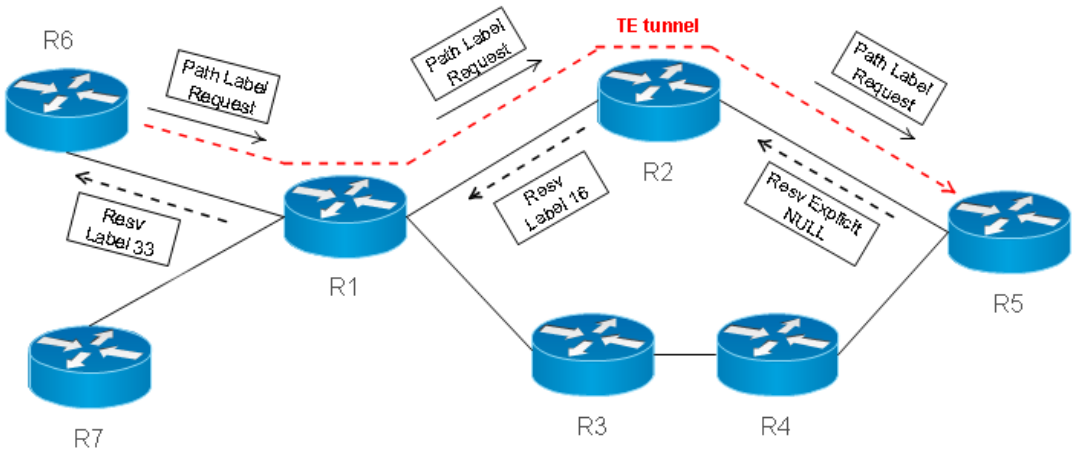

Figure 1.2 - - RSVP label distribution

1.4 Quality of Service

QoS (Quality of Service) provides methods to guarantee a certain level of performance to a data flow. It is possible to coordinate the overall behavior in the network by assigning a priority to certain type of data. There are two general approaches to QoS:

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

When using IntServ, each application that wants to have any guarantee has to make a reservation. Typical example is RSVP mentioned earlier – the path for data flow is reserved by using PATH and RESV messages.

The newer form of signaling is represented by NSIS (Next Steps in Signaling), which has extended RSVP so its components are usable for different needs in different parts of the Internet and it does not require complete end-to-end deployment [7].

Several problems exist when using IntServ. Every router in the network needs to store many states for all the application requiring reservation. Another considerable drawback is the fact that if a router cannot reserve the required bandwidth, the connection fails.

DiffServ uses the 6-bit Differentiated Services Code Point (DSCP) field in IP header, which replaced the Type of Service (ToS). This approach operates on the principle of traffic classification, where each node in the network implements Pre-Hop Behavior (PHB). Based on a class of traffic PHB defines how the packet should be forwarded. There are five processes included in DiffServ: classification, marking, policing, shaping and queuing [8].

1.4.1 Classification and marking

In the classification part, every packet is analyzed and categorized based on defined parameters (source or destination address, type of application, etc.). Multiple groups are created, so the network traffic is divided into priority levels, or classes of service.

In the next step each packet is marked based on its classification. Marking can be done in variety of ways [21]:

- Layer-2: 802.1p, ATM CLP bit, Frame-Relay DE bit, MPLS EXP bits, etc.
- Layer-3: IP Precedence, DSCP field

1.4.2 Congestion Avoidance

Common procedure to provide the required level of quality of service is congestion avoidance. There are various techniques to monitor network traffic loads to predict and avoid congestion at different network bottlenecks:

- Tail Drop - this type of congestion avoidance treats all traffic equally and does not differentiate among the classes of service. Tail drop represents the most simple congestion avoidance technique since in time of congestion it drops all packets until the congestion is eliminated [20].
- Random Early Detection (RED) was created to address the problem of network congestion in a responsive way. It was meant to be used with transport protocols such as TCP which can react appropriately to sudden packet loss by slowing down their traffic transmission.

  RED avoids the congestion by controlling the average queue size and randomly drops packet when this average value is reached. As a reaction to this loss of packets, TCP starts slowing its transmission rate until the congestion is cleared [20].
- Weighted Random Early Detection (WRED) combines the capabilities of RED while it uses IP Precedence bits to differentiate among various classes of service. WRED provides separate thresholds and weights for each IP Precedence and therefore it selectively discards packets from lower priority class when the congestion occurs and provides differentiated performance characteristics for different classes of service [20].
- Flow-Based WRED provides greater fairness to all flows compared to WRED regarding how packets are dropped. Therefore, even flows which have just a few packets are susceptible to packet drop in case of congestion. To provide fairness to all flows, it ensures that flows that respond to packet drops (by slowing down the transmission) are protected from flows that do not respond to packet drops. Also, it prohibits a single flow from using all available resources [20].

1.4.3 Congestion Management

In case congestion avoidance does not provide the required protection and the congestion occurs, it is necessary to apply queuing techniques to ensure that the critical

applications get the required forwarding treatment. Many different queuing techniques exist such as FIFO, LLQ, PQ, WFQ, and CBWFQ [21].

- FIFO (First In First Out) – packets are forwarded in the same order in which they arrived at the interface
- PQ (Priority Queuing) – offers four sub queues with fixed priority (low, medium, normal, high). Received packets are stored in these queues according to their priorities. When the congestion clears packets with highest priority are send first, followed by lower priority packets. This queuing technique can create a flow starvation for low priority packets since these can be waiting forever.
- WFQ (Weighted Fair Queuing), CBWFQ (Class-based WFQ) – this method offers the opportunity to create number of sub queues and define the used bandwidth for each one. With CBWFQ there are also classes considered.
- LLQ (Low Latency Queuing) – is a combination of PQ and CBWFQ. There is one priority class (PQ used) for the most important traffic flow and number of classes according to CBWFQ. Therefore when there are no packets from the priority class, CBWFQ is used. On the other hand, when packets from the priority class arrive everything else stops and the priority flow is served.

1.4.4 Policing and shaping

Policing and shaping procedures are used to limit the amount of the traffic flow. The main difference between them is that policing drops all traffic that exceeds given limit, whereas shaping regulates the traffic by delaying and queuing packets. This difference is shown in Figure 1.3 and Figure 1.4.

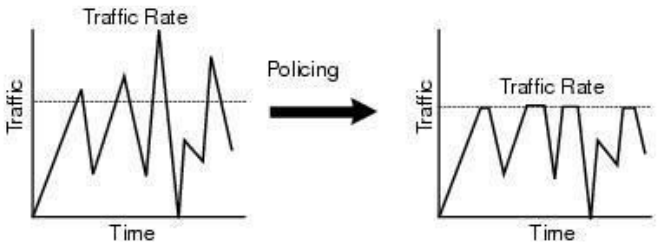

Figure 1.3 - Policing

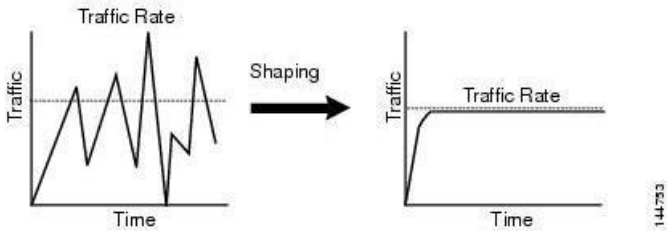

Figure 1.4 - Shaping

1.4.5 Measuring the quality factors

The quality of service provided by the network is often measured in terms of delay, jitter or packet loss. Each of these network parameters contributes to the overall quality of service. Provided QoS in the network is crucial especially for real-time traffic such as voice or video. To effectively provide required QoS the measurement of these network parameters has to be done, usually using tools such as NetFlow [10], IP SLA [9] and SNMP.

IP SLA is an embedded agent in Cisco IOS software designed to measure and monitor network parameters (jitter, delay and packet loss). IP SLA operations are based on active probes, which generate synthetic network traffic for the purpose of measuring network performance. It has two main components – the source and the target. The source defines the IP SLA operations and generates the synthetic network traffic. It also analyzes the results of the measurements so it can be accessed by SNMP.

The target can wary depending on the type of IP SLA operation. It can be a FTP or HTTP server when FTP/HTTP operations are used. For measuring UDP jitter, the target has to be a device with the responder feature enabled, since it has to participate on the measurement by inserting timestamps into the packet payload.

IP SLA offers variety of types of measurements such as ICMP Echo, ICMP Path Echo, ICMP Jitter, ICMP Path Jitter, UDP Echo, UDP Path Echo, TCP Connect, etc. Each of these types is dedicated to a specific network parameter.

NetFlow is a technology available on Cisco devices which provides monitoring of IP traffic flows in the network used to determine the bandwidth usage in the network and therefore provides accurate capacity planning. It also helps to choose the best place for applying QoS, optimize resource usage and detect DoS (Denial of Service) attacks.

The basic operation of NetFlow is dividing the traffic into IP Flows. IP Flow contains similar packets, where the similarity is defined in terms of 5-7 packet attributes (IP source address, IP destination address, source port, destination port, Layer 3 protocol type, Class of Service, router or switch interface). Grouping packets into flows is scalable because a large amount of network information is stored in NetFlow cache.

Data produced by NetFlow can be accessed either by CLI (show commands) or by a reporting server called “NetFlow Collector”. Network Collector is used for assembling and understanding the exported flows and producing valuable reports for traffic and security analysis.

2 MPLS

Multiprotocol Label Switching (MPLS) presents a popular networking technology that is widely used nowadays. It forwards packets through the network based on special labels attached to them and does not use IP addresses for packet routing. This functionality together with number of benefits has led to the popularity of MPLS [6]. The benefits of MPLS include better IP over ATM integration, BGP-free core, use of VPNs, the use of unified network infrastructure and traffic engineering. Details on MPLS evolution and architecture are explained in next sections.

2.1 History of MPLS

The idea of switching or using labels instead of IP addresses to forward the traffic has not been brought with MPLS. Frame Relay and ATM use switching to forward frames or cells through the network. Both Frame Relay and ATM use identification of the virtual circuit which the frame or cell resides on. The main difference between them is that the frame in Frame Relay can have variable length, whereas the cell in ATM has fixed length of 53 bytes.

With the popularity of Internet, IP became widely used. At that time, ATM was used as layer-2 protocol in the core of service provider networks. Service providers began deploying IP backbones and the integration of IP over ATM was required. Since this process was not trivial, the networking community came up with a number of solutions [3].

One of the solutions was to implement IP over ATM using ATM Adaptation Layer 5 (AAL 5) as described in RFC 2684. This solution offers two ways of carrying connectionless traffic over the ATM network: the “LLC Encapsulation” and the “VC Multiplexing” method. The LLC Encapsulation method allows multiplexing of multiple protocols over a single ATM VC. The protocol type of each PDU is identified by a prefixed LLC header. In general, LLC encapsulation tends to require fewer VCs in a multiprotocol environment. In the VC Multiplexing method, each protocol type is carried by one ATM VC. Therefore, if there are multiple protocols used, there is a separate VC for each. This method tends to reduce fragmentation overhead [11].

Another solution for the integration of IP over ATM was the LAN Emulator (LANE). Ethernet as layer-2 protocol became popular at the edge of the network, but has never been used in large service provider networks. In general, LANE makes the network look like an emulated Ethernet network and ATM WAN network looks like an Ethernet switch.

The tightest but most complex solution for integrating IP over ATM was Multiprotocol over ATM (MPOA) proposed by ATM Forum.

All of these methods were difficult to implement and troubleshoot, which led to the invention of MPLS. The only condition was for ATM switches to become more intelligent – to run an IP protocol and implement a label distribution protocol [3].

2.2 MPLS architecture

The operation of MPLS is based on using labels for forwarding of the packets. Therefore, the MPLS label is the most important item in this architecture. These labels are packed in a label stack in the packet. For the labels to be correctly processed, special equipment is needed. Routers supporting MPLS are called Label Switched Routers (LSRs) and a sequence of LSRs is called a Label Switched Path (LSP). For proper operation of MPLS a label distribution protocol is needed, such as LDP.

2.2.1 MPLS label

The MPLS label consists of 32 bits and its structure is shown in Figure 2.1. The first 20 bits represents the value of the label, next three bits are experimental and used for QoS. The next bit indicates the bottom of the stack (BoS). The label stack of a packet can consists of multiple different labels and only the bottom one has this bit set to 1. Bits 24 to 31 are used as Time To Live (TTL) field similar to TTL field in IP header.

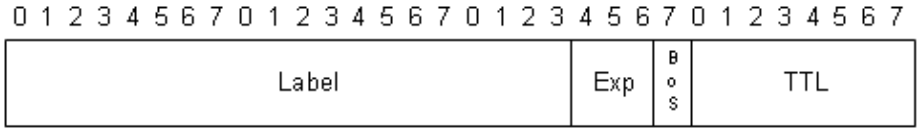

Figure 2.1 - MPLS label

2.2.2 Label stack

Some MPLS applications, such as MPLS VPN or AToM need more than one label to forward the packets through the network. In this case, more labels are grouped and create a label stack, which is attached to the packet. The first label in the stack is called the *top label* and the last one is called the *bottom label*. The bottom label has the BoS bit set to 1 [22].

The label stack is attached to the packet between the layer-2 header and layer-3 packet. Because of this placement of the label stack is MPLS often classified as layer-2,5 protocol. The location of the MPLS label in a layer-2 frame is shown in Figure 2.2.

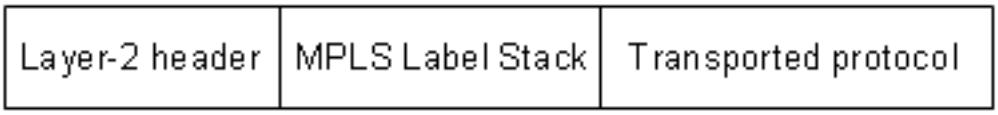

Figure 2.2 - The location of label

2.2.3 Label switched router

A router that supports MPLS is called a Label Switched Router (LSR). There are three types of LSR used in the MPLS network:

- Ingress LSR
- Egress LSR
- Intermediate LSR

The ingress and egress LSR are the *edge LSRs* because they are at the edge of the network. The ingress LSR receives a packet that is not labeled from a non-MPLS network. It inserts a label (or more labels) into the packet and sends it to the MPLS network. The intermediate LSRs receive a labeled packet, perform an operation on it, switch the packet and forward it to the next router. The egress LSR receives a label packet from the MPLS network, removes the labels and forwards it outside the MPLS network.

The LSR can do three operations with the packet: push the label onto the packet, swap the labels and pop the label from the packet. When an LSR pushes the label onto the packet that was not labeled yet, it is called an *imposing LSR*. This operation is provided by the ingress LSR. When the LSR is removing all labels from a labeled packet, it is called a *disposing LSR*. The disposition is done by egress LSR [3].

2.2.4 Label switched path

The sequence of LSRs that switch a labeled packet through the MPLS network is called a Label Switched Path (LSP). The first LSR of an LSP is the ingress LSR for that LSP and the last LSR is the egress LSR for that LSP. Important to mention is the fact, that LSP is unidirectional so for bidirectional communication two LSPs are needed [3].

The ingress LSR of the LSP does not have to be the first LSR to label the packet. In case of nested LSP can one LSP be inside another LSP. Then, packet in the nested LSP has to have the minimum of two labels in the label stack to identify both LSPs. An example of nested LSPs is shown in Figure 2.3.

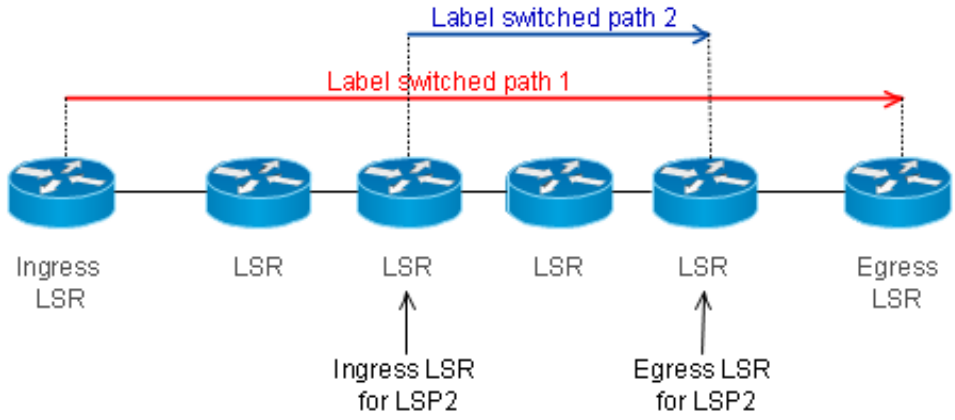

Figure 2.3 - Nested LSP

Packets that are forwarded along the same LSP create a Forwarding Equivalence Class (FEC). All packets in one FEC have the same label, but not all packets that have the same label have to belong to the same FEC. The classification of the packets into the FECs is done by the ingress LSR and the criteria for the classification can be for example [22]:

- Multicast packets belonging to one group
- Packets with layer-3 destination IP address matching a certain prefix
- Packets with same Precedence or DSCP field

2.2.5 Label distribution

For LSRs to successfully forward the packets through the network it is necessary for each LSR to know, which label to use for which packet. The labels have no global meaning and are local for each pair of neighboring LSRs. Therefore, a form of distributing the label information is needed. There are two ways to distribute this information: use of existing IP routing protocol or have a separate distribution protocol [3].

The first method has the advantage that no new protocol is needed and therefore also no synchronization. The possibilities of extending routing protocols OSPF and IS-IS were analyzed in Section *1.4.1 IGP extensions for TE.*

The use of separate protocol for distributing the label information has the advantage of being protocol independent. Two protocols can be used for this purpose: LDP or RSVP. The use of RSVP was described in Section *1.4.3 Signaling for TE tunnels*. With LDP, each LSR creates a local binding [label, IGP IP prefix] and distribute this binding to all of its neighbors. The neighbors then store this information in Label Information Base (LIB). The LSR chooses the accurate label based on the next-hop IP address in its IP table (called Routing Information Base - RIB). The distribution of labels across the network is shown in Figure 2.4.

Figure 2.5 shows the labels attached to a packet being forwarded through the network.

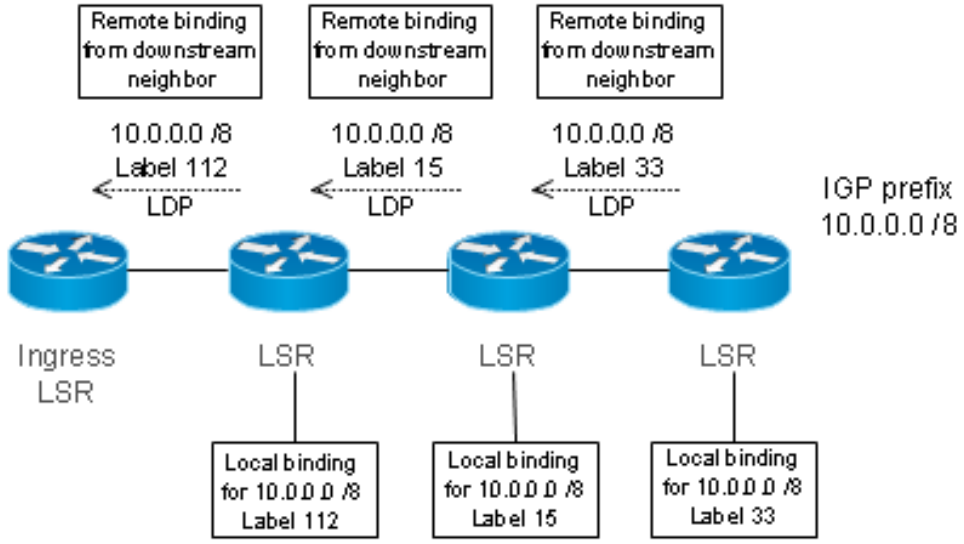

Figure 2.4 - LDP label distribution

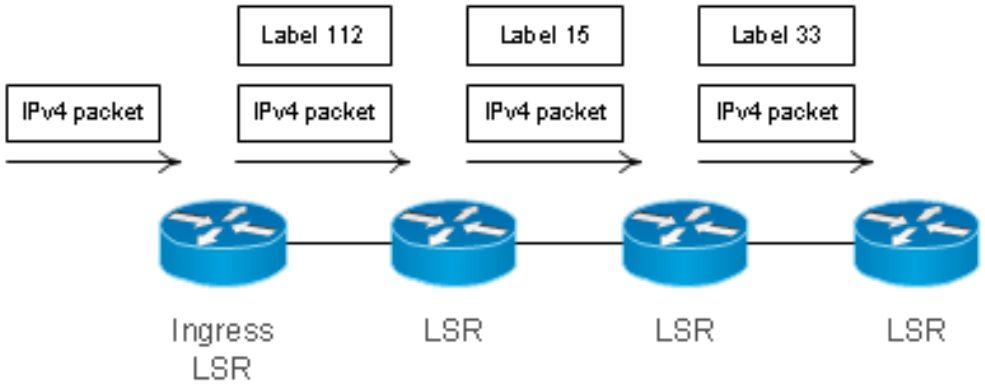

Figure 2.5 - Attached label

2.2.6 Cisco Express Forwarding

Cisco Express Forwarding (CEF) is a packet switching and forwarding method used in Cisco IOS. It was developed as a new and better switching method in routers, since with fast switching the switching cache was only built on demand. Therefore, the first packet of a flow had to be process-switched, which can be time consuming [23].

With CEF the switching table is not created on demand, but it is build in advance. Each prefix added to the routing table is also added into the CEF table. In MPLS, CEF is used to switch the IP packets, while labeled packets are switched according to the LFIB (label forwarding information base) on the router.

CEF consists of two main components: the Forwarding Information Base (FIB, also called a CEF table) and the adjacency table. The information from the adjacency table is used to rewrite the layer-2 header of a packet when it is being switched. The CEF table is responsible for the forwarding of the packet at layer-3. It is filled from the IP routing table and contains for each IP prefix its next hop IP address and outgoing interface. If needed, the next hop IP address is recursively learned from the IP table.

The operation of CEF consists of stripping off the layer-2 header, looking up the destination IP address in the CEF table (FIB), creating new layer-2 header and switching the packet onto the outgoing interface. A label is pushed into the label stack in case of IP-to-Label switching [3].

3 MPLS TE Algorithms

One of the main problems in providing QoS guarantees in MPLS networks is how to select paths for traffic flows to satisfy their QoS constraints. This issue is known as the QoS routing or constrained-based routing. To solve this problem number of QoS routing algorithms have been proposed and created.

The routing process in general consists of two main entities – the routing protocol and the routing algorithm. In this work, we will provide an overview of several routing algorithms used for QoS routing in MPLS network.

To provide scalable QoS guarantees DiffServ approach is being used. The DiffServ networks can support different service models such as Expedited Forwarding (EF), Assured Forwarding (AF) or Best Effort (BE). Integrating DiffServ with MPLS creates effective architecture with scalable edge-to-edge QoS and TE capabilities. DS-TE (DiffServ-aware MPLS Traffic Engineering) has been implemented as one of the MPLS TE methods. It can automatically adjust the LSP bandwidth and dynamically reroute the LSP when needed. The main drawback of this method is the need of high functionality on the routers, since each LSP needs to monitor the traffic and compute the required bandwidth, which leads to large load on the routers [25].

Therefore it is desirable to have the edge routers to configure the LSPs, determine the routes through the network and distribute the traffic optimally. The actual traffic in the network should also be taken into consideration within this process.

3.1 Routing algorithms

In this part of the work we will review the routing algorithms used, such as the Min-Hop Algorithm (MHA), Widest-Shortest Path algorithm (WSP), Shortest-Widest Path algorithm (SWA), Dynamic Online Routing Algorithm (DORA), Minimum Interference Routing Algorithm (MIRA) and Profile-Based Routing (PBR).
The advanced routing algorithms such as RATES, or other proposed algorithms are described in later sections.

3.1.1 Min-Hop Algorithm (MHA)

The MHA is based on Dijkstra's algorithm and routes a new connection along the path with the minimum number of links between the source and destination. It is very simple and computationally efficient algorithm. Since it does not take into account the current load on the links when computing the path, the best paths are used until congestion is reached. This approach leads to overutilizing some paths while the others are left unused, which creates unbalanced routing with congestion and bottlenecks.

3.1.2 Widest-Shortest Path algorithm (WSP)

The WSP is an improvements of the MHA, since it load-balances the traffic among number of feasible paths. WSP chooses the best path based on the minimum number of links and if there are more such links, it chooses the one with largest residual bandwidth. This approach helps lower the load on often used links but has similar disadvantages as MHA. The best paths are used until the congestion occurs before switching to less utilized links.

3.1.3 Shortest-Widest Path algorithm (SWP)

The SWP algorithm is very similar to WSP. The main difference between these two algorithms is that SWP chooses the best path first based on the maximum residual bandwidth and if there is more than one possible option, the path with smaller number of links is chosen.

3.1.4 Minimum Interference Routing Algorithm (MIRA)

The MIRA takes into account the location of ingress and egress routers which can be potential traffic source and destination pair. The key idea of MIRA is to make the routing decision effectively based on the interference, so a new connection will be routed over a path which has minimum interference with possible future flows. This interference level is used as a link weight to calculate the shortest path for a new demand.

MIRA keeps an updated list of the critical links and therefore can be considered as online routing algorithm. The critical links represent links, which usage reduces the opportunity to route other flows. Compared to algorithms described earlier, MIRA provides more sophisticated functions and results in less chosen the critical links.

However, it has main disadvantages [26]:

- MIRA takes into account all flows that can use a specific link without verifying if these flows actually use the link. This leads to suboptimal use of the network.
- The link weights are set in a static way and they are redistributed only if saturation of some links occurs.
- When choosing the path, MIRA does not take into account how this connection will affect the future request of the same ingress/egress pair.

3.1.5 Dynamic Online Routing Algorithm (DORA)

DORA represents a dynamic online routing algorithm for construction of bandwidth guaranteed paths in MPLS networks. It places the paths evenly across the network in order to allow the creation of future paths and to balance the traffic load in the network. During the computation of optimal path DORA avoids links that can be part of any other path or have not enough residual bandwidth. When computing the paths the algorithm assumes that request for paths arrive one by one and there is no a priori knowledge of these requests.

The operation is divided into two stages. In the first part DORA calculates the PPV (path potential value) array for each source-destination pair. This array represents information about each link between the source-destination pair which takes into account the possibility of using this particular link by other source-destination pair. The

algorithm considers only disjoint paths in the network. The actual computation is as follows [27]:

"When a path could be constructed over a link L for a given source–destination pair (S1, D1), we reduce $PPV_{(S1;D1)}(L)$ by 1. When a path could be constructed over the same link L for a different source–destination pair (S2, D2), we increment $PPV_{(S1;D1)}(L)$ by 1."

The second stage of DORA consists of removing all links which have less residual bandwidth than the required bandwidth. The link weights are then computed as the combination of PPV and residual bandwidth for each link. The combination of these two parameters is controlled by BWP (bandwidth proportion) which is set between values of 0.0 and 1.0. For example, BPW=0.7 implies that 70% of the link weight is affected by the residual bandwidth of the link and 30% is affected by the PPV value.

In the final stage, the Dijkstra's algorithm is run to compute a weight-optimized path from the source to the destination [27].

3.1.6 Profile-Based Routing (PBR)

The PBR algorithm uses the information about the ingress and egress routers in the network. It also takes into consideration the network traffic statistics by creating network traffic profiles. These profiles are used as a way of prediction of future traffic demands and distribution. PBR is based on an offline preprocessing step which determines the allocated bandwidth to each traffic class on each link in the network. This information is used for admission control provided on incoming connections. This approach effectively reduces the computation performed online upon a new connection request [28].

The performance of PBR is limited since the admission control can reject the incoming request even if there is a feasible path in the network.

3.2 Advanced routing algorithms employment

Since routers have limited memory and CPU power advanced routing algorithms are difficult to implement on them. Therefore these algorithms are usually implemented using a server which creates a centralized model. Such a server can obtain required information for computing the optimal paths from distributed protocols such as OSPF, IS-IS. To control the functions of the server or the set-up of LSPs it usually uses SNMP or telnet.

This part of work will focus on implementations used in MPLS environments. Possibility of using external server for traffic engineering is analyzed. Later different approaches to providing desired QoS in MPLS networks are described.

3.2.1 RATES

Routing and Traffic Engineering Server (RATES) is a software system developed for MPLS traffic engineering. Its implementation consists of a policy and flow database, a interface for the setting of policies and a COPS (Common Open Policy Service). COPS represents a client-server system created to enable the communication between the server and edge routers.

RATES uses information from OSPF protocol to dynamically obtain the link-state information in the network. RATES can set-up LSPs in the network with user-specific bandwidth guarantees based on this information. It uses its own "minimum-interference" routing algorithm to gain the optimal utilization of the network resources. This approach takes into consideration the possibility of new future requests which could arrive.

The main characteristics and design decisions are as follows [29]:

- Centralized approach – RATES is implemented in centralized manner, although the information used in its operations is obtained in a distributed way.
- Obtaining topology information – the server uses OSPF peering with one of the nodes in the network to get all the required information. It does not use SNMP mainly because in time of its development there were no SNMP MIBs for QoS attributes standardized as mentioned in [29]. RATES has also a graphical user interface, which can be used by network administrator to provide parameters such as bandwidth, preference or constraints. RATES keeps track of the

information about reserved and available bandwidth on the links since it is responsible for all bandwidth reservations.

- Route computation – can be triggered by a new incoming request from an ingress router to the server or by a network administrator through the graphical interface.
- Knowledge of ingress and egress points of LSPs – the path selection algorithm can use the knowledge of ingress and egress points in the network which are potential beginnings and ends of a LSP. Although LSPs can be created also between different nodes, this possibility is quite low. Therefore, the algorithm does not have to assume that each node in the network can be used as ingress or egress point.
- Re-routing performance – in case of link failure, it has to be possible to create an alternative route for the affected LSPs.
- Policies – are used for managing the use of created LSPs. It can be implemented in form of packet classifiers, which redirect the packets into the LSP tunnels bypassing the lookup in the routing table. Another way is to implement it directly in the routing table so the routing table will use the LSP tunnel as the next-hop. The administrator can specify these policies in the graphical user interface in RATES.
- Installing the LSP route – RATES provides the installation of the computed route by communicating only with the ingress router. The LSP is then signaled through the required path in the network. RATES uses the COPS (Common Open Policy Service) with added extensions to communicate with the routers.
- The database – RATES uses a relational database as its information base.
- Scale – the server operates in a single area within OSPF. The main reason is the summarization of information from other areas, which in case of traffic engineering information could not be the best option.
- LSP restoration – the paths in the network can be protected by pre-created backup paths or by re-routing in case of failure. When backup paths are used, they can have associated bandwidth reservation or can share the reservation with other paths.
- Network re-optimization – RATES supports the opportunity of manual re-routing of LSP even without any network failure. The network administrator can

use the “make before break” approach which set up a new path before removing the old one.

3.2.2 Multiple path selection algorithm

The authors in [30] proposed a new per-class bandwidth constrained algorithm for a DiffServ-aware traffic engineering called multipath selection algorithm (MSA). This algorithm consists of three steps: firstly, MSA is used to find number of LSPs from the source to the destination for specific class type (CT); in the second step the source allocates the initial traffic to the selected LSPs and in the last part the source adjust the traffic dynamically to the LSPs according to their round trip time (RTT).

In the first step the MSA uses two metrics to find the LSPs: the RTT of the path and the available bandwidth of each link. The algorithm always prefers the path with minimum number of hops, since it expects that the queuing delay on the router dominates the overall transmission delay. Therefore the path with lower number of hops is expected to have shorter RTT [30]. When selecting the LSP several principals have to be met:

- The LSP cannot contain a loop
- The source selects the path which has minimal number of hops and enough usable bandwidth. Therefore, each link can be used by multiple LSPs if it has the required bandwidth available.
- The algorithm does not distribute the link state database to all nodes, only the source node records the changes in available bandwidth of links.

The selected path is computed by combining the number of hops and available bandwidth among all possible combinations of paths in the network.

In the next step the source allocates the initial traffic to each selected LSP according to its available bandwidth. The allocation is proportional to individual maximum available bandwidth of each LSP. Then, the source measures the RTT of each LSP. This information is used in the last step to define a range of RTT which is compared to the pre-defined threshold. LSPs with high load of traffic (high RTT) must release part of the traffic to LSPs with lighter load which leads to gaining an average value of RTT by each LSP.

The proposed algorithm proved that the delay from the source to the destination is minimal when the RTT of each LSP is the same or similar. Another advantage is the

adjustment of the traffic on each LSP based on the RTT which decreases the overall delay from source to destination. Simulation results proved that the proposed algorithm gains better average delay, packet loss rate, throughput than those based on CSPF [30].

3.2.3 QoS Routing algorithm with delay and bandwidth constraints

The authors in [31] have proposed a new QoS routing algorithm which uses both bandwidth and delay constraints. The main idea is the computation of optimal path based on avoiding critical links, deleting links that do not satisfy the constraints and using shortest path algorithm to select the best path. The designing objectives of this algorithm are as follows:

- Minimize interference levels among ingress-egress nodes
- Load-balancing the traffic through underutilized paths
- Optimize the utilization of the network by using Dijkstra's algorithm

The algorithm uses the idea of "criticality" to select links with high possibility of future requests routed through them. This parameter is directly dependent on the total number of demands per link. Avoiding the critical links can help to reduce network congestion.

The next step of the algorithm is to compute the link weight which is directly proportional to criticality of the link and inversely proportional to the residual bandwidth of the link. Therefore the weight of the link is higher with higher criticality or lower residual bandwidth and vice versa. Links selected for each request are selected according to this link weight creating the weight for the path from source to destination.The algorithm uses MIRA to obtain the path with minimum path weight. Then, all paths which do not satisfy the constraints are removed and Dijkstra's algorithm is used to select the shortest path among the rest of paths.

The simulation results proved that this algorithm can lead to improved performance and provides better network utilization for bandwidth and delay guaranteed constrained applications. The proposed algorithm performs better for complex network in terms of call blocking ratio and CPU time.

3.2.4 Load Balancing Algorithm Using Deviation Path

The LBDP (Load balancing algorithm using deviation path) mentioned in [32] uses the spanning tree concept, deviation path and the idea of isoline. The isoline represents a

line on a network map connecting nodes with equal distance to the destination of certain flow. The deviation path represents a path in the network which shares number of nodes with original path but deviates at certain point and ends in the same destination as the original path. This concept is used to avoid critical or congested links in the network.

The proposed algorithm consists of these steps:

- For each new flow the algorithm creates its spanning-tree and the isolines
- In case of congestion threat on the link one of the flows using this link is selected to be the switch flow
- The algorithm searches for a possible backup path for the switch flow
- The selected flow is switched to balance the load

The algorithm uses periodic checking of the outcome links performed by each LSR. If the computed bandwidth utilization of certain link exceeds the pre-defined threshold, possibility of congestion occurs. In this case the algorithm determines which flow is most suitable to be switched to another path based on calculated minimum bandwidth that must be relocated.

In the next step a new path for the selected flow has to be found. For this process the isolines and deviation paths are used. Since the new path has to avoid the congested link, the deviation point has to be placed before this link. If the new path is found successfully the MPLS explicit routing technology is used to establish the LSP and map the flow onto it.

This algorithm has been tested in the environment of network simulator ns-2. The results of the simulation have proved obvious advantages of proposed algorithm compared to Shortest Path First algorithm (SPF) and Load Balance by Sideway Algorithm (LBAR). The LBDP algorithm proved better performance in terms of packet drop ratio after avoiding congestion, increased throughput of the network and network delay. Based on these simulation results LBDP is able to effectively balance the load in the network and can improve the network performance.

3.2.5 Flow distribution and flow splitting algorithm

The authors in [33] defined the flow distribution as selecting one of the available LSPs to carry one aggregated traffic flow. Flow splitting is defined as a mechanism designed for multiple parallel LSPs to share one single aggregated flow.

Incoming traffic requests to the ingress routers can arrive from different subnetworks, with different amount of traffic and with different QoS requirements. Therefore it is necessary to perform the flow aggregation. The aggregated traffic flows are created based on the CoS value in the MPLS label stack, source and destination address and ports.

The paper proposed three algorithms: the flow distribution algorithm (FD), flow splitting algorithm (FS) and the integration algorithm (IA). FD searches for the least utilized LSP. If this LSP cannot serve the flow, FD tries next least utilized LSP. FS is designed for multiple parallel LSPs to share the load of one aggregated traffic flow. The main idea is to optimally allocate the traffic among all LSPs so their load will be similar. IA is used to choose which option will be used – either flow distribution or flow splitting. The decision is based on the utilization or load of the network. In case the network is heavily loaded, the flow distribution is used, other whiles the flow splitting is used.

The proposed algorithms and ideas were studied through mathematical analysis and by simulations. The results proved that implementing this new concept of routing based flow shaping leads to avoiding network problems such as bottlenecks and mismatch problems [33].

3.3 Summary

The first part of this work covered the analysis of traffic engineering. Basic principals are defined together with the categorization of TE based on different approaches used. Each category is described briefly. The considerations needed when employing traffic engineering are analyzed as well as different possibilities of the cooperation between different TE categories. The possibility of using MPLS TE is described with its characteristics and supporting extensions of IGPs. Part of the work was also dedicated to the question of quality of service since it is the main reason of deploying the traffic engineering.

In the next part of the work the MPLS technology was analyzed in detail. Basic characteristics and principles of the MPLS architecture were described. The MPLS VPNs were analyzed in detail.

Last part of the analysis was dedicated to MPLS TE algorithms. Basic algorithms which are widely used were described. Advanced algorithms proposed by various authors were analyzed and characterized to provide brief overview of possible approaches developed in the area of MPLS traffic engineering.

4 Proposal

The service provider's networks are used to transmit a number of different types of traffic, such as data, voice or video. These types of traffic require various QoS to be provided. The transmission of data may be influenced by many negative factors, such as slow links and congestion which leads to increased packet loss, delay or jitter. Traffic engineering is used as a way to minimize this negative influence and maximize the network's utilization by balancing the traffic load distribution in the network.

Many different algorithms to provide the selection of desirable paths through the network were proposed and implemented, as described in the chapter 3 *MPLS TE Algorithms*. Each of these algorithms uses a different approach to analyze the network and to choose appropriate distribution of LSPs. However, most of these algorithms are not concerned about classification of traffic or providing different QoS for the traffic traversing the network. Therefore, the only requirement for the traffic trunks is the bandwidth which can be insufficient for special types of traffic such as voice.

From the view of the end user (customer) the most important issue is the provided quality of different services. Since the customer usually uses variety of different applications, it is the responsibility of service provider to take care of the provided QoS in the network.

In this work we propose a system to provide quality of service for different classes of traffic in the network. Based on created LSPs this system will provide efficient utilization of network resources together with optimal traffic distribution.

The main goal is to use the LSPs to transmit the classified traffic with regard to required QoS for each class. This process will include periodic measurements of different quality parameters in the network, such as delay, jitter or packet loss. The most optimal LSPs will be selected based on these parameters for each traffic class. Load sharing of traffic among number of LSPs will be used to maximize the utilization of available resources. Optimization and reoptimization will also be used in order to ensure sufficient QoS for each traffic class. In case of congestion and exhaustion of network resources, the less important classes of traffic will be limited to allow the resources to be used by the traffic class with higher priority.

Details about this process and steps to achieve the final goal are described in following chapters together with the topologies and measurements used to verify it.

4.1 System requirements

The proposed system has to meet these requirements to provide an efficient, manageable and scalable tool to be used in large networks of service provider:

- Effective way of measuring specified network parameters
- Reasonable computation of the cost of LSP
- Reasonable choice of LSP for different classes of traffic
- Effective use of load balancing among multiple LSPs
- Protection against congestion and degradation of provided QoS
- Not computationally intensive process of optimization

The whole system should be easy to use with Cisco routers since it has to use some of the information from the router. Also, it has to be able to communicate with the router as it will actively affect the decisions concerning the distribution of the traffic. We focus on Cisco devices for a simple reason: the experimental evaluation of the proposed server in laboratory environment is needed. Since the available laboratory equipment consists mainly of Cisco devices we cannot implement our solution with other devices.

The successful operation of the proposed system has these prerequisites:

- Working MPLS network – all required configuration concerning MPLS has to be applied in advance for the system to work properly
- Telnet access and encrypted password (enable secret) configured on PE routers
- SSH server configured on one PE router to provide the connection for the server
- Configuration of LSPs – LSP for each class of traffic has to be created in advance
- Definition of QoS – QoS requirements for each traffic class have to be defined in advance to be used as a parameter in the proposed server

4.2 Proposed solution

The main goal of the proposed solution is to efficiently distribute the traffic across the network with respect to specific QoS requirements:

- Resource demands of traffic flows within the guaranteed bandwidth are satisfied
- QoS requirements of specific traffic classes are satisfied
- None of the links in the network is congested
- LSPs in the network are evenly utilized

The proposed system will not cover the possible suboptimal choice of paths for LSP. Also, it will not deal with creation of backup paths or optimization due to link failure. Any of these issues should be handled in the process of creation of LSPs.

Several steps have to be done to reach the optimal state using the proposed system:

1. Analyze the network and existing LSPs
2. Measure end-to-end quality parameters of LSPs
3. Calculate the cost of LSPs
4. Assign the traffic classes to LSPs
5. Optimize the assignment (if necessary)

The flow diagram in Figure 4.1 represents the arrangement of these steps in our solution. The next sections analyze this process in detail.

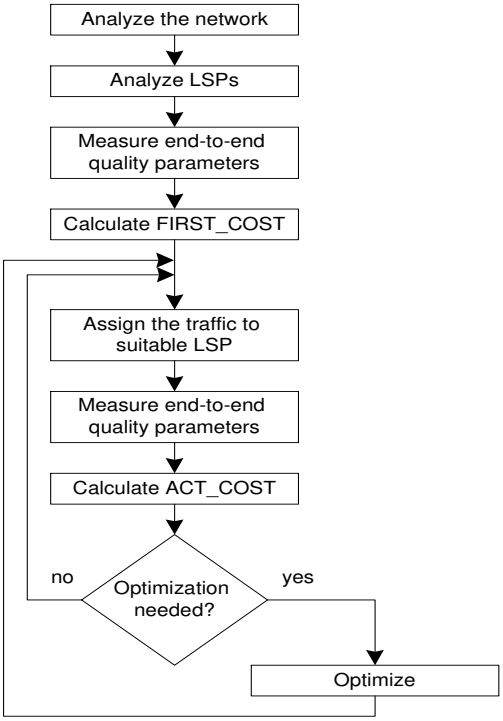

Figure 4.1 - The main system's processes

4.2.1 Main contribution of this work

The proposed system will take advantage of some already proposed and designed algorithms. For the first step – to create and apply the LSPs in the network – simulation of an existing algorithm will be used. The LSPs will be created manually, although the implementation of an algorithm can be considered as a future development of this work.

The main contribution lies in the proposed algorithm for efficient distribution of the traffic across the network with respect to various QoS requirements. We propose our own method of calculating cost of LSPs based on different quality parameters measured in real-time. We also propose the concept of optimizing of assigned traffic flows to achieve the best performance with provided QoS.

The implementation of the proposed algorithms in a form of online traffic-engineering server is also considered as an important contribution of this work. With the use of proposed network topologies we will be able to validate the functionality and effectiveness of our proposed system.

4.2.2 Classification of traffic

The traffic entering the network will be classified into four classes defined by the requirements it has. One class (Class1) will be dedicated for real-time traffic with strict default requirements to achieve sufficient QoS. All other classes will be used for non-real-time data traffic. The parameters for data classes will be set by the administrator in the system's options. The classes will be used as follows:

- Class1 – requires maximum delay of 150 ms, maximum jitter of 30 ms and maximum packet loss of 1,5%. This class will be dedicated to real-time applications, such as VoIP and will have the highest priority among all classes.
- Class2 – high priority class for data
- Class3 – medium priority class for data
- Class4 – low priority class for data

The traffic will be treated according to the class it belongs to in descending order. That means traffic of class with higher priority will be preferred and served prior to any other traffic. Every traffic class will be managed in terms of maximum bandwidth demands to avoid a critical situation, when traffic from one class would use all the resources in the

network. Each traffic class will have defined the amount of overall bandwidth it can use in the network to avoid the traffic-class starvation.

The traffic entering the network has to be classified to achieve proper treatment as mentioned earlier. The classification into four classes will be provided by the CE router at the customer network using IP Precedence bits. The mapping between IP Precedence values and MPLS EXP bits will be done at the PE router in order to provide QoS in the MPLS network. The EXP values of 7, 6, 5 and 4 will be used to mark the traffic based on its class within the guaranties. The EXP values from 3 to 0 will be used to define the traffic based on its class above the guaranties. The example of such mapping is shown in Table 5.1 although this mapping will be done dynamically and could change.

Traffic class	IP Precedence	EXP
Class1 – guaranteed traffic	1	7
Class2 – guaranteed traffic	2	6
Class3 – guaranteed traffic	3	5
Class4 – guaranteed traffic	4	4
Class1 – extra traffic	1	3
Class2 – extra traffic	2	2
Class3 – extra traffic	3	1
Class4 – extra traffic	4	0

Table 4.1 - The mapping between IP Precedence and EXP values

4.2.3 Creation of LSPs

Our solution requires the creation of LSPs in the network to be done in advance. In our work these LSPs will be created manually before the start of the server. The manual configuration of LSPs will simulate the work of a TE algorithm analyzed in chapter *3 MPLS TE algorithms.*

The LSPs will be created one per traffic class which means that four LSPs will exist per one customer. The bandwidth of these LSPs should be configured according to the agreement between the customer and service provider. The bandwidth of LSPs will be configurable by the administrator of the server to simulate this situation.

The created LSPs will not be changed during the operation of proposed system. The traffic however, will be assigned to the most suitable LSP at the time. Therefore the traffic load of the LSPs may change as may the traffic class using the LSP.

4.2.4 Measurements of quality parameters

When the LSPs are created, it is important to periodically measure various network performance parameters, such as end-to-end delay, jitter or packet loss. These values will be used in the next step to calculate the cost of each LSP. The cost will be then used to choose the best path for each traffic trunk, depending on its QoS requirements.

There are two versions of calculated cost of LSPs. First cost is calculated after the creation of LSP before any traffic uses it. This value (called FIRST_COST) is used in the optimization process as the "last hope" – in case high priority traffic has not been assigned to any LSP and the LSP with best FIRST_COST is used (with eliminating all traffic using the LSP earlier). The second cost value is the actual cost updated by each measurement during normal operation of the system.

The measurements will be carried out using IP SLA probes on the edge routers. The measured values will be stored in the database for further usage.

Since the values of delay, jitter and packet loss are variable in time, it is preferable to work with their statistical values instead of actual values. We propose computation shown in Formula 1, Formula 2 and Formula 3 to provide trustworthy values of these parameters to be used. The variables $delay_t$, $jitter_t$ and $loss_t$ represent the actual values of delay, jitter and packet loss respectively. Each value is calculated using basic statistical approach of finding the mean value among last three measured values (in times t-3, t-2 and t-1).

$$delay_t = \frac{delay_{t-3} + delay_{t-2} + delay_{t-1}}{3} \quad (1)$$

where: $delay_t$ represents the actual value of delay is ms

$delay_{t-3}$, $delay_{t-2}$, $delay_{t-1}$ represent values of last 3 measurements of delay in ms

$$jitter_t = \frac{jitter_{t-3} + jitter_{t-2} + jitter_{t-1}}{3} \quad (2)$$

where: $jitter_t$ represents the actual value of jitter is ms

$jitter_{t-3}$, $jitter_{t-2}$, $jitter_{t-1}$ represent values of last 3 measurements of jitter in ms

$$loss_t = \frac{loss_{t-3} + loss_{t-2} + loss_{t-1}}{3} \quad (3)$$

where: $loss_t$ represents the actual value of packet loss in %

$loss_{t-3}$, $loss_{t-2}$, $loss_{t-1}$ represent values of last 3 measurements of packet loss in %

4.2.5 Calculating the cost of LSP

The cost of LSP is used to decide whether it is suitable for specific traffic class or not. Since there are different types of traffic with different requirements, the cost has to reflect the parameters for every traffic class. The main difference is between Class1 and other classes since the first class has specific demands on values of delay, jitter and packet loss along with the bandwidth demand. There is no possibility of including network performance parameters of the link into the cost used for path computation at the time of writing this work. Although there is an effort to develop extensions for including the network performance criteria into OSPF, it is not usable at the moment [34]. Due to this fact we decided to use two cost values for each LSP – one as characteristic of network performance parameters and one to describe the bandwidth usage of LSP. Formula 4 shows the basic mathematical representation of cost value for Class1 and Formula 5 shows the representation of cost value for classes Class2, Clas 3 and Class4. The variables of C_{voice} and C_{data} are considered to be non dimensional.

$$C_{voice} = C_{delay} + C_{jitter} + C_{loss} \quad (4)$$

where: C_{voice} represents the actual value of the cost of LSP for Class1 traffic
C_{delay} represents the actual value of the cost of LSP according to the actual delay
C_{jitter} represents the actual value of the cost of LSP according to the actual jitter
C_{loss} represents the actual value of the cost of LSP according to the actual loss

$$C_{data} = \text{free\_bw\_of\_LSP} \quad (5)$$

where: C_{data} represents the actual value of the cost of LSP for data traffic
free_bw_of_LSP represents the actual value of the unused bandwidth of LSP

The C-values for delay, jitter and packet loss will be obtained from a reference tables shown in Table 5.2, Table 5.3 and Table 5.4. It is crucial to have all three parameters (delay, jitter, packet loss) in a specific range to be able to guarantee specific QoS. The proposed reference tables are created in such a way, that even one parameter out of range changes the LSP's cost significantly. No other information is then needed to select the suitable LSP.

The final ranges of the cost values for Class1 traffic are defined in Table 5.5. The cost in range from 0 to 3 represents the optimal conditions for real-time traffic. The cost in range from 3.1 to 12 represents that the conditions on the specific LSP are still within a suitable range according to [35]. Values of cost above 12.1 mean that the

quality parameters of LSP are not sufficient to provide required QoS with values above 40 representing absolutely unusable LSP for real-time traffic.

LSP which is used by Class1 and Class3 with the values of quality parameters:

- Delay = 20ms
- Jitter = 10ms
- Packet loss = 0,2%
- Free bandwidth = 120Mbps

will have cost values C_{voice} = 0,60 and C_{data} = 120.

DELAY [ms]	0	5	10	15	20	25	30	35	40	45	50
C_DELAY	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
DELAY [ms]	55	60	65	70	75	80	85	90	95	100	105
C_DELAY	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00	3.10
DELAY [ms]	110	115	120	125	130	135	140	145	150	160	170
C_DELAY	3.15	3.20	3.25	3.30	3.35	3.50	3.65	3.75	4.00	12.10	12.15
DELAY [ms]	180	190	200	210	220	230	240	250	260	270	more
C_DELAY	12.20	12.30	12.40	12.50	12.60	12.70	12.80	13.00	40.00	40.00	40.00

Table 4.2 - Reference table for delay values

JITTER [ms]	0	1	2	3	4	5	6	7	8	9	10
C_JITTER	0.00	0.02	0.04	0.06	0.09	0.12	0.15	0.18	0.22	0.26	0.30
JITTER [ms]	11	12	13	14	15	16	17	18	19	20	21
C_JITTER	0.35	0.40	0.45	0.50	0.60	0.65	0.70	0.80	0.90	1.00	3.10
JITTER [ms]	22	23	24	25	26	27	28	29	30	31	32
C_JITTER	3.15	3.20	3.25	3.30	3.40	3.50	3.65	3.80	4.00	12.10	12.15
JITTER [ms]	33	34	35	36	37	38	39	40	41	42	more
C_JITTER	12.20	12.30	12.40	12.50	12.60	12.70	12.80	13.00	40.00	40.00	40.00

Table 4.3 - Reference table for jitter values

LOSS [%]	0.00	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
C_LOSS	0.00	0.02	0.04	0.06	0.10	0.14	0.18	0.22	0.25	0.28	0.31
LOSS [%]	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	1.00	1.05
C_LOSS	0.34	0.38	0.43	0.50	0.55	0.60	0.65	0.75	0.85	1.00	3.10
LOSS [%]	1.10	1.15	1.20	1.25	1.30	1.35	1.40	1.45	1.50	1.60	1.70
C_LOSS	3.15	3.20	3.25	3.30	3.35	3.40	3.55	3.75	4.00	12.10	12.15
LOSS [%]	1.80	1.90	2.00	2.10	2.20	2.30	2.40	2.50	2.60	2.70	more
C_LOSS	12.20	12.30	12.40	12.50	12.60	12.70	12.80	13.00	40.00	40.00	40.00

Table 4.4 - Reference table for packet loss values

Conditions	OPTIMAL	GOOD	BAD	UNUSABLE
Cvoice	0 - 3	3.1 - 12	12.1 - 39	40 and more

Table 4.5 - The meaning of cost values

4.2.6 Assigning traffic trunks to LSPs

The incoming traffic flows have to be served according to their traffic class. It means that if more traffic flows arrive in one time, they will be assigned to LSPs based on their priority. It is important to emphasize that different approach is used for voice and data traffic. The flow diagrams describing the process of assigning data and voice traffic flows are shown in Figure 4.2 and Figure 4.3 respectively.

Every traffic class has defined the maximum guaranteed bandwidth in the network. With the use of optimal distribution of traffic in the network however, more traffic can be served and use the network resources. In this case it is crucial to ensure that all traffic within the guaranteed bandwidth is treated in preference of the traffic beyond the guaranties. In other words, traffic from Class4 which is within the guaranteed bandwidth is of higher importance then traffic from Class2 which is above the guaranteed bandwidth.

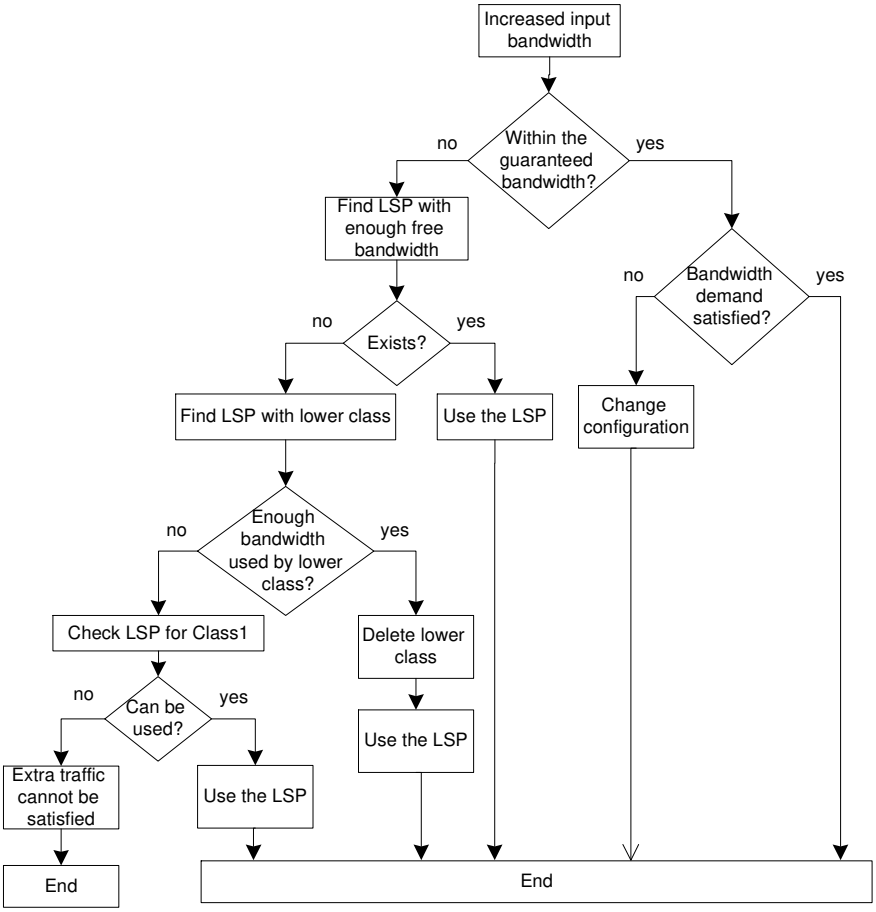

Figure 4.2 - Assigning data to LSP

We use the terms “guaranteed traffic” to describe the traffic within the guaranteed bandwidth for each class and “extra traffic” to describe the traffic beyond these

guarantees. For data traffic classes the cost for data of LSP is used and the assignment process goes as follows:

- If the traffic is within the guaranteed bandwidth, there are enough resources in the network to satisfy its demands. If no LSP has enough free bandwidth for this traffic it means that extra traffic from another class is using it. This extra traffic can be therefore deleted to release the resources for the guaranteed traffic.
- If the traffic is beyond the guarantees (extra traffic) it is still possible to be assigned to one of the LSPs. This possibility results from the fact that the guaranteed traffic does not consume all the network resources. If some LSP has enough unused bandwidth this LSP can be used for the extra traffic.
- If extra traffic cannot be assigned to any LSP due to lack of resources its traffic class can prioritize it among other extra traffic already using the network resources. In practice this means that if extra traffic of Class2 (ET2) cannot be assigned to any LSP and extra traffic of Class3 (ET3) is assigned, ET3 will be deleted to provide resources for ET2. The prioritization of traffic classes is then further extended beyond the guarantees since it is applied for extra traffic too.
- The extra traffic can use the LSP for Class1 if no LSP has enough free bandwidth and not enough extra traffic of lower class is assigned in the network.
- If the data traffic – guaranteed or extra – is for some reason to be assigned to LSP which carries the voice traffic special care has to be taken. Since voice traffic has specific QoS requirements the data traffic has to be assigned carefully to not degrade the actual quality parameters of the LSP.

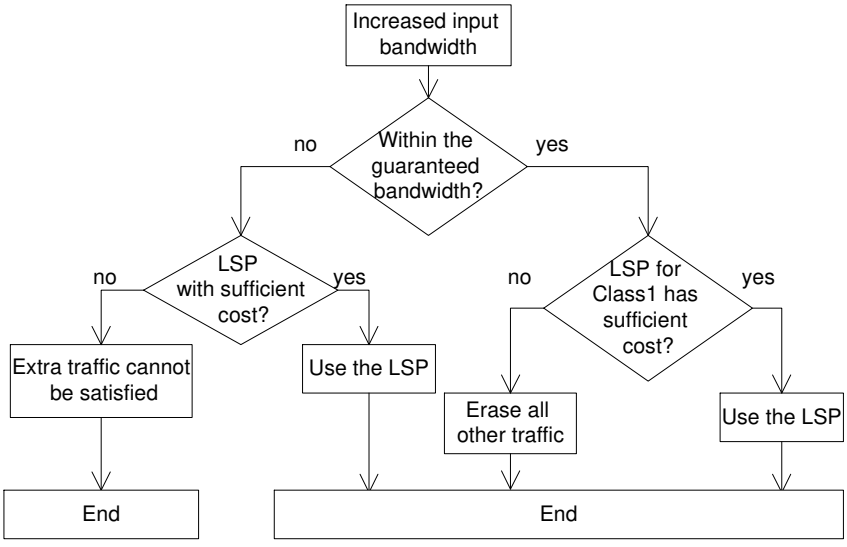

Figure 4.3 - Assigning Class1 to LSP

When assigning Class1 traffic (voice) the cost values for voice are used to choose the best LSP. The process of assigning traffic to LSPs is similar as for data traffic with the only change: the voice traffic requires specific QoS parameters to be satisfied besides the bandwidth requirements. All the other steps for assigning the guaranteed and extra traffic are identical with the process for data traffic:

- If the traffic is within the guarantees, there are enough resources in the network to satisfy its demands. If no LSP has enough free bandwidth and sufficient QoS parameters for this traffic it means that traffic from another class (data traffic) is using it. This data traffic has to be therefore deleted to release the resources for the guaranteed voice traffic.
- If the traffic is beyond the guarantees (extra traffic) it is still possible to be assigned to one of the LSPs although it can be problematic since it requires QoS. If an LSP is found with sufficient QoS this LSP is used for extra traffic of Class1. If no LSP has sufficient parameters the extra traffic of Class1 cannot be served.
- If the extra voice traffic cannot be assigned to any LSP due to lack of resources it can be prioritized among other extra traffic already using the network resources similarly to extra data traffic.

Using this process the optimal distribution of traffic is achieved in the network. The maximum bandwidth settings ensure that problem with traffic-class starvation will not occur. The possibility to assign traffic above the maximum guarantees increases the utilization of the network. The prioritization among traffic classes ensures that the traffic with higher priority will get better treatment before the traffic with lower priority.

4.2.7 Optimization of traffic flows

The process of optimization may be considered as the most important part of the whole system. Its purpose is simple – to achieve efficient distribution of traffic across all LSPs with preserved QoS. It will be triggered when the LSPs are unevenly utilized.

If extra traffic is assigned to a LSP its data cost (free bandwidth) is decreased. In some situation this can lead to overutilizing one LSP while others are underutilized. The traffic of lower priority classes may be deleted and later assigned to another LSP to achieve more effective traffic distribution. This process will be triggered by the results of periodic measurements of all LSPs in the network.

The flow diagram of the whole process of optimization in the first case is described in Figure 4.4. The diagram uses number of expressions which will be described in detail in the next section, such as:

- Critical unused bandwidth – state of LSP if its percentual value of unused bandwidth is lower than a half of average unused bandwidth of all LSPs. This state is represented by mathematical expression shown in Formula 6.

$$unused\ bandwidth[\%] < \frac{average\ unused\ bandwidth[\%]}{2} \tag{6}$$

- Splitting of guaranteed traffic – if a class is using two LSPs to take traffic which is within the guarantees

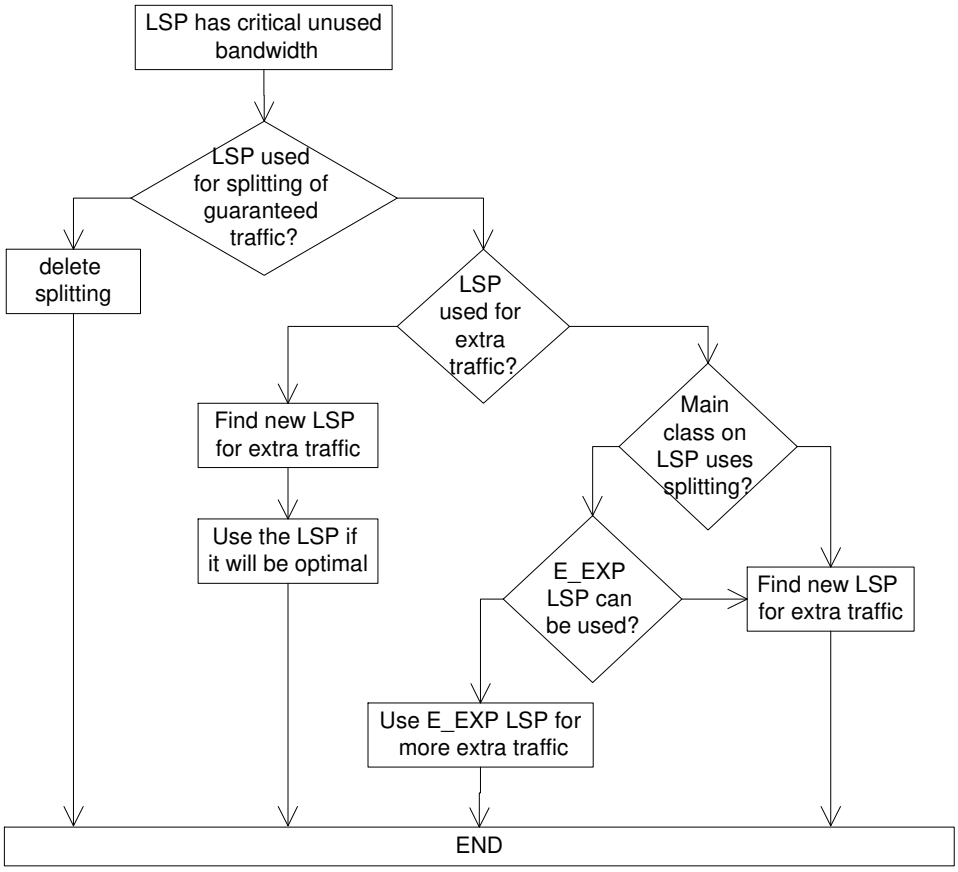

Figure 4.4 - The process of optimization

4.3 Proposed implementation

The whole system will be implemented as a server application running on a host connected to one of the PE routers in the network. The functions it will cover can be divided into two groups:

- One-time actions:
    - analyze the topology
    - gather information about LSPs
    - compute the maximum bandwidth of each LSP
    - set the maximum bandwidth for each class
- Periodic actions:
    - measure and update end-to-end quality parameters on each LSP
    - calculate and update the cost values of each LSP
    - choose and apply the traffic flows on an LSP
    - measure and update actual amount of used bandwidth per LSP per class
    - optimize the traffic flows if necessary

The one-time actions will be used in the very beginning of the server's process. These functions provide information about the network and its nodes, and then use this information to run the algorithm for LSP calculations and apply these LSPs in the network. The maximum bandwidth for each traffic class will be set by the administrator.

The periodic actions will provide the effective distribution of traffic through the network by measuring end-to-end quality parameters, calculating the cost values and applying the traffic flows on chosen LSP. Optimization will be used to provide QoS for each traffic class.

4.3.1 Communication

The crucial part of the server's performance is the ability to communicate with the PE routers. This communication has to provide a method for gathering information from the routers and also a method for applying new configurations to the routers. An important aspect of this communication is its security. Two possibilities exist to provide such a secure connection between the server and router: SNMPv3 and SSH. The scheme of such a connection is shown in Figure 4.5.

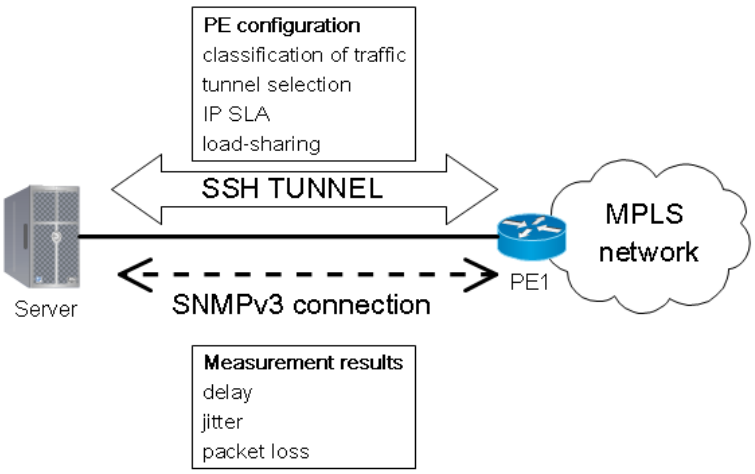

Figure 4.5 - The scheme of communication

SNMPv3 can be used to gather required information from the router such as quality parameters of LSP. It supports authentication, privacy and access control which provide the required security. To make the connection most secure the security level "AuthPriv" will be used. With this security level all messages will be authenticated and encrypted. The configuration on both sides (server and router) is required to enable the secure communication. The SNMP server has to be configured on the PE routers with specified username, authentication method and encryption algorithm. An access list should be used to provide the access control and permit only the connection from the server. The scheme of such a connection is shown in Figure 4.6.

However, it cannot be used to gather some specific information due to the permissions of some SNMP objects set to non-accessible.

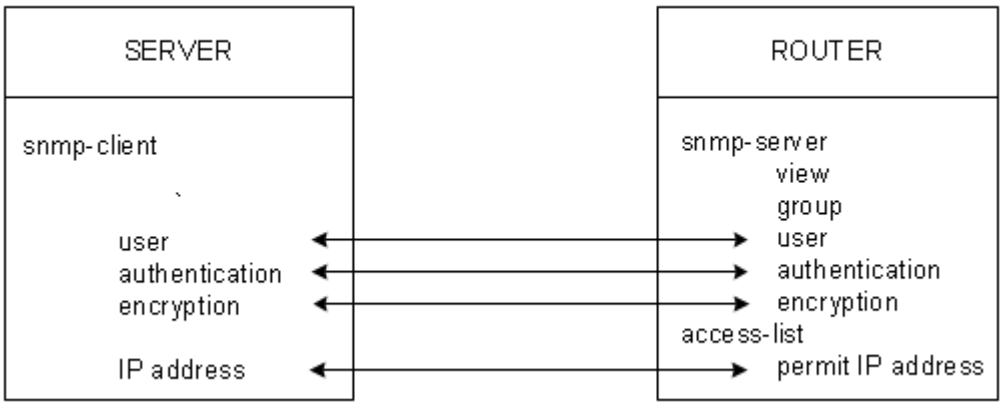

Figure 4.6 - The scheme of SNMPv3 connection

SSH connection can be used to get direct access to the CLI of the PE routers by creating a SSH tunnel between the endpoints. It provides user authentication and encryption to secure the connection. The scheme of such a connection is shown in Figure 4.7. With SSH the server would have access to the whole router's configuration which will be used to gather all necessary information and also change or update the router's configuration.

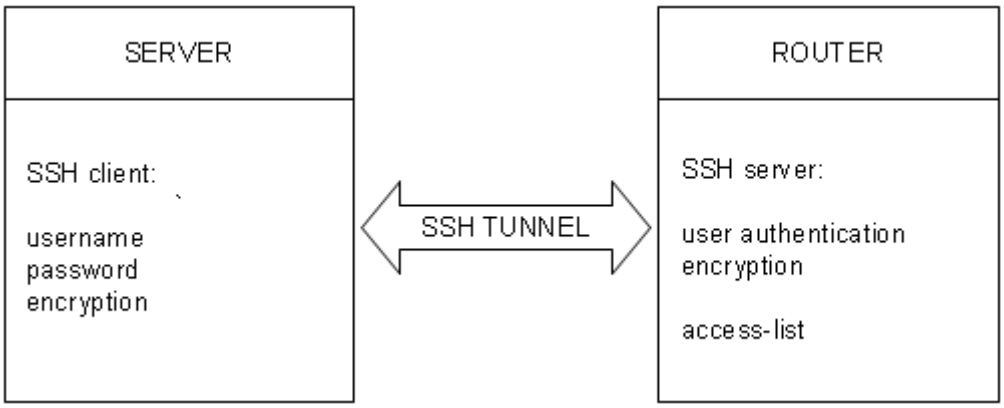

Figure 4.7 – The scheme of SSH connection

Additional configuration on the PE routers is required to allow the SSH connection similarly to SNMPv3. The PE router has to be configured as an SSH server to perform the user authentication. The terminal-line access has to be configured to allow the SSH connection. The security can be enhanced by applying an access list to the terminal-line configuration to allow only connections from the server. Since there will be only one connection to each PE router no separate authentication server has to be used. The authentication will be performed by configuring the user name and password locally on the PE routers.

4.3.2 The measurements

The knowledge of actual values of quality parameters (delay, jitter and packet loss) for each LSP is crucial for the server's performance. The measured values have to be accurate and up-to-date according to the actual situation in the network since they are used for the cost calculation for Class1 for each LSP. The quality parameters on LSPs change dynamically in time. These changes occur randomly and can cause a difference between the measured values and actual values on the LSP. Therefore the measurements have to be performed and evaluated in a very short time to provide accurate result. As

mentioned earlier the server calculates with the average values to mitigate the potential difference in case of sudden variance of parameter's value.

The tool used for measuring required quality parameters on the LSPs will be IP SLA. It supports many different operations for measuring different aspects of quality. The server requires information about one-way delay, jitter and packet loss for each LSP. To gather this information the UDP Jitter operation can be used. This operation generates packets with time stamps which can be used to monitor:

- per-direction jitter
- per-direction packet loss
- per-direction delay
- round-trip delay

The UDP Jitter operation supports the configuration of different parameters such as number of generated packets, payload size per packet, time between packets (in ms) or frequency of the operation (in seconds) [35].

The monitoring requires additional configuration on all endpoints (PE routers). This configuration has to be maintained by the server using the SSH connections. Since the operation is unidirectional it requires the configuration of both source and destination devices. The configuration on the destination device requires only the enabling of IP SLAs Responder functionality. The source device has to have all the parameters for measurement configured such as IP SLA operation number, type of IP SLA operation, destination IP address and port, source IP address and port, number of packets sent, the inter-packet interval and the frequency of the operation. Additional settings such as the threshold and timeout have to be configured as well to provide accurate results. The IP SLA operation has to be scheduled to start the measurements by setting the start time and time of execution.

The scheme of all configurations is shown in Figure 4.8. The important aspect of this configuration is the fact that IP SLA operation has to be configured for each LSP between the pair of source and destination PE routers. This means four active measurements between the pair of PE routers – one for each LSP.

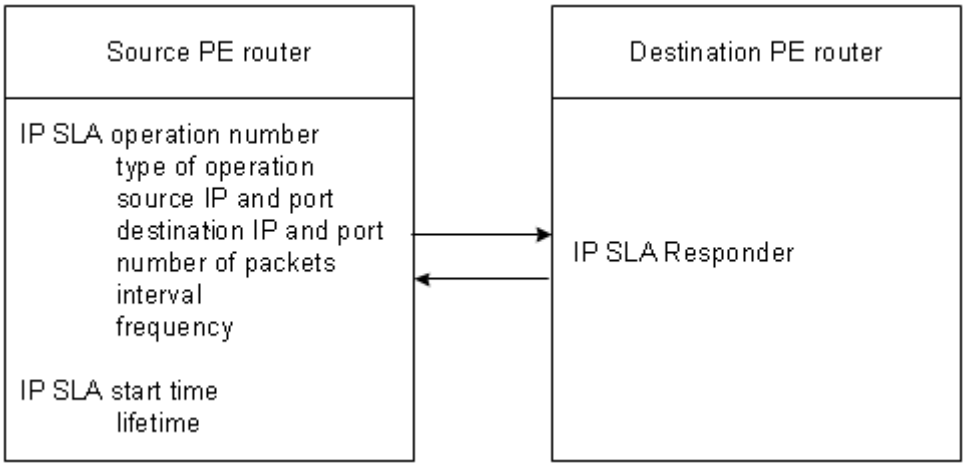

Figure 4.8 - IP SLA configuration scheme

To gather all the measured values SNMPv3 will be used as mentioned earlier. For the periodic measurements standard SNMP MIB will be used to get the data from the PE router. SNMP traps will be used to signal significant change of QoS parameters which results in degradation of LSP's quality. If any QoS parameter exceeds the specified threshold a SNMP message will be sent to the server which will take a proper action as described earlier. Using this mechanism the response time to the degradation of QoS can be minimized since the server will be informed immediately.

4.3.3 The generation of traffic

The traffic traversing the network will be generated by a software traffic generator. This approach will ensure the control over the experiments since the experiments will take place in laboratory environment. A traffic generator (such as Iperf) will provide the creation of exact traffic patterns used in our experiments together with measurements of bandwidth and quality of used LSPs. Multiple instances of the traffic generator will be created to simulate real network load with traffic flows of all classes.

Additional advantage of using a software traffic generator is its analysis of the network which could contribute to the precision and correctness of other evaluation tools used within the experiments.

4.4 Topologies and experiments

The functionality of the proposed server can be verified by its implementation into the real topology. For this purpose we proposed a topology consisting of seven routers and the layout of LSPs as is shown in Figure 4.9. This topology will be implemented in laboratory environment using Cisco devices. The networks LAN1 and LAN2 will be used for the traffic generation.

The proposed LSPs will be configured in advance since they are a prerequisite for the server's functioning. The bandwidth requirements for each LSP will be set according to the capacity of links and will not change. The number of LSPs is set to four since our design classifies the traffic into four classes with different priorities. Each LSP is therefore created primary for one class of traffic. During the server's operation however, different traffic classes can share resources of one LSP as was explained earlier. The exact values of bandwidth requirements per LSP together with its association with the traffic class are shown in Table 4.6. The bandwidth values of each LSP represent also the maximum guaranteed bandwidth per given class.

LSP	**Bandwidth**	**Traffic class**
LSP1	800 kbps	Class1
LSP2	1200 kbps	Class2
LSP3	500 kbps	Class3
LSP4	500 kbps	Class4

Table 4.6 - Bandwidth of LSPs

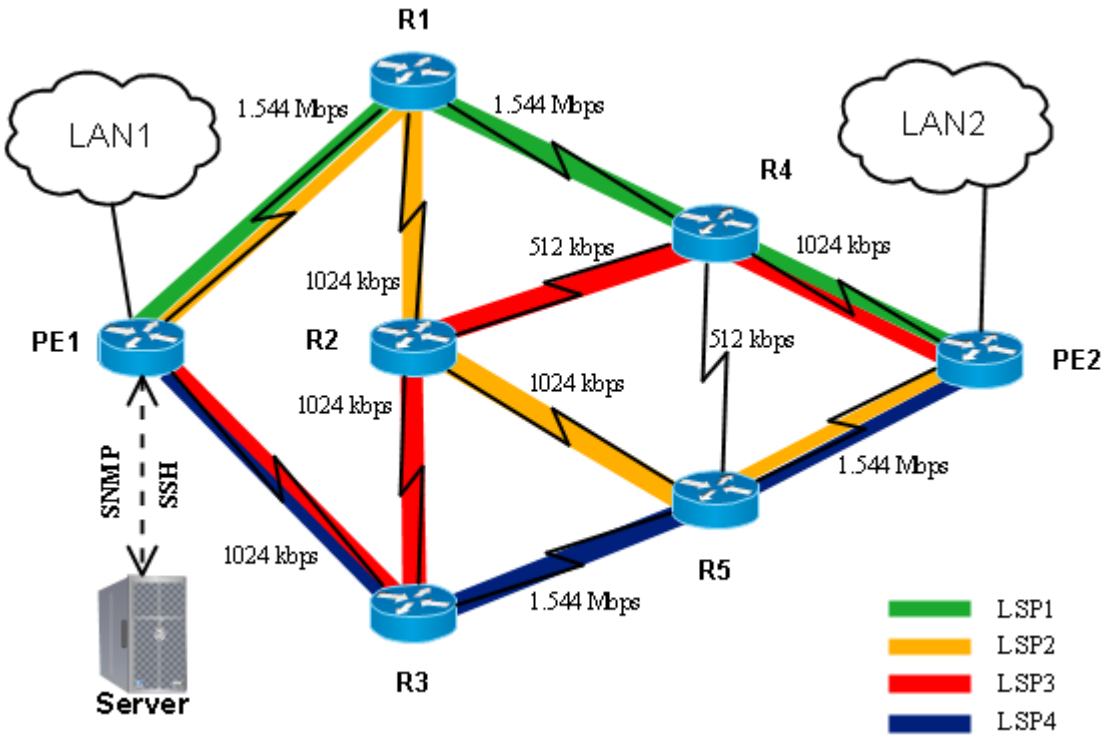

Figure 4.9 - Implemented topology 1

The server will be connected to router PE1 by SSH and SNMPv3 clients as described earlier in this work. Through this connection the server will gather information about the LSPs, configure the IP SLA probes on both PE1 and PE2 routers and provide the measurements.

The experiments will contain set of traffic flow demands which will be created pseudo-randomly. All traffic flows will be set with source in LAN1 network and destination in LAN2 network with the use of a traffic generator as described in chapter *5.3.4 The generation of traffic*. The server will process these demands and act accordingly to the actual situation in the network. The main goal is to provide optimal network utilization with satisfied QoS demands of Class1.

The evaluation of experiments will consist of measurements of QoS parameters, utilization of LSPs, overall throughput of data per class and average time needed for optimization. The results will be processed into graphical form for easy evaluation. The results will be also compared to the network performance without implemented server in the same environment and conditions.

4.5 Software components

The server has to be able to accomplish multiple tasks simultaneously as mentioned earlier. It has to maintain and process different data, perform computations, keep track of the current situation in the topology, perform measurements and apply suitable decision logic to take actions. To achieve effectiveness, optimal performance and scalability of our solution it is reasonable to use a modular scheme as an implementation method. Using this approach it will be very simple to provide an update to any part of the server or to change one whole module for another.

The server's architecture will therefore be divided according to specific functions of each module. As shown in Figure 4.11 the server's implementation will consist of following modules:

- Server daemon – the main part of server
- Network analyzer – module for analyzing the connected network
- Traffic handler – module for handling all traffic trunks together with their requirements
- Measurement engine – module providing the measurements of quality parameters per LSP using IP SLA
- Calculator – module for cost calculations per LSP
- Database – central storage of all necessary information
- SSH client – module providing the connection to the PE router
- SNMP client – module providing the gathering of measured parameters

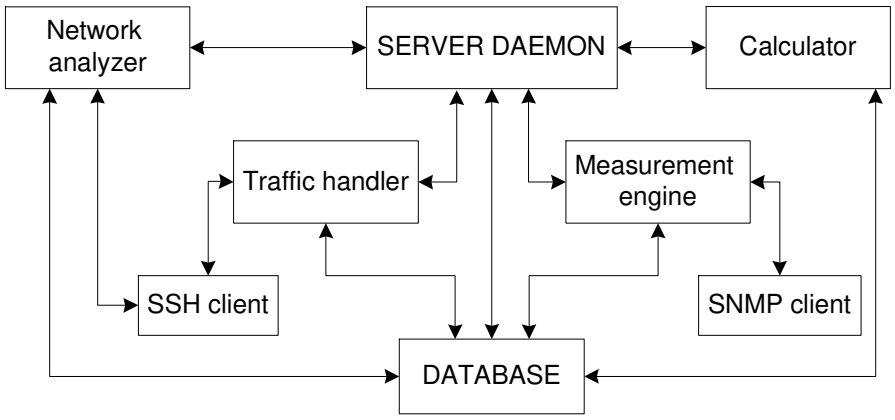

Figure 4.10 - The architecture of the server

4.5.1 Server daemon

The central part of the server will be the daemon which will be used to start up the server's performance by starting all the other components. The daemon will have a graphical user interface through which the administrator will be able to configure the settings of the server such as IP address and authentication data for SSH connection, system variables and many other settings. The daemon will store all configurations in the database to be used by other modules.

4.5.2 Network analyzer

The network analyzer will use information stored by the daemon to connect to the PE router by a secure SSH connection. To achieve this it will use another module (SSH client) to create and maintain the connection. The SSH client will use the authentication data to access the router configuration and deliver this data to the network analyzer.

The network analyzer will use the router's configuration to get information about the network such as number of LSPs, source and destination of LSPs, bandwidth usage of LSPs and other. This data will be also stored in the database since it will be used by other components of the server.

4.5.3 Traffic handler

The main purpose of the traffic handler module is to manage all traffic demands in the network. That includes:

- to manage all new incoming traffic flows
- to manage deleted traffic flows due to optimization or other reason
- to choose the most suitable LSP for the traffic
- to control the cost values of each LSP
- to optimize the traffic load – delete a traffic flow if necessary

The traffic handler can be considered as the main component of the server since it will manage all the logic about the distribution of the traffic in the network. It will use the information in the database created by other server's modules to practically realize the main goal of the whole server.

4.5.4 Measurement engine

The measurement engine will be used to manage the SNMP connection to the router and collect the results of measurements in the network. To create the connection it will use the module SNMP client which will maintain the channel. The SNMP client will gather the data from the router and pass it to the measurement engine. The measurement engine will then store the measured data in the database to be used for the LSP's cost calculations.

4.5.5 Calculator

The main work of the calculator will be to calculate the cost of each LSP based on mathematical formulas mentioned earlier. Information about the measured values of quality parameters stored in the database by the measurement engine will be used for these calculations. The results – cost values of LSPs – will be also stored in the database to be used by other modules of the server.

4.5.6 Database

The server has to maintain and work with a lot of different information. It is reasonable to use a database to keep the amount of data and to easily work with it. The use of database also contributes to the modular architecture of the solution since individual parts of the system do not have to communicate directly.

The database represents the central data point of the architecture. Every other module uses information from the database to perform its functions. The results of each module's operation are stored in the database so they can be used by any other module.

The structure of the database is shown in Figure 4.12. It consists of seven tables which store all necessary information about the network, measurements and used resources.

The table **Tunnel** stores all information about one LSP:

- Name – the name of the tunnel
- Description – the description of the tunnel
- Source_IP – the source IP address of the tunnel
- Destination_IP – the destination IP address of the tunnel

- Bandwidth – the overall amount bandwidth of LSP which is set when creating the LSP. This value does not change during the system's operation.
- Unused_bw – the actual amount of unused bandwidth. This value has to be updated according to actual usage of the LSP.
- Unused_bw_percent – percentual value of the unused bandwidth according to the bandwidth of the tunnel
- First_cost – the cost value of LSP for Class1 traffic calculated when the LSP is not used for any traffic. This calculation takes place in the very beginning of system's performance and it does not change.
- Act_cost – the actual cost value of LSP for Class1 traffic. This value has to be calculated periodically according to the actual values of quality parameters of the LSP.
- PhysicalInterface – the description of the physical interface used by the tunnel
- IfIndex – the index of the tunnel interface (used by SNMP)
- PhysicalIfIndex – the index of the physical interface which is used for the tunnel (used by SNMP)
- Policy_index – the index of the policy attached to the physical interface (used by SNMP)
- G_EXP – EXP value used for guaranteed traffic on this tunnel
- E_EXP – EXP value used for extra traffic on this tunnel

The table **IP SLA** stores all information about configured measurements in the network. Each LSP has one such a measurement configured. This information is set by the administrator and should not be changed during the server's operation:

- ID_tunnel – the identification of the tunnel this measurement is assigned to
- Source_lo – the number of the source loopback interface used
- Source_IP – the source IP address of the probe
- Source_port – the source port of the probe
- Destination_lo – the number of the destination loopback interface used
- Destination_IP – the destination IP address of the responder
- Destination_port – the destination port of the responder
- Number_of_packets – the number of packets to be generated
- Packet_interval – the interval between sending the packets

- Frequency – the rate at which the IP SLA operation repeats
- Start_time – the time for the operation to begin
- Lifetime – the amount of time for the operation to run for

The table **Measurements** stores information about last three measurements of the quality parameters of each LSP together with the average values used in the cost calculations. All of these values have to be updated periodically:

- ID_IP_SLA – the ID of the actual IP SLA probe used for the measurement
- Delay_avg – the average value of measured delay
- Jitter_avg – the average value of measured jitter
- Loss_avg – the average value of measured loss

The table **Class** stores information about each class of traffic together with some configuration details:

- Description – the name of the class
- G_bw – the amount of guaranteed bandwidth in bits per second
- In_bw – the amount of bandwith measured at the input interface
- G_exp – the EXP value used for the traffic within the guaranties
- E_exp – the EXP value used for the extra traffic
- Used_e_bw – the amount of bandwidth used by the extra traffic
- Cir – value of Commited Information Rate (CIR) for the policy configuration
- Pir – value of Peak Information Rate (PIR) for the policy configuration

The table **Class_maps** stores information about the data rate counters on the configured policies. This information is used to calculate accurate bandwidth usage on each tunnel:

- Policy index – the index of the specific policy-map (used by SNMP)
- Object_index – the object index of the specific class-map used by the policy-map (used by SNMP)
- Config_index – the configuration index of the specific class-map used by the policy-map (used by SNMP)
- Name – the name of the class-map
- Data_rate – the actual data rate in bits per second of the class-map

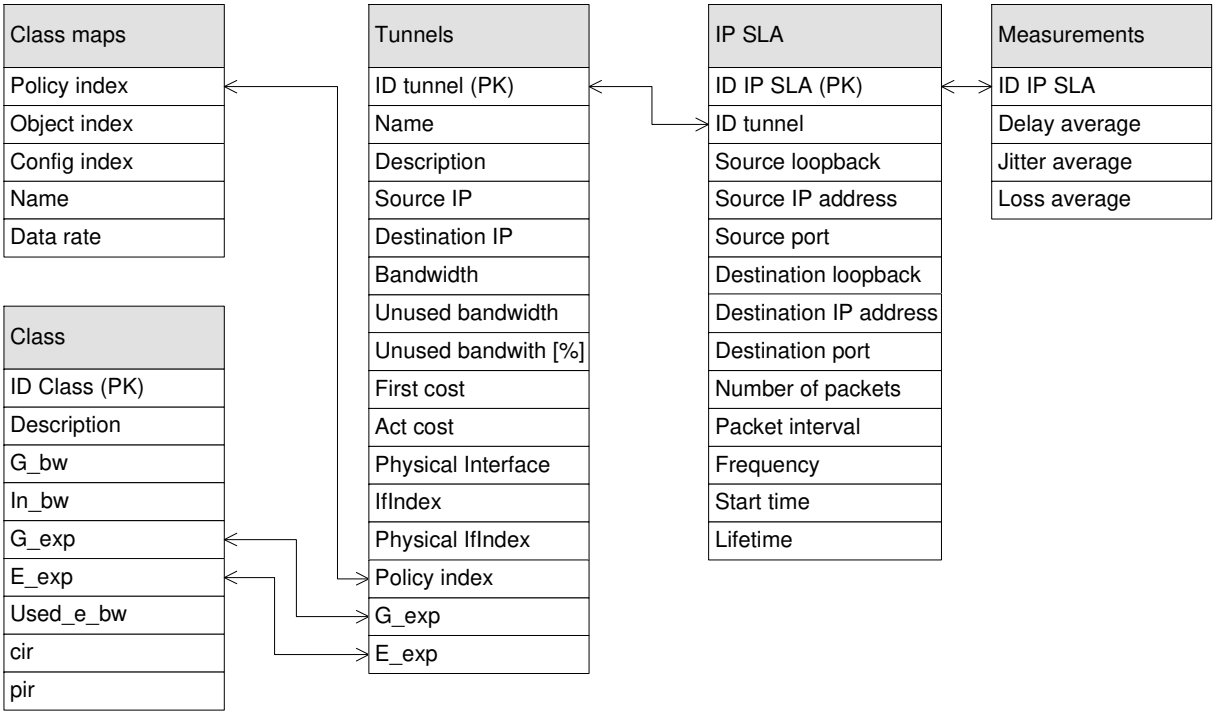

Figure 4.11 - The database

4.6 Summary

In this section the proposal of a traffic-engineering server was described. The system requirements for successful operation of the server were defined together with the classification principles which have to be used. Basic logic of the operations of the proposed server was analyzed and described in form of flow diagrams. Individual steps in the operation of the server were described: the measurement of the quality parameters, the calculation of LSP cost, the logic applied in assigning traffic onto the LSPs, the process of optimization. Reference table used for calculating the cost of LSP were proposed and explained.

The proposal of the implementation covered methods of communications between the server and the Cisco router, detailed use of IP SLA for network measurements and methods for generating traffic to simulate real network utilization.

The next part covered proposed testing topologies and proposed layout of LSPs. Individual bandwidth requirements for each class were also defined together with brief description of the testing scenarios.

The last part of this section described the software components of the proposed solution. Each component was briefly identified with its functions and responsibilities. The structure of the database used as a storage entity was described and described.

5 Implementation

The proposed online traffic-engineering server for optimal distribution of traffic and even utilization of LSP in the network was implemented as an interactive application in C#. It provides an interface for the user to access the results of the server operations. Based on the proposed architecture in 4.5 Software components the implemented server consists of several autonomous components which communicate through the database.

In this section details of the implementation will be discussed. The configuration of communication between the server and the router will be described; details of packet classification and implementation of measuring will be analyzed. The components of the server will be described in detail to provide complex information about the server functionalities.

The implementation and testing scenarios work with the network configured as shown in Figure 5.1.

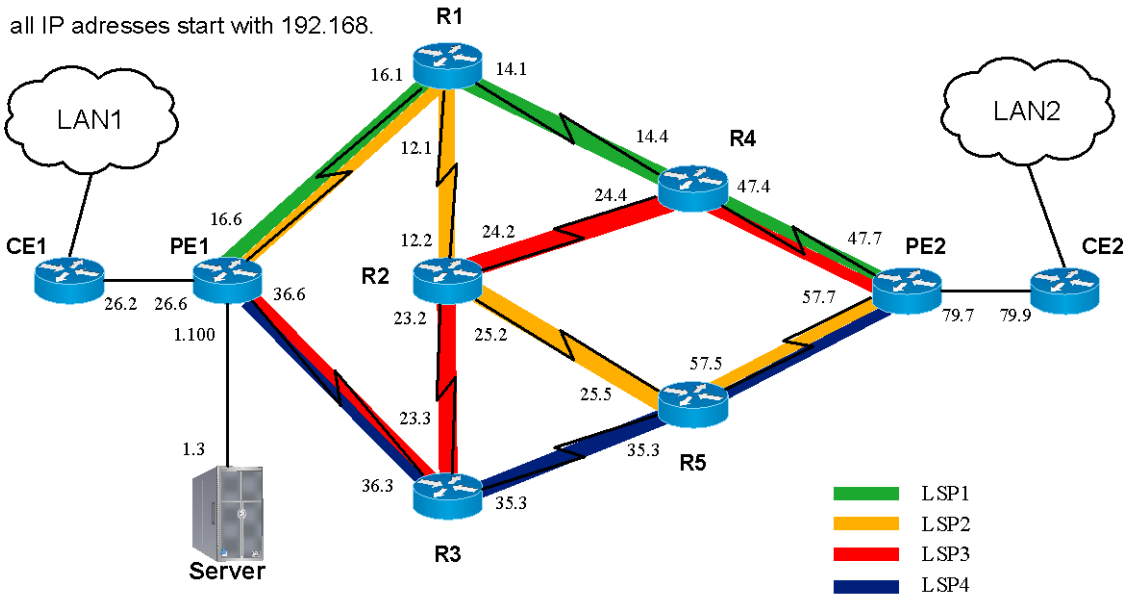

Figure 5.1 - Network IP adressing

5.1 Communication

The communication between the server and the PE router is implemented using SNMPv3 and SSH as described in section *4.3.1 Communication*. SNMPv3 was implemented using SnmpSharpNet library. It was used to gather measured values of delay, jitter and packet loss together with some information about the topology layout and LSPs. SSH was implemented using the SharpSSH library and it was used to gather some additional information and to configure necessary changes on the PE routers.

The configuration necessary to provide the SNMPv3 connection to the PE router consists of following:

- Configuration of SNMP server view
- Configuration of SNMP server community
- Configuration of SNMP server group

The configuration for SSH connection consists of following:

- Configuration of local username and password
- Configuration of virtual lines
- Enabling of SSHv2 with generating of RSA keys

The detail of the configurations is shown in Figure 5.2 and Figure 5.3 respectively.

```
snmp-server group priv1 v3 priv match exact read VIEW
snmp-server view VIEW iso included
snmp-server view VIEW cisco included
snmp-server community COMM view VIEW RO
```

Figure 5.2 - The configuration of SNMP

```
username ivana privilege 15 password 0 cisco
line vty 0 4
 login local
 transport input ssh
line vty 5 15
 login local
 transport input ssh
```

Figure 5.3 - The configuration of SSH

5.2 Network configurations

The server has some prerequisites as described earlier in this work. The network has to be operational and the classification of the traffic has to be configured. After the server is started it uses SSH to change the configuration if necessary. In this section the configuration details are described.

5.2.1 Classification of traffic

The traffic flowing through the network has to be classified to get the appropriate attention. The first classification is done at the CE router at customer's side of the network. Traffic is here classified into four classes of traffic using four IP Precedence values (1 to 4). The configuration of this classification and marking can be found in Appendix A.

Traffic marked with IP Precendece value is then processed at the PE router. Here is the traffic marked with EXP bits since it is entering the MPLS network. The marking is done dynamically by the server according to current situation in the network. Example of the classification on the CE router is shown in Figure 5.4. Details of this logic will be described later in this section.

```
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.2.1.0 0.0.0.255

class-map match-any class1
 match access-group 101

policy-map set_ip_prec_out
 class class1
  set ip precedence 1

interface FastEthernet0/0
 service-policy output set_ip_prec_out
```

Figure 5.4 - Example of classification on CE router

5.2.2 Implementation of IP SLA

The measurements of network performance parameters are done using IP SLA probes as proposed earlier in this work. The PE routers are used for the IP SLA ICMP-jitter operation to measure one-way delay, jitter and packet loss on each configured LSP. Since our proposal works with four fixed LSPs there are four IP SLA operations

running. Each operation has configured one dedicated loopback on each PE router. These loopback interfaces are used as source and destination IP addresses for the measurements. To measure the parameters on the specific tunnel static routes are used to direct the IP SLA probes on the correct tunnel interface. The configuration of IP SLA parameters is done dynamically by the server after the network is analyzed. Example of this configuration is shown in Figure 5.5. Details of this configuration can be found in Appendix A.

```
interface Loopback100
 ip address 66.66.66.1 255.255.255.255
ip sla 1
 icmp-jitter 66.66.66.2 source-ip 66.66.66.1
 frequency 15
ip sla schedule 1 life forever start-time now
ip route 66.66.66.2 255.255.255.255 Tunnel1
```

Figure 5.5 - The example of IP SLA configuration

5.2.3 Implementation of the tunnels

Four tunnels are used in our proposed testing scenario. These tunnels are configured in advance, before the server is started. The layout of tunnels is shown in Figure 4.9. Each tunnel has defined its bandwidth and explicit path through the network. MPLS Class-based tunnel selection (CBTS) is used to choose the right tunnel for each traffic class marked with EXP value. All four LSPs are configured as tunnel members of one tunnel master. The tunnels and CBTS have to be operational before using the TE server.

The server assigns two EXP values to each tunnel after the network analysis is done. These EXP values are used to distinguish between traffic within the guarantees and the extra traffic. The first EXP value is used for guaranteed traffic on the tunnel and has its priority configured on the physical interface. The second EXP value is used for the extra traffic which can use this tunnel if there is free bandwidth. The example of such configuration is shown in Figure 5.6. The details of the configuration can be found in Appendix A.

```
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 1.1.1.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 7 7
 tunnel mpls traffic-eng bandwidth 1544
 tunnel mpls traffic-eng path-option 1 explicit name LSP1
 tunnel mpls traffic-eng exp 0 4
 no routing dynamic

ip explicit-path name LSP1 enable
 next-address 192.168.16.1
 next-address 192.168.14.4
 next-address 192.168.47.7

interface Tunnel1234
 ip unnumbered Loopback0
 tunnel destination 1.1.1.7
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng exp-bundle master
 tunnel mpls traffic-eng exp-bundle member Tunnel1
 tunnel mpls traffic-eng exp-bundle member Tunnel2
 tunnel mpls traffic-eng exp-bundle member Tunnel3
 tunnel mpls traffic-eng exp-bundle member Tunnel4
 no routing dynamic
```

Figure 5.6 - The example of tunnel configuration

5.3 Implementation of the server

The server is implemented as a C# application running on PC connected to the PE router. The server consists of several parts as described earlier. Each component is defined as one class with its own functions and variables. Functions of these components will be described in this section.

5.3.1 Database

For the purpose of storing all information needed for the operation of the server the SQLite database was used. Its structure was defined earlier in section 4.5.6 Database. The connection to the database is done at the beginning of the server operation. All functions performed with the database (connection, reading, writing or updating) are done using the Devart.Data.SQLite namespace.

The database is used to trigger actions of the server since it stores all required data. Tables tunnels and class have set triggers which call functions from the class Database in the code of the server.

The table tunnels has set trigger for updating the value of act_cost of LSP and for updating the value of unused_bw_percent. The function called after updating the cost of LSP controls if the LSP carries Class1 guaranteed traffic and if the cost reached the value of 3. If this condition is met the server reacts with appropriate actions. The function called after updating the unused bandwidth values controls if the unused bandwidth is critical according to Formula 6.

The table class has set trigger for updating the in_bw value. The function called controls if the value is not bigger than the guaranteed bandwidth for this class. If it is, appropriate actions take place. The process of triggering actions for each trigger is show in Figure 5.7, Figure 5.8 and Figure 5.9. All actions triggered by the database belong to the Traffic Handler component and will be described in detail later in this section.

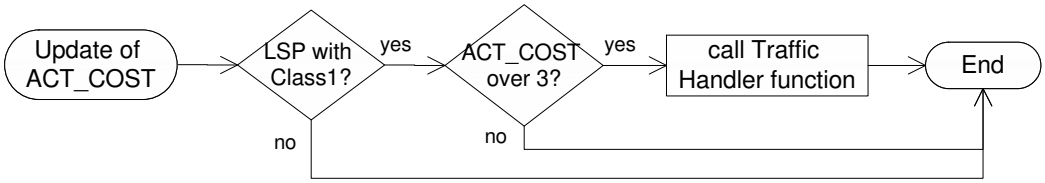

Figure 5.7 - Trigger for updating act_cost on LSP

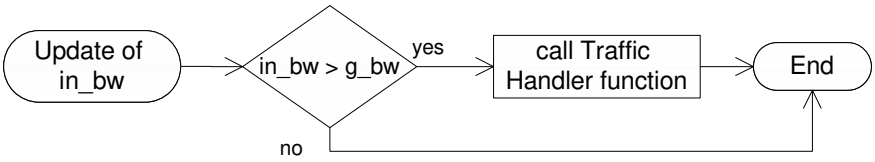

Figure 5.8 - Trigger for updating unused_bw of LSP

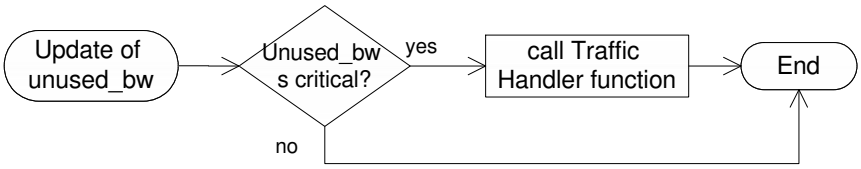

Figure 5.9 - Trigger for updating in_bw of class

5.3.2 Daemon

The Daemon functions as the main interface with menu. It starts all other processes and functions based on the input given by the user. It has a menu of commands which can be used to trigger specific actions:

- Analyze network – starts the Network Analyzer
- Start – starts the Traffic Handler
- Show tunnels – shows information about LSPs learned by the Network Analyzer
- Reconnect – new SSH connection (in case the default connection fails)
- Help – shows the menu

5.3.3 Network analyzer

The network analyzer is used to analyze the MPLS network and obtain following information:

- Name of the tunnel
- Source and destination IP address of the tunnel
- Physical interface used by the tunnel
- Reserved bandwidth for the tunnel
- Index of the tunnel interface and physical interface

Some of the information is retrieved by analyzing the output of various show commands issued using the SSH connection. Other information such as the index of the

interface is obtained through the SNMPv3. All information is automatically stored in the database.

The MIBs used by SNMPv3 are:

- ifDescr (1.3.6.1.2.1.2.2.1.2)
- ifIndex (1.3.6.1.2.1.2.2.1.1)

The Network Analyzer then fills the database with configuration details about IP SLA entered by the used or loaded with default values.

This component is called only once, right after the start of the server and it fills the database with information about the configured topology. The flow diagram describing the processes of this component is show in Figure 5.10.

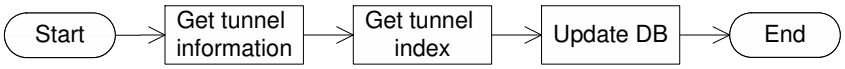

Figure 5.10 - Operation of the Network Analyzer

5.3.4 Measurement engine

The measurement engine is responsible for configuring IP SLA and periodically retrieving results of the measurements. It also periodically obtains the load on the tunnel interfaces using the policy counters on interfaces. It can be called after the network analyzer is finished since it needs the information about the tunnels. The measurement results are stored in the database for further processing.

For IP SLA configuration both PE routers have to be configured. On the tail-end router following has to be configured:

- Loopback interface for each IP SLA operation with unique IP address
- IP SLA responder

On the head-end router following has to be configured:

- Loopback interface for each IP SLA operation with unique IP address
- IP SLA operation with parameters
- IP SLA scheduling
- Static route for each IP SLA operation into one of the tunnels
- Load-interval for accurate results of interface counters

The measurement engine then controls if there is some policing configured. This configuration is automatically erased since it could affect the measurements.

The next step is the initial distribution of EXP values among the tunnels. Each class has one EXP for guaranteed traffic assigned. The choice of tunnel for each class is made based on the requirements of classes. The classes are served according to their priorities. These assignments are stored in the database.

Input policing on the PE router has to be configured for these assignments to be used. According to the data in the database class-maps, input policy-map and output policy-map is configured. Also the mapping of EXP values to the tunnels has to be configured. The example of the configuration of class-maps, input policy and output policy is shown in Figure 5.11. The details of this configuration can be found in Appendix A.

```
class-map match-any ip_prec_1
 match ip precedence 1

policy-map input_policy
 class ip_prec_1
   police cir 300000 pir 300000
     conform-action set-mpls-exp-imposition-transmit 7
     exceed-action drop
     violate-action drop

class-map match-any mpls_exp7
 match mpls experimental topmost 7

policy-map output_policy_Serial0/0/1
 class mpls_exp7
   priority 300
```

Figure 5.11 - Example of policy configuration

After the initial distribution of EXP values is done, the index of each policy on each interface has to be obtained. This is done using the SNMPv3 MIBs:

- cbQosIfIndex (1.3.6.1.4.1.9.9.166.1.1.1.1.4)
- cbQosCMCfgTable (1.3.6.1.4.1.9.9.166.1.7.1)
- cbQosObjectsTable (1.3.6.1.4.1.9.9.166.1.5.1)

When all necessary information is gathered the measured results can be obtained. The SNMP MIBs for delay, jitter and packet loss used are:

- rttMonLatestIcmpJitterAvgSDJ (1.3.6.1.4.1.9.9.42.1.5.4.1.45)
- rttMonLatestIcmpJitterOWAvgSD (1.3.6.1.4.1.9.9.42.1.5.4.1.47)
- rttMonLatestJitterOperPacketLossSD (1.3.6.1.4.1.9.9.42.1.5.2.1.26)

The measured values are added to two last measurements so the average value can be used instead. These average values are then stored in the database.

After the measurement the cost of each tunnel is updated. For this function the component Calculator is used.

The last operation of the measurement engine is the update of data rates on each tunnel interface together with data rate on the input interface. These values are obtained from the policy-maps attached to physical interfaces. The SNMPv3 MIB used is:

- cbQosCMPrePolicyBitRate (1.3.6.1.4.1.9.9.166.1.15.1.1.7) with the index of the policy and object index of the class-map

All information gathered by this component is stored in the database.

The functions for gathering measured values, updating the cost of tunnels and getting the data rates are periodically repeated while the server is running. The flow diagram of the Measurement engine functions is shown in Figure 5.12.

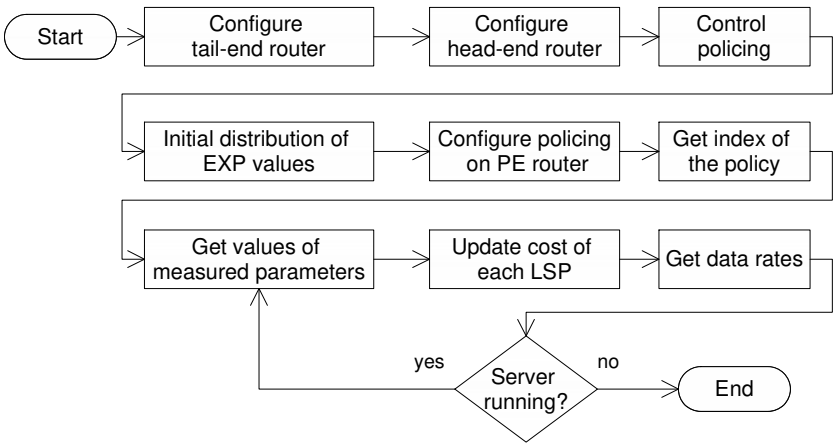

Figure 5.12 - Operation of the Measurement engine

5.3.5 Traffic handler

The class Traffic Handler consists of three main functions:

- Act_cost_alarm
- Class_in_bw_high
- Optimize

These three functions are called by the triggers in the database when the conditions are met. These functions provide the main logic of the server operations since they are responsible for managing the traffic flows using the optimal distribution of traffic while preserving QoS for Class1 traffic.

The Traffic Handler manages the flow of the traffic by configuring the input policy-map at the PE router. By default, at the start of this module each class has set the CIR and PIR values to its guaranteed bandwidth. Only one EXP value is used and therefore, only one LSP per class is used. By setting the values of CIR and PIR to different values and by setting various EXP values in the conform-action and exceed-action of the police command optimal distribution of traffic can be obtained.

Function Act_cost_alarm is called when the act_cost value of the LSP carrying Class1 traffic reaches value of 3 or value of 12. These values are considered to be the borderlines for optimal QoS (for guaranteed and extra traffic). The cost value of 3 and higher on LSP carring the guaranteed traffic of Class1 represents that data traffic is using the LSP for Class1 traffic. The data traffic has to be therefore removed to decrease the load on this LSP and subsequently decrease also the cost of the LSP. This is done by finding any data traffic class which uses e_exp value of the considered LSP and removing the usage of this e_exp (update the database and change the policy configuration).

If the cost of LSP carrying extra traffic of Class1 reaches the value of 12, the extra traffic cannot use this LSP and therefore has to be removed. The details of this process are shown in Figure 5.13.

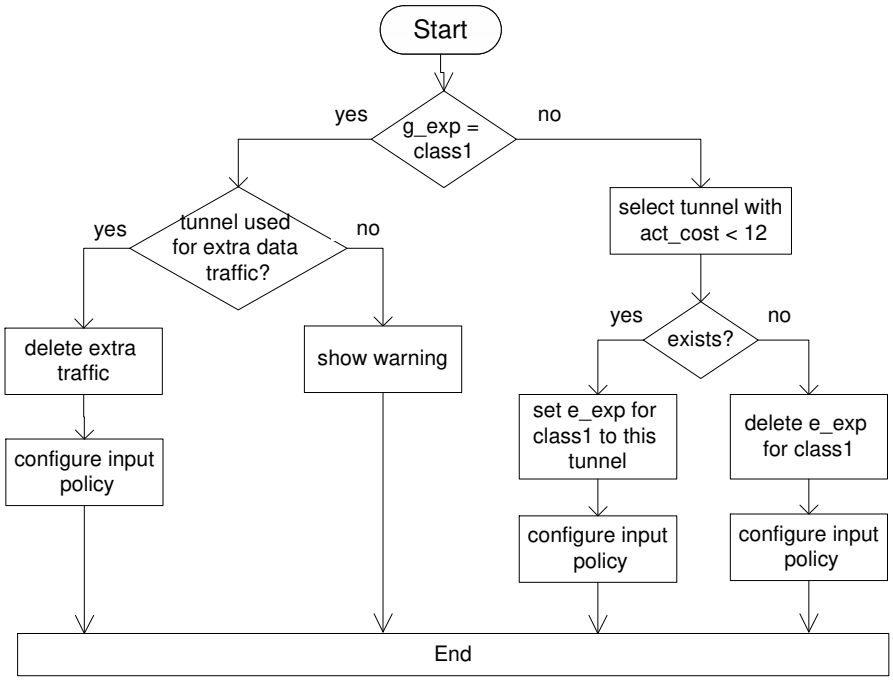

Figure 5.13 - Function act_cost_alarm

The function Class_in_bw_high is called by the trigger in the database when the input bandwidth of some class is higher than its guaranteed bandwidth. The logic of this function is shown in **Chyba! Nenašiel sa žiaden zdroj odkazov.**. The flow chart uses these expressions:

- g_exp – EXP value used for guaranteed traffic
- e_exp – EXP value used for extra traffic
- in_bw – input bandwidth of class
- g_bw – guaranteed bandwidth of class
- used_e_bw – amount of bandwidth used for extra traffic
- cir, pir – values of Commited and Peak Information Rate
- new_cir, new_pir – new calculated values of cir and pir
- g_bw_split – flag of class detecting if the class is using splitting of guaranteed traffic
- t_e_exp – tunnel used for extra traffic of class

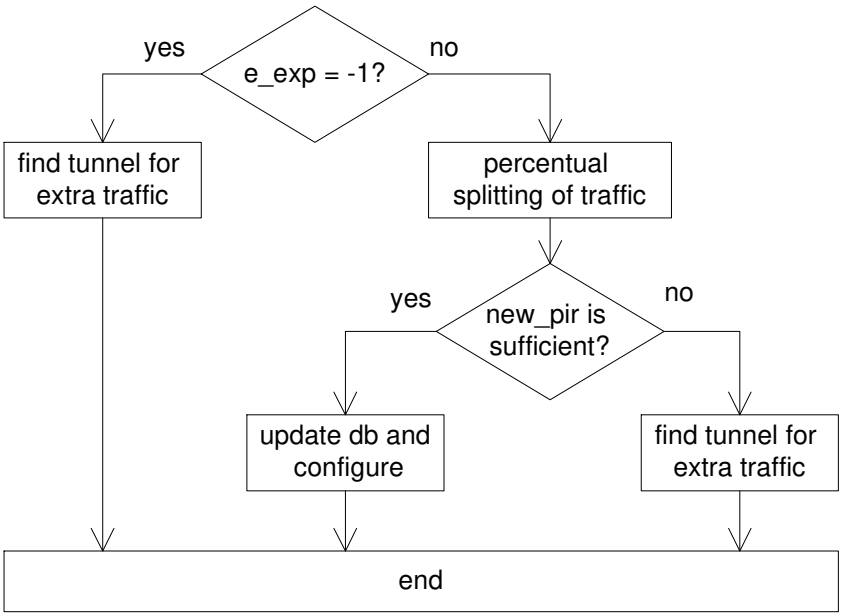

Figure 5.14 - The function in_bw_high

The function deals with assigning the extra traffic of any class to suitable LSP. If the class has not set e_exp the server tries to find LSP to carry the extra traffic. If the class has set e_exp the server tries to use this LSP. The server tries to find new LSP if the e_exp LSP cannot be used due to any reason. If no LSP has enough resources for

carrying the extra traffic, LSPs with lower classes can be used while the lower class traffic is deleted from this LSP.

The whole logic of this function is very complex due to all possible combinations of conditions that can happen. The logic of smaller function used in the function in_bw_high is shown in Figure 5.15. Flow diagrams of functions used in this function can be found in Appendix A.

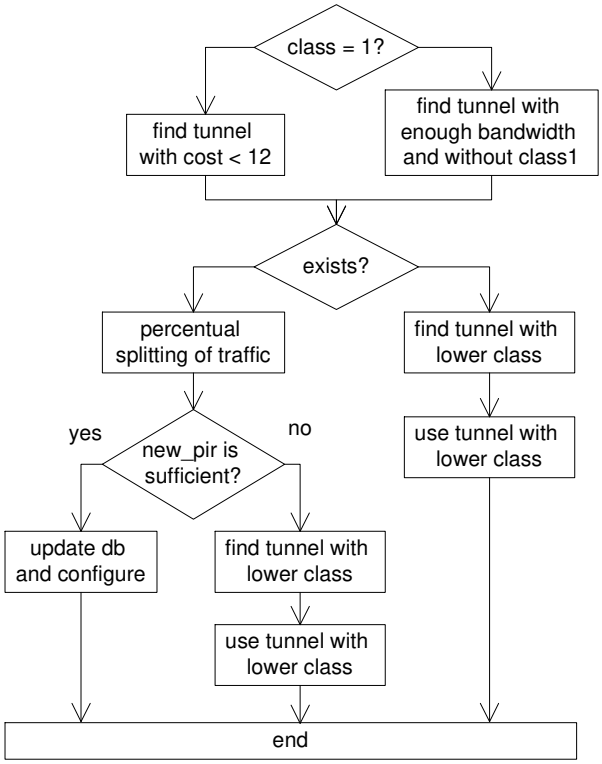

Figure 5.15 - Function find_tunnel_for_extra_traffic

The function Optimize is called by the trigger in the database if a LSPs are unevenly utilized. This state is detected if some LSP has its unused bandwidth value according to the Formula 6.

The main idea of the process of optimization is to steer the traffic across more LSPs to provide more equal LSP utilization. Since the server works by changing the cir and pir values, optimization is also done using this approach. By setting the values and choosing the suitable LSP for the traffic the required distribution of traffic is obtained.

If the LSP is carrying extra traffic of some class, this traffic is deleted and will be mapped to another LSP by the function in_bw_high.

If the class using the LSP for guaranteed traffic has set e_exp, this e_exp is used to take more traffic if possible. If it is not possible, new LSP is found. In case that no LSP has enough resource, prioritization takes place and extra traffic of lower class can

be deleted to create space for traffic of higher class. The whole process of optimization is shown in Figure 5.16.

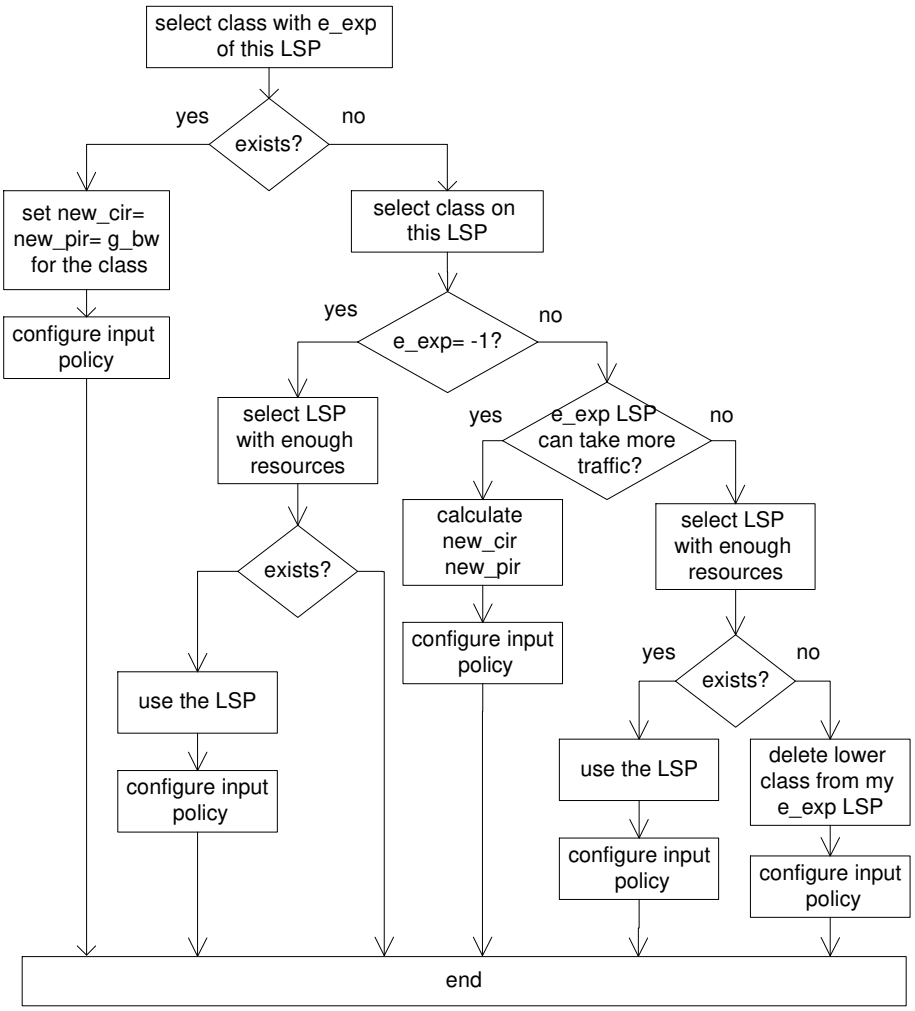

Figure 5.16 - The function optimize

5.3.6 Calculator

This component contains all mathematical calculations of the server. In separate functions it calculates:

- the cost of LSP
- unused bandwidth of LSP
- average unused bandwidth

The calculations are done after updates in the database are made – update of network performance parameters or unused bandwidth of LSP. The calculated values are stored in the database and can trigger further actions.

6 Experiments

Number of experiments had to be performed to prove its effectiveness. The main goal of the server was to optimally distribute the traffic among the LSPs in the network while providing required QoS for Class1 traffic. To verify the results of the server, proposed testing topology show in Figure 5.1 was used.

In each experiment different set of generated traffic was used. Traffic was generated to simulate different classes with different bandwidth demands. To evaluate the results of the server, in each experiment following values were measured:

- Input data per class (in bits per second)
- Throughput of the traffic per class (in bits per second)
- Tunnel utilization (in %)
- Loss per class (in %)
- QoS parameters for Class1 traffic (delay in ms, jitter in ms, loss in %)

The results were compared to the same experiment done on the network without the use of the server. The desired result was to gain maximal throughput of the traffic, equal utilization of LSPs and minimal loss while preserving optimal QoS for Class1 traffic.

In every experiment the network and LSPs were identically configured. The guaranteed bandwidth for each class was set according to the bandwidth of LSPs and physical possibilities of the network:

- Class1 – 700kbps
- Class2 – 1100kbps
- Class3 – 460kbps
- Class4 – 460kbps

In each experiment, two versions are presented – one without the use of TE server and one with TE server used. The conditions for both versions of the experiment were identical although the input data rate may have slightly different progress. This difference is caused by the generation of traffic which was done manually.

6.1 Experiment 1

6.1.1 Testing scenario

In the first experiment following set of traffic was generated:

- Class1 – 650 kbps
- Class2 – 514 kbps
- Class2 – 514 kbps
- Class2 – 514 kbps
- Class2 – 300 kbps
- Class3 – 514 kbps

The traffic of Class4 was intentionally not generated to show how the unused space can be optimally used by traffic of other classes. The amount of input data entering the PE router is shown in **Chyba! Nenašiel sa žiaden zdroj odkazov.** and **Chyba! Nenašiel sa žiaden zdroj odkazov.**.

The throughput achieved during this experiment is shown in **Chyba! Nenašiel sa žiaden zdroj odkazov.** and **Chyba! Nenašiel sa žiaden zdroj odkazov.**. It is obvious, that each class is using only one tunnel without the use of TE server. This behavior is caused by CBTS which can select only one tunnel for one traffic class and therefore each class can use only the bandwidth of this tunnel. All traffic above this bandwidth is dropped as shown in Figure 6.5. When the TE server is used, each class can use additional bandwidth from other (unused) tunnel. In this scenario, Tunnel4 is not being used by Class4 (since no traffic of Class4 is generated) and can be use by traffic of Class2. The packet loss in this case is lower as shown in Figure 6.6. It is not absolutely eliminated since the amount of traffic generated is more than these two tunnels can take.

The utilization of tunnels is optimized with the use of TE server when compared to the experiment without the server as shown in Figure 6.7 and Figure 6.8. All tunnels are equally utilized in the end of the experiment, which leads to lower packet loss and higher throughput.

The values of delay and jitter were comparable in both experiments and did not exceed the threshold as shown in Figure 6.9 - Experiment 1, Delay, without TE serverFigure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12.

6.1.2 Evaluation

The first experiment shows that without the use of TE server traffic is limited by the tunnel it uses. This leads to constant packet loss for Class3 and Class4 traffic. The utilization of tunnels is not optimal since one of the tunnels is not used during the whole experiment.

Using the TE server helps to equally utilize all existing tunnels and therefore use all resources available in the network. As a result, the throughput of all classes of traffic is maximized and the packet loss is minimized or completely eliminated.

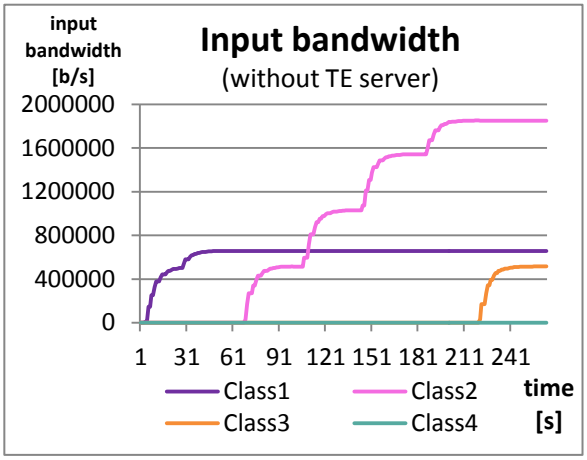

Figure 6.1 – Experiment 1, Input bandwidth, without TE server

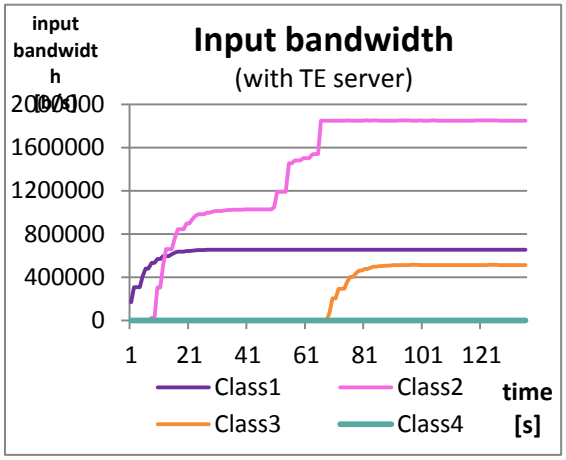

Figure 6.2 – Experiment 1, Input bandwidth, with TE server

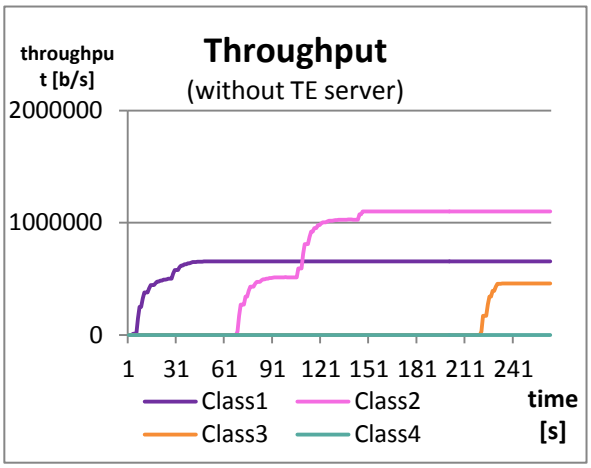

Figure 6.3 – Experiment 1, Throughput, without TE server

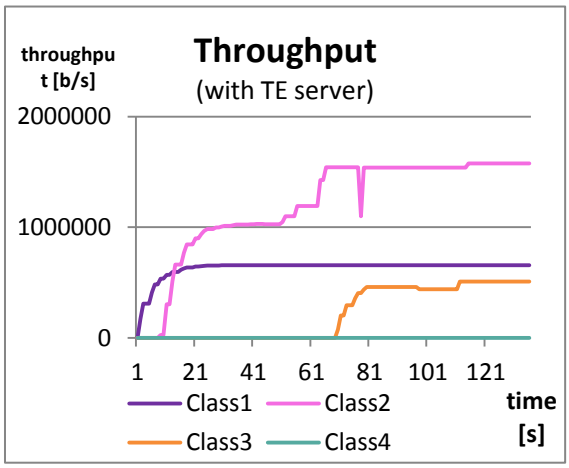

Figure 6.4 – Experiment 1, Throughput, with TE server

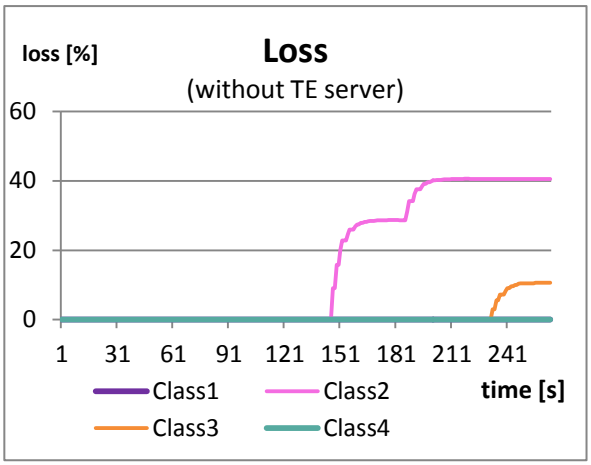

Figure 6.5 – Loss, without TE server

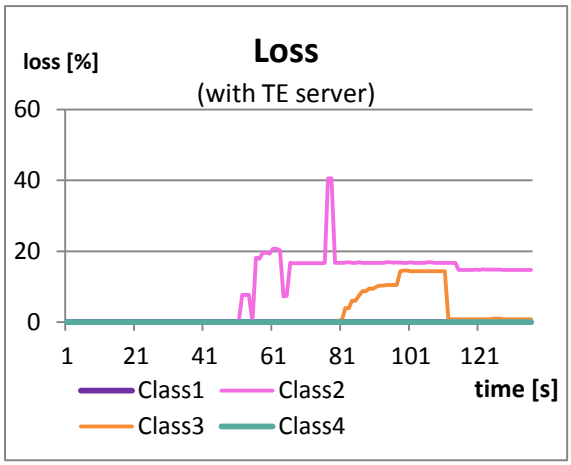

Figure 6.6 – Loss, with TE server

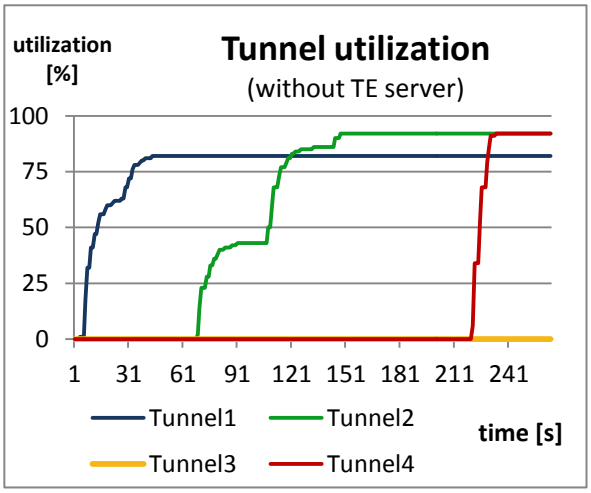

Figure 6.7 – Experiment 1, Utilization of tunnels, without TE server

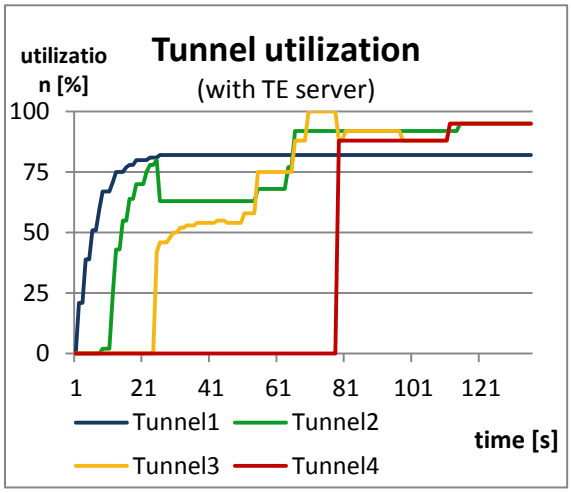

Figure 6.8 – Experiment 1, Utilization of tunnels, without TE server

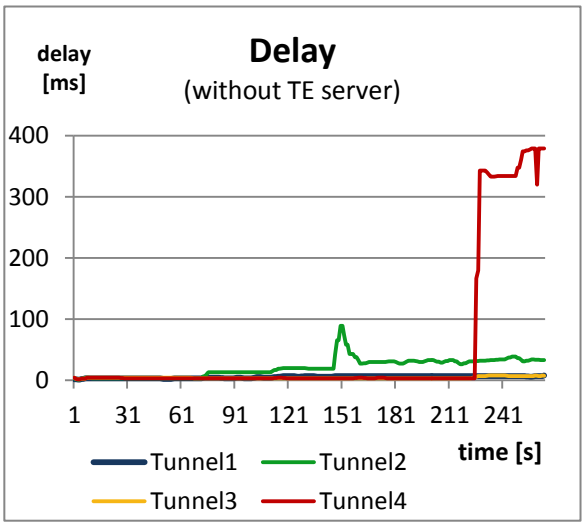

Figure 6.9 - Experiment 1, Delay, without TE server

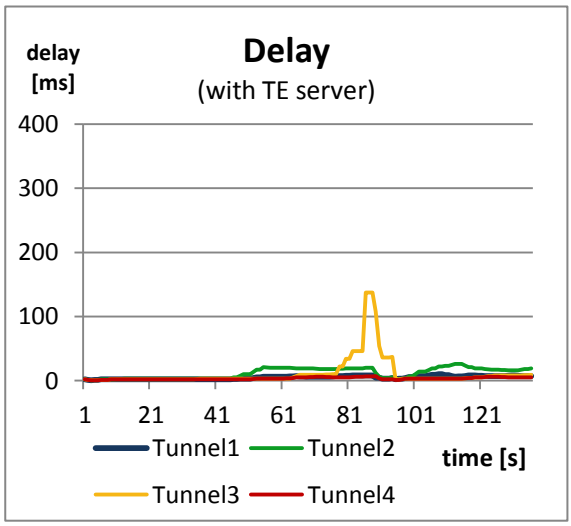

Figure 6.10 - Experiment 1, Delay, with TE server

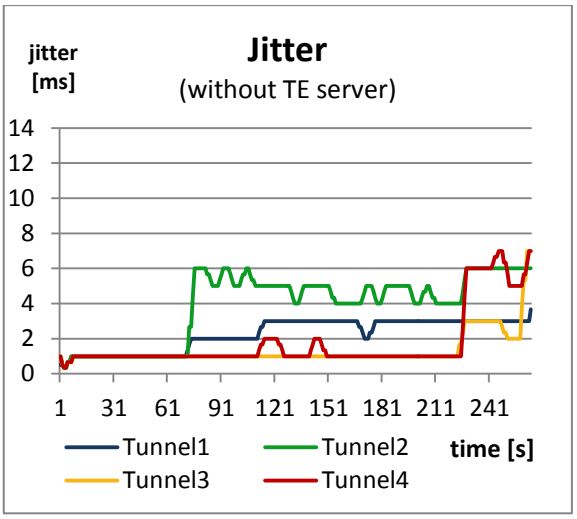

Figure 6.11 - Experiment 1, Jitter, without TE server

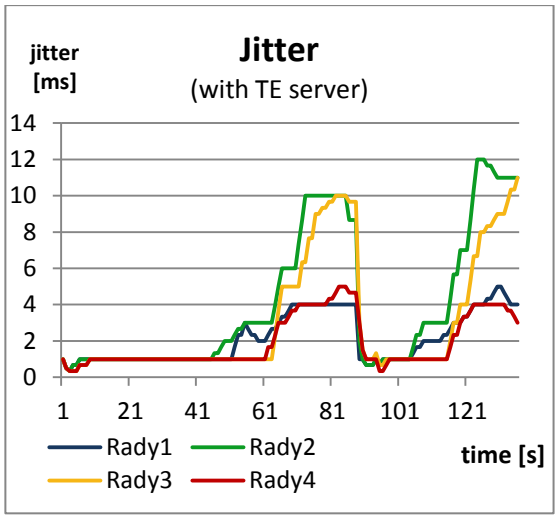

Figure 6.12 - Experiment 1, Jitter, with TE server

6.2 Experiment 2

6.2.1 Testing scenario

In the second experiment, following traffic was generated:

- Class4 – 600 kbps
- Class2 – 1130 kbps
- Class3 – 514 kbps
- Class1 – 650 kbps
- Class3 was stopped

In this scenario traffic of Class2, Class3 and Class4 is generated above the guaranteed bandwidth. The input bandwidth requirements are shown in Figure 6.13 and Figure 6.14. In the experiment without the TE server, each class is allowed to use only specific amount of bandwidth according to the tunnel it uses. When the TE server is used, however, all traffic can be satisfied because all network resources are used. The throughput of data in each experiment is shown in Figure 6.15 and Figure 6.16.

The throughput of Class4 shows the optimization and prioritization among classes. Class4 uses the tunnel of Class1 (throughput rises up to 920 kbps). When the traffic of Class1 enters the network, the traffic of Class4 is limited to its guarantees (460kbps). When the Class3 is removed from the network, the traffic of Class4 uses the release bandwidth and its throughput rises again to 920 kbps.

The packet loss is constant when the TE server is not used as shown in Figure 6.17. The optimization of traffic flows done by the TE server causes that the traffic can use the network resources even when it is above the guaranteed bandwidth. The momentary packet loss of Class2, Class3 and Class4 is caused by the delay in reaction of the server and re-routing the traffic flows to another tunnel as shown in Figure 6.18.

The utilization of tunnels is much more efficient with the use of the TE server since it utilizes all four tunnels compared to only three used without the TE server. The comparison is shown in Figure 6.19 and Figure 6.20.

The values of delay and jitter were comparable in both experiments and did not exceed the threshold as shown in Figure 6.21, Figure 6.22, Figure 6.23 and Figure 6.24.

6.2.2 Evaluation

In this experiment we showed that the equal utilization of all tunnels in the network preserves the priorities among different classes of traffic. Without the TE server all classes are limited by the tunnel bandwidth as in the previous experiment. It causes constant packet loss for classes Class2, Class3 and Class4 which have generated traffic above their guarantees.

The throughput of traffic is maximized with the use of the TE server. In this experiment, Class4 is using Tunnel1 while traffic of Class1 is not generated. As Class1 enters the network, traffic of Class4 is limited down to its guarantees to free the Tunnel1 for Class1 traffic. When Class3 traffic is stopped, Class4 traffic renews its throughput as it uses the Tunnel3. As a result, the packet loss of all classes is minimized.

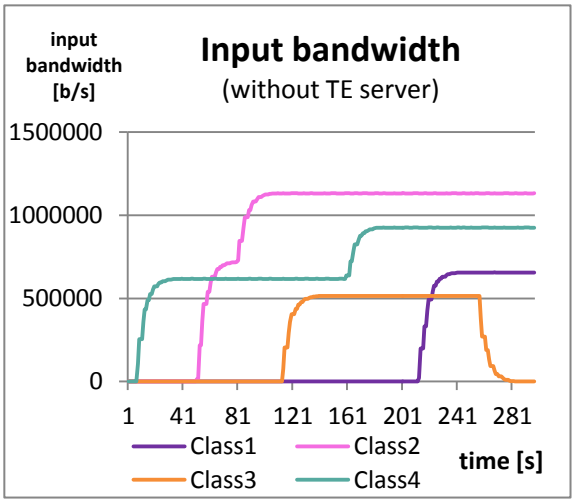

Figure 6.13 – Experiment 2, Input bandwidth, without TE server

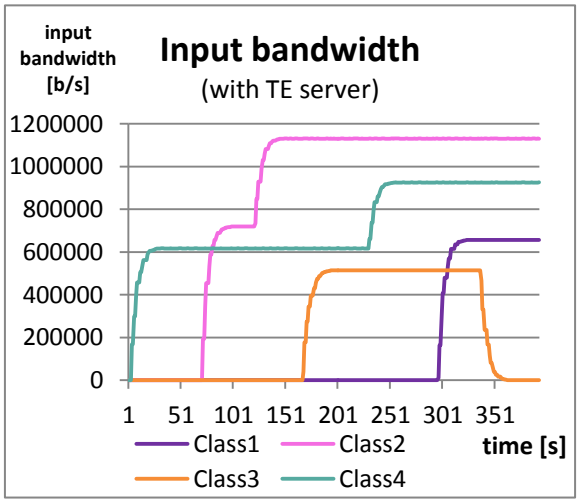

Figure 6.14 - Experiment 2, Input bandwidth, with TE server

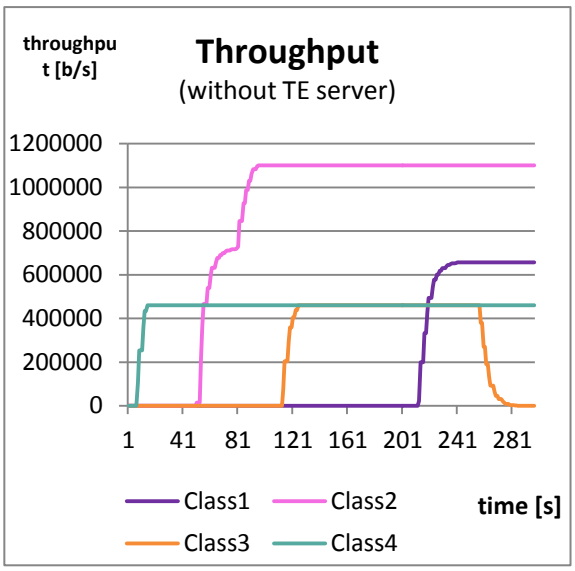

Figure 6.15 - Experiment 2, Throughput, without TE server

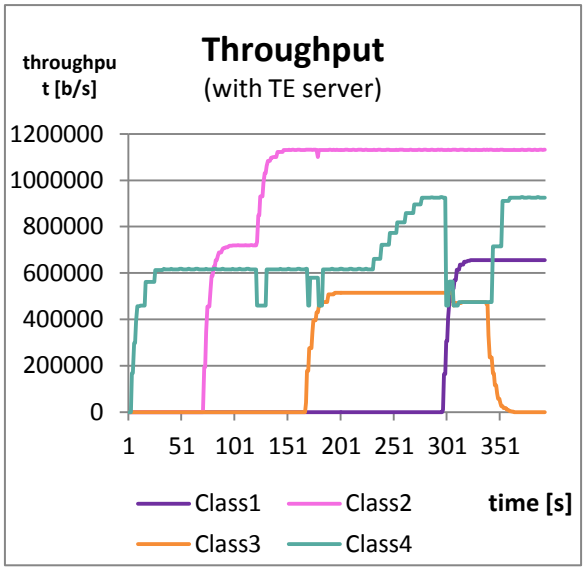

Figure 6.16 - Experiment 2, Throughput, with TE server

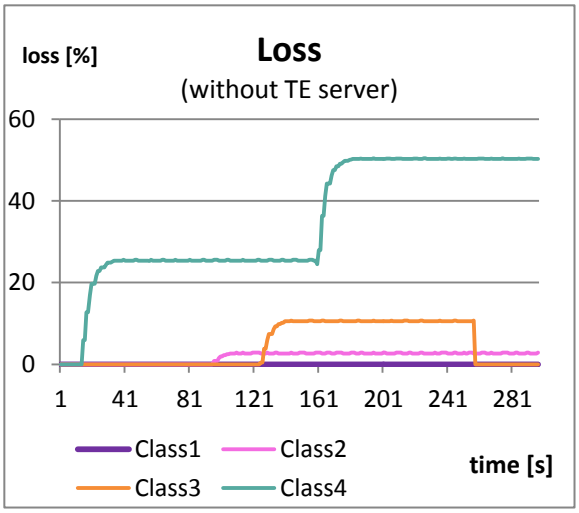

Figure 6.17 - Experiment 2, Loss, without TE server

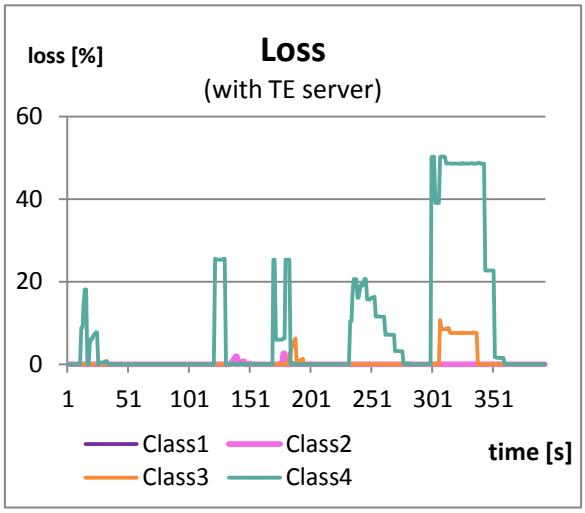

Figure 6.18 - Experiment 2, Loss, with TE server

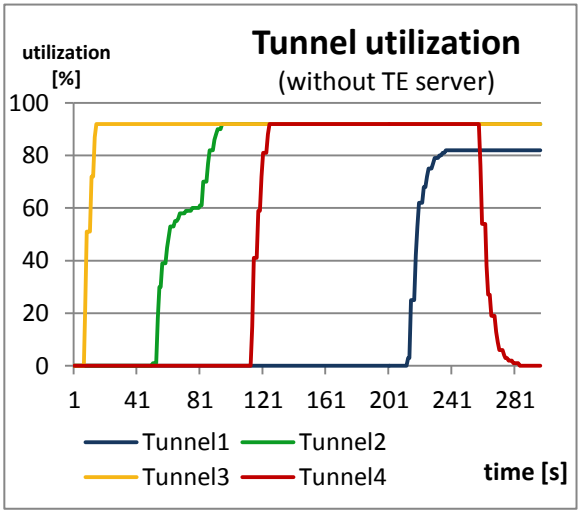

Figure 6.19 – Experiment 2, Utilization of tunnels, without TE server

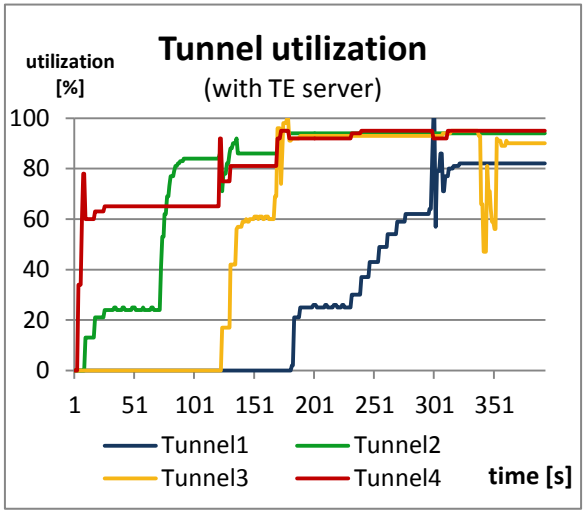

Figure 6.20 – Experiment 2, Utilization of tunnels, with TE server

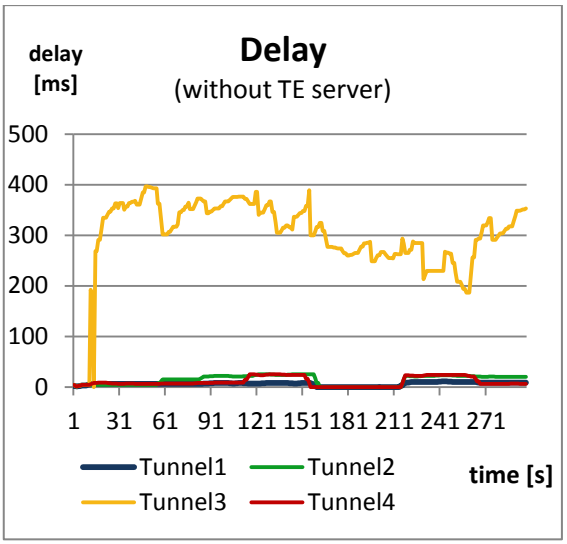

Figure 6.21 - Experiment 2, Delay, without TE server

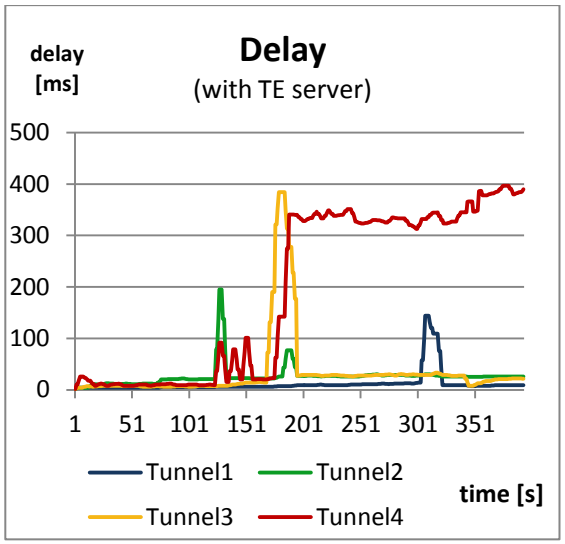

Figure 6.22 - Experiment 2, Delay, with TE server

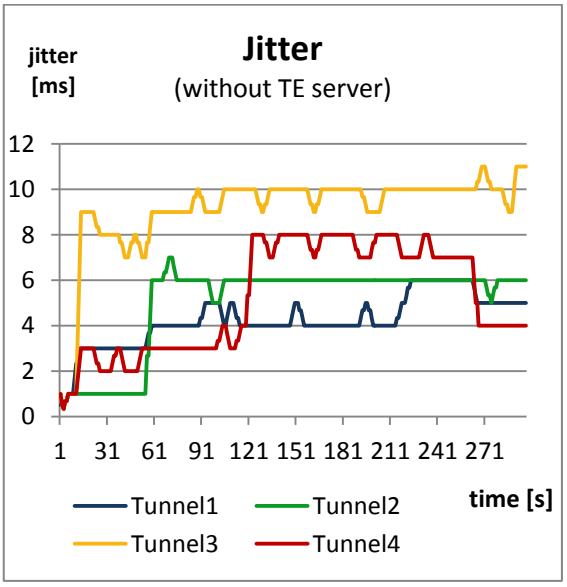

Figure 6.23 - Experiment 3, Jitter, without TE server

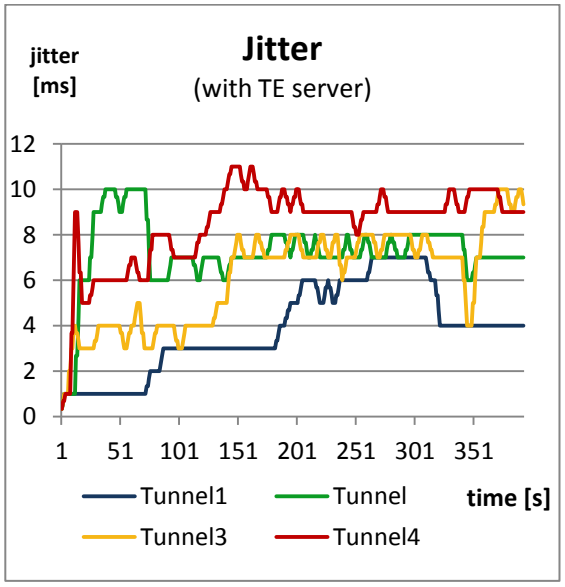

Figure 6.24 - Experiment 3, Jitter, with TE server

6.3 Experiment 3

6.3.1 Testing scenario

The third experiment shows that the TE server is capable of optimizing the utilization of all tunnels together with ensuring required QoS for Class1 traffic. The traffic was generated as follows as shown in Figure 6.25 and Figure 6.26:

- Class1 – 1160 kbps
- Class4 – 615 kbps
- Class3 – 412 kbps
- Class2 – 421 kbps
- Class1 – 110 kbps

The traffic of Class1 and Class4 exceeds the guarantees (700 kbps for Class1 and 460 kbps for Class4). The traffic is therefore limited according to the LSP as shown in Figure 6.27. The packet loss reaches 45% for Class1 and 25% for Class4 as shown in Figure 6.29. Figure 6.31 shows that the utilization of tunnels is not optimized since Tunnel2 has 65% of unused bandwidth while all other tunnels are utilized over 80%.

The unequal utilization of tunnels and not enough bandwidth for Class1 traffic are reflected also on the values of delay on Tunnel1 as shown in Figure 6.33. The values of delay reach almost 200 ms which is not acceptable for real-time traffic such as VoIP.

When the TE server is used, all traffic entering the PE router is able to traverse the network since all four tunnels are used as shown in Figure 6.28. Only occasional packet loss is present due to the process of optimizing and re-routing the traffic as show in Figure 6.30. The utilization of all tunnels is shown in Figure 6.32. The Figure 6.34 shows the values of delay, which are in this case minimized. The values of jitter were comparable in both cases and did not exceed the threshold as shown in Figure 6.35 and Figure 6.36.

6.3.2 Evaluation

In this experiment traffic of Class1 and Class4 is generated high above its guarantees. This leads to high packet loss when the TE server is not used. The packet loss together with unequal utilization of resources in the network causes that delay of Class1 traffic

exceeds its threshold of 150 ms. We consider this state as a violation of required QoS guarantees.

When the TE server is used, the throughput of each class is maximized and therefore the packet loss of each class is minimized or completely eliminated. With the same testing environment resulting values of delay are significantly better since they did not exceed the threshold.

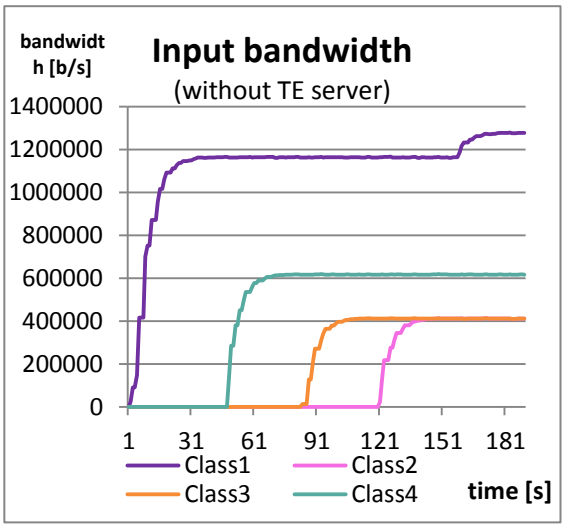

Figure 6.25 - Experiment 3, Input bandwidth, without TE server

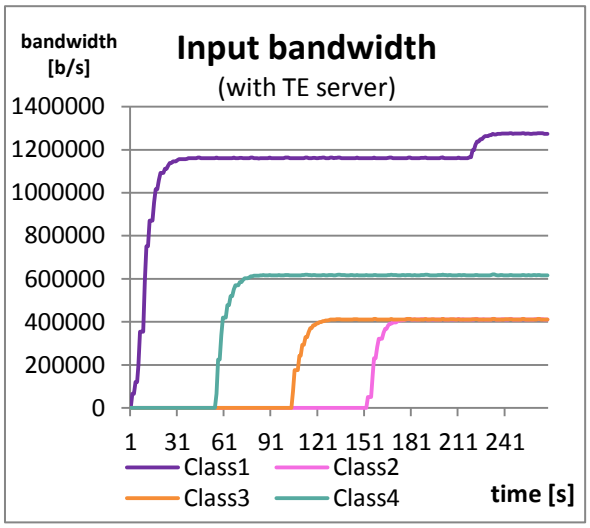

Figure 6.26 - Experiment 3, Input bandwidth, with TE server

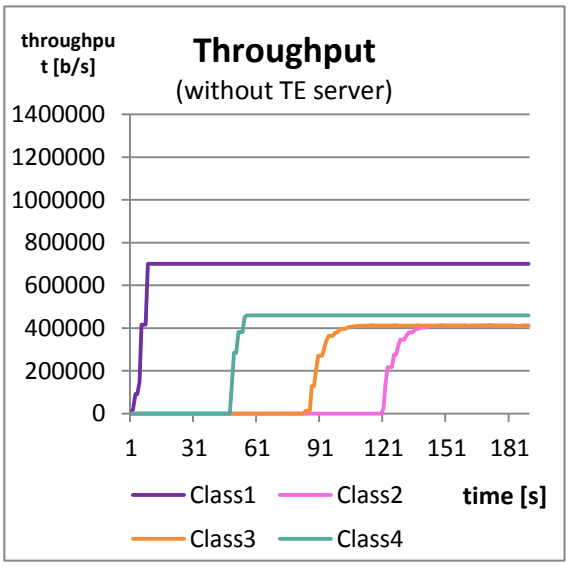

Figure 6.27 - Experiment 3, Throughput, without TE server

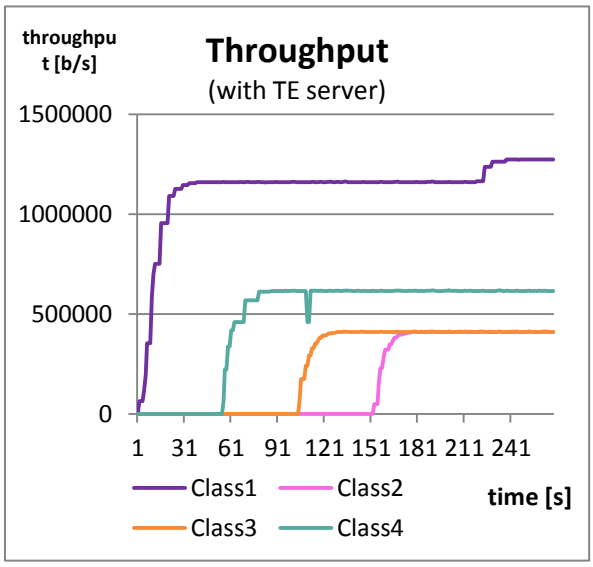

Figure 6.28 - Experiment 3, Throughput, with TE server

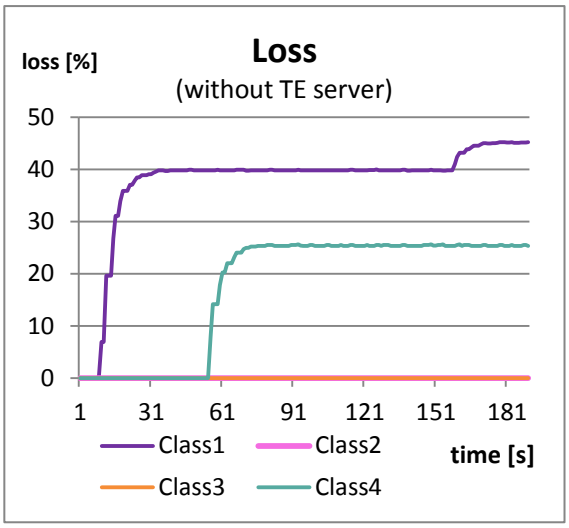

Figure 6.29 - Experiment 3, Loss, without TE server

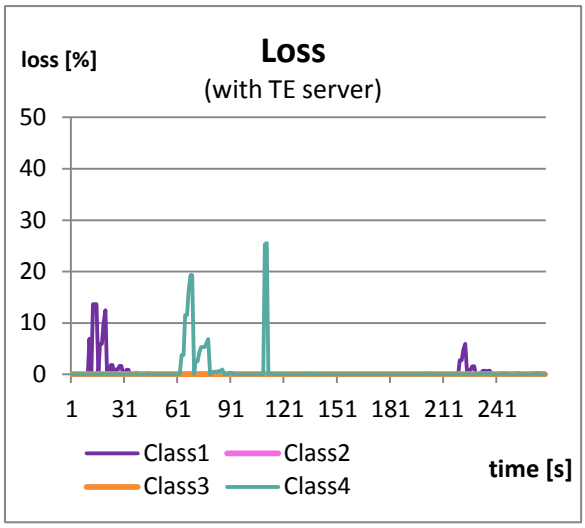

Figure 6.30 - Experiment 3, Loss, with TE server

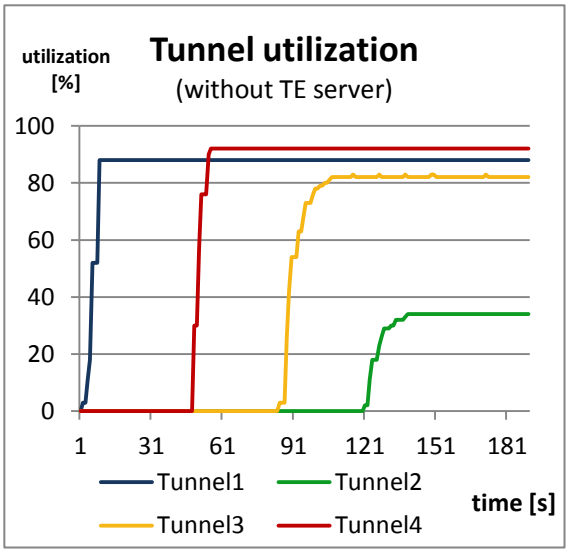

Figure 6.31 - Experiment 3, Utilization of tunnels, without TE server

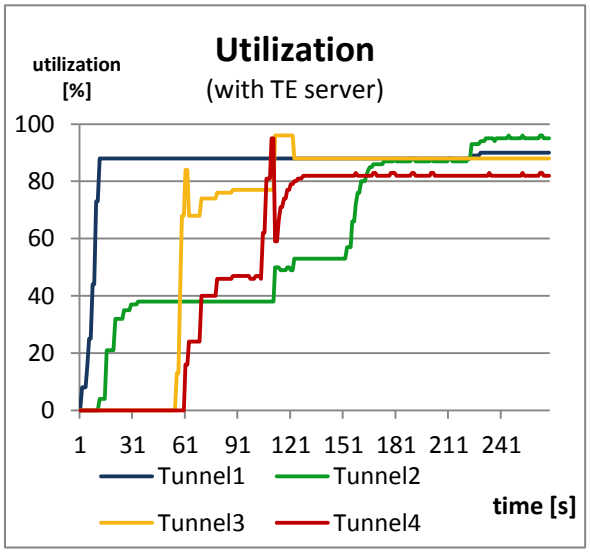

Figure 6.32 - Experiment 3, Utilization of tunnels, with TE server

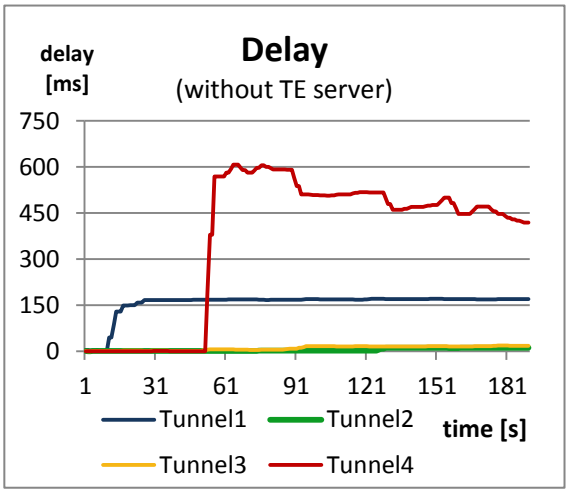

Figure 6.33 - Experiment 3, Delay, without TE server

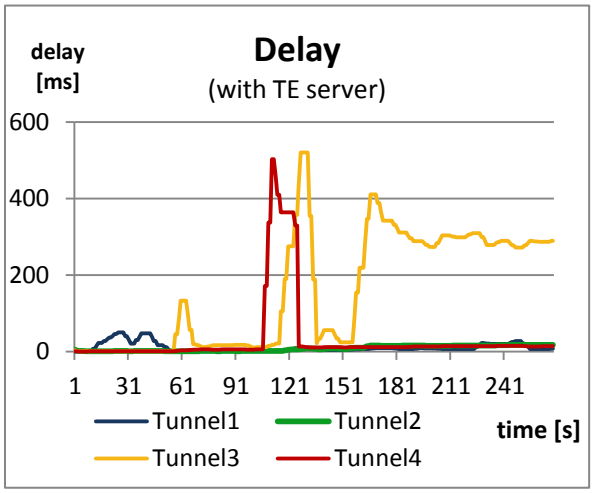

Figure 6.34 - Experiment 3, Delay, with TE server

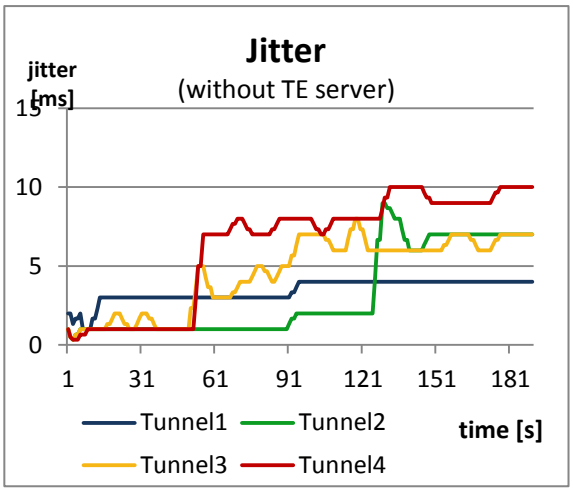

Figure 6.35 - Experiment 3, Jitter, without TE server

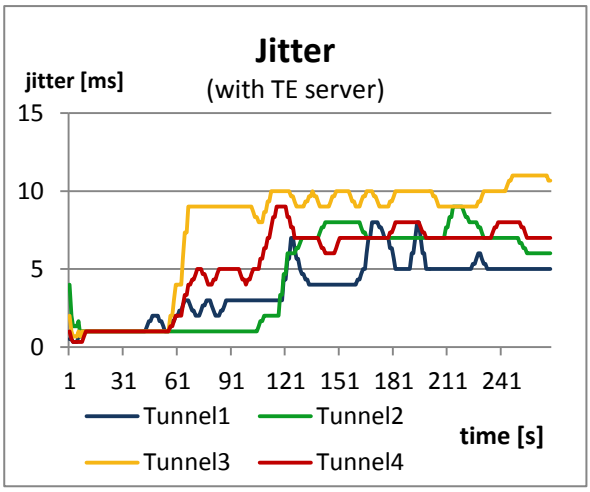

Figure 6.36 - Experiment 3, Jitter, with TE server

6.4 Experiment 4

6.4.1 Testing scenario

In the fourth experiment, the traffic was generated as follows:

- Class3 – 200 kbps
- Class1 – 500 kbps
- Class2 – 770 kbps
- Class3 – 300 kbps
- Class1 – 240 kbps
- Class3 – 120 kbps

The traffic of Class1 and Class3 was generated above their guaranties. Traffic of Class4 was intentionally not generated. The values of input bandwidth required are shown in Figure 6.37 and Figure 6.38.

Each class is limited by its guarantees when the TE server is not used as shown in Figure 6.39. This leads to constant packet loss of Class3 and Class1 as shown in Figure 6.41. One of the tunnels is not used since no traffic of Class4 was generated. This causes unequal utilization of tunnels as shown in Figure 6.43. The values of delay for Class1 traffic are slightly higher (up to 80 ms) as shown in Figure 6.45 but we do not consider this as a violation of QoS guarantees.

Optimal utilization of tunnels with the use of the TE server shown in Figure 6.44 leads to elimination of packet loss for each class as shown in Figure 6.42. The throughput of each class is therefore higher as shown in Figure 6.40. The occasional packet loss is caused by the reaction time of the server. Lower packet loss and equal distribution of traffic leads to lower values of delay for Class1 as shown in Figure 6.46.

6.4.2 Evaluation

In this experiment we showed that even small packet loss together with unequal utilization of tunnels leads to slightly decreased QoS. Since one of the tunnels is not used even when it has free bandwidth, the packet loss of Class3 reaches up to 25%.

With the use of TE server all tunnels are optimally used and the traffic is equally distributed. As a result, packet loss is eliminated and QoS are preserved.

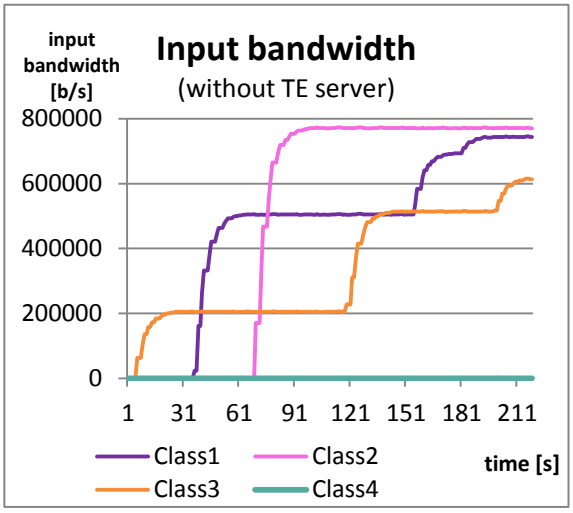

Figure 6.37 – Experiment 4, Input bandwidth, without TE server

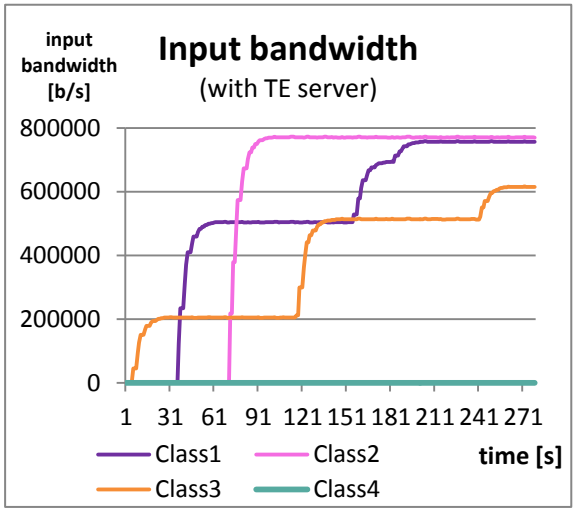

Figure 6.38 - Experiment 4, Input bandwidth, with TE server

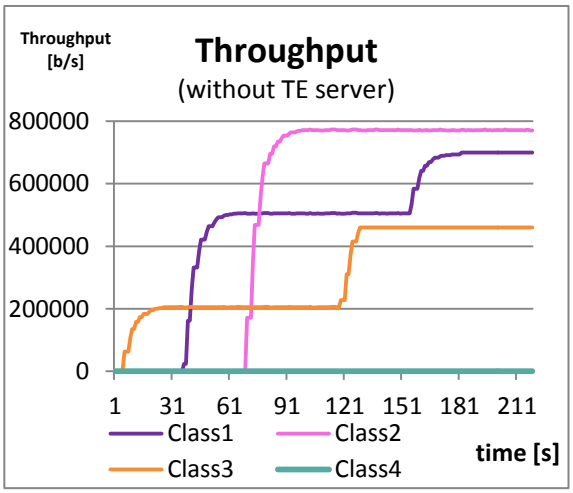

Figure 6.39 - Experiment 4, Throughput, without TE server

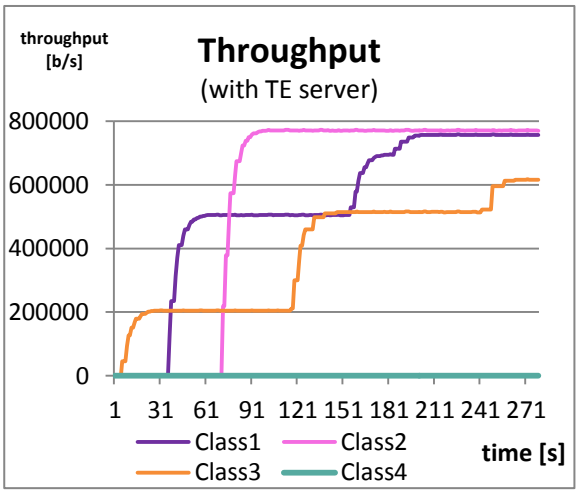

Figure 6.40 - Experiment 4, Throughput, with TE server

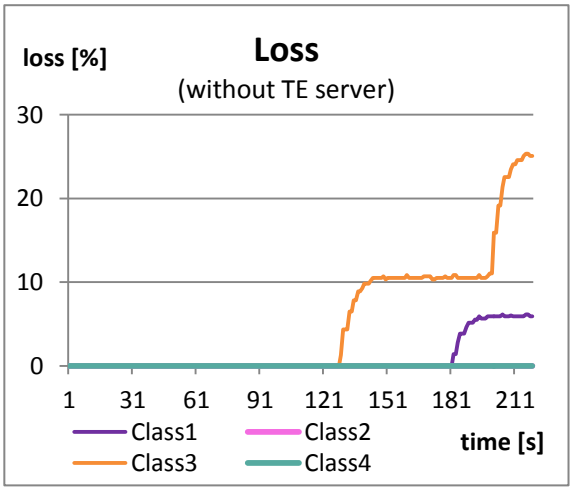

Figure 6.41 - Experiment 4, Loss, without TE server

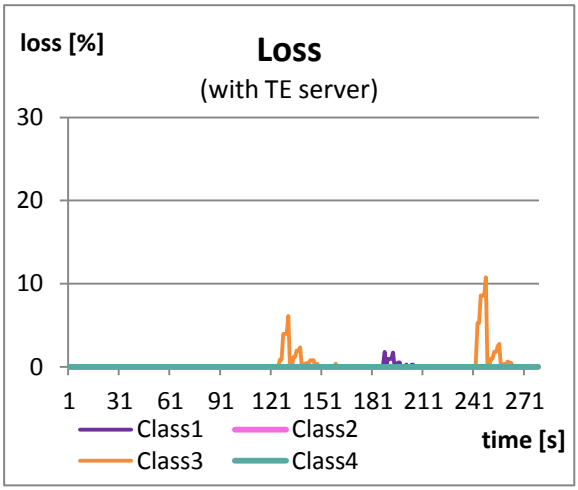

Figure 6.42 - Experiment 4, Loss, with TE server

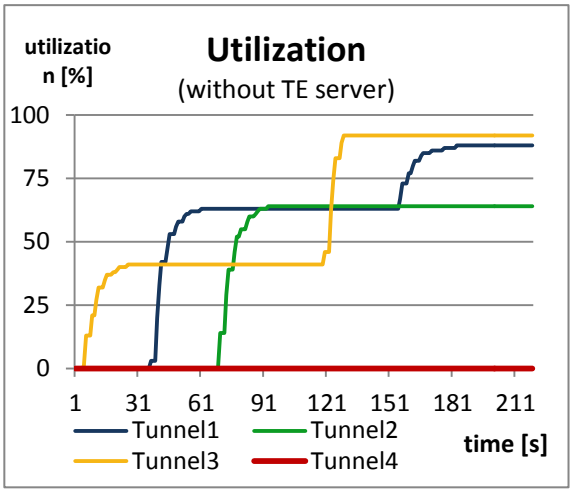

Figure 6.43 - Experiment 4, Utilization of tunnels, without TE server

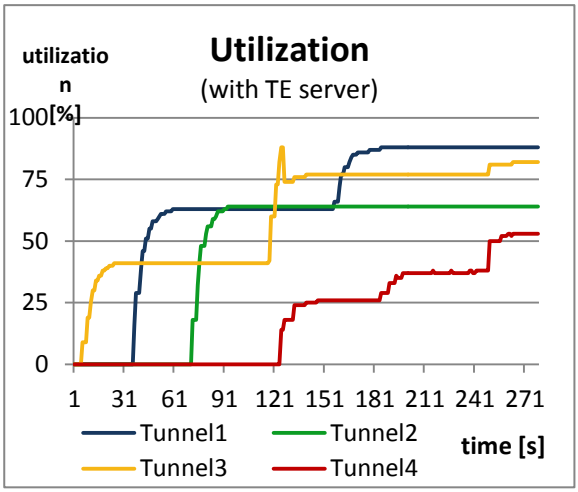

Figure 6.44 - Experiment 4, Utilization of tunnels, with TE server

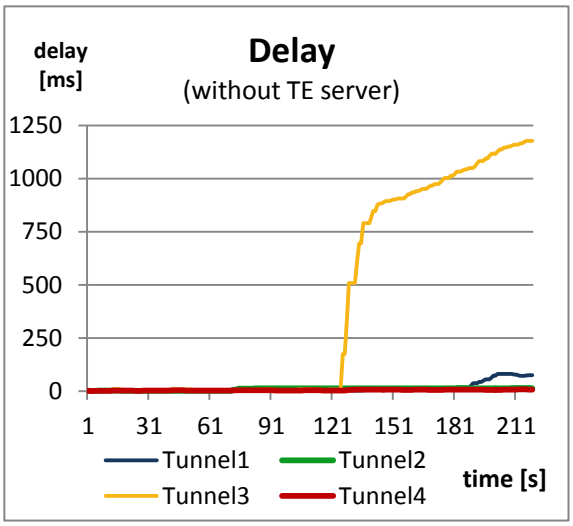

Figure 6.45 - Experiment 4, Delay, without TE server

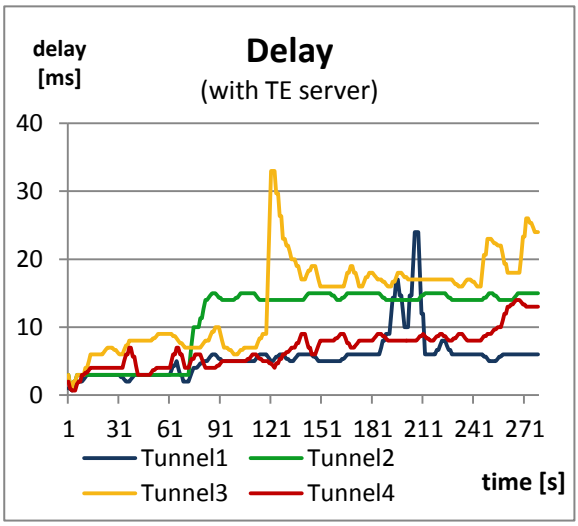

Figure 6.46 - Experiment 4, Delay, with TE server

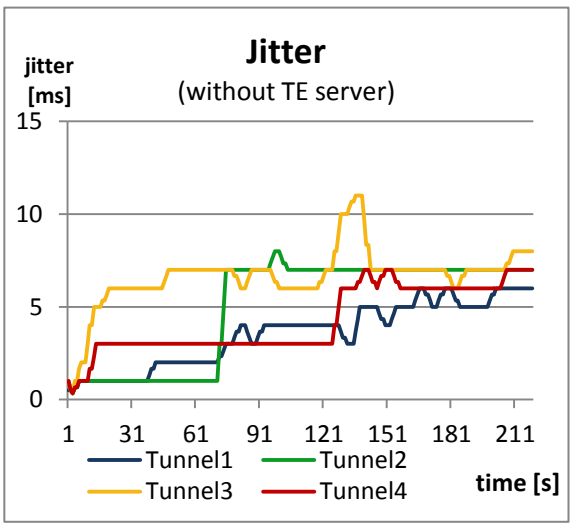

Figure 6.47 - Experiment 4, Jitter, without TE server

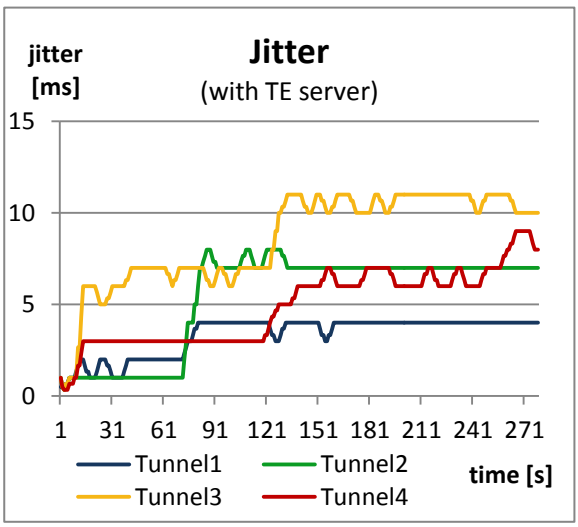

Figure 6.48 - Experiment 4, Jitter, with TE server

6.5 Summary

This part of the work describes the testing scenarios and results of the testing process. Each experiment was performed using the same testing topology and conditions. Each experiment was performed with and without the use of proposed TE server and each of these versions were repeated three times to ensure the objectivity of results.

Each experiment proved that the use of proposed TE server helps to equally utilize all tunnels in the network and therefore maximizes the throughput of each traffic class and minimizes the packet loss. This leads to preserved QoS guarantees and optimal distribution of traffic.

The second experiment showed that when utilizing the tunnels the priorities of each class are taken into account. This approach ensures the protection against class-based starvation, where one traffic class would use all network resources.

The third experiment proved that with optimal distribution of traffic and reduced packet loss QoS guarantees are preserved compared to the same scenario without the TE server.

7 Conclusion

This work describes the principles of traffic engineering, the classification of TE and details of each TE type. The building blocks of traffic engineering, the extensions for IGP, TE tunnels and the signaling used are described. Basic description of QoS toolset is provided with possibilities to measure important parameters of the network.

The next section of this work covers the MPLS architecture, its history and building blocks which are described in detail. The implementation of MPLS VPN is analyzed.

The last section of analysis is dedicated to different algorithms used for MPLS TE. Basic approaches together with complex solutions for path selection in MPLS network are described. Different proposals dealing with the selection of the optimal path for MPLS LSP are analyzed and compared.

Next part of this work covers the proposal of an online TE server. Since the analysis showed that number of different algorithms for path selection were already proposed we focused on optimal distribution of traffic among multiple LSPs. System requirements, details of the proposal and proposed testing topologies are described in detail.

The implementation covers all details of network configuration used in testing environment. The communication of the server and the network and functions of all components are described.

The last part of this work covers the experiments performed to prove the effectiveness of the proposed server. Four different scenarios were proposed and performed. The results were collected, analyzed and evaluated. Based on the experiments performed the TE server maximizes the throughput, optimizes the traffic flows and achieves optimal utilization of all tunnels in the network while preserving QoS.

8 Resumè

Internet v súčasnosti predstavuje obrovský multifunkčný nástroj na prepojenie, komunikáciu, vzdelávanie, zábavu, zdieľanie a množstvo ďalších aktivít. Je založený na hierarchickom modeli, čo mu umožňuje neustále rásť. Na každej úrovni hierarchického modelu sú využívané rôzne prístupy a technológie a spravujú ich rôzni sieťoví operátori.

Pre koncového používateľa, ktorý chce využívať možnosti internetu je jeho štruktúra a fungovanie nepodstatné. Dôležitou vlastnosťou je funkčnosť a kvalita, ktorú používateľ vníma. Na zabezpečenie a zlepšenie tejto kvality boli navrhnuté a vyvinuté rôzne mechanizmy a prístupy, ktoré sú implementované na rôznych miestach v sieti.

Táto práca sa zameriava na využitie architektúry MPLS v sieti prevádzkovateľa služieb a možnosti zabezpečenia kvality služby (QoS). Riadenie premávky (Traffic Engineering - TE) je využívané na zabezpečenie QoS a optimalizáciu využitie sieťových zdrojov.

V prvej časti práce je analyzovaný princíp riadenia premávky v sieťach. Existujú rôzne prístupy riadenia premávky, ktoré sú kategorizované na základe ich aplikácie. Každý z prístupov je stručne charakterizovaný. Ďalej sú v práci analyzované jednotlivé kroky a funkcie vykonávané pri použití riadenia premávky. Opísané sú existujúce rozšírenia pre protokoly OSPF a IS-IS, charakteristika TE tunela a využitie protokolu RSVP na jeho signalizáciu. Záver prvej časti práce sa zaoberá poskytovaním kvality služieb. Opísané sú operácie nutné pre použitie prístupu DiffServ – klasifikácia a značkovanie paketov, policing a shaping, prístupy používané na vyhýbanie sa zahlteniu, techniky výstupných radov (queueuing). Taktiež sú tu opísané možnosti meranie výkonnostných parametrov siete s použitím technológie IP SLA a NetFlow.

Druhá časť práce obsahuje analýzu architektúry MPLS, jej história, stavebné prvky, používanie značiek na vytváranie LSP a distribúcia značiek použitím protokolu LDP. Implementácia MPLS VPN je detailne analyzovaná – použitie VRF, Route Target, Route Distinguisher.

V tretej časti práce sú opísané existujúce návrhy a implementácie rôznych prístupov riešenia riadenia premávky. Základné algoritmy ako Min-hop Algorithm (MHA), Widest-shortest Path (WSP) a Shortest-Widest Path (SWP) sú stručne opísané spolu s komplikovanejšími algoritmami ako napr. MIRA, DORA a PBR. Komplexné

riešenia riadenia premávky s použitím servera sú opísané spolu so základným prehľadom ich funkcionalít a výsledkov uverejnených v rôznych odborných publikáciách.

Ďalšia časť práce sa zaoberá návrhom servera na riadenie premávky v MPLS sieti. Sú tu definované systémové požiadavky servera a predpoklady, ktoré musia byť pre úspešné fungovanie servera splnené. Navrhnutý server pozostáva z nasledujúcich operácií:

- Analýza siete a existujúcich LSP
- Meranie výkonnostných parametrov na LSP
- Výpočet ceny každého LSP
- Priradenie tokov premávky na vhodné LSP
- Optimalizácia rozloženia premávky

Pre správne fungovanie servera musí byť premávka klasifikovaná do štyroch tried. Prvá trieda (Class1) predstavujú premávku a aplikácie v reálnom čase, ktoré majú požiadavky na oneskorenie, variáciu oneskorenia a stratovosť. Ostatné tri triedy predstavujú dátovú premávku s rôznymi nárokmi na šírku pásma. Cieľom je rozložiť premávku všetkých tried v sieti tak, aby trieda Class1 mala zabezpečené požadované parametre a aby ostatné triedy mali maximum použiteľnej šírky pásma.

Predpokladom sú vytvorené LSP v sieti, o ktorých sa server naučí počas analýzy siete. V ďalšom kroku server nastaví meranie výkonnostných parametrov na každom LSP použitím IP SLA operácie. Na základe meraných hodnôt a aktuálne využívanej šírky pásma na každom LSP sú počítané hodnoty ceny tunelu (cost). Definované boli dve hodnoty ceny tunelu – jedna vyjadruje vhodnosť tunela pre Class1 a druhá vyjadruje použiteľnú šírku pásma tunelu. Hodnoty sú počítané na základe funkcií (4) a (5). Na výpočet ceny tunelu pre Class1 boli použité navrhnuté referenčné tabuľky, ktoré vyjadrujú aktuálny stav tunelu vzhľadom na hodnoty oneskorenia, variácie oneskorenia a stratovosti. Referenčné tabuľky sú zobrazené v tabuľkách Table 5.2, Table 5.3,Table 5.4 a Table 5.5.

Priraďovanie premávky na LSP je riadené algoritmom, ktorého zjednodušený diagram je zobrazený na obrázku Figure 4.2. Základným princípom je zabezpečenie garantovanej šírky pásma pre každú triedu a QoS parametrov pre triedu Class1. Keďže každá trieda nemusí vždy využívať celé garantované pásmo, môže byť táto nevyužitá šírka pásma použitá inou triedou. Pri priraďovaní sa berie do úvahy priorita danej

triedy. Výsledným efektom je maximálna priepustnosť pre všetky triedy premávky, pričom trieda Class1 má zabezpečené QoS požiadavky.

Proces optimalizácie toku dát je riadený podľa diagramu na obrázku Figure 4.4. Hlavným cieľom je dosiahnutie optimálneho rozloženia premávky v sieti s použitím všetkých LSP. Tento proces je aktivovaný, ak niektoré LSP má nevyužitú šírku pásma nižšiu ako je polovica priemernej nevyužitej šírky pásma na všetkých LSP. Matematické vyjadrenie tohto vzťahu je zobrazené v (6).

Navrhnutý TE server bol implementovaný v jazyku C#. Funkcie, ktoré vykonáva, môžu byť rozdelené do dvoch skupín:

- Jednorazové funkcie (vykonané iba pri spustení servera):
    - analýza topológie
    - analýza LSP
    - nastavenie garancií pre každú triedu
- Periodické funkcie:
    - meranie výkonnostných parametrov na každom LSP
    - výpočet ceny LSP
    - priradenie premávky na LSP
    - výpočet aktuálne využívanej šírky pásma na každom LSP
    - optimalizácia toku dát

Navrhnutý server bude komunikovať s okrajovým PE smerovačom pomocou protokolu SSHv2 a SNMPv3. SSH bude použité na konfiguráciu smerovača a aplikáciu zmien na základe výpočtov. SNMP bude použité na získanie výsledkov meraní IP SLA, ktoré sú použité na výpočet ceny cesty.

Implementácia testovacej topológie na overenie funkčnosti servera pozostávala z nasledovných krokov:

- Konfigurácia topológie zobrazenej na obrázku Figure5.1 – základná IP konektivita, MPLS, tunely
- Konfigurácia SNMP servera na PE1
- Konfigurácia SSHv2 na PE1 a PE2
- Klasifikácia premávky na CE1

Softvérový návrh servera je zobrazený na obrázku Figure 4.13. Pozostáva z niekoľkých tried, pričom všetky komunikujú s externou databázou, v ktorej sú uložené všetky

potrebné informácie. Návrh databázy je zobrazený na obrázku Figure 4.14. Každý komponent je realizovaný jednou triedou a každá obsahuje funkcie pre svoju činnosť:

- Databáza je použitá na volanie troch základných funkcií servera – funkcia na priradenie premávky na LSP, funkcia optimalizácie pre dáta a funkcia optimalizácie pre Class1.
- Komponent Network analyzer slúži na vykonanie prvej analýzy siete a zistenie všetkých potrebných informácií o sieti a LSP tuneloch. Získané informácie uloží do databázy.
- Komponent Measurement Engine konfiguruje operácie IP SLA pre každý z tunelov a potom periodicky zaznamenáva namerané hodnoty získané cez SNMP do databázy.
- Komponent Traffic Handler obsahuje tri základné funkcie, ktoré sú volané z databázy - funkcia na priradenie premávky na LSP, funkcia optimalizácie pre dáta a funkcia optimalizácie pre Class1. Na základe aktuálnych informácií v databáze tento komponent nastavuje hodnoty CIR a PIR na vstupnom rozhraní pre jednotlivé triedy a shaping na výstupných rozhraniach.
- Komponent Calculator obsahuje všetky funkcie na výpočet ceny cesty, aktuálne využitých zdrojov v sieti, hodnoty priemerne voľnej šírky pásma na tuneloch.

Funkcionalita servera bola overovaná na testovacej topológii zobrazenej na obrázku Figure 5.1. Testovanie prebiehalo generovaním premávky rôznych tried v rôznych množstvách a v rôznych kombináciách. Na vyhodnotenie výsledkov boli zaznamenávané hodnoty priepustnosti pre každú triedu, stratovosť pre každú triedu, percentuálne využitie tunelov a QoS parametre každého tunela. Výsledok bol porovnaný s rovnakým testovacím scenárom pri použití funkcie Class-based tunnel selection (CBTS).

8.1 Experiment č. 1

V prvom experimente bola generovaná premávka v nasledovnom poradí a množstve:

- Class1 – 650 kbps
- Class2 – 514 kbps
- Class2 – 514 kbps
- Class2 – 514 kbps

- Class2 – 300 kbps
- Class3 – 514 kbps

Premávka triedy Class4 bola zámerne vynechaná, za účelom preukázania optimálneho využitia všetkých tunelov. Množstvo premávky, ktorá vstupovala do siete je zobrazené na obrázkoch Figure 6.1 a Figure 6.2.

Výsledkom experimentu boli nasledovné zistenia:

- Ak nie je v sieti použitý TE server, každá trieda premávky má priradený iba jeden tunel, ktorý definuje jej maximálnu priepustnosť. Výsledná priepustnosť je zobrazená na obrázku Figure 6.3. Z toho vyplýva aj stratovosť triedy Class2 a Class3 konštantná (40% a 10%), ako je vidieť na obrázku Figure 6.5.
- Pri použití TE servera je premávka optimálne rozdelená medzi viacero tunelov, čím je efektívne zvýšená jej priepustnosť zobrazená na obrázku Figure 6.4. Výsledná priepustnosť pre triedu Class2 je takmer o 50% vyššia ako pri testovaní bez servera. Vďaka optimálnemu využitiu zdrojov v sieti je stratovosť triedy Class3 eliminovaná a stratovosť triedy Class2 je minimalizovaná (zobrazená na obrázku Figure 6.6). Chvíľková stratovosť triedy Class3 je spôsobená časom potrebným na zistenie danej situácie serverom a jeho reakciu – nájdenie vhodného tunela a presmerovanie časti premávky.
- Využitie tunelov nie je optimálne, pokiaľ v sieti nie je použitý TE server. Ako vidieť na obrázku Figure 6.7, bez použitia servera sú využívané iba tri tunely zo štyroch, čo vplýva na zvýšenú stratovosť a zníženú priepustnosť premávky.
- Pri použití TE servera sú všetky tunely v sieti optimálne využívané, ako je vidieť na obrázku Figure 6.8.
- Hodnoty oneskorenia a variácie oneskorenia boli porovnateľné v oboch experimentoch a neprekročili definované hranice.

8.2 Experiment č. 2

V druhom experimente bola premávka generovaná nasledovne:

- Class4 – 600 kbps
- Class2 – 1130 kbps
- Class3 – 514 kbps
- Class1 – 650 kbps

- Premávka triedy Class3 bola zastavená

V tomto experimente bola generovaná premávka tried Class2, Class3 a Class4 nad rámec garantovaných hodnôt, ako je zobrazené na obrázkoch Figure 6.13 a Figure 6.14. Výsledky experimentu sú nasledovné:

- Bez použitia TE servera je každá trieda limitovaná použitým tunelom, ako vidieť na obrázku Figure 6.15. Z toho vyplýva aj konštantná stratovosť každej z tried Class2, Class3 a Class4 zobrazená na obrázku Figure 6.17. Všetky tunely boli využívané až po zastavenie generovania premávky triedy Class3, kedy jeden z tunelov prestal byť používaný, ako je zobrazené na obrázku Figure 6.19. Hodnoty oneskorenia a variácie oneskorenia boli porovnateľné v oboch experimentoch a nepresiahli kritické hodnoty pre tunely určené pre premávku triedy Class1.
- S použitím TE servera v sieti je priepustnosť všetkých tried zvýšená, ako je vidieť na obrázku Figure 6.16. Stratovosť tried Class2, Class3 a Class4 je iba občasná, pričom je spôsobená reakčným časom servera. Z obrázkov Figure 6.16 a Figure 6.18 je očividné, že premávka triedy Class4 využívala tunel určený pre triedu Class1. V okamžiku, kedy začala byť generovaná premávka triedy Class1, bola priepustnosť triedy Class4 znížená na jej garantovanú hodnotu. Po zastavení generovania premávky triedy Class3 bola priepustnosť triedy Class4 opäť zvýšená, pretože premávka bola presmerovaná na tunnel Tunnel3. Hodnoty stratovosti a využitia jednotlivých tunelov na obrázku Figure 6.20 to potvrdzujú.
- Hodnoty oneskorenia a variácie oneskorenia boli porovnateľné a nepresiahli hraničné hodnoty pre premávku triedy Class1.

8.3 Experiment č. 3

V treťom experimente bola premávka generovaná nasledovne:

- Class1 – 1160 kbps
- Class4 – 615 kbps
- Class3 – 412 kbps
- Class2 – 421 kbps
- Class1 – 110 kbps

Premávka tried Class1 a Class4 presahuje garantované hodnoty (700 kbps pre Class1 a 460 kbps pre Class4). Výsledky experimentu sú nasledovné:

- Premávka všetkých tried je obmedzená použitým tunelom, podobne ako pri predchádzajúcich experimentoch. Priepustnosť jednotlivých tried je zobrazená na obrázku Figure 6.27. V závislosti od priepustnosti jednotlivých tried dosiahla ich stratovosť až 45% pre triedu Class1 a 25% pre triedu Class4. Presné hodnoty sú zobrazené na obrázku Figure 6.29. Obrázok Figure 6.31 jednoznačne ukazuje, že využitie tunelov nebolo optimálne, keďže tunel Tunnel2 je využívaný iba na 35%, zatiaľ čo ostatné tunely sú využité nad 80%.
- S použitím TE servera je priepustnosť premávky všetkých tried maximalizovaná, ako vidieť na obrázku Figure 6.28. Občasná stratovosť, zobrazená na obrázku Figure 6.30 je spôsobená časom reakcie servera na vzniknutú situáciu a presmerovaním premávky. Využitie tunelov je optimálne, ako vidieť na obrázku Figure 6.32.
- Vysoká stratovosť triedy Class1 a neoptimálne využitie zdrojov v sieti spôsobilo, že oneskorenie pre triedu Class1 presiahlo hraničnú hodnotu 150 ms. Presné hodnoty oneskorenia sú zobrazené na obrázku Figure 6.33. Z toho dôvodu považujeme QoS pre triedu Class1 za rapídne zhoršený.
- Pri použití TE servera je oneskorenie pre triedu Class1 minimalizované, ako vidieť na obrázku Figure 6.34.

8.4 Experiment č. 4

Vo štvrtom experimente bola premávka generovaná nasledovne:

- Class3 – 200 kbps
- Class1 – 500 kbps
- Class2 – 770 kbps
- Class3 – 300 kbps
- Class1 – 240 kbps
- Class3 – 120 kbps

Premávka triedy Class1 a Class3 bola generovaná nad rámec ich garancií. Premávka triedy Class4 nebola zámerne generovaná. Presné hodnoty vstupných požiadaviek sú zobrazené na obrázku Figure 6.37 a Figure 6.38.

Bez použitia TE servera je každá trieda limitovaná svojimi garanciami, ako vidieť na obrázku Figure 6.39. To vedie k zvýšenej stratovosti zobrazenej na obrázku

Figure 6.41. Využitie tunelov zobrazené na obrázku Figure 6.43 taktiež nie je optimálne, keďže jeden z tunelov nie je vôbec použitý. Hodnoty oneskorenia zobrazené na obrázku Figure 6.45 pre triedu Class1 sú mierne vyššie, avšak neprekračujú hraničnú hodnotu 150 ms.

S použitím navrhnutého TE servera je priepusnosť všetkých tried maximalizovaná, ako vidieť na obrázku Figure 6.40. Stratovosť je iba občasná, z dôvodu oneskorenej reakcie servera. Na obrázku Figure 6.44 je zobrazené využitie všetkých tunelov v sieti, ktoré zabezpečuje maximálnu priepustnosť pre všetky triedy premávky. Hodnoty oneskorenia a variácie oneskorenia zobrazené na obrázku Figure 6.46 a Figure 6.48 dokazujú zachované QoS požiadavky pre triedu Class1.

References

[1] Wang, N., Ho, K. H., Pavlou, G., Howarth, M.: *An Overview of Routing Optimization for Internet Traffic Engineering*, IEEE Communication Surveys. In: The Electronic Magazine of Original Peer-Reviewed Survey Articles, 2008, 21 p.

[2] *IP-MPLS-Traffic-Engineering*, White Paper, Cariden Technologies, Inc., May 2012

[3] De Ghein, L.: *MPLS Fundamentals*. Cisco Press, 2006, 672 p. ISBN 978-1587051128

[4] Berger, L., et. al.: *The OSPF Opaque LSA Option*, RFC 5250, July 2008

[5] Awduche, D., et al.: *RSVP-TE: Extensions to RSVP for LSP Tunnels,* RFC 3209, December 2001

[6] *Advanced Topics in MPLS-TE Deplyment*, Cisco Systems, White Paper, 2009

[7] HANCOCK, R.; et al.: *Next Steps in Signaling (NSIS): Framework*; RFC 4080, June 2005.

[8] *Diffserv – The Scalable End-to-end quality of service model*, Cisco Systems, White Paper, August 2005

[9] *IP Service Level Agreement (IP SLA)*, Cisco Systems, White Paper, September 2007

[10] *Introduction to Cisco IOS NetFlow*, Cisco Systems, White Paper, May 2012

[11] Grosmann, D. et al.: *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, RFC 2684, September 1991

[12] Fortz, B. et al.: *Traffic Engineering with Traditional IP Routing Protocols*, IEEE Commun. Mag., vol. 40, no. 10, Oct. 2002

[13] Trimintzios, P. et al.: *A Management and Contral Architecture for Providing IP Differentiated Services in MPLS-Based Networks*, IEEE Commun. Mag., vol. 39, no. 5, May 2001

[14] Kandula, S. et al.: *Walking the Tightrope: Responsive Yet Stable Traffic Engineering*, ACM SGICOMM Comp. Commun. Review, vol. 35, no. 4, Oct. 2005

[15] Yang, Y. R. et al.: On *Route Selection for Interdomain Traffic Engineering*, IEEE Network, vol. 19, no. 6, Nov./Dec. 2005

[16] Wang, H. et al.: *COPE: Traffic Engineering in Dynamic Networks*, Proc. ACM SIGCOMM, 2006.

[17] Ho, K. H. et al.: *Joint Optimization of Intra- and Inter-Autonomous System Traffic Engineering*, Proc. IEEE/IFIP NOMS, 2006

[18] Bagula, A.: *Hybrid Routing in Next Generation IP Networks*, Comp. Commun., vol. 29, no. 7, Apr. 2006

[19] Andersson, L. et al.: *The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols*, RFC 3468, February 2003

[20] *Congestion Avoidance Overview*, Cisco IOS Quality of Service Solutions Configuration Guide

[21] *Cisco IOS Quality of Service*, Cisco Systems, White Paper, September 2008

[22] Osborne, E., Simha, A.: *Traffic Engineering with MPLS*, Cisco Press, July 2002, 608 p. ISBN 1-58705-031-5

[23] *Cisco Express Forwarding Overview,* Cisco Systems, Cisco IOS Switching Services Configuration Guide

[24] *Layer 3 MPLS VPN Enterprise Consumer Guide Version 2,* Cisco Systems, 2006

[25] Kulkarni, S. et al.: *Recent Research and Comparison of QoS Routing Algorithms for MPLS Networks*, International Journal of Advanced Computer Research, vol. 2, March 2012

[26] Capone, A. et al: *Dynamic online QoS routing schemes: Performance and bounds*, Computer Networks, Elsevier, 2005

[27] Boutaba, R. et al.: *DORA: Efficient Routing for MPLS Traffic Engineering*, Journal of Network and Systems Management, vol. 10, no. 3, September 2002

[28] Suri, S. et al.: *Profile-based routing and traffic engineering*, Computer Communications 26, Elsevier, May 2002

[29] Aukia, P. et al.: *RATES: A Server for MPLS Traffic Engineering,* IEEE Network, vol. 14, no. 2, April 2000

[30] Hsu, W. et al.: *Multiple path selection algorithm for DiffServ-aware MPLS traffic engineering*, Computer Communications, Elsevier, 2010

[31] Kulkarni, S. et al.: *New QOS Routing Algorithm for MPLS Networks Using Delay and Bandwidth Constraints*, ICT Journal, vol. 2, no. 3, March 2012

[32] Li, F., Chen, J.: *MPLS Traffic Engineering Load Balance Algorithm Using Deviation Path*, International Conference on Computer Science and Service System, 2012

[33] Shi, T., Mohan, G.: *An efficient traffic engineering approach based on flow distribution and splitting in MPLS networks,* Computer Communications 29, Elsevier, 2006

[34] Giacalone, S.; et. al.: *OSPF Traffic Engineering (TE) Metric Extensions*, Internet draft, June 2013

[35] *Cisco IOS IP SLAs Configuration Guide*, Release 12.4, Cisco Systems, Inc. 2008

Appendix A

Contents of the electronic medium.

The electronic medium attached to this work includes these folders:

\Documentation

\Cofiguration files

\P1.txt

\P2.txt

\P3.txt

\P4.txt

\P5.txt

\PE1.txt

\PE2.txt

\CE1.txt

\CE2.txt

\SW1.txt

\SW2.txt

\TE server

\TE-server

\TE server.sln

Appendix B

There are several prerequisites for a successful usage of the implemented TE server :

- Correct version of the .NET Framework installed
- Libraries used in the project:
  - System.Data.SQLite.dll
  - Tamir.SharpSSH.dll
  - SnmpSharpNet.dll

After the start of the application, the TE server requires the configuration of several parameters necessary for its operation. As shown in Figure 0.1 these include:

- IP address of the directly connected PE router
- Username and password used for the SSH connection

The TE server supports following commands:

- analyze network – initial network analysis
- log – information about input data rates and configuration changes
- unlog – stops logging
- debug – more detail information about operations in the server
- undebug – stops debugging
- start – start the TE server
- stop – stop the TE server
- reconnect – create new SSH connection
- ? – show help

Figure 0.1 – Initial settings

If the initial settings are not required, default setting will be used. In the next step the TE server tries to connect to the router. It is possible to change the connection settings if the connection is not successful as shown in Figure 0.2. If connection is successful the user is notified as shown in Figure 0.3.

In the next step the analysis of the network is necessary. This is done by typing the command “analyze network” as shown in Figure 0.4. The TE server uses it’s SSH connection to get all information about the network from the PE router.

User can use the “?” to get help from the server as shown in Figure 0.5. In the next step the user can start logging to get response from the server during its operation. This is done by typing the command “log” as shown in Figure 0.6.

To actually start the server functionalities, the user has to type the command “start”. After this command is typed, the TE server configures the PE router and starts all necessary processes. Then it starts monitoring the input data rates of each class. This process is shown in Figure 0.7.

When the input data rate of some class exceeds its guarantees, the TE server reacts by re-configuring the policing on the PE router. This process can be shown in Figure H. It is obvious that the input data rate of Class2 is rising and therefore the values of CIR and PIR are re-configured.

When the input data rate of some class exceeds its guarantees, the TE server reacts by re-configuring the policing on the PE router. This process can be shown in Figure 0.8. It is obvious that the input data rate of Class2 is rising and therefore the values of CIR and PIR are re-configured.

Re-configuration can occur also in the process of optimization, which is triggered by the utilization level of one of the tunnels. An example of such situation is shown in Figure 0.9.

The TE server can be used as a monitoring system with periodic information about actual input data rates and information about all re-configrations done in the network. To stop the server, the user has to type the command “stop”.

Figure 0.2 – Connection fail

Figure 0.3 - Successful connection

Figure 0.4 - The analysis of the network

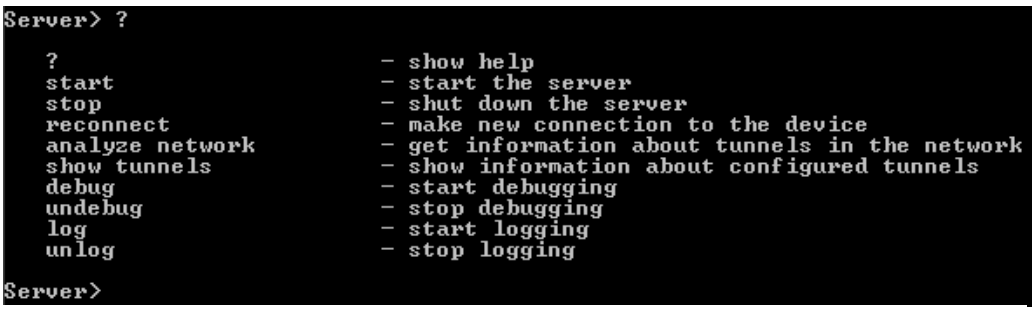

Figure 0.5 - Help

Figure 0.6 - Logging enabled

```
Server> start
Server> Starting Measurement engine (1 of 6)
Server> Configuring tail-end (2 of 6)
Server> Measuring first cost on tunnels (3 of 6)
Server> Configuring initial distribution of traffic (4 of 6)
Server> Configuring head-end (5 of 6)
Server> Starting Traffic handler (6 of 6)
Server>
```

Figure 0.7 - The start of TE server

```
Input rate:
Class 1 = 0 b/s
Class 2 = 1130000 b/s
Class 3 = 0 b/s
Class 4 = 616000 b/s


Input rate:
Class 1 = 0 b/s
Class 2 = 1132000 b/s
Class 3 = 0 b/s
Class 4 = 616000 b/s


10/05/2014 14:47:43
 Changed configuration
    Class:   ip_prec_2
    Cir:     1103938
    Pir:     1132000
    Conform: set-mpls-exp-imposition-transmit 6
    Exceed:  set-mpls-exp-imposition-transmit 1
```

Figure 0.8 - Re-configuration due to high input rate

```
Input rate:
Class 1 = 0 b/s
Class 2 = 240000 b/s
Class 3 = 0 b/s
Class 4 = 618000 b/s


10/05/2014 14:44:25: Optimize for tunnel Tunnel4


10/05/2014 14:44:26
 Changed configuration
    Class:   ip_prec_4
    Cir:     337500
    Pir:     618000
    Conform: set-mpls-exp-imposition-transmit 4
    Exceed:  set-mpls-exp-imposition-transmit 2
```

Figure 0.9 - Re-configuration due to optimization