The MPLS Network Simulators in the Computer Network Education

Veneta P. Aleksieva[1]

Abstract **–In this paper are discussed the MPLS network simulators as teaching tool. It is presented some criteria of the simulators comparison. The main goal is to find the best tool for teaching innovation.**

Keywords **– simulator, MPLS, network education, networking tool**

I. INTRODUCTION

Next generation communication networks are migrating towards a unified network architecture where both wired and wireless network segments will co-exist. This is accompanied by the growing demand on networks to provide QoS, due to the rise in popularity of real time and multimedia applications. Future networks should accommodate a variety of services, satisfying different traffic types, providing support for user mobility, and will be able to guarantee QoS.

MPLS/Multi protocol Label Switching/ is a connection oriented technology that arises to palliate the problems that current networks have related to speed, scalability and traffic engineering [1]. In fact, MPLS packets forwarding is based on labels and not in the analysis of encapsulated data from upper levels. It is a multi protocol technology that supports any network protocol as well as any technology in lower layers (link or physical). MPLS is used as a traffic engineering tool to direct traffic in a network in a more efficient way than original IP shortest path routing. Path in the network can be reserved for traffic that is sensitive, and links and router that are more secure and not known to fail can be used for this kind of traffic.

MPLS recovery provides different levels of service, based on their service requirements. It should give the flexibility to select the recovery mechanism, choose the granularity at which traffic is protected and choose the specific types of traffic that are protected in order to give operations more control over that tradeoff. [2]

Actually, there is a growing need for end-to-end QoS mechanisms with mobility support to address the requirements of heterogeneous network environments. The wireless segments in the next generation networks are likely to be based on multi-hop ad-hoc or infrastructure-less networks. Also, in wired networks, paths once established, hardly change. On the other hand in mobile networks, calculated routes have limited lifespan since the network connection structure changes due to nodal mobility. The topological changes affect the resource availability directly. Further, wireless resources and topological connectivity are even harder to track accurately because of the nature and transmission effects associated with the shared wireless media. Consequently, adhering to the negotiated QoS guarantees is even more challenging than in static network topologies. It will be good to analyze and test these cases with simulation, before applying in the real networks.

The recovery of the MPLS network is based on the algorithm that is applied in order to detect the faults and route the data flow in an alternative path. There are various algorithms. However, each algorithm employs only one of the two basic techniques:

- **protection switching**, where a recomputed alternative path, which is usually disjoint from the working path, is set up for every flow
- **rerouting**, where an alternative path is dynamically recomputed after a fault is detected.

For both techniques, the alternative path can be either global or local. The protection of data flows in case of link or router failures is very important, especially for real time services and multimedia applications.

In this research a comparative between some common teaching simulators is made. Some researchers have already made proposals in this way [2,3,4,5]. It is given an overview of the available network simulators with MPLS implemented, and it is described by which criteria must choose MPLS simulator as teaching tool. The primary purpose of this research is to find a simulator that enables to simulate various MPLS applications without constructing a real MPLS network.

II. CHOICE OF CRITERIA OF MPLS SIMULATORS COMPARISON

When surveying various simulation tools which could be used in evaluation studies of service availability and resiliency mechanisms in MPLS network, their following features should be taken into account:

- Modelling capabilities
- Credibility of simulation models
- Credibility of simulation results
- Extendibility
- Usability
- Cost of licenses

Each potential simulator will be used in a simple simulation of MPLS recovery model, to gain practical experience and to

[1]Veneta P. Aleksieva is with the Department of Computer Science and Engineering, Technical University of Varna, str.”Studentska ”1, 9010 Varna, Bulgaria, e-mail: ven7066@abv.bg

assess the level of user-friendliness of simulators considered [6].

In this research the main goal is to be found an MPLS simulator that allows designing and setting MPLS domains and their components and give different MPLS recovery mechanism opportunities. When the simulation is run, it must perform statistical analysis of its results, from all educational points of view. Their main features must:

- be a teaching simulator that allows an elementary statistical analysis of the network traffic;
- have a visual editor to design scenes;
- be a multiplatform software;
- be free-of-charge;
- be easy to install;
- have not been designed to work in real MPLS networks using current manufacturers' components.

In the research[2] it is used some additional criteria of comparison of MPLS simulators:

- Interactive simulation
- Multilanguage
- Free software
- QoS simulation
- Real environments applicability
- Installation and execution easiness

Moreover, it can be used four criteria which are very important for the performance of the restoration mechanisms in MPLS networks:

- fault recovery time,
- packet loss,
- packet reordering,
- tolerance of multiple faults.

Simulation results must indicate the performance advantages of the proposed MPLS recovery algorithms, when compared with other restoration mechanisms, based on these criteria. In the research [7] are used some new criteria:

- Latency
- Full restoration time
- Vulnerability
- Quality of protection

In this research it is used combination of all these criteria, because in the computer network education of MPLS it is important to present advantages and disadvantages of different MPLS recovery schemes.

III. Comparison of MPLS simulators

In this section it is given a short overview of the available network simulators with MPLS implemented. Each simulator is individually assessed and mutually compared.

- **J-Sim** (Java Sim) [8] is a component based network simulator developed entirely in Java, by Hung-ying Tyan and some other people at the Ohio State University. J-Sim is an open source simulator. There has been one MPLS module contributed to this simulator, developed by the Infonet Group of the University of Namur. This model consists of two components: a forwarding table component and a MPLS component. It associates an IP prefix or an incoming label with an outgoing interface and an outgoing label. The MPLS component forwards packets according to the configuration of the forwarding table. This model does not include any label distribution protocol so LSPs must be setup statically.
- **OMNeT++ [**9] is a discrete event simulation environment programmed in C++ and developed by András Varga. The simulator is open source and free to use for academic and non-profit users. The simulator is component-based and there are many models in development for this simulator. The MPLS model in OMNeT++ was originally developed by Xuan Thang Nguyen from the University of Technology in Sydney, but now the model is maintained by András Varga [10]. The model includes components for MPLS forwarding, LDP, CR-LDP and RSVP-TE. This model was developed to study the forwarding mechanisms in MPLS so there where some simplification done in the way that the Label Switched Routers was implemented. This made this model, as it is today, not suitable for failure recovery simulation. The forwarding tables for all routers in the network is implemented in one table, which means that all routers have the same copy of the network topology at the same time. If a failure happens in the network than all routers will be aware of the failure at the same time. This is not what happens in a real case scenario where topology changes like a failure has to be indicated to nodes by sending routing update information from the point of failure. There were also some problems with the implemented TCP and UDP for IP on time. RSVP-TE was implemented, but after some testing of the included MPLS model and the transport protocols, it is noticed that a lot of the implemented functionality in the model had to be rewritten. It has a professional technical support, growing customer base, hardware requirements are moderate and relatively easy to learn. It has some weakness – not so good documentation, model design GUI is not detailed enough to be useful, simulation results reporting is not adequate.
- **GLASS-** GMPLS Lightwave Agile Switching Simulator is a Java based network simulator [11]. The simulator uses the basic framework of the Scalable Simulation Framework (SSF) with its extension SSFNet. Although this is as the name implies a simulator that is developed for GMPLS simulations, it can be used for MPLS simulations as well. The simulator has implemented MPLS forwarding and label distribution with LDP, CR-LDP and RSVP-TE. It is designed for studying IP+MPLS network and multilayer resiliency, and growing customer base. But documentation is not adequate, a relative new widely used simulation tool, smaller library of rebuilt modules and protocols than in other simulators, simulation result reporting is not adequate, poor post-technical support.
- **OPNET** [12] – This simulator has large library of simulation models of communication protocols and equipment, professional support and good documentation. The main disadvantages are relatively high price and it doesn't support on-line analyses if simulation output data.
- **NS-2** - (Network Simulator 2) is a discrete event simulator targeted at networking research [13]. The simulator has been in development since 1989 and there are many

different contributed models available for the simulator. This simulator is also open source and there are a lot of users that have contributed to make this simulator. The main implementation is made by the VINT project at BL, Xerox PARC, UCB, and USC/ISI. NS-2 is coded in C++ and TCL. The simulator has one contributed MPLS module for NS-2. This module is called MNS (MPLS Network Simulator) and was developed by Gaeil Ahn. This model had implemented MPLS forwarding and label distribution by LDP and CR-LDP. The model has also some functionality implemented for recovery mechanisms, like the ability to setup a backup path and associate it with a working path. Unfortunately this model did not include an implementation of RSVP-TE. The main advantages here are: well known and widely used, powerful and flexible scripting and simulation setup, many protocols implemented, available source code, easy to extend. But some things are no so good: some protocols and features are not well documented, not well known technical support, patching extensions in is not easy.

The results of comparison are presented in Table1.

TABLE I
COMPARISON OF MPLS SIMULATORS

	J-Sim	OMNeT++	GLASS	OPNET	NS-2
VoIP protocol stacks	missing	satisfy	missing	satisfy	satisfy
MPLS-TE and RSVP signaling protocols	poor	poor	excellent	excellent	satisfy
Models of typical failures and different routing protocols	satisfy	satisfy	satisfy	excellent	excellent
Models of optical layer components	satisfy	satisfy	excellent	satisfy	poor
Models of different teletraffic scenarios	poor	satisfy	excellent	excellent	satisfy
QoS mechanisms	satisfy	satisfy	satisfy	excellent	excellent
Different network architectures	excellent	excellent	excellent	excellent	excellent
Analysis of typical performance measures	poor	poor	satisfy	excellent	satisfy
Credibility of simulation models	missing	excellent	missing	satisfy	satisfy
Quality of sources of randomness	poor	excellent		poor	excellent
Analysis of simulation output data		excellent		poor	excellent
Ability of extending or adding new simulation models of protocols	excellent	excellent	excellent	excellent	excellent
Existence of GUI, which be used as input/ output interface	excellent	excellent	excellent	excellent	poor
Existence of a good manual and an introductory tutorial	poor	satisfy	poor	excellent	excellent
The cost of license	excellent		excellent	poor	excellent

As it is possible to see in the Table1, none of these simulators satisfies all requirements. This problem could be solved by designing a custom-made simulator. The best start of research to this new simulator is with OPNET,OMNET++ and free of charge NS-2.

IV. CONCLUSION

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. It is possible to check the level of reliability provided by the simulator. These results could be taken into account later to simulate real-time critical services over MPLS networks.

Simulation models are credible if they are valid and verified. A model is valid if it represents a given system accurately, at the required level of details. Verification is concerned with determining whether the simulation model has been correctly translated into a computer program. If simulator is free of charge, nobody takes responsibility for offering valid models encoded in verified programs.

Assuming that a network simulator uses credible simulation models, credibility of the final results it produces depends on the quality of its sources of randomness of failures or statistical representativeness of traces of real traffic and statistical accuracy of the final simulation results.

Extendibility of simulator with new features to existing simulation models is important, because the amount of work or time which is needed to extend the existing simulation models must be less.

A user-friendly simulator should be equipped with GUI, because during simulation the GUI could be used for showing evolution of simulated processes and intermediate values of analyzed performance measures.

In this research are presented simulators, which are not suitable to satisfy all these requirements. This problem could be solved by designing a custom-made simulator. This simulator will be supporting tool to research projects related to MPLS as well as in teaching subjects related to this technology.

ACKNOWLEDGEMENT

The work presented in this paper was partially supported within the project BG 051PO001-3.3.04/13 of the HR Development OP of the European Social Fund 2007-2013.

REFERENCES

[1] M. Kodialam. T. V. Lakshman. Restorable Dynamic QoS Routing. IEEE Communications Magazine, Vol 40, Issue 6, June 2002, pp 72-81.

[2] M. Domínguez-Dorado, F. J. Rodríguez-Pérez, J. Carmona-Murillo, J. L. González-Sánchez "Educational Improvements Applying an MPLS Network Simulator: A Technical Approach",IEEE Multidisciplinary Engineering Education Magazine, VOL. 2, NO. 4, DECEMBER 2007, http://www.ieee.org/edsocsac

[3] Gaeil Ahn, Woojik Chun, "Overview of MPLS Network Simulator: Design and Implementation", http://heim.ifi.uio.no/~johanmp/mpls/MNS_v1.0_arch.pdf

[4] V. Sharma, Ed. Metanoia, Inc. ,etc. "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery", http://www.faqs.org/rfcs/rfc3469.html

[5] Maria Hadjiona, Chryssis Georgiou, Maria Papa, and Vasos Vassiliou "A Hybrid Fault-Tolerant Algorithm for MPLS Networks", Wired/Wireless Internet Communications, Volume 5031/2008, Tuesday, May 20, 2008, http://www.springerlink.com/content/q01x1q865337t8t0/

[6] http://www.cosc.canterbury.ac.nz/research/reports/TechReps/2006/tr_0605.pdf

[7] http://www-entel.upc.es/xavierh/mpls (07/05/2007).

[8] http://www.j-sim.zcu.cz/

[9] http://www.omnetpp.org/

[10] http://www.tdx.cesca.es/TESIS_UPC/AVAILABLE/TDX-0323104-125205//Chapter2.pdf

[11] http://www.glasssimulator.com/simulators.html

[12] http://www.opnet.com/solutions/

[13] http://www.isi.edu/nsnam/ns/