LAYER 4 SWITCHING IN A TCP/IP ENVIRONMENT FOR THE ATM SOURCES

Virgil DOBROTA, Daniel ZINCA, Cristian Mihai VANCEA, Aurel VLAICU
Technical University of Cluj-Napoca, Department of Communications
26 Baritiu Street, 3400 Cluj-Napoca, Romania
Tel: +40-64-191689, 195699/208 Fax: +40-64-191689
E-mail: {Virgil.Dobrota, Daniel Zinca, Mihai.Vancea, Aurel.Vlaicu}@com.utcluj.ro

Abstract: This paper is focused on the results of a Layer 4 switching experiment, aiming to evaluate the performances at the interface between the applications and the nonblocking stream-oriented sockets in TCP/IP. One major objective is to apply the traffic models for burst traffic and video sources, initially designed for ATM sources, to user applications requesting transport layer services.

Keywords: *ATM, burst traffic, Fast-Ethernet, Layer 4 switching, TCP/IP, video sources*

I. INTRODUCTION

Prior to our studies on Fast Ethernet and ATM traffic parameters, presented at LANMAN'96 and LANMAN'98, we are trying to obtain better results for burst traffic and video sources by involving departure schedules for cells or frames. This means that the applications should not send the information directly to the sockets without taking into account the behaviour of TCP/IP entities within a broadband network. Preliminary results of this work were presented at LANMAN'99 [1]. We have selected the real-time experiments, carried out on both Classical IP over ATM and IP over Fast Ethernet, in order to get the answers to the following questions:

1. *Is it possible to apply ATM traffic models to the TCP/IP environment?*
2. *Which are the advantages of Layer 4 switches implemented by software for point-to-point, point-to-multipoint and broadcast services?*
3. *What is the influence of the lower layers technologies against the transport layer exchange of information?*

II. MODELS FOR BURST TRAFFIC AND VIDEO SOURCES

The first paragraph is devoted to burst traffic generated by ON/OFF sources of constant throughput. A Matlab-based scheduler is able to determine the number of ON cells to the number of OFF cells ratio, for every burst, until the transmission process is completed [2].

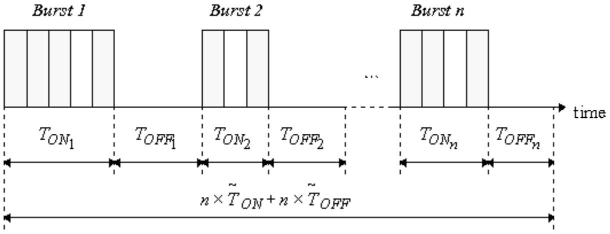


Figure 1. Geometrical distribution of burst traffic

Due to different types of correlation between successive frames, the video services are mainly different than voice and data, involving discrete-state continuous-time Markov models. *M1_X* is the unidimensional model, whilst *M2_X* is the bidimensional one. *X* represents the type of experiment:

(A) Probability of being in a given state versus average throughput (state *i*, where *i=0,1,...N* for unidimensional model, or state *(i,j)*, where *i=0,1,...N-low* and *j=0,1,..N-high*, for bidimensional model);

(B) Average throughput D versus activation/deactivation rates (α , β for unidimensional model, respectively α , β , γ , δ for bidimensional model);

(C) Average throughput D versus probability of being in a given state.

A detailed description of these video models is given in [3],[4].

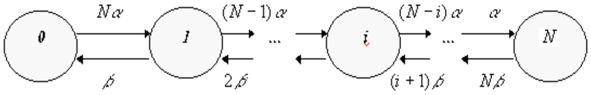


Figure 2. Unidimensional discrete-state Markov model

III. TESTING CONFIGURATION AND FILES

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 collisions or congestions, obviously the results presented herein could be considered as the maximum we can get from the network.

The testing configuration included four workstations connected either to ATM 25.6 Mbps ports of VIRATAswitch 1000, either to Fast Ethernet 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 three different workstations (with Celeron 366 MHz, Pentium 233 MHz MMX and Pentium 120 MHz). Note that these machines were not connected simultaneously to ATM and Fast Ethernet, in order to avoid the uncontrolled influences.

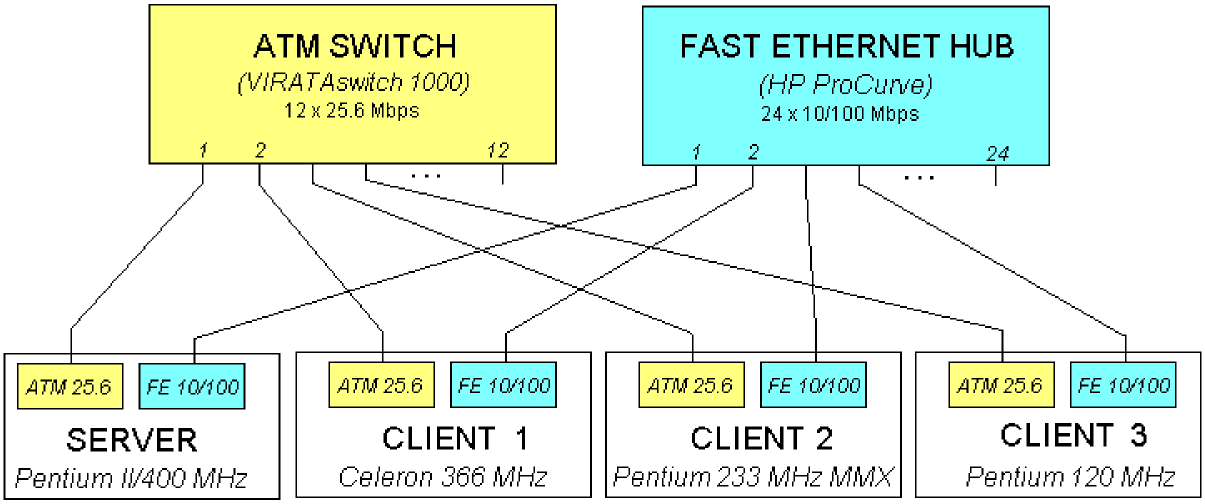
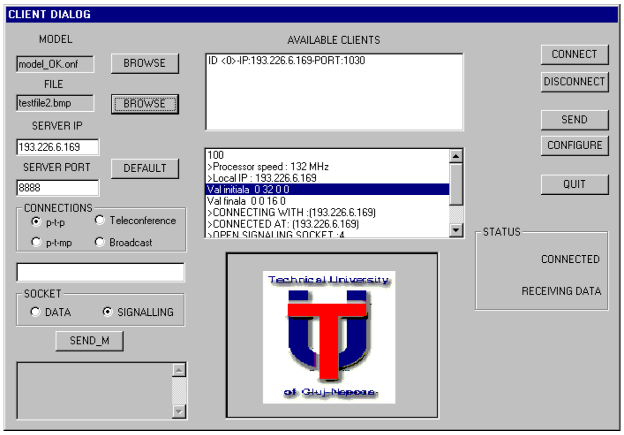


Figure 3. The testing configuration

The *testfile1* presented in *Figure 4* 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 (8 KB); 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 4. Testfile1 (7,990 bytes)

Being larger than the previous one, the *testfile2* is suitable for on/off models.



MODEL: model_OK.onf

FILE: testfile2.bmp

SERVER IP: 193.226.6.169

SERVER PORT: 8888

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

SOCKET: ○ DATA ● SIGNALLING

```
ID <0>-IP:193.226.6.169-PORT:1030
```

```
100
>Processor speed : 132 MHz
>Local IP : 193.226.6.169
Val initiala  0 32 0 0
Val finala  0 0 16 0
>CONNECTING WITH :(193.226.6.169)
>CONNECTED AT: (193.226.6.169)
>OPEN SIGNALING SOCKET :4
```

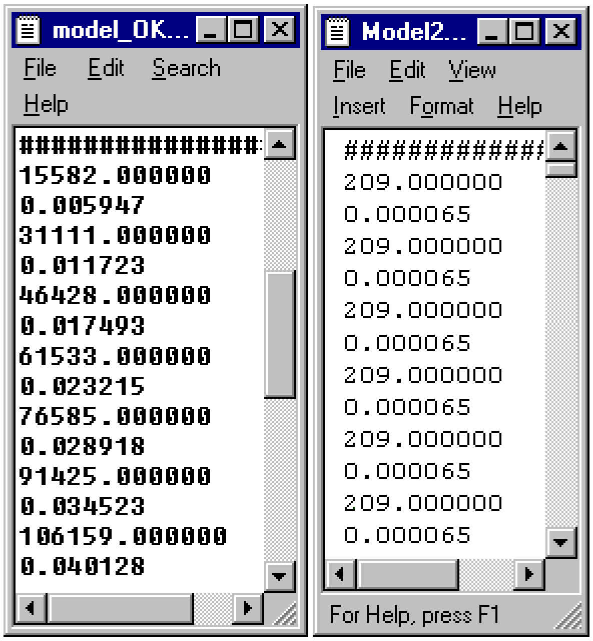
Figure 5. Screen capture of the client's GUI used as Testfile2 (240,118 bytes)

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).

The evaluation 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.

IV. EXPERIMENTAL RESULTS FOR BURST TRAFFIC

The first experiments are dedicated to the study of burst traffic generated by the ON/OFF sources. *Figure 6* presents the numerical description of *Model 1* and *Model 2*. 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*). The second article is the total duration of the ON+OFF period (for example 0.005947 seconds in *Model 1*, 0.000065 seconds in *Model 2*).



```
################
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
0.000065
209.000000
0.000065
209.000000
0.000065
209.000000
0.000065
209.000000
0.000065
209.000000
0.000065
```

Figure 6. Model 1 and Model 2 for burst traffic

The models should take into account the Layer 4's specific behaviour. For instance the TCP entity is able to follow the *Model 1* for both Classical IP over ATM and IP over Fast Ethernet, as in *Figure 7*. However, the actual sixth PDU is different from the model for the very simple reason that it collected the remained bytes from *testfile2*, after the sending of previous 5 PDUs.

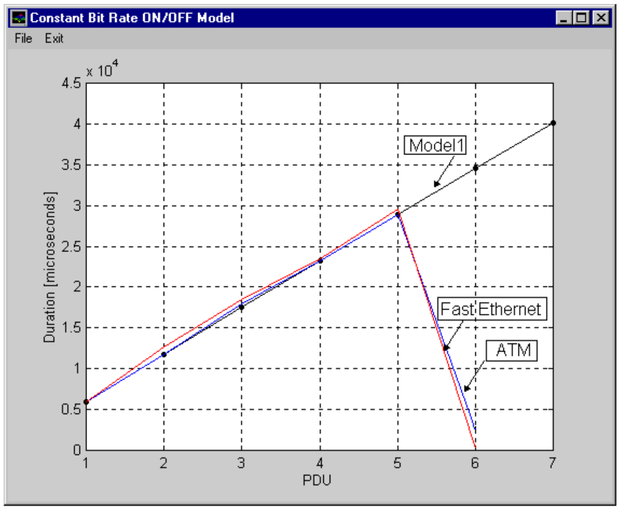


Figure 7. Model 1 for burst traffic, testfile2, Celeron 366

In opposition, the sending TCP entity will never be able to follow the *Model 2* (for both studied transport services), as in *Figure 8*. This it happens because of the minimum software loop which takes at least 100 microseconds, whilst the model is requesting 65 microseconds per ON+OFF period.

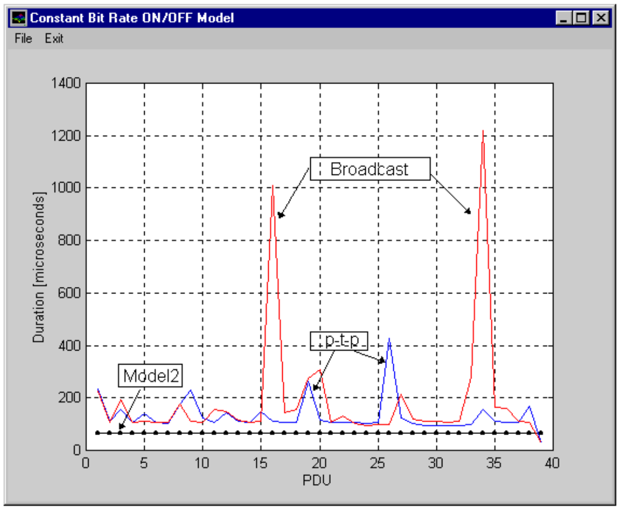


Figure 8. Model2 for burst traffic, testfile2, Classical IP over ATM: point-to-point (Celeron 366 -> server-> Pentium 233) and broadcast (Celeron 366->server-> Celeron366, Pentium 233, Pentium 120).

Interval	Model	Measured time [μs]	Throughput [Mbps]
r_f - s_i ELAPSED	-	252512…260383	7.37…7.60
	1	223194…236111	8.13…8.60
	2	309603…326569	5.88…6.20
s_f - s_i SENDING	-	35987…39626	48.47..53.37
	1	90362…91211	21.06..21.25
	2	265194…277370	6.92..7.24
r_f - r_i RECEIVING	-	195191…208155	9.22..9.84
	1	200324…220329	8.71..9.58
	2	305731…322555	5.95..6.28
r_i - s_i	-	52227…57321	N.A.
	1	14465…22870	N.A.
	2	3652…4014	N.A.
r_f - s_f	-	216524…220756	N.A.
	1	132100…144899	N.A.
	2	44384…49198	N.A.

Table 1. Classical IP over ATM on Celeron366, testfile2, point-to-point (Celeron366->server-> Celeron366). The planned sending time/ throughput were 75036 μ s/25.6 Mbps without model and for Model2, respectively 90070 μ s/21.32 Mbps for Model1.

Interval	Model	Measured time [μs]	Throughput [Mbps]
r_f - s_i ELAPSED	-	434926…436824	4.39… 4.41
	1	409365…418930	4.58… 4.69
	2	419113…431807	4.44 …4.58
s_f - s_i SENDING	-	20408…20534	93.54..94.12
	1	87871…88381	21.73..21.86
	2	191061…257949	7.44…10.05
r_f - r_i RECEIVING	-	405426…408241	4.70…4.73
	1	398744…402074	4.77…4.81
	2	416132…428589	4.48…4.61
r_i - s_i	-	28583…29499	N.A.
	1	10621…16856	N.A.
	2	2980…3218	N.A.
r_f - s_f	-	414391…416416	N.A.
	1	321494…330549	N.A.
	2	161163…240746	N.A.

Table 2. Classical IP over ATM on Celeron366, testfile2, broadcast (Celeron366 -> server -> Celeron366, Pentium233, Pentium120). The planned sending time/ throughput were 75036 μ s/25.6 Mbps without model and for Model2, respectively 90070 μ s/21.32 Mbps for Model1

Elapsed time

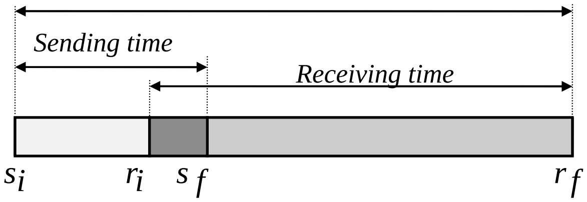


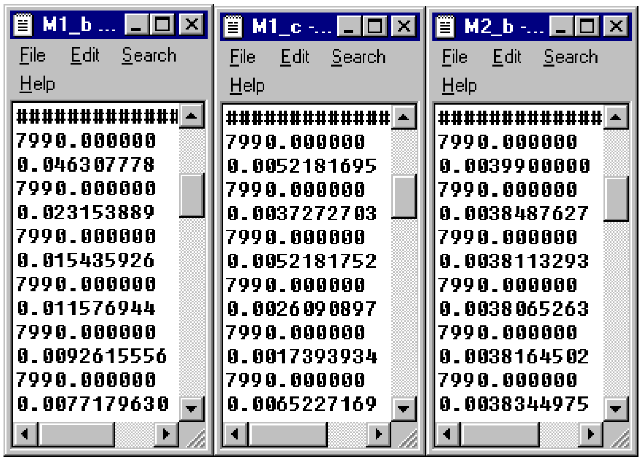
Figure 9. Four time stamps for measuring the sending, receiving and elapsed times

Sometimes it is more efficient to send the information using a model, as in *Table 1*. However the general throughput for point-to-point service could be higher (about 15 % for *Model1*) or lower (about 20 % for *Model2*) compared to the case of classical one-block sending. This observation is not valid for point-to-multipoint and broadcast services (see *Table 2*) because in this case it seems that any model generates better performances.

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.

V. EXPERIMENTAL RESULTS FOR VIDEO SOURCES

Next experiments are focused on video sources, choosing the *Models M1_B, M1_C, M2_B*, as in *Figure 10*. These notations are according to the second paragraph of this paper. For software implementation reasons, the description includes the same articles as for ON/OFF models, although there are constant PDUs *(Protocol Data Units)* of 7,990 bytes for each frame. It is not the purpose of this paper to make experiments with variable PDUs for video streams (at departure), by involving compression techniques.



```
##############
7990.000000
0.046307778
7990.000000
0.023153889
7990.000000
0.015435926
7990.000000
0.011576944
7990.000000
0.0092615556
7990.000000
0.0077179630
```

```
##############
7990.000000
0.0052181695
7990.000000
0.0037272703
7990.000000
0.0052181752
7990.000000
0.0026090897
7990.000000
0.0017393934
7990.000000
0.0065227169
```

```
##############
7990.000000
0.0039900000
7990.000000
0.0038487627
7990.000000
0.0038113293
7990.000000
0.0038065263
7990.000000
0.0038164502
7990.000000
0.0038344975
```

Figure 10. Models M1_B, M1_C, M2_B for video

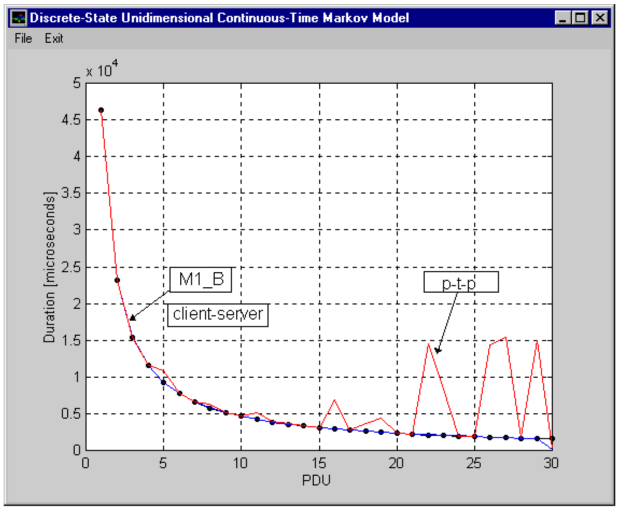


Figure 11. M1_B for video, ATM: client-server (Celeron 366->server) and point-to-point (Celeron 366->server-> Celeron 366).

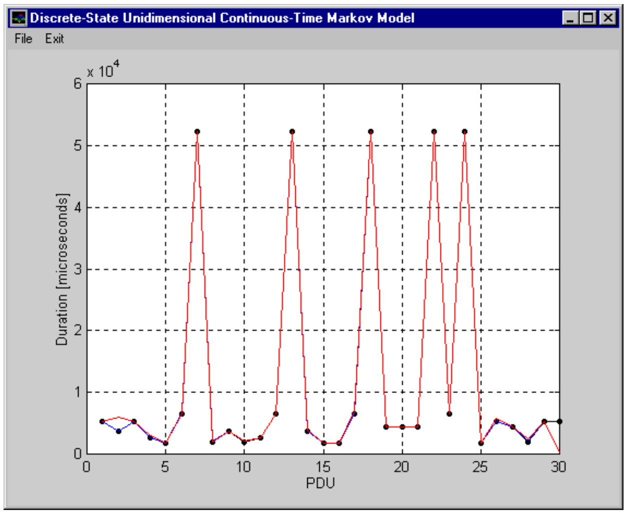


Figure 12. M1_C for video, Classical IP over ATM: client-server (Celeron 366->server) and point-to-point (Celeron 366->server-> Celeron 366).

According to *Figure 11* and *Figure 13*, the sending TCP entity could generally follow *M1_B, M2-B* for client-server applications, but there are some differences at the actual departure schedule for point-to-point applications. On the other hand, *M1_C* seems to be suitable for both client-server and point-to-point, as in *Figure 12*.

A special attention was paid for broadcast transport service from Client 1 (Celeron 366), through server, to all clients (Celeron 366, Pentium 233, Pentium 120). The planned sending time and throughput were 139825 μ s, respectively 13.71 Mbps.

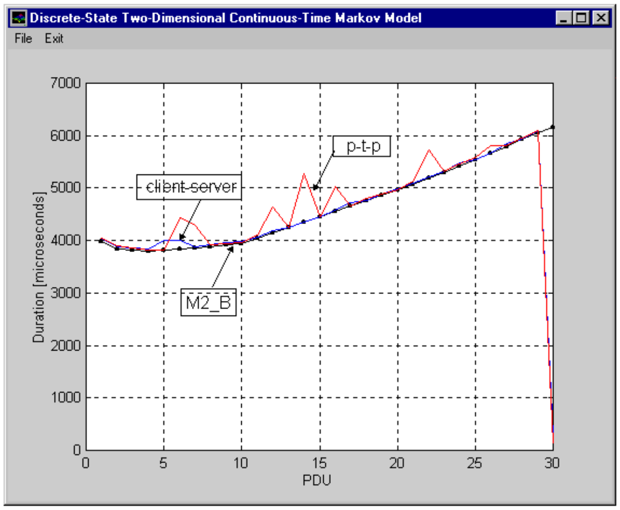


Figure 13. M2_B for video, Classical IP over ATM: client-server (Celeron 366->server) and point-to-point (Celeron 366->server-> Celeron 366).

Interval	Station	Measured time [μ s]	Throughput [Mbps]
$r_f - s_i$ ELAPSED	Cel366	401870…405505	4.72…4.77
	P233	N.A.	N.A.
	P120	N.A.	N.A.
$s_f - s_i$ SENDING	Cel366	138185…145189	13.20..13.87
	P233	-	-
	P120	-	-
$r_f - r_i$ RECEIVING	Cel366	389869…393386	4.87..4.91
	P233	391094…394638	4.85..4.90
	P120	390398…392943	4.88..4.91

Table 3. Classical IP over ATM on Celeron366, M2_B video model, broadcast (Celeron366 -> server -> Celeron366, Pentium233, Pentium120.

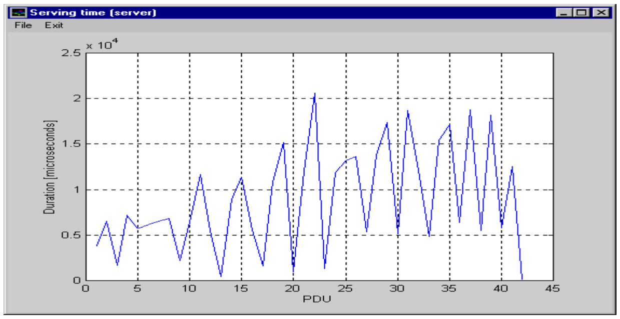


Figure 14. The serving time for Layer 4 switch performing 3-station broadcast. The incoming traffic is the result of the model M2_B for video sources.

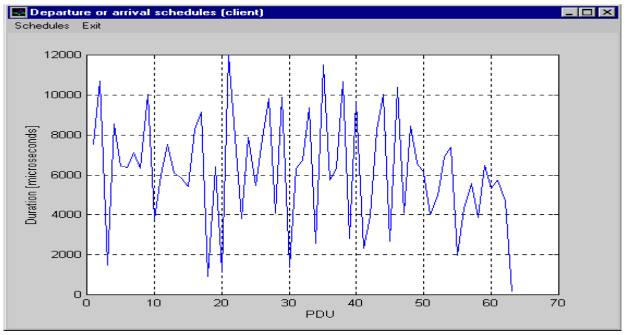


Figure 15. Classical IP over ATM on Celeron366, M2_B video model, broadcast (Celeron366->server-> Celeron366, Pentium233, Pentium120). The arrival schedule is different from the departure schedule

A very interesting comparison between the departure schedule *(Figure 13)*, the serving time *(Figure 14)* and the arrival schedule *(Figure 15)* could be done. This valuable information shows that although there were 7,990 byte-frames with a precised timing, due to non-linear behaviour of TCP/IP, the server and the receiver worked with different schedules, i.e. variable lengths and different numbers of PDUs.

The model-based transmission could also reduce the network congestion. For instance, the experiment described in *Table 3* (broadcasting video frames to three workstations, including the transmitter) shows that the serving rate of about 4.94…5.11 Mbps (Pentium II/400 MHz) is comparable to the incoming rate of any station. We come to the conclusion that the CPU’s frequency of the sender does not have a great influence at the level we are discussing in this paper. The elapsed time is less than 5% higher for Pentium 120 MHz, compared to Celeron 366 MHz, in a 3-station broadcast trial.

VI. CONCLUSIONS

1. Some of the ON/OFF and video models, usually describing the departure schedules for ATM sources, could be used also for nonblocking stream-oriented sockets in TCP/IP.
2. The Layer 4 switching has advantages due to its status information about the sockets traffic. By exploiting the specific non-linear behaviour of TCP/IP-based networks, it can reduce the traffic congestion. The resulting switching and arrival schedules are significantly different from the departure ones.
3. The highest throughput, calculated at the application/Windows Sockets interface, is less than 10 Mbps for 25.6 Mbps ATM, and less than 20 Mbps for 100 Mbps Fast Ethernet.

VII. FUTURE WORK

The next step is to include the results of the voice and variable video streams experiments. 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. It is for further work to determine the optimum model by anticipating the consequences of the self-similar behaviour of the network.

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, “Experimental Results of Traffic Models for Burst Data and Voice Sources in ATM Networks”, *ACTA Tehnica Napocensis*, ISSN 1221-6542, Vol.39, No.1, 1999, pp. 5-12.

[3] V. Dobrota, *Digital Networks in Telecommunications. Volume 2: B-ISDN with ATM, SS7*, Mediamira Science Publishers, Cluj-Napoca 1998

[4] V. Dobrota, D. Zinca, A. Vlaicu, K.Pusztai, “Evaluation of ATM Traffic Parameters in Heterogeneous Networks”, *Proceedings of the 9th IEEE Workshop LANMAN’98*, Banff, Canada, May 17-20 1998, pp.304-309

[5] D. Zinca, V. Dobrota, M. Cosma, A. Vlaicu, “Software Traffic Analyzer and Frame Generator for IEEE 802.3u”, *Proceedings of the 8th IEEE Workshop LANMAN’96*, Potsdam, Germany, August 25-28, 1996, pp. 243-248

[6] *http://193.226.6.174/people/dobrota/book9.htm*