An Experimental Testbed for Evaluation Topics in Converged Networks

P. H. P. DE CARVALHO[1], H. ABDALLA JR[1], A. M. SOARES[1], P. SOLÍS. BARRETO[2], P. TARCHETTI[2], R. LAMBERT[2], G. AMVAME-NZE[2]
Department of Electrical Engineering, University of Brasilia,
Campus Universitário Darcy Ribeiro - Asa Norte CEP 70910-900
BRASÍLIA - DF – BRASIL

http://www.labcom.unb.br

Abstract: -In this paper we present the development and validation of an experimental platform based in open source software and the development of a package for monitoring purposes. Also we show a group of tests executed on this platform and their results, which intend the validate the functioning and support for future research which includes new ideas in the NGN (Next Generation Networks) topics, such as QoS (Quality of Service) mechanisms, traffic characterization and MPLS (Multi-protocol Label Switching) hybrid routing. We believe that the procedures shown in this paper, may give other research groups an overview of the primary steps for the implementation of a research lab with similar interests.

Key –Words: MPLS, self-similar, multiservice, converged networks, Diffserv.

1 Introduction

This work begins with the implementation of a reliable laboratory for converged networks topics research. This environment was developed during the last year at the Labcom-ENE-FT-University of Brasilia, a laboratory founded from the collaboration of the academic community and the telecom companies, through a national fund (Fundo das Telecomunicações) with the scope to produce new ideas for the network convergence period.

The new branch of telecommunication services expected during and after the convergence process, need the establishment of platforms and management functions in common for the two worlds: data, domined by the Internet Protocol and the traditional telephone system, i.e. circuit switched networks. The research topics include new protocols, performance improvement through new routing paradigms, end-to end quality of service and mobility, between others[1].

Considering the above topics, we designed a testbed configuration, that in a first moment will allow to interact with different protocols, in a second moment will integrate different equipments of different vendors in a close to reality environment, and finally, under simulation studies of different kinds of traffic flows will prove its accuracy and the necessary adjustments for future research.

This manuscript is organized as follows: in section 2 we describe the testbed as well as the equipments involved and their functionality. In section 3 we present a group of tests, which intend the validate our testbed functionability: a comparison of functionality of protocols like MPLS and Diffserv, a failure recovery test for the MPLS and IP environments, the measure of QoS in SIP applications such as VoIP, and finally, the measure of packet losses and latency in the presence of self similar traffic In section 4 we present our conclusions and future work.

2 The Testbed

The testbed structure, named Labcom is shown in figure 1. Basically, it is formed by five different networks: a PSTN (Public Switched Telephony Network), an ADSL (Assymetric Digital Subscriber Line) access network, two local area networks (LANs), a wireless LAN and a MPLS/Diffserv core.

The PSTN is formed with two local exchanges, Tropico RA and a S12, both from Alcatel. The ADSL network, two local area networks and a wireless LAN are both interconnected by the MPLS core, so in this way, we concentrate the traffic from different sources in an unique point. The main goal is to have in the MPLS core the forwarding process of different types of traffic, from several applications and with different

This work was supported by Anatel (Agência Nacional de Telecomunicações), CNPQ (Conselho Nacional de Pesquisa e Desenvolvimento Tecnológico) e CAPES (Conselho de Aperfeiçoamento de Ensino Superior).

QoS needs, which simulates in some manner a real multiservice network. The MPLS/Diffserv core has four routers, based on the Linux Operating System Kernel 2.4.21 and the open source MPLS implementation, developed by the Broadband Communication Networks of the Information Technology Department of the Gent University of Belgium[2].The routers are four computers Pentium IV 2.1GHz and are interconnected by 10/100 Mbps links. The first router, LSR01, connects three LANs to the core, and the LSR03, via a radio link of 2 Mbps, connects the fourth LAN. The routers LSR02 and LSR04 are the forwarding elements of the core.
The VoIP service was implemented using an open source VOCAL 1.5 server[3].
The MPLS core has support for Diffserv as well as MPLS tunnels establishment. The Diffserv feature allows traffic classification such as BE (Best Effort), AF (Assured Forwarding) and EF (Expedited Forwarding), implemented as queuing disciplines. To automate the operating features of the MPLS core, web based software has been implemented with Perl scripts, SNMP (Simple Network Management Protocol), TELNET and MRTG tools. The Performance of MPLS can be accessed for SLA(Service Level Agreement) policies, Diffserv and tunnels set-up.
The QoSLabCom web page, as we named the monitoring tool, has several links for choosing diffserv's queuing disciplines. Each router receives 3 AFs classes, 1 EF, 1 BE and, 1 OR (OSPF & RSVP) at each Ethernet interface card, using a specific command. All routers at the MPLS core receive the same configuration, as the incoming and outgoing traffic should obey to the same policies restrictions. Packets can be marked according to their TOS/DSCP (Type of Service/Diffserv Code Point) target fields for example using the iptables command suite.

3 The Experiments

3.1. MPLS and Diffserv

For the MPLS and Diffserv tests some practical adjustments were made, which appeared to be important for the production of results that closely fit reality.
The first adjustment concerns a bandwidth reduction to ease the data processing of traffic results. Since the links have 10/100 Mbits, and we wanted to produce an overloaded system, with losses and delays, we reduced them to 1.2 Mbps.

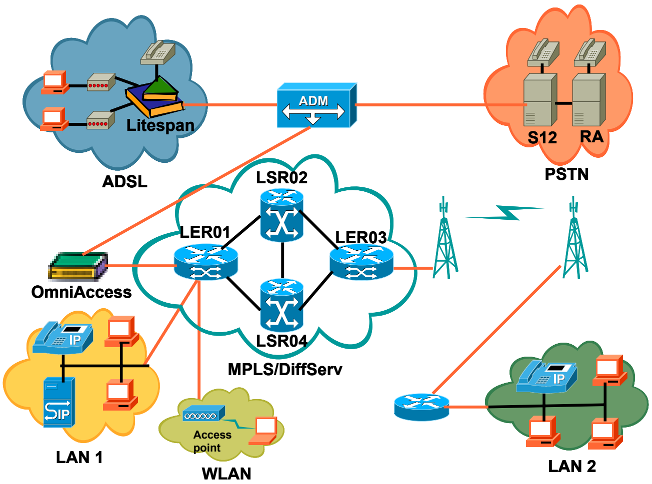

Fig.1 The Testbed

The configuration files for this purpose were implemented using the CBQ (Class Based Queuing) discipline for traffic control in Linux systems.
The second adjustment concerns to traffic patterns. We work with four applications. The first two applications are called "disturbing" and the other two are called of "evaluation". The disturbing applications produce VBR (Variable Bit Rate) traffic patterns, with several bursts, in order to produce an overload of links in random intervals. The evaluation applications are CBR (Constant Bit Rate) traffic patterns, as a simulation of VoIP and video streaming applications.
In table 1 appears a consolidation of the traffic patterns used for the experiment. The time interval for all traffics is 60 seconds. Both of traffics VBR have periodical bursts with an exponential distribution. For pattern VBR1 we define periodical bursts of 0.5secs on intervals of 3secs. For VBR2 the burst lasts 1secs in intervals of 5secs.
For the DiffServ/MPLS scenario, the same network adjustments are preserved but with different traffic patterns, as shown in table 2. The main change concerns an additional CBR traffic classified as BE
In figure 3 and 4 are shown the measures of latency and packet losses for the MPLS experiment. In figure 5 and 6 are shown the same measures for the MPLS/Diffserv experiment.

Table 1. Traffic patterns for MPLS

Type	Bandwidth	Packet size
CBR1	64Kbps	256 bytes
CBR2	384Kbps	512 bytes
VBR1	1000Kbps	1024 bytes
VBR2	1000Kbps	1024 bytes

Table 2. Traffic patterns for MPLS/Diffserv

Traffic	Bandwidth	Type
CBR1	64Kbps	EF
CBR2	384Kbps	AF11
CBR12	1000Kbps	BE
VBR2	1000Kbps	AF21

Table 3. Traffic flows

Traffic flow ID	Packet size	Data Rate (in kbps)
CBR01	256 KB	384
CBR02	512 KB	64
CBR03	300 KB	384
CBR04	256 KB	800
VBR01	1024 KB	500

Comparing the latency graphs, we observe that there is a processing overhead due to the packet classification process which results in a higher latency for the MPLS/Diffserv implementation.
Regarding the packet losses, we verified that the higher packet losses for unprivileged flows in the Diffserv scenarios also show that the priority procedure works as expected.. In both experiments, the packet losses have a consistency with the adopted politics. This behavior can be an important issue for the definition of billing politics and marketing programs.

3.2. Failure Recovery

In the failure recovery process, we aim to evaluate the testbed for recovery times in a link fault event and compare if the reactive process of LSP recovery has advantages in latency metrics and packet losses over the same process in an IP network.
The traffic flows are specified in table 3. We work with five sources of traffic. The VBR01 is a VBR (Variable Bit Rate) traffic pattern with several bursts that attempt to overload the link in random intervals. The VBR traffic has periodical bursts with an exponential distribution. Each burst lasts 0.5secs on intervals of 3secs.
The CBR01, CBR02, CBR03 and CBR04 are CBR (Constant Bit Rate) traffic patterns from applications such as VoIP and video streaming.

This first scenario simulates an IP network with Best Effort politics. We decided to create routing tables with more or less 65500 entries on each router. Also, the routing cache was extremely reduced to less than 256 entries and garbage loop traffic of 640 KB was simulated on each router to overload the routing cache update process. We decided to create an overloaded cache instead of disabling the cache[4], because we believe this approach simulates in a better way a real network environment in which nodes usually operate with certain process load.
The IP network uses the OSPF protocol for routes advertisement. Considering the OSPF operation, the standard values for the hello interval and the dead interval are 10 seconds and 40 seconds respectively. So, after 40 seconds without a reload, a LS (link state) packet is sent with information about other routes.
In our experiment we tested a reduction of the hello intervals and dead delay intervals values to 2 seconds and 5 seconds respectively. We used these values to produce an experimental result, but lower values could be used. We observed that in a loaded link, if these values are very small, the functionality of the OSPF protocol may interpret missing advertising packet as an unreachable network situation.
For the MPLS environment, we defined three LSPs on the network. Table 4 shows the LSP configurations and the traffic mapping for each flow which intend to provide an initial load balance of traffic in the backbone.
Regarding the open source implementation used for this experiment, the LSPs can be established in two ways: static and dynamic. For the static feature, the nodes that form the LSP should be specified. For the dynamic feature, the route information for a destiny provided by the OSPF protocol is used. After the experimentation of different configurations, we verified that the dynamic LSP establishment does not work satisfactory in the open source implementation. This feature misses the migration of the traffic mapping from the LSP that goes down to the substitute LSP, and even when there exists redundancy (an alternative LSP with the same mappings) the software is unable to redirect the traffic flow on the redundant LSP. We implemented this feature in the MPLS platform, as shown in the following algorithm:

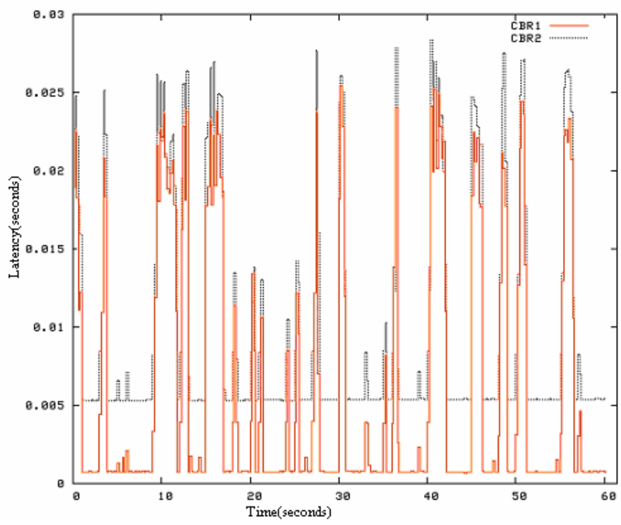

Fig. 3. Latency – MPLS

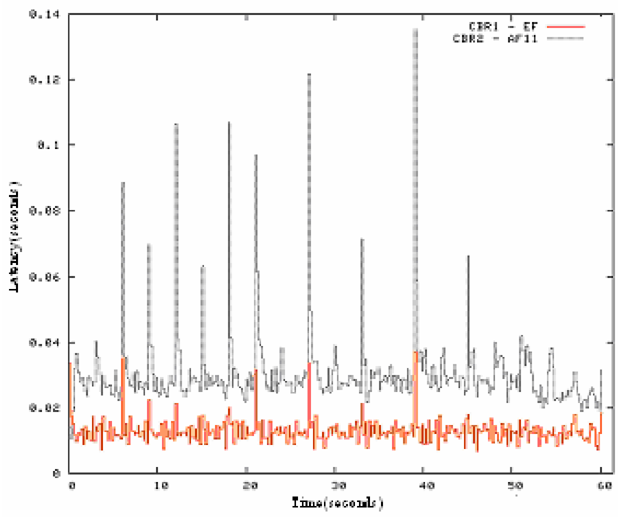

Fig. 4. Latency – MPLS/Diffserv

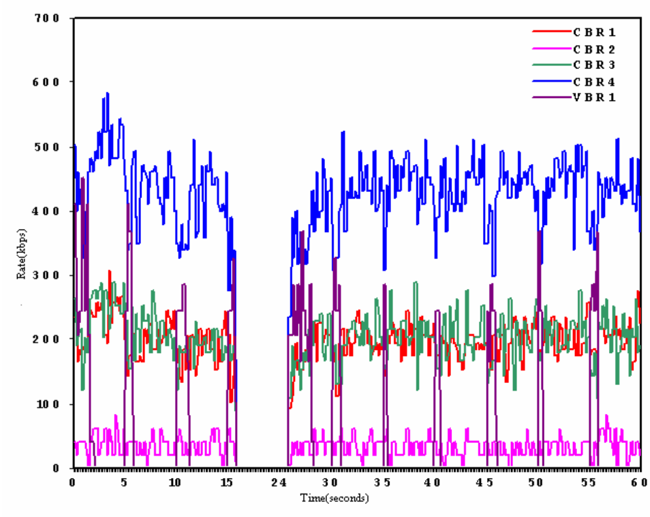

Fig. 5. IP Network Failure interval

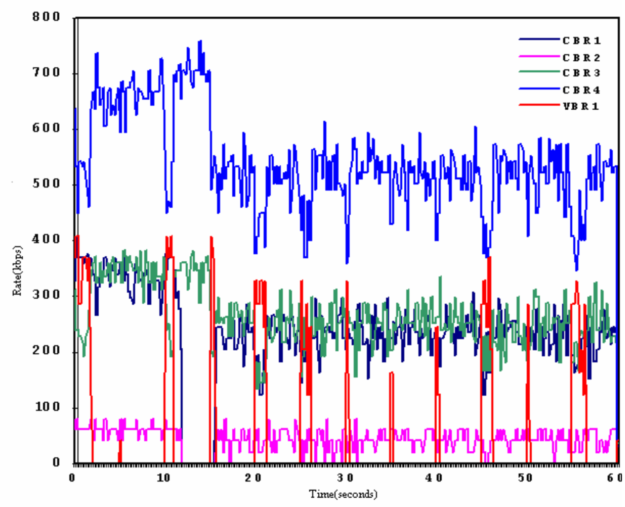

Fig. 6. MPLS : Failure interval

```
While (true)
 Listen the LSA packets
 LSP_falha = matches_node (LSA, LSP
table)
 For each LSP_falha
 Then Get LSP_R =
LSP_Redundancy(LSP_falha(i))
Remap_flows (LSP_falha(i), LSP_R)
```

To measure the failure recovery time the duration of the traffic flows is 60 seconds, and within this interval, a link crash occurs specifically between the 10 - 20 seconds time interval. This crash is showed in figure 5 and figure 6 for both IP and MPLS experiments respectively.

Regarding the latency measures we can see that the results show that the IP experiment has a lower value than the MPLS experiment. This result was unexpected since in previous experiments we observed that the latency in the MPLS platform was lower than in the IP platform. The latencies results show that that the mean latency measures for the MPLS platform are better than the ones in the IP platform. Regarding the losses, for the IP network the period of losses is greater and all the traffic flows suffer losses. We observed that after the link failure, the behavior of losses remains the same. For the MPLS platform, the 100% losses occur only for CBR1 and CBR2 flows in the failure interval and the losses appear to increase after the failure.

For every flow, the MPLS platform shows a better performance regarding the losses percentage.

The results show that the MPLS environment has a better recovery time in the event of a failure as well as less packet losses, but has a higher latency. Even when the difference of latency between the IP platform and the MPLS platform is little, we have to consider that the routing process of the IP platform is much more of a hard labor.

Table 4. LSP mappings

LSP ID	LSP nodes	Traffic mappings (see table 2)
100	1,2,3	CBR01, CBR02
200	1,4,3	CBR03, VBR1
300	1,4,2,3	CBR4

We interpret these results as a specific characteristic relative to the link failure and the traffic mapping process for each LSP. In the previous experiment, shown in section 3.1, this behavior did not appear. But we worked with an unique LSP and all the traffic mapped on it. Without remapping the flows, the MPLS platform showed a better latency result than the IP platform.

3.3 VoIP Experiments

The goal in these experiments is to evaluate the behavior of multimedia traffic in the testbed, especially the VoIP traffic. Within this section, we present the results of experiments which analyze the latency and packet losses of a VoIP application, as well as the MOS (Mean Opinion Score)[5][6].

Our interest in the MOS evaluation is to verify the influence of using a certain decoder, so we also analyze two different decoders available in the CISCO 7490 IP phones.

The VoIP service, was implemented using a VOCAL 1.5 server, which uses the SIP (Session Initiation Protocol) to implement an IP telephony system. The UA (User Agents) used in this experiment are IP CISCO 7940, but any software UA could be used.

We configured two environments for the tests: an MPLS/Best Effort (BE) and a MPLS/Diffserv, as well as in the experiments shown in section 3.1, and using the same traffic patterns, as shown in table 1 and table 2. The test evaluates the quality of a 60 seconds duration VoIP phone call, executed simultaneously with the traffic flows of table 1. All traffic flows from LAN1 to LAN2 (see figure 1).

For the MPLS/Best Effort environment the MOS results for coders G711 and G729 were 2,5 and 1,8, respectively. The results of latency and packet losses are already shown in figures 3 and 4.

For the MPLS/Diffserv we used the traffic patterns of table 2, and the results were for the G711 and G729 coders 4,3 and 3,5 respectively. The latency and packet losses are already shown in figures 5 and 6.

We can verify that the results of loss and latency correspond to the user's perception. Exists an influence regarding the type of decoders but a good network performance is fundamental for a fine subjective evaluation.

3.4 Self Similar Traffic

In this platform we are also interested to study more widely the influence of self-similar traffic in the performance of real time applications, as well as the chance to manipulate the different traffic flows using their self-similarity characteristics in order to improve the network performance. So, we did an experiment to validate the testbed in the presence of traffic with different levels of burstiness.

It is known that in the case of bursty traffic, the aggregation of different flows also results in a bursty traffic, i.e with an H parameter higher that 0.5. This behavior is the opposite of what happens for Poisson traffic, in which the aggregation of different traffic flows tend to smooth the resulting flow[7].

To verify this premise in our platform, we propose the evaluation of a CBR application, in terms of latency, jitter and packet losses in a MPLS core in two different situations: a) with 10 aggregated traffic flows, with a resulting H equal to 0.8, in a resulting average data rate between 1,1Mbps and 1,2Mbps routed in a unique LSP, the LSP01, in the MPLS core; b) with 14 aggregated traffic flows, with a resulting H between 0.5 and 0.55 (low burstiness) with an average data rate between 1.1Mbps and 1.2Mbps, routed as in (a). For both experiments, we used a synthetic self similar traffic generator and the CBR application is a 512kbps synthetic flow with packet sizes of 256 bytes.

For this particular experiment, we did not limit the links bandwidth capacity to 1.2 Mbps, since the queuing disciplines used for this purpose, appeared to disturb the self-similarity characteristics of the original traffic flows.

In both cases we collected 50,000 packets to analyze the self similarity which represents a time interval of 10 minutes All the traffic sources were in LAN1 (see figure 1) and the targets are in LAN2 in several ports. The traffic trace of the intended self similar traffic is collected in LSR01 and the H parameter was measured using the method verified in [9].

The bandwidth utilization has an average data rate of 1.1 Mbps. The calculus of the H parameter for the flow in 0.1 sec time scale is 0.54, using the method of the variance time plot and the method verified in [8].

The traffic trace of the second experiment has an H parameter of 0.8 calculated in a 0.1 seconds time scale. A very little variation was observed in the value of the H parameter in different time scales which is the time invariance behavior expected for the self similar traffic. On table 5 we show the results of mean latencies and losses for this experiment. The results confirm the expectation of higher packet losses and higher latency since we have a higher H parameter. The latency shows twice the mean latency of the first experiment and the packet losses are five times higher. This result has coherence with results obtained in some other research works[9], and for our case, validates the functioning of the testbed.

Table 5: Comparative Losses and Latency

	Mean Latency	Mean Packet Losses (%)
H=0.8	0.029	3.6
H=0.55	0.013	0.7

4 Conclusions and Future Work

In this work we showed a set of experimental tests that intend to validate the functioning of an environment for research topics for the NGN era. The results of all tests, mainly made with open source tools, showed consistency and were predictable, in comparison with some other works in the area.

As verified in our tests, the benefits offered by the technologies available in present networks may not cover the incremental needs of future services. We consider that the initial point for the development of new technologies, protocols and algorithms is based on reliable testbeds, with support to several protocols, diversity of equipments, networks and protocols.

Also we concentrate part of our effort in the open source software development that may have the contribution and exchange of several groups. We hope that the methodology and results showed in this work may offer some ideas for the development of other comparative testbeds that may become in the future interesting points of exchange for the development of the converged network technologies.

References:

[1] Lambert R.., *Evolução das Tecnologias e Protocolos para Comunicação Multimídia*, Master Thesis, Departamento de Engenharia Elétrica, Universidade de Brasília, Brazil., March 2004)

[2] http://dsmpls.atlantis.rug.ac.be

[3] http://www.vovida.org

[4] S. Avallone, M. Esposito, A. Pescapè, S.P. Romano, G. Ventre, *An Experimental Analysis of Diffserv-MPLS Interoperability*, Proceedings of the 10th International Conference on Telecommunications (ICT 2003), - Papeete , 2003.

[5] ITU-T: Recommendation P.800: *Methods for subjective determination of transmission quality*. Genève, August 1996

[6] ITU-T Recommendation G.729. *Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP).*Genève, March 1996

[7] Leland W. Taqqu M, & Willinger W, *On the self-similar nature of Ethernet Traffic*, Computer Communications Review, 23 . Proc. Of the ACM/SIGCOMM 93, San Francisco, pp 183-193.

[8] Kettani H. & Gubner J. *A Novel Approach to the Estimation of the Hurst Parameter in Self Similar Traffic,* Proceedings of IEEE Conference on Local Computer Networks, Tampa, Florida. (November 2002).

[9]Sahinoglu Z. & Tekinay S.,. *On Multimedia Networks: Self-Similar Traffic and Network Performance,* IEEE Communications Magazine, January 1999, pp 48 – 52.