

# Does ns2 really simulate MPLS networks?

Hugo Coll<sup>1</sup>, Jaime Lloret<sup>2</sup> and Francisco Javier Sanchez<sup>3</sup>

<sup>1,2</sup>*Department of Communications, Polytechnic University of Valencia, Camino Vera s/n, 46022, Valencia (Spain)*

<sup>3</sup>*ADIF, Molino de las Fuentes s/n, 46026, Valencia (Spain)*

<sup>1</sup>*hucolfer@posgrado.upv.es,* <sup>2</sup>*jlloret@dcom.upv.es,* <sup>3</sup>*jsbolumar@adif.es*

## Abstract

*Multi Protocol Label Switching (MPLS) is being used in many corporate networks and public infrastructures and as a backbone technology of many Autonomous Systems. Because of its importance, what is needed is to find out simulators able to simulate MPLS networks whose results reflect the real environment as much as possible. In this article, we will show real measurements from the Railway Infrastructure Administrator (ADIF, the Spanish railway infrastructure manager) MPLS network and we will compare them with the results obtained by the freeware simulator ns2. We will check the level of reliability provided by the simulator and we will know under which parameters its results will be similar with those obtained from the real network. These results could be taken into account later to simulate real-time critical services over MPLS networks.*

## 1. Introduction

MPLS is a standard data-carrying mechanism, created by the IETF and defined in the RFC 3031 [1], designed to transport different types of traffic. Operates at layers 2 and 3 of the OSI model and belongs to the family of packet-switched networks. MPLS works in conjunction with IP and Interior Gateway routing protocols such as OSPF and IS-IS. Any communication is identified using a label placed between level-2 and level-3 headers and labels are changed in every hop. A MPLS network is formed by Label Switching Routers (LSR). Each LSR analyzes the label of the packets to determine the output interface. MPLS gives QoS regardless of the type of network where it is being implemented. Because of routing is done using the level-2 label, it offers multiprotocol support and can be portable over other data link technologies such as ATM and Frame Relay. MPLS uses Application Specific Integrated Circuit

(ASIC) technology, providing fast searches in the routing table and classifying the packets based on the Forwarding Equivalence Class and inbound interfaces with highest criteria. MPLS runs regardless the network architecture and the type of network connected with it. It gives better convergence times than routing protocols, and it is a scalable mechanism. VPNs could be easily deployed over MPLS networks providing tunnels more efficient than IP ones.

One of the main benefits of MPLS architecture is that it provides traffic engineering (each flow could have different labels for different users), so forecasts and estimations to optimize the resources and to avoid the congestion could be performed [2]. These features make MPLS as one of the best options for high performance networks.

Normally, Autonomous Systems (medium and high corporate and public data networks) are based on Ethernet access to corporate MPLS/VPLS network. The main structure and technology for this network responds basically to multimedia traffic requirements. Sometimes, in order to avoid occasional blocking problems, network services are provisioned through local Ethernet networks (most of them in ring topology to provide resiliency). These solutions increase complexity and budget, and present many difficulties for integrated management. Figure 1 shows an example. In the case of railway systems (where we are going to do our tests and simulations), railway station auxiliary services (such as automatic stairs, lifts, evacuation and emergency facilities, IP phones, video, remote alarm and supervision and so on) are provisioned through direct access to corporate MPLS network. Station services and equipments are connected to third level nodes (usually MPLS routers) organized in double ring topology. Both ends of the ring are connected to regional gigabit ring, through physical link between third level node and second level node (this is typically a layer 3 switch). Figure 2 shows station services and equipments topology.

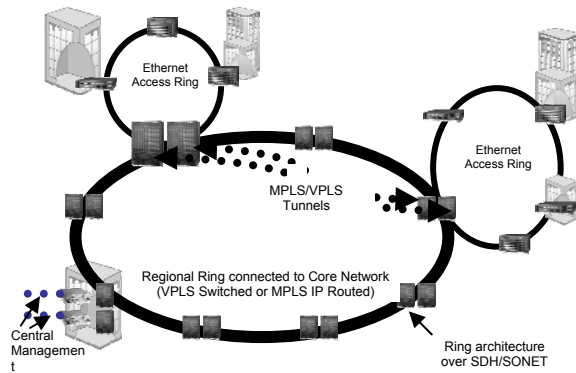


Figure 1. Corporate and integrated network

Once we have described briefly the MPLS technology and that it is widely used in corporate and public networks, and how is it implemented in a real environment, we have considered the necessity of to simulate such technology and to find out which differences exist between two simulator measurements and measurements taken from the real world.

The rest of the paper is structured as follows. Section 2 describes ns2 main features and shows how MPLS capabilities are introduced. The comparison between real measurements and the measurements taken from the simulator are shown in section 3. Finally, section 4 gives the conclusions and future works.

## 2. Ns2 main features

Network Simulator 2 (ns2) [2] is an open source discrete events simulator that is developed in C++ and OTcl (a Tcl object oriented version). The simulator is fully oriented to the research, although it is also used for educational purposes. It is able to simulate many types of architectures and protocols for any layer of the OSI model (802.3, 802.11, routing protocols, TCP, UDP, http, telnet, ftp, cbr and so on), for many types of networks such as (satellite, wireless, wired, etc.). Parameters such as RTT, delay, jitter and statistics can be measured between any nodes for many types of protocols of these networks. In fact, new protocols and algorithms can be developed to check their operation.

Ns appeared as a REAL network simulator variant in 1989, but in 1995 DARPA took on its development through the VINT (Virtual InterNetwork Testbed) project with SAMAN (Simulation Augmented by Measurement and Analysis for Networks), through NSF with CONSER (Collaborative Simulation for Education and Research) and many other researchers. In 1999 Gaeil Ahn and Woojik Chun developed MPLS functions for ns, setting up MNS module [3], which is a part of the code since 2.1b6 version.

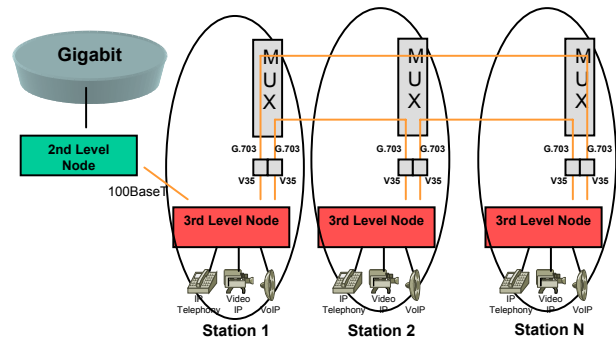


Figure 2. Railway station equipment connection to multi-service MPLS network

The last stable version (ns-2.31, March 2007), has the following MPLS features:

- Label switching
- LDP (Label Distribution Protocol), CR-LDP (constraint-based LDP), RSVP-TE Resource ReSerVation Protocol Extension) support.
- Explicit and dependent routing
- QoS support, FEC (Forwarding Equivalence Class), stream aggregation, Diffserv.
- On demand and ordered label distribution.
- Data-driven and control-driven triggered labels.
- Global and local route recovery mechanisms support.
- Support for different recovery models (Haskin, Makam, fast-reroute, Reliable and Fast Rerouting (RFR)).

MPLS design and implementation in ns2 can be found in [4] and [5]. Furthermore, other modules can be programmed to support special protocols that are not provided till now [6][7].

This simulator have been used to simulate MPLS networks in many relevant works such as the study of load balancing algorithms in MPLS traffic engineering [8], protection performance components in MPLS networks [9], routing protocols for MPLS networks [10], and even traffic engineering [11].

## 3. Real measurements vs simulations

This section shows the measurements taken from the real network and the results obtained from the Network Simulator. Then, we will compare them.

### 3.1. Testbed

In order to test a MPLS network, we have chosen the area 3 (Madrid-Albacete-Játiva-Valencia) of ADIF corporate network. 31 routers form this zone. The network is shown in figure 3.

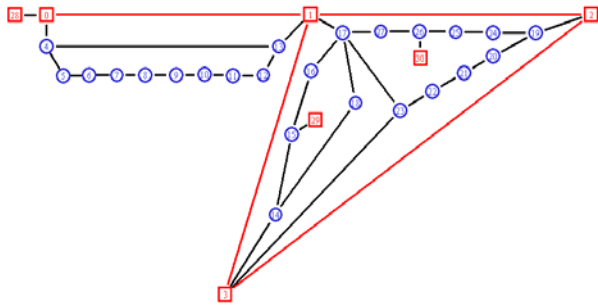


Figure 3. Railway network used for comparison.

The network has two levels. The second level can be distinguished by lines in black in figure 3. There, stations are connected to the rings with 2 Mb/s, for deliver service in short distances (15-60 kilometers). Above this topology we can find 1 GB fiber optic links that connect main network nodes. Lines in red form the first level, that is, nodes 0, 1, 2 and 3 (Madrid, Albacete, Valencia and Jativa). Each ring has 2 connected nodes to this fiber network through 100Mb/s links. Thus, when a station needs to communicate with a station of another ring or other control node, data will go through fiber network. Distances between nodes in the same ring not exceed 15 Kilometers. However, first level node links are about 250 Kilometers. The devices used as nodes in the MPLS network was routers from the manufacturers Cisco, Alcatel and Teldat.

### 3.2. Real measurements

This network is usually used to carry video traffic for railway surveillance between the stations and the main station of their ring and between the stations and the central server that is located in Madrid. There is also VoIP traffic to allow people from different stations have an IP telephone call between neighbour stations and between any station of the ring and the main station of the ring. We can also find http, telnet, ftp, pop3 and smtp.

We have measured the real environment to know which delays are in a real MPLS network when there is some specific traffic through it. In order to do that, we have gathered RTT from node 21 to node 9, that is, the nodes more remote in terms of distance.

Figure 4 shows the real response time obtained by 100 regular pings between node 21 and node 9. The trace followed by this ping was 21-20-19-2-1-13-12-11-10-9, that is, 9 hops between the source and the destination node. The minimum value was 22 msec, the maximum value has been 316 msec. and the average time has been 26.37 msec. We think that that

peak at around the 60<sup>th</sup> ping has been because there has been an sporadic excess of traffic in one of the nodes in the path, so the queue of that device needed more time to process all packets. Figure 5 shows the delay time for 100 regular pings from node 9 to node 21. This time the trace followed has been 9-10-11-12-13-1-17-23-22-21, that is, 9 hops between the source and the destination node. The minimum value has been 23 msec, the maximum value has been 98 msec. and the average value has been 24.84 msec. This time has also been several peaks due to congestions in some queues of the devices involved in that path.

Now, we wanted to know how much the time is increased between 2 nodes when there is more traffic between several nodes inside the path. We have divided our procedure in two stages.

In the first stage, we have measured the response time between node 15 and node 28 (nodes from different rings) and then between node 9 and 28 (nodes from the same ring) without any additional traffic. In the first case, we have obtained a RTT minimum value of 7 msec, maximum of 12 msec. and an average of 7.16 msec. The trace followed has been 15-16-17-1-0-28. RTT is shown in figure 6. We can see that there are congestion peaks once again. This time there are several peaks, but their peak value is very low compared with the results obtained from the measurements taken before. In the second case (between node 9 and 28), we have obtained a RTT minimum value of 12 msec. a maximum of 18 msec. and an average of 13.22 msec. The trace followed has been 9-8-7-6-5-4-0-28. Figure 7 shows RTT values for 100 pings. There are more peaks than before and, although the difference between the minimum value and the maximum value is higher than the case measured before, it is less constant.

In the second stage, we have measured the response time between node 15 and node 28 and then between node 9 and node 28 but with additional traffic. The traffic introduced has been packets with a size of 556 bytes between nodes inside their path. In the first case, we introduced traffic between node 14 and node 29, between node 3 and node 14, between node 1 and node 3 and between node 0 and node 1. RTTs obtained between node 15 and 28 while this traffic was running is shown in figure 8. There was just one main peak in the 57<sup>th</sup> ping that could be given because a sporadic congestion. The average time is 12.15 msec. In the second case, we introduced traffic between node 13 and node 12, between node 12 and node 11, between node 11 and node 10 and between node 10 and node 9. Figure 9 shows the RTTs measured. In this case, values vary a lot and there are several main peaks. The average time is 33.64 msec.

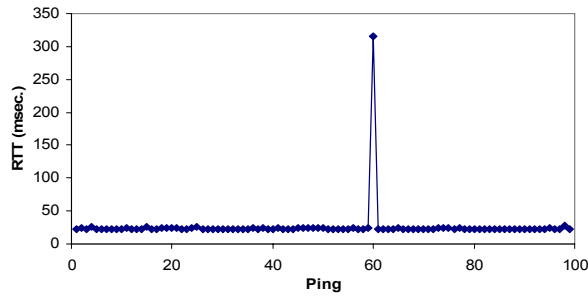


Figure 4. Delay time from node 21 to node 9.

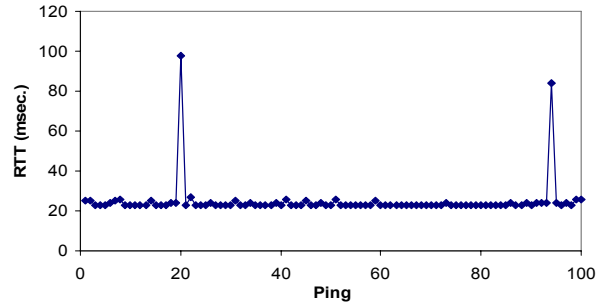


Figure 5. Delay time from node 9 to node 21.

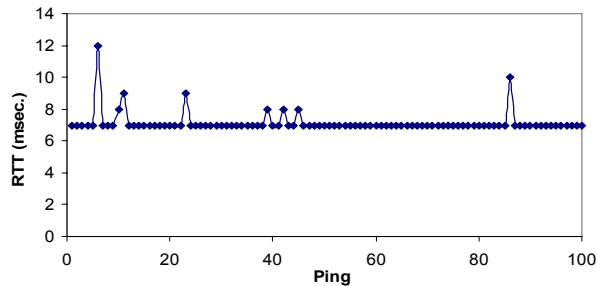


Figure 6. Delay time from node 15 to node 28.

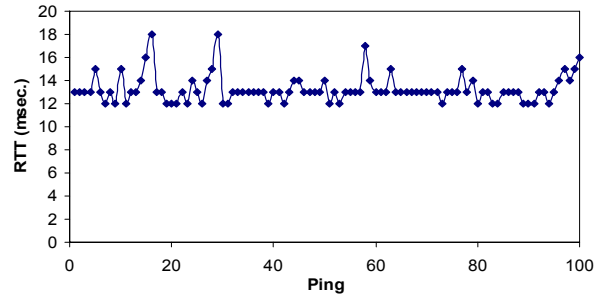


Figure 7. Delay time from node 9 to node 28.

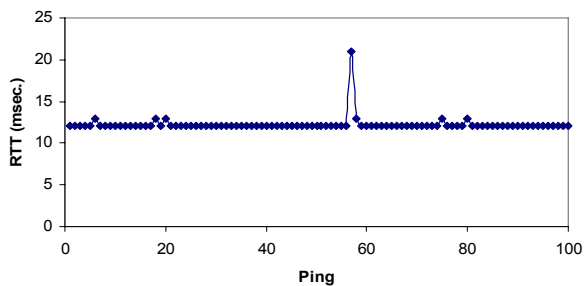


Figure 8. Delay time from node 15 to node 28 with additional traffic.

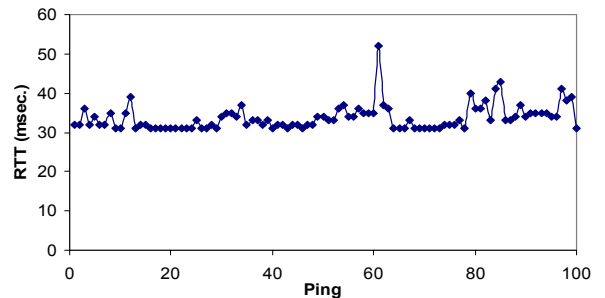


Figure 9. Delay time from node 9 to node 28 with additional traffic.

### 3.3. NS2 simulations

In order to measure network parameters, we have implemented the ADIF railway network as really it is in the ns2 simulator. Nodes 29 and 30 were the sources of video streaming and node 28 was the destination of the video streams. Video streams were simulated using UDP protocol with a maximum packet size of 512 Kb and a constant bit rate (cbr) of 256 Kb/s. The video stream sources simulate traffic from a video-vigilance system placed in two stations and a central control node respectively. On the other hand, we have implemented several VoIP streams between three pairs of nodes in order to simulate regular telephonic conversations over IP between collateral stations. These VoIP streams were placed between nodes of the same ring (nodes 15 and 18, nodes 21 and 1 and nodes 5 and 6). The streams were UDP packets with

200 bits of length and 70 Kb/s of bidirectional audio bit rate. Figure 10 and figure 11 are obtained from the ns2 and give the packets received by the node 28 and node 1 respectively using the traffic aforementioned.

Once network topology is created, we obtain the following simulation measurements. In the first simulation, when there isn't any traffic in the network, we obtained the same RTT from node 21 to node 9 than from node 9 to node 21, it was 5 msec., but when there is traffic, we obtained different RTTs, 5 msec when the ping was from node 21 to node 9 and 5.5 msec from node 9 to node 21. This difference was because the path was different from node 9 to node 21 (it was 9-10-11-12-13-1-17-23-22-21) than from node 21 to node 9 (it was 21-22-23-3-1-13-12-11-10-9). It makes sense since both cases (the real and the simulated environments) had different paths for every RTT and different values of RTT.

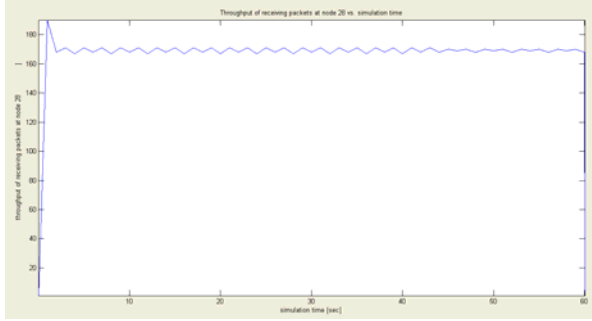


Figure 10. Packets received per second in node 28.

In the second simulation, in the first stage, first we have obtained a RTT average of 3.34 from node 15 to node 28. The maximum value was 5.0 msec. and the minimum value was 3.1 msec. Second, between node 9 and node 28 we obtained 3.41 msec. as the average RTT value. The maximum value was 3.7 msec. and the minimum value was 3.4 msec. Both cases had the same tracer route than the real environment. In the second stage, we added more traffic to the network between nodes inside the path of node 15 and node 28 and between node 9 and node 28. We have observed that in several cases the additional traffic in the network doesn't make to change the results. We even had had lower RTT measurements in some of those simulations than when there isn't additional traffic, so these measurements seem to be wrong. Figures 12 and 13 shows RTT measurements taken from ns2 simulator in the first and in the second stage respectively.

Finally, we have simulated a failure in the network in order to know how ns2 performs the path recovery in MPLS. We didn't do it in the real environment because to do it in a real railway network could imply serious problems such as the loose of the train control, so we forgot the idea of doing this failure in the real environment.

Simulation was performed with the following events: VoIP streams began in  $T=0.2$  sec. Video streams began in 0.7 sec. There is a link down between node 0 and node 1 at  $T=1$  sec. The link is recovered at  $T=1.3$  sec. Then, there is a link down between node 17 and node 1 at  $T=1.5$  seconds and it is recovered at  $T=1.7$  seconds. Figure 14 shows the delay of the MPLS-LDP packets received in node 28 during the described events. Just at the beginning their delay were about 1 msec, but they decreased quickly till the link failed in  $T=1$  sec. Then, there was a sporadic peak of around 4.5 msec. When the link was recovered in  $T=1.3$ , there was a peak of around 3.7 msec. (lower than the previous one). The link failure between node 17 and node 1 didn't have the same effect, so the delay was lower than the half of the other link (2 msec. when

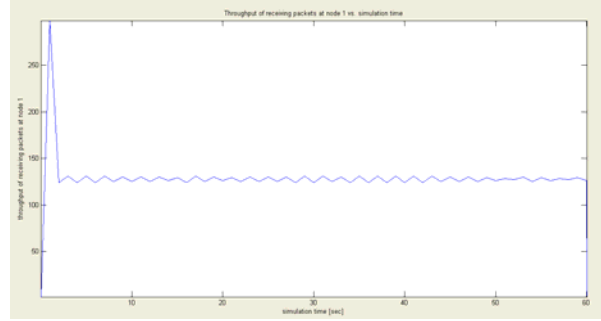


Figure 11. Packets received per second in node 1.

it failed down and 0.9 msec. when it was recovered). It seems that it is needed more time for MPLS-LDP packets to recover the path when the link fails down than when it is recovered.

### 3.4. Comparison analysis

Now, we are going to compare real environment results with the ones obtained with the simulator ns2 for the same network.

The difference between the real and the simulated environments was high in the measurements taken between node 9 and node 21. It was between 4 and 5 times more in the real environment. But, when we compare RTT average values between node 15 and node 28 and between node 9 and node 28, the relationship between the real environment was 3.6 and 10 times more respectively, so we can't extract any general rule between real environment and simulations.

Nevertheless, we have observed a rule between real measurements and ns2. It includes three data (Real RTT, RTT simulation result and number of nodes in the path between the source and the destination). So we will be able to scale simulated results to real ones. The relationship between them is given by expression 1.

$$R = \frac{\text{Real\_time}}{\text{Simulated\_time} \cdot N} \quad (1)$$

Where  $N$  is the number of nodes involved in that path. Applying it to our measurements, we always obtain a result close to 0.6.

On the other hand, when we introduce regular traffic in the network, ns2 doesn't give us reliable results. Making the same comparison with the peaks of the RTT values, we obtained very different results, 4.4 more between node 15 and node 28 and 57.5 more between node 9 and node 28. So, once again there is no relationship between real measurements and simulations.

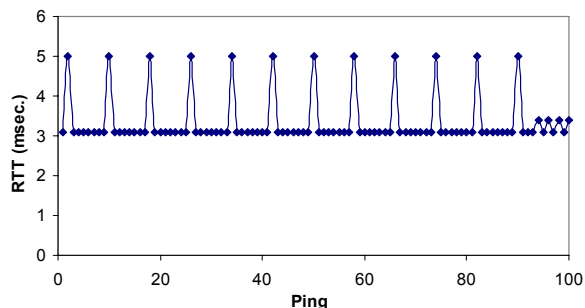


Figure 12. Delay time from node 15 to node 28 with additional traffic measured in ns2.

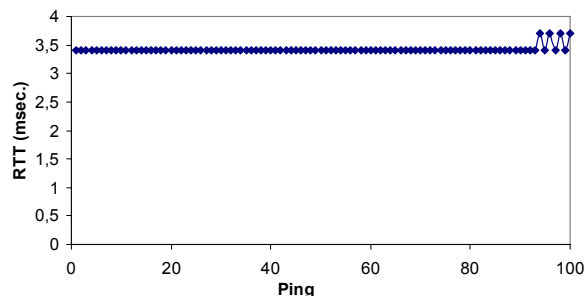


Figure 13. Delay time from node 9 to node 28 with additional traffic measured in ns2.

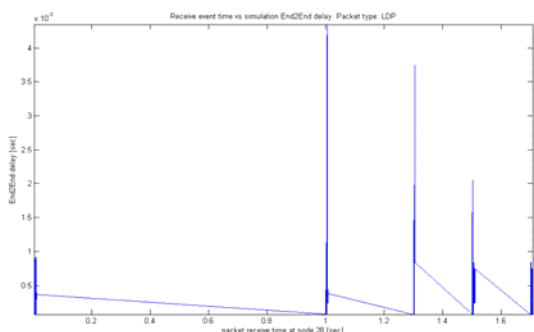


Figure 14. Delay of the MPLS-LDP packets in node 28.

Finally, we have observed that there is higher impact in the number of LDP-MPLS packets when there are link failures that when the link is recovered, but the convergence time is done in a very low time, which seems to be “ideal”, instead of simulating the real world.

## 4. Conclusions

Nowadays, MPLS networks are widely deployed. Many researchers use ns2 to simulate them, but the results obtained are far from the real measurements. We have compared both measurements and ns2 gives us an approximation of what will happen in the real world if we take into account an expression which depends on the number of hops involved in the path between the source and the destination. But, when we introduce traffic between nodes inside the path between a source and a destination, results are wrong.

Taking into account results obtained in this work, we can state that ns2 is not a good option if we want to simulate MPLS networks using critical real-time services.

We think that this work could encourage the developers to improve the MPLS module in the ns2 simulator. We also wanted to advise researchers about

the differences between simulations in MPLS networks using ns2 and the real world measurements.

## 5. References

- [1] E. Rosen, A. Viswanathan, and R. Callon, “Multiprotocol Label Switching Architecture”, RFC 3031. January 2001. Available at <http://www.ietf.org/rfc/rfc3031.txt>.
- [2] William Stallings, MPLS. Internet Protocol Journal Volume 4, Issue 3. Available at [http://www.cisco.com/warp/public/759/ipj\\_4-3/ipj\\_4-3\\_mpls.html](http://www.cisco.com/warp/public/759/ipj_4-3/ipj_4-3_mpls.html)
- [3] ns2 Simulator. Available at <http://www.isi.edu/nsnam/ns/>
- [4] Gaeil Ahn and Woojik Chun. Overview of MPLS Network Simulator: Design and Implementation. Technical report, Department of Computer Engineering, Chungnam National University, Korea, December 1999.
- [5] Gaeil Ahn and Woojik Chun, Design and Implementation of MPLS Network Simulator Supporting LDP and CR-LDP. Proceedings of the 8<sup>th</sup> IEEE International Conference on Networks. Singapore. September 2000.
- [6] J. Chen, C. Wangy, F. C. Tsaix, C.-W. Chang, S.-S. Liu, J. Guo, W.-J. Lien, J.-H. Sum, and C.-H. Hung, “The design and implementation of WiMAX module for ns2 simulator,” ACM VALUETOOLS. Pisa, Italy October, 2006.
- [7] Frida Eng., New Modules for ns-2, 2004. Available at <http://www.control.isy.liu.se/~frida/nsmodules>
- [8] Keping Long, Zhongshan Zhang, Shiduan Cheng. Load balancing algorithms in MPLS traffic engineering. Proceedings of the IEEE Workshop on High Performance Switching and Routing. Pp. 175-179. Dallas, TX, USA. 2001
- [9] Eusebi Calle, José L. Marzo and Anna Urra, Protection performance components in MPLS networks. Computer Communications. Volume 27, Issue 12. Pp. 1220-1228. July 2004.
- [10] R. Boutaba, W. Szeto and Y. Iraqi, DORA: Efficient Routing for MPLS Traffic Engineering. Journal of Network and Systems Management. Volume 10, Number 3. Pp. 309-325. September 2002.
- [11] D.A. Barlow, H.L. Owen, V. Vassiliou (USA), J. Grimminger, H.-P. Huth, and J. Sokol, Simulating Router-based Traffic-engineering in Radio Access Networks using NS-2. Proceedings of the Communications, Internet and Information Technology. St. Thomas, US Virgin Islands November, 2002.