QoS-Based Optimization of Data Flow in MPLS Networks

Ivana Hucková, Martin Hrubý
Institute of Computer Systems and Networks
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava, Ilkovicova 3, 842 16 Bratislava 4, Slovakia
huckova@fiit.stuba.sk, hruby@fiit.stuba.sk

Abstract — Network resources in terms of available bandwidth are not always sufficient to provide desired quality of service, especially for real-time traffic. Network performance parameters such as delay, jitter and packet loss are significant indicators of suitability of the path for this type of traffic. To provide QoS traffic engineering methods are used. In this paper we focus on traffic engineering in MPLS networks and the question of quality of service, since it is the main reason of deploying TE. We propose an online TE server to optimize the data flow in the network and maximize the utilization of the network resources. The proposed server was implemented and experimentally tested in laboratory environment. The results prove the efficiency and optimal resource utilization in the network provided by the proposed TE server.

Keywords. Optimization, QoS, Traffic engineering, MPLS, network.

I. INTRODUCTION

Customer traffic often suffers from congestion due to the bottlenecks in the network which leads to degradation of service’s quality. Traffic engineering (TE) 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 [9, 11].

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, 12].

TE routing approaches can be classified into various categories based on different aspects: IP-based and MPLS-based TE, online and offline TE, interdomain and intradomain TE, unicast and multicast TE [9]. This work is focused on MPLS-based TE with regard to QoS.

II. RELATED WORK

There were many different approaches, mechanisms and solutions proposed and developed in the area of MPLS traffic engineering so far. Two main aspects of MPLS TE can be defined as

A. finding of suitable path in the network for the creation of Label Switched Paths (LSPs)
B. distributing the traffic among the LSPs to maximize network utilization.

Many studies and research has been done in the question of routing the LSPs [3], [7], [9]. The main advantage of these algorithms is their simplicity. They can, however create bottlenecks and lead to network under-utilization [2].

Relatively little research has been done in the area of distributing the traffic among created LSPs. Authors in [8] proposed a method based on traffic flow distribution and splitting for traffic engineering in MPLS networks. MPLS Adaptive Traffic Engineering (MATE) proposed in [5] represents another algorithm focused on distributing the traffic over multiple LSPs. Our work is focused on the traffic distribution and network utilization since this area is not being as analyzed as the routing of LSPs.

III. PROPOSED SOLUTION

In our work we proposed an online TE server to provide traffic distribution, sufficient QoS for real-time traffic and maximize network utilization. The proposed server uses LSPs created in advance (manually or using some of existing algorithms) and it will steer the traffic across the network using these LSPs. The operation of the server is divided into several steps:

1. The analysis of the network and existing LSPs
2. The end-to-end measurement of network performance parameters of LSPs
3. Calculation of the cost of each LSP
4. The assignment of the traffic to LSPs
5. Optimization (if necessary)

The connection of the server to the network was provided as shown in Figure 1. The server communicates with the Provider Edge (PE) router via SSH and SNMP connection. SSH connection is used mainly to obtain specific router configuration details and to apply changes in configuration. SNMP is used to gather measured parameters via IP SLA probes [4].
A. Classification of the traffic

Traffic entering the network is classified into four classes defined by the requirements it has. One class (Class1) is dedicated for real-time traffic with strict default requirements to achieve sufficient QoS. All other classes (Class2, Class3 and Class4) are used for non-real-time data traffic and the requirements of each class are defined only by specific bandwidth guarantees. Traffic is treated according to the class it belongs to in descending order. Each class has defined the amount of overall bandwidth it can use in the network to avoid the traffic-class starvation.

To mark each of the classes MPLS EXP bits were used. Every class has two MPLS EXP values reserved – one to mark traffic within defined guarantees and another to mark traffic above the guarantees. Using this logic the server can easily influence how the traffic is treated according to requirements.

B. Network performance parameters

It is important to periodically measure various network performance parameters, such as end-to-end delay, jitter or packet loss to provide up-to-date information about each LSP. The measurements are carried out using IP SLA probes on the edge routers. Since the values of delay, jitter and packet loss are variable in time, it is preferable to work with their statistical values to provide trustworthy values of these parameters to be used. The measured values are stored in the system’s database for further usage.

C. LSP cost

The cost of LSP is used to decide whether the LSP is suitable for specific traffic class and therefore it 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 [6]. 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.

The basic mathematical representation of cost value for Class1 is shown in Formula 1 where \( C_{\text{voice}} \) represents the actual value of the cost of LSP for Class1 traffic, \( C_{\text{delay}} \) represents the actual value of the cost of LSP according to the delay, \( C_{\text{jitter}} \) represents the actual value of the cost of LSP according to jitter, \( C_{\text{loss}} \) represents the actual value of the cost of LSP according to packet loss.

The representation of cost value for classes Class2, Class 3 and Class4 is shown in Formula 2 where \( C_{\text{data}} \) represents the actual value of the cost of LSP for Class2, Class3 and Class4 traffic and free_bw_of_LSP represents the amount of unused bandwidth of LSP.

The variables of \( C_{\text{voice}} \) and \( C_{\text{data}} \) are considered to be non-dimensional. The main advantage of this approach is the simplicity and computational modesty.

\[
C_{\text{voice}} = C_{\text{delay}} + C_{\text{jitter}} + C_{\text{loss}} \quad \text{(1)}
\]

\[
C_{\text{data}} = \text{free_bw_of_LSP} \quad \text{(2)}
\]

The \( C \)-values for delay, jitter and packet loss (\( C_{\text{delay}}, C_{\text{jitter}}, C_{\text{loss}} \)) are obtained from a reference tables shown in Table 1, Table 2 and Table 3 respectively. Reference tables were proposed to accurately express the quality requirements defined in [10]. 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 calculated by Formula 1 for Class1 traffic are defined in Table 4. 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 [4]. 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 unusable LSP for real-time traffic.

### Table 1. Reference table for values of delay

<table>
<thead>
<tr>
<th>delay[ms]</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{\text{delay}} )</td>
<td>0.00</td>
<td>0.05</td>
<td>0.10</td>
<td>0.15</td>
<td>0.20</td>
<td>0.25</td>
<td>0.30</td>
</tr>
<tr>
<td>delay[ms]</td>
<td>35</td>
<td>40</td>
<td>45</td>
<td>50</td>
<td>55</td>
<td>60</td>
<td>65</td>
</tr>
<tr>
<td>( C_{\text{delay}} )</td>
<td>0.35</td>
<td>0.40</td>
<td>0.45</td>
<td>0.50</td>
<td>0.55</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>delay[ms]</td>
<td>70</td>
<td>75</td>
<td>80</td>
<td>85</td>
<td>90</td>
<td>95</td>
<td>100</td>
</tr>
<tr>
<td>( C_{\text{delay}} )</td>
<td>0.70</td>
<td>0.75</td>
<td>0.80</td>
<td>0.85</td>
<td>0.90</td>
<td>0.95</td>
<td>1.0</td>
</tr>
<tr>
<td>delay[ms]</td>
<td>105</td>
<td>110</td>
<td>115</td>
<td>120</td>
<td>125</td>
<td>130</td>
<td>135</td>
</tr>
<tr>
<td>( C_{\text{delay}} )</td>
<td>3.10</td>
<td>3.15</td>
<td>3.20</td>
<td>3.25</td>
<td>3.30</td>
<td>3.35</td>
<td>3.50</td>
</tr>
<tr>
<td>delay[ms]</td>
<td>140</td>
<td>145</td>
<td>150</td>
<td>156</td>
<td>160</td>
<td>170</td>
<td>180</td>
</tr>
<tr>
<td>( C_{\text{delay}} )</td>
<td>3.65</td>
<td>3.75</td>
<td>4.00</td>
<td>12.10</td>
<td>12.15</td>
<td>12.20</td>
<td>12.30</td>
</tr>
<tr>
<td>delay[ms]</td>
<td>200</td>
<td>210</td>
<td>220</td>
<td>230</td>
<td>240</td>
<td>250</td>
<td>260</td>
</tr>
<tr>
<td>( C_{\text{delay}} )</td>
<td>12.40</td>
<td>12.50</td>
<td>12.60</td>
<td>12.70</td>
<td>12.80</td>
<td>13.00</td>
<td>14.00</td>
</tr>
<tr>
<td>delay[ms]</td>
<td>270</td>
<td>more</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C_{\text{delay}} )</td>
<td>40.00</td>
<td>40.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Reference table for values of jitter

<table>
<thead>
<tr>
<th>jitter [ms]</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_jitter</td>
<td>0.00</td>
<td>0.02</td>
<td>0.04</td>
<td>0.06</td>
<td>0.09</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>jitter [ms]</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>C_jitter</td>
<td>0.12</td>
<td>0.22</td>
<td>0.26</td>
<td>0.30</td>
<td>0.35</td>
<td>0.40</td>
<td>0.45</td>
</tr>
<tr>
<td>jitter [ms]</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>C_jitter</td>
<td>0.50</td>
<td>0.60</td>
<td>0.65</td>
<td>0.70</td>
<td>0.80</td>
<td>0.90</td>
<td>1.0</td>
</tr>
<tr>
<td>jitter [ms]</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>C_jitter</td>
<td>3.10</td>
<td>3.15</td>
<td>3.20</td>
<td>3.25</td>
<td>3.30</td>
<td>3.40</td>
<td>3.50</td>
</tr>
<tr>
<td>jitter [ms]</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>C_jitter</td>
<td>3.65</td>
<td>3.80</td>
<td>4.00</td>
<td>12.10</td>
<td>12.15</td>
<td>12.20</td>
<td>12.30</td>
</tr>
<tr>
<td>jitter [ms]</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>C_jitter</td>
<td>12.40</td>
<td>12.50</td>
<td>12.60</td>
<td>12.70</td>
<td>12.80</td>
<td>13.00</td>
<td>40.00</td>
</tr>
<tr>
<td>jitter [ms]</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C_jitter</td>
<td>40.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Reference table for values of packet loss

<table>
<thead>
<tr>
<th>loss [%]</th>
<th>0.00</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
<th>0.30</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_loss</td>
<td>0.00</td>
<td>0.02</td>
<td>0.04</td>
<td>0.06</td>
<td>0.10</td>
<td>0.14</td>
<td>0.18</td>
</tr>
<tr>
<td>loss [%]</td>
<td>0.35</td>
<td>0.40</td>
<td>0.45</td>
<td>0.50</td>
<td>0.55</td>
<td>0.60</td>
<td>0.65</td>
</tr>
<tr>
<td>C_loss</td>
<td>0.22</td>
<td>0.25</td>
<td>0.28</td>
<td>0.31</td>
<td>0.34</td>
<td>0.38</td>
<td>0.43</td>
</tr>
<tr>
<td>loss [%]</td>
<td>0.70</td>
<td>0.75</td>
<td>0.80</td>
<td>0.85</td>
<td>0.90</td>
<td>0.95</td>
<td>1.00</td>
</tr>
<tr>
<td>C_loss</td>
<td>0.50</td>
<td>0.55</td>
<td>0.60</td>
<td>0.65</td>
<td>0.75</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>loss [%]</td>
<td>1.05</td>
<td>1.10</td>
<td>1.15</td>
<td>1.20</td>
<td>1.25</td>
<td>1.30</td>
<td>1.35</td>
</tr>
<tr>
<td>C_loss</td>
<td>3.10</td>
<td>3.15</td>
<td>3.20</td>
<td>3.25</td>
<td>3.30</td>
<td>3.35</td>
<td>3.40</td>
</tr>
<tr>
<td>loss [%]</td>
<td>1.40</td>
<td>1.45</td>
<td>1.50</td>
<td>1.60</td>
<td>1.70</td>
<td>1.80</td>
<td>1.90</td>
</tr>
<tr>
<td>C_loss</td>
<td>3.55</td>
<td>3.70</td>
<td>4.00</td>
<td>12.10</td>
<td>12.15</td>
<td>12.20</td>
<td>12.30</td>
</tr>
<tr>
<td>loss [%]</td>
<td>2.00</td>
<td>2.10</td>
<td>2.20</td>
<td>2.30</td>
<td>2.40</td>
<td>2.50</td>
<td>2.60</td>
</tr>
<tr>
<td>C_loss</td>
<td>12.40</td>
<td>12.50</td>
<td>12.60</td>
<td>12.70</td>
<td>12.80</td>
<td>13.00</td>
<td>40.00</td>
</tr>
<tr>
<td>loss [%]</td>
<td>2.70</td>
<td>more</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C_loss</td>
<td>40.00</td>
<td>40.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Final values of cost

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Optimal</th>
<th>Good</th>
<th>Bad</th>
<th>Unusable</th>
</tr>
</thead>
<tbody>
<tr>
<td>C voice</td>
<td>0 - 3</td>
<td>3.1 - 12</td>
<td>12.1 - 39</td>
<td>40 and more</td>
</tr>
</tbody>
</table>

D. Concept for traffic distribution

The incoming traffic has to be served according to its traffic class. That means that if more traffic flows arrive in one time, they are assigned to LSPs based on their priority. It is important to emphasize that different approach is used for voice and data traffic. 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 guarantees. In other words, traffic from Class4 which is within the guaranteed bandwidth is of higher importance then traffic from Class2 which is beyond the guaranteed bandwidth.

The initial distribution of traffic is done by assigning one LSP to each traffic class. The Unused bandwidth of LSP decreases as the amount of traffic increases. When the utilization of one LSP reaches the limit defined in Formula 3 the process of optimization takes place.

The process of optimization is used to achieve efficient distribution of traffic across all LSPs with preserved QoS. It can be triggered by three events:
1. new traffic flow within the guarantees cannot be assigned to any LSP
2. the network performance parameters of LSP carrying Class1 traffic have become insufficient
3. LSPs are unevenly utilized

The first event is solved by reducing the traffic flows of lower classes above guarantees. This reduction frees network resources and the new traffic flow can be assigned to LSP. The second event is solved either by reducing lower class traffic from the LSP for Class1 or by finding more suitable LSP for Class1 traffic and re-routing the traffic. We defined the uneven utilization of LSPs as shown in Formula 3. Unused bandwidth ($Ub(A)$) refers to the free bandwidth of LSP A and the average unused bandwidth ($Ab$) is calculated as shown in Formula 4 where N is number of used LSPs.

$$Ub(A) = \frac{Ab}{2}$$

$$Ab = \frac{N \sum Ub(x)}{N}$$

The process of the proposed TE server is summarized in following steps:
1. Analyze existing LSPs and their bandwidth
2. Start periodic IP SLA monitoring on each LSP
3. Calculate $C_{data}$ and $C_{voice}$ for each LSP
4. Assign each traffic class one LSP
5. Configure PE router (policing on each LSP) according to input bandwidth per class
6. Optimize bandwidth allocation if necessary

The optimization process follows strict rules. The increasing of bandwidth is always done on every LSP the traffic class is using proportionally to unused bandwidth of the LSP. Although the whole process is more complex following set of rules are the basic outline.

1. If input bandwidth > configured bandwidth and input bandwidth <= guaranteed bandwidth then increase configured bandwidth
2. If input bandwidth > configured bandwidth and input bandwidth > guaranteed bandwidth then
   a. Increase configured bandwidth only if there is unused bandwidth
   b. If there is lower class traffic above guarantees, lower this traffic amount
3. If Formula 3 is not reached for LSP A follow these steps:
   a. Find LSP with highest Ub (LSP B)
   b. Divide the amount of traffic between LSP A and LSP B so that Formula 3 is valid
4. LSP chosen for real-time traffic may be considered for data traffic only if its $C_{voice}$ is in Optimal conditions defined in Table 4.
IV. IMPLEMENTATION

The whole system was implemented as a server application running on a host connected to one of the PE routers in the network. To achieve effectiveness, optimal performance and scalability of our solution it was reasonable to use a modular scheme as an implementation method as shown in Figure 2.

The central part of the server is the daemon which is used to start up the server’s performance by starting all the other components. The network analyzer uses information stored in the database by the daemon to connect to the PE router by a secure SSH connection using the SSH client module. It will use the router’s configuration to get information about the network and configured LSPs. The main responsibility of the trunk handler is to apply the algorithms and manage all traffic demands in the network. The measurement engine is used to manage the SNMP connection to the router and collect the results of measurements in the network. The main work of the calculator is to calculate the cost of each LSP based on mathematical formulas (1) and (2). In the database all required information about LSPs, measurements, costs of LSPs, IP SLA configuration and assigned traffic flows is stored.

![Figure 2. Architecture of the server](image)

The proposed TE server performs following actions on incoming traffic:
- Classification and marking
- Analysis of input bandwidth vs. output bandwidth per class
- Re-configuration of the PE router if necessary
- Optimization if necessary

The configuration of the PE router which provides the distribution of the traffic consists of specific policy map configured on the input interface. This map consists of several class-maps, each for every traffic class defined.

Every traffic class has defined the maximum bandwidth which can be guaranteed in the network. All traffic exceeding this value is considered an extra traffic and does not have any guarantees. The traffic distribution is then performed by three-color policer used at PE router as follows:
- Traffic within defined guarantees is marked with specific MPLS EXP value
- Traffic above defined guarantees but below maximum currently possible used bandwidth is marked with another specific MPLS EXP value
- All other traffic is dropped

With this logic traffic of each class is marked with one of two possible values – one for guaranteed traffic and one for extra traffic.

Each LSP has strictly defined which MPLS EXP values can be transmitted by it. With the use of Class-based tunnel selection (CBTS) is the traffic sent to the correct LSP selected by the traffic distribution algorithm.

V. EXPERMENTS AND RESULTS

The proposed server was implemented and experimentally tested in a laboratory environment. We proposed a testing topology with the layout of LSPs defined in advance. The topology is shown in Figure 3. The traffic in the network was simulated by a traffic generator Iperf. The evaluation of the server’s functionality was based on following parameters:
- Throughput of the traffic per class [b/s]
- Utilization of LSPs [%]
- Packet loss per class [%]
- QoS parameters for Class1 traffic (delay [ms], jitter [ms] and loss [%])

We performed four experiments, each with different data flow patterns. The results were compared to results of the same scenario in the same network without the use of proposed TE server. The results of one of the experiments are shown in Figures 4-10.

The generated traffic was identical in both scenarios and is represented in Figure 4 and Figure 5 as input bandwidth. The amounts of traffic of each class were increased or decreased gradually to show the reaction of server to different situations.

The actual throughput gained during the experiment without TE server is shown in Figure 6 and with TE server in Figure 7. It is obvious that the throughput for Class3 and Class4 is higher when the TE server is used. Also the throughput for Class2 is slightly higher. This behavior is the result of sharing the traffic load across multiple LSPs. When the TE server is not used each traffic class uses only its dedicated LSP and is therefore limited by its bandwidth. The TE server periodically measures the load on LSPs and re-configures the PE router so that the traffic load is distributed more evenly among the LSPs.

Figures 8 and 9 show the packet loss of each class during the experiment. Without the TE server values of packet loss for Class4 reach 50 %, packet loss of Class3 reaches 10 %. The use of TE server minimizes the packet loss of each class. The values can be high for short period of time which reflects the reaction time of the server. The packet loss drops to minimum after the TE server re-distributes the traffic flows.

The utilization of LSPs during the experiments is shown in Figure 10 and Figure 11. Without the TE server each of tunnels Tunnel2, Tunnel3 and Tunnel4 (LSP2, LSP3 and LSP4 respectively) is fully utilized while Tunnel1 (LSP1)
is not used at all. All of the LSPs are evenly utilized with the use of TE server and therefore network resources are optimally used.

As the input bandwidth of Class4 traffic increases Tunnel4 is utilized up to 80%. Since the input traffic increases even more another LSP – Tunnel2 is used to share the traffic. Similar situation repeats when Class3 traffic enters the network. In the end of experiment all four LSPs are in use (with TE server) although only three classes of traffic are generated.
VI. CONCLUSION

In this paper we proposed an online traffic-engineering server (TE server) to measure network performance parameters and optimally distribute traffic flows across multiple LSPs in the network. The proposed server uses classification of traffic into four classes although more classes may be used if necessary. The solution is based on periodic measurements of amount of incoming traffic, used bandwidth of LSPs, delay, jitter and packet loss on LSPs. The measured data is used by the traffic distribution algorithm which optimizes the utilization of network resources while providing required QoS for sensitive traffic.

Our results prove the effectiveness and successful utilization of network resources while distributing various classes of traffic among multiple paths in the network. The results show higher throughput for every class of traffic with the use of proposed TE server when compared to the same scenario without the TE server.

VII. ACKNOWLEDGMENTS

The support by Slovak Science Grant Agency (VEGA1/0676/12 “Network architectures for multimedia services delivery with QoS guarantee”) is gratefully acknowledged.

VIII. REFERENCES


Figure 7. Utilization of LSPs without TE server(A) and with TE server (B)