Layer 4 Switching Experiments with IPv6 versus IPv4

Virgil Dobrota, *Member, IEEE,* Daniel Zinca, *Member, IEEE,* Cristian Mihai Vancea
Technical University of Cluj-Napoca, Department of Communications,
26 Baritiu Street, 3400 Cluj-Napoca, Romania, Tel/Fax: +40-64-197083
e-mail: {Virgil.Dobrota, Daniel.Zinca, Mihai.Vancea}@com.utcluj.ro

Abstract **- This paper is focused on a Layer 4 switching experiments of IPv6 over Fast Ethernet, running under Windows 2000 Professional. Prior to our studies on IPv4 over ATM with TCP relaying, firstly introduced at IEEE LANMAN'99, we are trying to evaluate the performances at the interface between the applications and the nonblocking stream-oriented sockets in TCP/IP. The first major objective is to get consistent results for a IPv6 versus IPv4 debate, even the implementation phase of this new version of the Internet Protocol is under progress.**

Index Terms **- burst traffic, Fast-Ethernet, IPv4, IPv6, Layer 4 switching, TCP/IP**

I. INTRODUCTION

To help the industry-wide conversion from IP version 4 (IPv4) to IP version 6 (IPv6), Microsoft has announced a four-phase execution plan. The current phase (delivery of a preview IPv6 stack for applications conversion) will be followed by the delivery of a pre-production version for laboratory testing and, finally, the production release to be deployed [5].

All the experiments, concerning both IPv6 and IPv4 presented herein, involved Fast Ethernet technology and Windows 2000 Professional. An updated version of the software tool from [1],[2] offered the facilities to evaluate the sending time, the receiving time and the elapsed time at the interface between the application and the non-blocking stream-oriented socket in TCP/IP. Obviously each Layer 3 protocol version requested its own TCP implementation on top of it.

Although the types of models applied have a great influence on the overall performances, it is not the subject of this paper to present the reasons to choose them. The reader is kindly advised to read [2],[3] for more details.

To understand the experiments carried out by comparing IPv6 implementation to currently running IPv4, under the same physical and data link testing conditions, a short note is necessary.

According to the approved commentaries regarding the OSI Reference Model, transport relays could not guarantee the transport service, except under very constrained circumstances [4]. Therefore it was generally agreed that the Layer 4 switching is prohibited. Despite of this statement, the results presented in [1],[2] demonstrated that the TCP/IP environment should be exploited by involving scheduling and relaying mechanisms even at the transport layer.

Recently, Cisco Systems Inc. has introduced its own concept of Layer 4 switching, rather different from that one we are using in this paper. They have implemented a Server Load Balancing (SLB) over Layer 3 switching for their Fast-Ethernet/Gigabit-Ethernet Catalyst 4840G. Cisco's Layer 4 switch is a re-distributor of the requests and hits from clients evenly among all the server in the server farm, in order to achieve a balanced load for each server. It was mainly designed for increasing Web traffic and access reliability of multiple Web servers, offering the appearance of one virtual server, with one IP address and a single Universal Resource Locator (URL) for an entire server farm [6].

In this paper, we are trying to obtain better results in TCP/IP for burst traffic by involving departure schedules for TPDUs (Transport Protocol Data Units). This means that the applications should not send the information directly to the sockets without taking into account the non-linear behaviour of the TCP/IP entities within a broadband network. On the other hand, at the server site, a pure Layer 4 switching will be performed, by redistributing the TPDU from the incoming socket to the outgoing socket, as faster as possible, without any additional scheduling or checking. As soon as the optimum model, i.e. a frame departure schedule, will be determined for a given application, under a given network, it is for sure that a Layer 4 switching schedule (or at least a QoS mechanism) has to be added at the server site, too.

II. TESTING CONFIGURATION AND FILES

Due to the fact that Microsoft's IPv6 implementation is based on Windows 2000 technology only and the available ATM cards drivers (VIRATAlink) were written for Windows95/98/NT only, we were forced to perform the experiments on Fast Ethernet, instead of ATM.

Let us suppose the most favorable networking conditions, i.e. there will be no other workstations connected, except those involved in trial. As the entire bandwidth is at our disposal, without unexpected collision or congestion, obviously the results presented herein could be considered as the maximum we can get from the network.

The testing configuration in *Figure 1* included three workstations connected to the 100 Mbps ports of HP ProCurve hub. The most powerful station within the tested network was based on Intel's Pentium II/400 MHz, running the server and acting as a Layer 4 switch. The client software was installed on two different workstations (with Celeron 366 MHz and Pentium 233 MHz MMX). Note that by the time this experiments were done, more powerful machines would have been available, but it was decided to keep the same testing configuration as in [1],[2] for IPv4 over ATM.

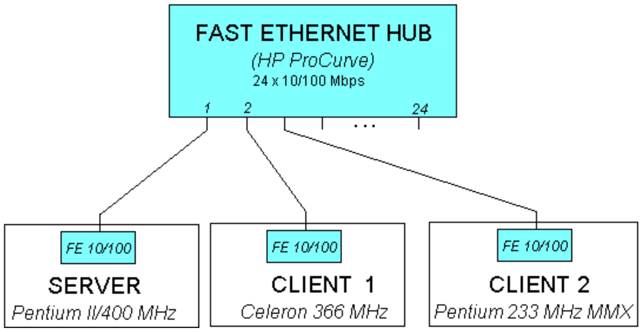
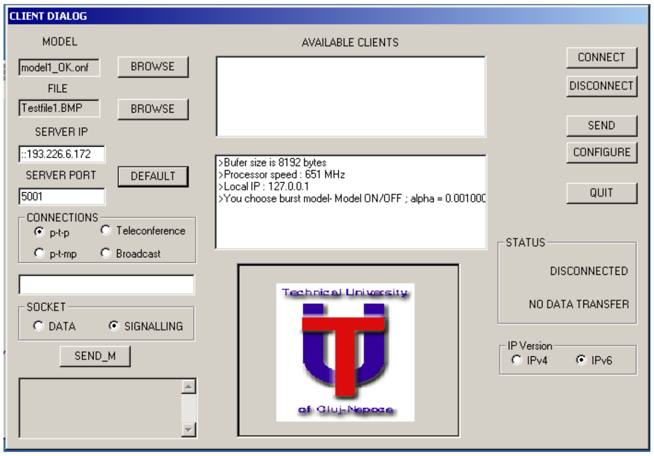


Figure 1. The testing configuration

The *Testfile1* presented in *Figure 2* was intentionally chosen due to several reasons: a) It is a part of the stream used for the study of video sources; b) It has a number of bytes which is less than the implicit buffer size for TCP socket sends (8192 bytes); c) It is suitable for those models requiring the sending within one single burst, which is in fact the current pattern used by the existing applications to communicate with the sockets (i.e. without any model).

Figure 2. Testfile1 (7,990 bytes)

Being larger than the previous one, the *Testfile2* is suitable for on/off models.



MODEL model1_OK.onf

FILE Testfile1.BMP

SERVER IP ::193.226.6.172

SERVER PORT 5001

CONNECTIONS ● p-t-p ○ Teleconference ○ p-t-mp ○ Broadcast

SOCKET ○ DATA ● SIGNALLING

AVAILABLE CLIENTS

```
>Bufer size is 8192 bytes
>Processor speed : 651 MHz
>Local IP : 127.0.0.1
>You choose burst model- Model ON/OFF ; alpha = 0.001000
```

STATUS DISCONNECTED NO DATA TRANSFER

IP Version ○ IPv4 ● IPv6

Figure 3. Screen capture of the client's GUI used as Testfile2 (240,118 bytes)

The evaluation's accuracy of the proposed software tool (client and server) is given by the clock period of the CPU (2.5 ns. at Pentium II/400 MHz). The measurement of the sends and receives on the sockets is also dependent on RDTSC *(Read Time Stamp Counter)* and other instructions included in the loop. Obviously the processes are guided by the TCP/IP entity, as we rely on the Windows Sockets *select* function to determine the status of the sockets and to perform synchronous I/O.

III. EXPERIMENTAL RESULTS

The first experiments are dedicated to the influence of the application's buffer size (see *Table1* and *Table2*).

Applica-tion's buffer size [Bytes]	Average switching time *(IPv4)* [μ s]	Average switching speed *(IPv4)* [Mbps]	Average receiving time *(IPv4)* [μ s]
8192	111	575.84	217
5000	266	240.30	721
3000	333	191.95	1074
1500	617	103.59	2458
750	937	68.21	4945
375	1738	36.77	9502

Table 1. Testfile1, point-to-point, Celeron366->server -> Celeron366, without model. The average sending time/ throughput was 165 μ s/387.39 Mbps for IPv4.

Note that the application's buffer for sending information and the application's buffer for receiving information are different from those of Windows Sockets related to TCP/IP. The last ones could be modified through *setsockopt* function (integer values SO_SNDBUF and SO_RCVBUF).

We tried also the influence of disabling the Nagle's algorithm (by enabling TCP_NODELAY option), but the general suggestion is to leave it enabled (by default).

Application's buffer size [Bytes]	Average switching time *(IPv6)* [μ s]	Average switching speed *(IPv6)* [Mbps]	Average receiving time *(IPv6)* [μ s]
8192	59	1083.38	102
5000	317	201.64	565
3000	486	131.52	1371
1500	N.A.	N.A.	N.A.
750	1249	51.17	5498
375	2601	24.57	10843

Table 2. Testfile1, point-to-point, Celeron366->server->Celeron366, without model. The average sending time/ throughput was 264 μ s/242.12 Mbps for IPv6.

The results for the application's buffer size of 8192 bytes are very important for evaluating the highest Layer 4 switching speed of about 575 Mbps (IPv4), respectively 1083 Mbps (IPv6). In general, supposing a theoretical transmission throughput of 100 Mbps (Fast Ethernet), it seems that the speed advantage is greater than 1 for a buffer size of at least 1500 bytes. Due to the protocols stack the actual sending or receiving throughput at the lower layers cannot reach the upper bound of 100 Mbps.

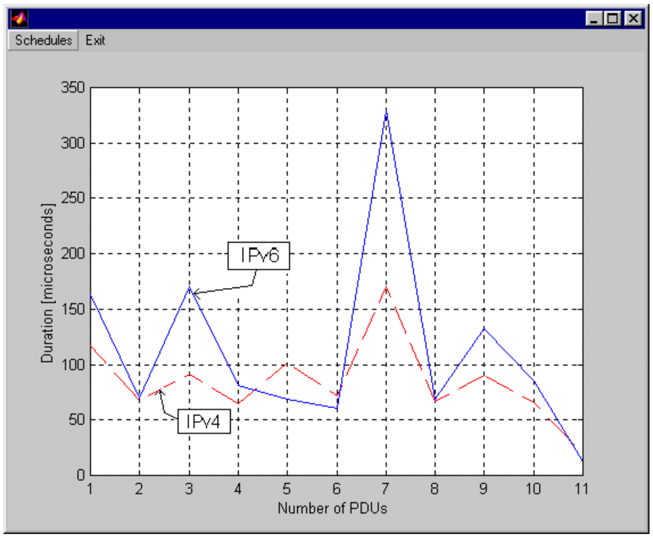
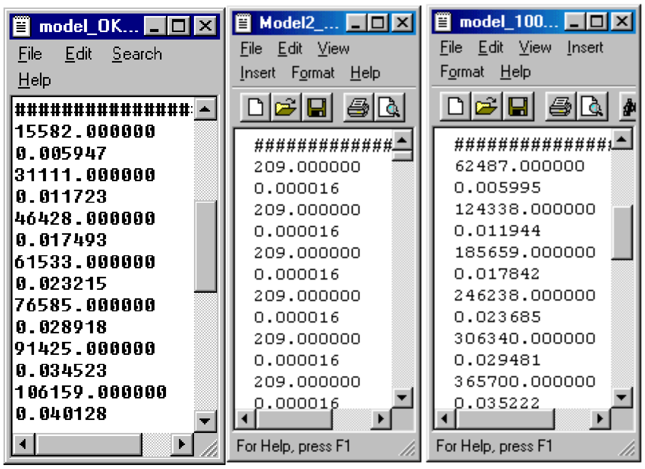


Figure 4. Switching time for application's buffer size of 750 bytes, Testfile1, point-to-point, Celeron366->server -> Celeron366, without model

Apparently, there is no advantage of using IPv6 instead of IPv4, as shown in *Figure 4*. This is a preliminary conclusion because several additional experiments should be performed. Let us involve now departure schedules.



```
################
15582.000000
0.005947
31111.000000
0.011723
46428.000000
0.017493
61533.000000
0.023215
76585.000000
0.028918
91425.000000
0.034523
106159.000000
0.040128
```

```
#############
209.000000
0.000016
209.000000
0.000016
209.000000
0.000016
209.000000
0.000016
209.000000
0.000016
209.000000
0.000016
```

```
###############
62487.000000
0.005995
124338.000000
0.011944
185659.000000
0.017842
246238.000000
0.023685
306340.000000
0.029481
365700.000000
0.035222
```

Figure 5. Model 1, Model 2, Model 100 for burst traffic

Note that there are two articles for each ON+OFF period. The first article represents the number of bytes during the burst (for example 15582 bytes in *Model 1*, 209 bytes in *Model 2*, 62487 bytes in *Model 100*). The second article is the total duration of the ON+OFF period (for example 0.005947 seconds in *Model 1*, 0.000016 seconds in *Model 2*, 0.005995 seconds in *Model 100*). Actually *Model 1* was designed for 25.6 ATM, as in [1],[2], but it seems that is also suitable for Fast Ethernet.

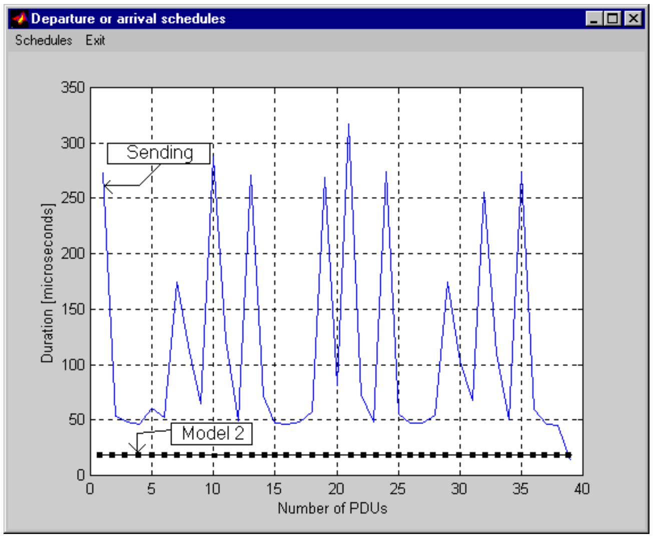


Figure 6. Model 2 for IPv6, Celeron 366 -> server-> Celeron 366, testfile1. The sending TCP entity cannot follow Model 2

Model 100 is the updated version of *Model 1*, but it is rather difficult to be accurately followed by the sending TCP entity,

Interval	Measured time [μs]	Throughput [Mbps]
s_f - s_i SENDING	18543…18877 (CELERON 366)	101.76..103.59
SWITCHING	74493…74643 (P400)	25.73…25.78
r_f - r_i RECEIVING	82671…84942 (P233MMX)	22.61…23.23

Table 3. IPv4, point-to-point, Celeron366->server-> Pentium233MMX, Testfile2, application's buffer size = 5000 bytes. The planned sending time/ throughput were 19209 μ s/100 Mbps without model and for Model2, respectively 22202 μ s/86.52 Mbps for Model100

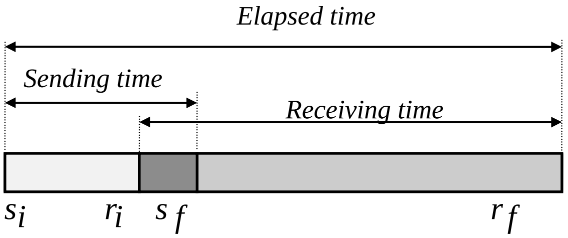


Figure 7. Four time stamps for measuring the sending, receiving and elapsed times at the client site

Note that the elapsed time could be evaluated only if the sending and receiving entities are on the same machine, otherwise a very complicated synchonization mechanism for time stamps should be involved. At the server site, the switching time is defined as the interval since the reception of the first TPDU started and the transmission of the last TPDU ended.

IV. CONCLUSIONS

1. The preliminary comparison between IPv6 and IPv4 proved that there are no relevant differences concerning the throughputs and the Layer 4 switching performances. This observation is valid for both Microsoft's Windows 2000 implementation (as from this paper) and Linux-based solution (as from our previous work).
2. It is more difficult to choose a departure schedule for Fast Ethernet comparing to equivalent conditions in ATM.
3. The Layer 4 switching performances could be improved by selecting the proper model at both sending client and the server.
4. Many users have an unrealistic expectation about the overall throughput, calculated within the interval since the first bit left the sender until the last bit reach the destination. The highest value, calculated at the application/Windows Sockets interface, is about 20…25 Mbps for 100 Mbps Fast Ethernet (i.e. 20-25% efficiency, compared to ATM's maximum efficiency of about 40%).
5. Anyway these results are better than the expected ones got by involving the classical one-block sending mechanism through sockets.
6. Supposing that no acquisition of the receiving data is performed, the overall throughput could be doubled, but this situation has no use in practice.

V. FUTURE WORK

Obviously the next step is to determine the proper model, depending on the specific application (burst traffic, voice, variable video streams etc). The overall performance of the Layer 4 switching is expected to be improved by running it on top of Layer 2/Layer 3 switches on the same machine.

REFERENCES

[1] V. Dobrota, D.Zinca, C.M. Vancea, A. Vlaicu - "Layer 4 Switching Experiments for Burst Traffic and Video Sources in ATM", *Proceedings of the 10th IEEE Workshop on LANMAN'99*, Sydney, Australia, 21-24 November 1999, pp.66-69.

[2] V. Dobrota, D. Zinca, C.M. Vancea, A. Vlaicu, "Layer 4 Switching Experiments in a TCP/IP Environment for the ATM Sources", *ACTA Tehnica Napocensis*, ISSN 1221-6542, Vol.40, No.1, 2000, pp. 13-18.

[3] V. Dobrota, *Retele digitale in telecomunicatii. Volumul 2: B-ISDN cu ATM, Sistemul de semnalizare cu canal comun SS7*, Editura Mediamira, Cluj-Napoca 1998.

[4] J. Day, "The (Un) Revised OSI Reference Model", *ACM/SIGCOMM Computer Communication Review*, vol.25, pp.39-55, No.5, October 1995

[5] ***, *http://msdn.microsoft.com/downloads/sdks/ platform/tcpip6.asp*

[6] ***, *http://www.cisco.com/univercd/cc/td/doc/product/ l4sw/index.htm*