Title: **Deliverable D4.2** **Final System Test Report**	**Document Version:** 1.9

Project Number: IST-2001-37611	**Project Acronym:** 6QM	**Project Title:** IPv6 QoS Measurement

Contractual Delivery Date: 31/05/2004	**Actual Delivery Date:** XX/XX/2004	**Deliverable Type* - Security**:** R – PU

* Type:	P - Prototype, R - Report, D - Demonstrator, O - Other
** Security Class:	PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author: Miguel Ángel Díaz	**Organization:** Consulintel	**Contributing WP:** WP4

Authors (organizations):

César Olvera (Consulintel), Jordi Palet (Consulintel).

Abstract:

This deliverable describes the final system performance and precision of the QoS parameters measured on the prototype developed by the project. The deliverable includes not only a description of the tests performed to evaluate the performance and the precision but also the results, graphic representations and conclusions about the prototype.

Keywords:

IPv6 Test-bed, Limit rate, OWD precision, Packet loss, QoS Measurement.

Revision History

The following table describes the main changes done in the document since his creation.

Revision	Date	Description	Author (Organization)
v1.0	23/04/2003	Document creation	Miguel Ángel Díaz (Consulintel)
v1.1	4/11/2004	Document update	Miguel Ángel Díaz (Consulintel) César Olvera (Consulintel)
v1.2	24/11/2004	Added content on sections 2.1 and 2.2	Miguel Ángel Díaz (Consulintel)
v1.3	26/11/2004	Added content on sections 2.3 and 2.4	Miguel Ángel Díaz (Consulintel)
v1.4	8/02/2005	Added content on sections 2.5 to 2.8	Miguel Ángel Díaz (Consulintel)
v1.5	14/02/2005	Added figures on section 2	Miguel Ángel Díaz (Consulintel)
v1.6	21/02/2005	Added section 3	Miguel Ángel Díaz (Consulintel)
v1.7	22/02/2005	Added figures on section 3	Miguel Ángel Díaz (Consulintel)
v1.8	23/02/2005	Added Summary, Conclusions and References	Miguel Ángel Díaz (Consulintel)
v1.9	29/03/2005	Minor corrections	Miguel Ángel Díaz (Consulintel)

Executive Summary

The 6QM Consortium identified several important kinds of tests to be performed on its prototype. Each one is needed to know different aspects of the accuracy on the most important parameters of the OpenIMP system, mainly the packet loss and precision of the measured one-way-delay.

The evaluation of those two parameters is enough because all the calculations made by the OpenIMP system are based on them, so no extra tests are required. The tests have been performed on laboratory not only to know the reliability of such parameter but also, and maybe the most important, how different aspects like type of CPU, RAM amount on the system, number of measurement tasks running on the probes, etc. can influence on precision of the obtained results.

As secondary objective, it was stated interesting to find out the limit traffic rate that the each architecture can support with reliable information. Regarding this point, it has been proved that 6QM probes installed on PIV 2.8 GHz are enough to capture packet on OC3 interface in a reliable way.

The description of the tests, result and conclusions are presented on this deliverable.

Table of Contents

Table of Figures

1. INTRODUCTION

The 6QM Consortium identified several important kinds of tests to be performed on its prototype. Each one is needed to know different aspects of the prototype behavior. It is not recommend making this kind of tests in uncontrolled networks where traffic is absolutely random. The best way to evaluate the system is to have a controlled environment where the network traffic can be configured and properly managed, not only with different packet rates but also with different values for fields like Traffic Class or Flow Label within the IPv6 Header.

For this reason these tests have been done into Consulintel’s laboratory with the proper laboratory equipments. In this way, it is easily possible to find the limits where the measurement system can properly work with reliable results in the field.

The laboratory equipment used in these tests was the SMB-600 system, which was integrated by Spirent SmartBits 600 with LAN-3101B modules, plus SmartFlow v3.00 software [1].

Such laboratory equipment has been useful and valuable to find out different parameters on the 6QM prototype that define its quality. Specifically, performance and time precision on the one-way-delay (OWD) are the main aspects that have been evaluated on the system.

The meaning of performance within the scope of these tests refers to the velocity of the system is able to work in order to capture and process each captured packet without producing packet loss, while the system is performing a capture measurement. In order to have reliable measurements (specially whether the system is measuring the total traffic on the network under test) the system (rather the probes which are included on the 6QM prototype) must be able to capture all the packets flowing through the network, that is, the only packet loss that the system must inform are the ones that the network has produced. The system must not introduce extra-loss due to lack of packets that the system was not able to process. This is only guarantied if each probe works below its limit rate. Performance tests try to find out such limit and how different aspect influence on the performance.

Accordingly, the performance tests have been planned to know how different hardware components, software load and traffic characteristics can influence the limit rate of the system.

On the other hand, time precision tests try to find out the quality parameters of the time-related measurements done by the 6QM prototype (basically OWD). These tests inform about the precision of the results provided by the prototype and how different aspects similar to the ones listed above influence on it.

Only performance and precision tests are enough to know the quality of the system because all the possible measurements that the prototype is able to perform are based on either or both the packet capture and the OWD computation.

In the following sections the characteristics, topologies and main results of these tests as well as conclusions are described.

1.1 Packet Size

To address the tests proposed on this deliverable there is an important question that requires to be treated carefully and it is the packet size used on each test. It is important because, a priori, the packet size might influence on the results obtained on each test. One could think that the lower size, the higher CPU load, so the minimum guarantied performance is achieved with the lowest packet size because the system is stressed at maximum. However, networks where the system is stressed at maximum because lowest packet sizes are very unlikely, so it is better to know the performance with packets having the size of the average Internet packet.

Hence making the tests with the average Internet packet seems to be reasonable to know the behavior of the 6QM prototype. However it is expected that the 6QM probes are going to be used on many types of networks to measure many type of traffics, so it would be even better to know the behavior of the probes not only for the average Internet packet but also for a wider range.

For this reason several papers have been analyzed [2-9] to know the average packet size on the Internet for the most usual applications and bellow are presented the results.

The most usual traffic is TCP which represents between 60% and 90% (75% in packets) of all the measured traffic on Internet. The most used applications are HTTP and FTP and the most used packet sizes are:

- 40 byte packets (the minimum packet size for IPv4-TCP) which carries TCP acknowledgments but no payload.
- 552 byte and 576 byte packets from TCP implementations that do not use path MTU discovery.
- 1500 byte packets (the maximum Ethernet payload size) from TCP implementations that use path MTU discovery.

The average size for TCP traffic is 471 bytes (IPv4), which should be increased by 20 bytes for IPv6 because the greater IP header size, so the average size for IPv6-TCP could be considered as 491 bytes.

On the other hand, the UDP traffic on Internet is between 10% and 40% (22% in packets) of all the measured traffic on Internet, being the average packet size of 157 bytes for IPv4-UDP which should be increased upto 177 bytes for IPv6-UDP packets because the greater IP header on IPv6. The UDP traffic is very diverse and it is composed of:

- DNS queries/replies.
- Media stream. Traffic coming from many streaming applications has been identified on the Internet, working all of them with different packet size, not only for video streaming but also for audio streaming. However the most used application is Real Audio which works with packets sizes of:
    - Audio: 304, 320 and 640 bytes
    - Video: from 547 to 1009 bytes

  And the average packet size is around 670 bytes.
- Peer-to-peer applications. There are many p2p protocols like eDonkey2000 (Implementations: eDonkey2000, eMule, MLDonkey, Shareaza), Fasttrack (Implementations: Kazaa Media Desktop, Grokster and iMesh), Gnutella (Implementations: Morpheus, Limewire, BearShare and XoloX) and Napster. The traffic generated by all of them have very different characteristics, but in general the packets use to be as large as possible. For example, the average UDP packet used on eDonkey is around 1.280 bytes and 1.040 bytes for Napster.
- Games. There are also much traffic coming form network games (Quake3, Starcraft, Quake2, Quake World, Unreal), being UDP as the usual transport protocol. However there is no much information about the average packet size.
- IP Telephony. This is one of the applications that have more interest on the community. The packet size is strongly bound to the coder used on the communication because the way that they work makes that the packet size can be larger or not. The most used codecs use to generate packet size of:
    - G.711: 120 bytes/pckt
    - G.729: 60 bytes/pckt
    - G.723.1: 70 bytes/pckt

Because the diversity of applications and packet size, it has been stated a range of size to be used on the tests performed on the laboratory. The range of sizes tries to be near to the most packet sizes used by the most of the applications. Such a range is:

- 84 bytes
- 300 bytes
- 600 bytes
- 900 bytes
- 1.200 bytes
- 1.500 bytes

Furthermore, on those tests where the average Internet packet size was required, the size of 300 bytes has been chosen because two reasons:

- It is near of the real average Internet packet size (491 bytes for IPv6)
- It is bellow the real average Internet packet size, so the system is stressed more than larger sizes.

2. PERFORMANCE TESTS

The first 6QM Measurement System is basically developed as software that can run in either Linux or BSD PC platforms. Consequently the system performance strongly depends on the hardware used to run it. However, some aspects of the software architecture chosen for implementing the prototype can strongly influence in the general system performance. Because of these points, it is necessary to know the rate limits where the prototype can successfully work.

One of the basic tasks that the Measurement System has to do is to sniff all the network traffic that flows through the network and to capture all the packets that properly match the filter defined in the configuration phase. As it was stated before, in order to successfully measure the QoS available to the user, it is crucial to avoid any packet loss inside the measurement system itself. This only can be achieved if the performance of the Measurement System is in accordance to the network environment where it is deployed.

In order to find out how different aspects influence the performance of probes[1] the testbed was setup as follows. The SMB-600 system was configured to send controlled IPv6 packets from the PortA1 to PortA2 (10/100 Ethernet each) as shown in figure 2-1. Both Ethernet ports were attached to a Full Duplex Fast-Ethernet Switch which is configured for doing port mirroring on the Ethernet port where the probe was attached. This is needed in order to let all the packets transmitted by the SMB-600 PortA1 be delivered not only to the SMB-600 PortA2, but also to the 6QM probe. The usage of a non-blocking switch instead of a hub is preferred in order to avoid reduced performance results due to packet collisions in the hub.

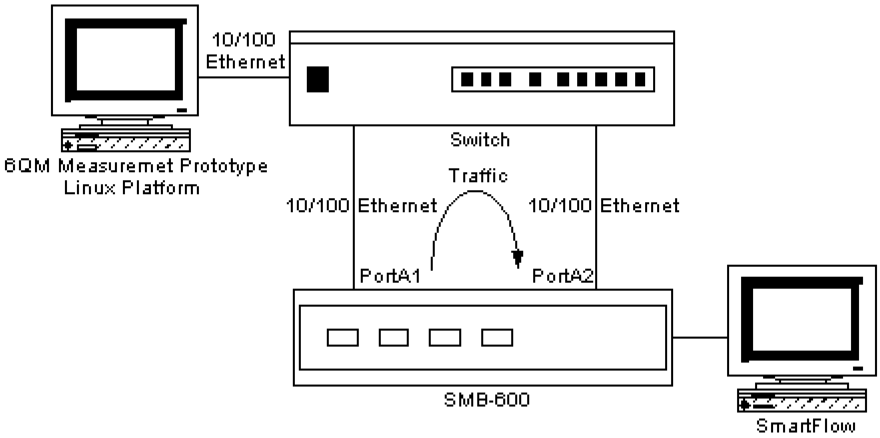


Figure 2-1: Performance Test Topology

All the delivered packets were equal (they had the same size, headers and payload) and were sent at the same packet rate, so the time interval between consecutive packets was always constant. Different packet sizes and rates were used on different tests as will be detailed further.

These packets were captured by the OpenIMP probe, which was setup with different configurations (different hardware, different packet size, etc) in order to know the influence of those factors on the probe performance. Finally, the report of the captured packets on the probe was compared to the generated packets by the SMB-600. Both of them inform about the total

---

[1] *The rest of components of the 6QM prototype only are used to configure the measurements and to collect the data from the probes*

number of captured packets on each device, so if they are different, we can conclude that the probe has reached its limit rate.

All the 6QM components (probes, controller and collector) were installed on the same Operating System which was Linux 2.4.20-8. Although the figure 2-1 does not show the 6QM Controller and the 6QM Collector components to avoid complicating too much the picture, they were necessary to setup all the measurements performed and to collect all the captured data on the probes.

Several tests have been done to evaluate the influence on the performance of the following aspects:

- Hardware factors
  - Type of CPU
  - Amount of RAM
  - Type of NIC
- Software factors
  - Configuration of filters on the measurement
  - Several tasks running on the probes
- Traffic factors
  - Packet Size
  - Time between packets
- Other
  - Test duration

During the tests the CPU load required by the impd daemon (which is the one in charge of perform the measurements on the probe) has been measured with the linux command `top`, to find out the correlation of the CPU load and performance on all the tests done.

Bellow the different tests, results and conclusions about each evaluated factor are detailed.

2.1 Influence of the type of CPU

Because all the work related to the traffic capture is made by the probes, one could believe at a glance that the type of CPU strongly influences on the system performance because the power probe system, the faster packet processing. In order to check the veracity of this assumption and the performance of different CPUs, the following test has been done which implements the network-test topology shown in figure 2-1.

In this tests all the delivered packets were equal, with 84 Bytes length (Layer 2 size, including CRC) and with the same IPv6 headers and payload. As stated before, it is important to note that small packets have been chosen considering that this is the traffic type which forces the hardware to work more intensively, so it is expected to get the worse performance in terms of bits per seconds. Consequently, we obtained the minimum performance guaranteeing that no loss is produced while capturing longer packets.

The SmartBits was configured to send traffic with different network load percentages ranged from 5% to 70 % which means for example that traffic delivered at 15% only occupied the 15% of the network capacity (100 Mbps), so although the packets were always delivered at a bit rate 100 Mbps, the mean bandwidth occupied on the network was 15 Mbps.

Figure 2-2 illustrates the concept of bit rate, packet rate and network load managed by the SmartBits.

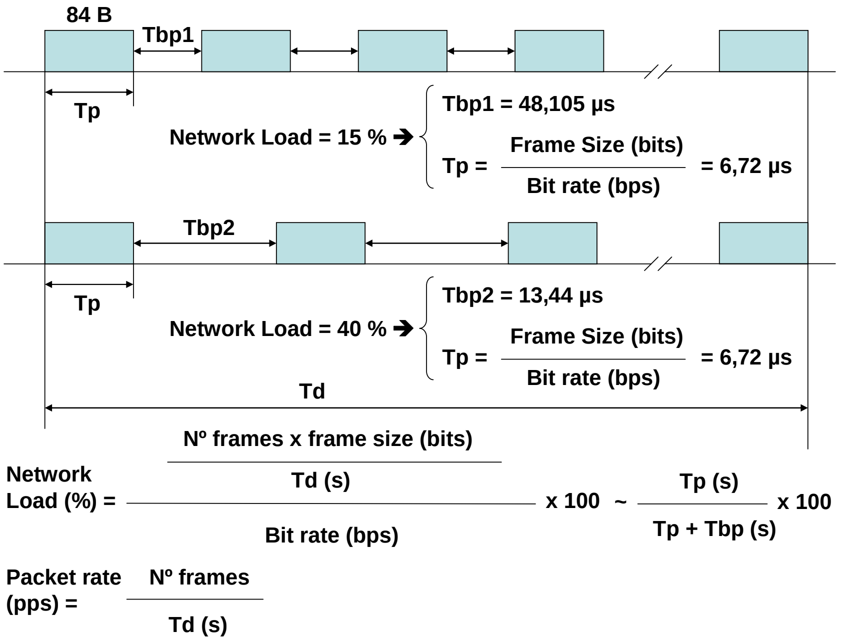


Figure 2-2: Network load concept

So, the number of packets transmitted varied from the percentage of network link used, and it ranged from 360.576 (6.010 packets per second (pps) for 5 % of network load) to 5.048.076 (84.135 packets per second (pps) for 70 % of network load).

All the packets were uniformly transmitted, which means that the time interval between two consecutive packets was always constant and it depended on the packet rate. These packets were captured by the 6QM probe, which was configured with different CPUs. The tests were repeated at different packet rates. Finally, the report of the captured packets on the probe was compared to the generated packets by the SMB-600 to find out when the probe has reached its limit rate.

The following platforms have been tested according the topology shown in figure 2-1 and the method explained above:

1) AMD-K6, 450 MHz, Cache 64 KB, RAM 64 MB, NIC 100 Mbps.
2) PIII, 750 MHz, Cache 256 KB, RAM 128 MB, NIC 100 Mbps.
3) PIV, 2.8 GHz, Cache 512 KB, RAM 1 GB, NIC 100 Mbps.

And the following results were obtained from the tests done as explained above.

2.1.1 AMD-K6 CPU

The configuration of the probe and results were as follows:

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))
- Test duration: 60 seconds

Network Load (%)	Traffic Rate (PPS)	Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
5	6.010	360.576	360.576	100,00%	0,00%	15,97%
10	12.019	721.153	721.153	100,00%	0,00%	29,99%
15	18.029	1.081.730	1.081.695	100,00%	0,00%	46,05%
20	24.038	1.442.307	1.364.836	94,63%	5,37%	99,90%
25	30.048	1.802.884	1.080.555	59,93%	40,07%	99,88%
30	36.058	2.163.461	826.439	38,20%	61,80%	99,85%
35	42.067	2.524.038	581.215	23,03%	76,97%	99,82%

Figure 2-3: AMD-K6 450 MHz CPU Performance Numeric Results

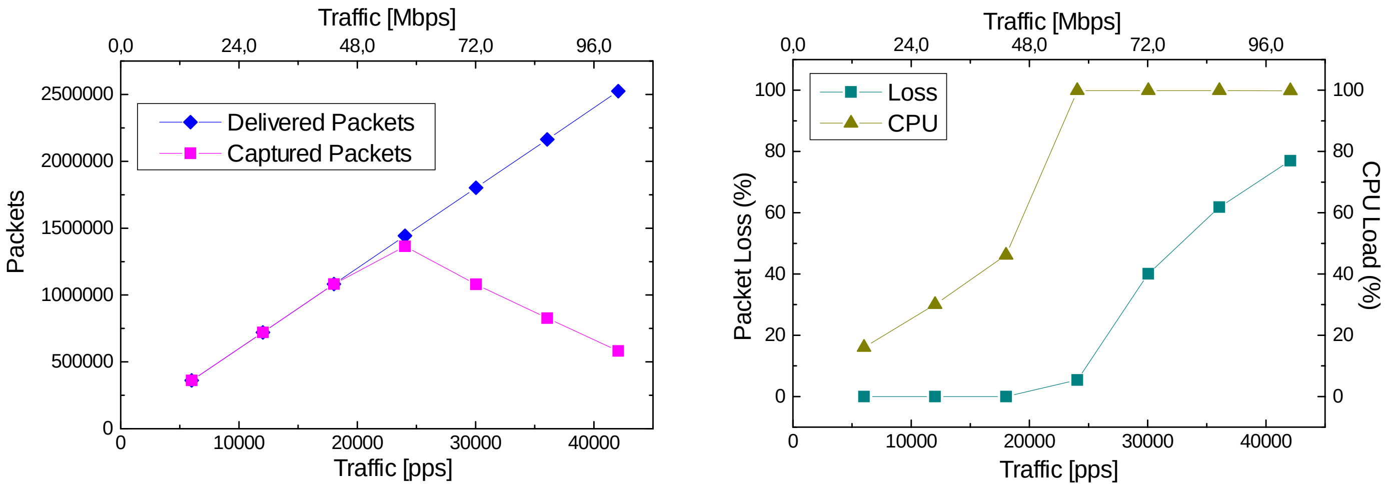


Figure 2-4: AMD-K6 450 MHz CPU Performance Results

Figure 2-4 left represents the evolution of captured packets (magenta line) when the packet rate increases. The bottom X axis represents the traffic rate in packets per second. It is important to note that the right value that indicates the performance is the traffic rate in packets per second because the probe must process each packet independently and, as it will be further demonstrate, the size of the packet not influence on the performance.

Taken this consideration into account, we can easily calculate the maximum traffic rate in bits per second for other frames with size similar to the average size on Internet. The top X axis represents the traffic rate in bits per second for frames of 300 Bytes (layer II, including CRC), which is near to the average packet size on the Internet, as stated in section 1.

According to figure 2-2 the calculation is based on the assumption that the frames with the new size (300 Bytes) have a Tp' lower than the sum of Tp plus Tbp gotten for frames of 84 Bytes, and the number of total packets delivered during the test remains constant. Furthermore, it is assumed that the bit rate for transmitting the packets of 300 bytes can be greater that the one used with packets of 84 bytes.

With packets of 84 bytes, Tp comes from:

$$T_p(\mu s) = \frac{Frame\ Size(b)}{Bit\ Rate(Mbps)} = \frac{84 \bullet 8}{100} = 6{,}72\ \mu s$$

and the value of Tbp is provided by the SmartBits according to the following expression:

$$T_{bp}\ (\mu s) = \frac{\left[\left(\frac{Bit\ Rate\ (Mbps)}{Network\ Load\ (\%)} - 1\right)(160 + Frame\ Size\ (b))\right] + 96}{100}$$

so, for packets of 84 bytes transmitted at the traffic rate which packet loss starts to appear (22.000 pps at 18,3% of network load) we have that Tp = 6,72 µs and Tbp = 38,10 µs.

Assuming that new frames have 300 bytes and the same traffic rate (aprox. 22.000 pps) the new traffic rate in bits per second is:

$$Traffic\ Rate\ (bps) = Traffic\ Rate\ (pps) \bullet Frame\ Size\ (b)$$

$$Traffic\ Rate = 22.000\ (pps) \bullet 2.400\ (b) = 52{,}8\ Mbps$$

If we consider now a frame size of 1.500 bytes (the maximum in Ethernet networks), we can see that the maximum traffic rate for this CPU would be 264 Mbps, which only would be reached in GigaEthernet networks.

$$Tr_{\max}^{K6-456\ MHz} = 264\ Mbps$$

The maximum traffic rate means that it is the maximum rate where there is guaranty that the probe is able to capture the packets flowing on the network without producing packet loss greater than 0,1%.

However, given that the most of packets that probes will capture are not the maximum ones (1.500 B), it has been considered more useful to represent the maximum bit rate for the frames of size (300 bytes) near to the average Internet packet, which is the scale represented at the top X axis of the graphs in all the pictures from now on.

On the other hand, figure 2-4 right shows that there is a strong correlation between CPU load and packet loss once a saturation point is overcome. Such a point represents the value where CPU is overloaded and packet loss might start to appear. For AMD K6 456 MHz architecture such point is around 40 % (22.000 pps) and for traffic rate greater, the CPU load and packet loss rate increase exponentially.

2.1.2 Pentium III CPU

The configuration of the probe with this CPU and results were as follows:

- CPU: Pentium III (Coppermine), 750 MHz
- CACHE: 256 KB
- Memory: 128 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.18-14 (gcc version 3.2 20020903 (Red Hat Linux 8.0 3.2-7))

Network Load (%)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
5	6.010	360.576	360.576	100,00%	0,00%	6,07%
10	12.019	721.153	721.153	100,00%	0,00%	12,35%
15	18.029	1.081.730	1.081.730	100,00%	0,00%	18,30%
20	24.038	1.442.307	1.442.307	100,00%	0,00%	24,80%
25	30.048	1.802.884	1.801.918	99,95%	0,05%	33,32%
30	36.058	2.163.461	1.806.731	83,51%	16,49%	99,57%
35	42.067	2.524.038	1.149.258	45,53%	54,47%	99,41%
40	48.077	2.884.615	485.102	16,82%	83,18%	98,65%
45	54.087	3.245.192	292.065	9,00%	91,00%	97,19%
50	60.096	3.605.769	208.374	5,78%	94,22%	95,73%

Figure 2-5: PIII 700 MHz CPU Performance Numeric Results

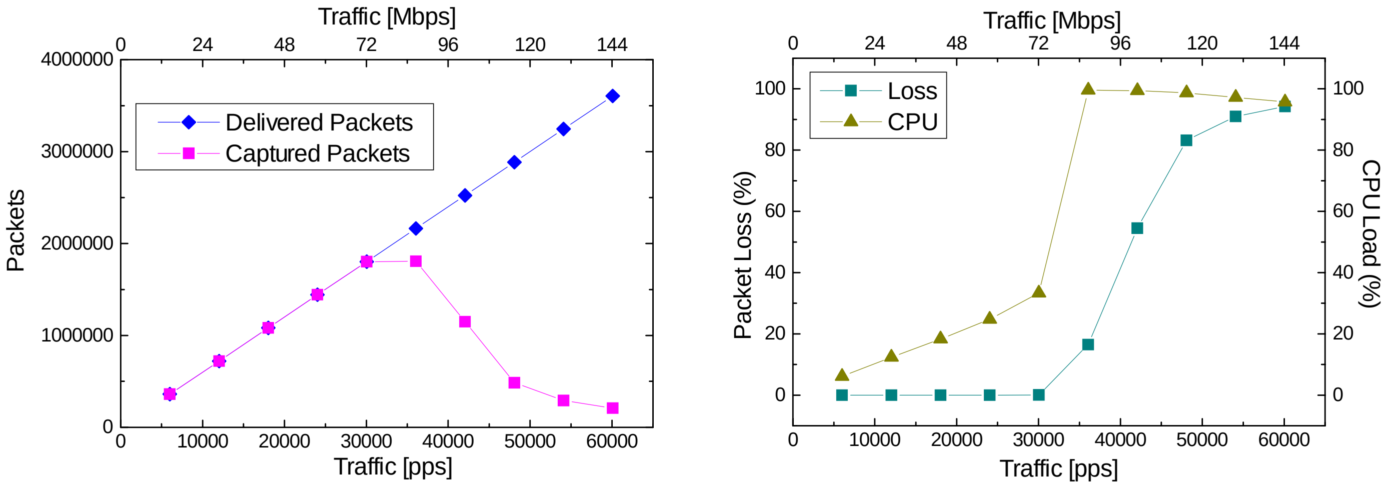


Figure 2-6: PIII 700 MHz CPU Performance Results

In this case, given the fact that the CPU is faster than the previous one we can see that the traffic rate which packet loss starts to appear is higher than the case of K6-456 MHz CPU. The maximum traffic rate where packet loss is below 0,1% in this architecture is around 32.000 packets per second (26,7% of network load).

With the same considerations pointed before, the maximum traffic rate in bits per seconds for PIII 700 MHz calculated for frames with 300 Bytes (near to the average packet size on Internet) is provided on the top X axis of figure 2-6 and the absolute maximum traffic rate (for packets of 1.500 Bytes) is:

$$Tr_{\max}^{PIII-700\ MHz} = 384\ Mbps$$

Regarding the CPU load evolution we can observe similar behavior than the case of K6-456 MHz CPU, with a saturation point near of 40% of CPU load which is reached with a traffic rate greater than the previous case, at 32.000 pps approximately.

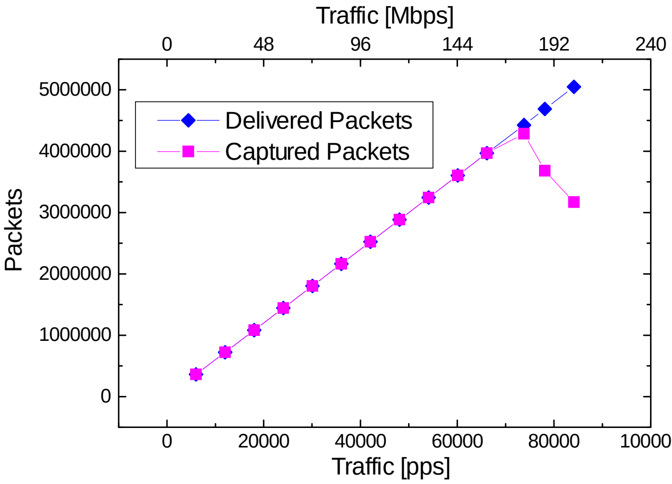
2.1.3 Pentium IV CPU

This was the last tested architecture which was the more powerful and gives us an idea of the maximum expected performance that 6QM probes have. The configuration of the probe with this CPU and results were as follows:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 1 GB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Network Load (%)	PPS	Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
5	6.010	360.576	360.576	100,00%	0,00%	4,79%
10	12.019	721.153	721.153	100,00%	0,00%	9,64%
15	18.029	1.081.730	1.081.730	100,00%	0,00%	13,68%
20	24.038	1.442.307	1.442.307	100,00%	0,00%	18,81%
25	30.048	1.802.884	1.802.884	100,00%	0,00%	22,93%
30	36.058	2.163.461	2.163.461	100,00%	0,00%	28,48%
35	42.067	2.524.038	2.524.038	100,00%	0,00%	32,78%
40	48.077	2.884.615	2.884.615	100,00%	0,00%	39,65%
45	54.087	3.245.192	3.245.192	100,00%	0,00%	53,40%
50	60.096	3.605.769	3.605.769	100,00%	0,00%	74,03%
55	66.106	3.966.346	3.966.346	100,00%	0,00%	88,29%
60	73.782	4.426.923	4.286.094	96,82%	3,18%	99,71%
65	78.125	4.687.500	3.682.418	78,56%	21,44%	99,90%
70	84.135	5.048.076	3.170.705	62,81%	37,19%	99,90%

Figure 2-7: PIV 2.8 GHz CPU Performance Numeric Results



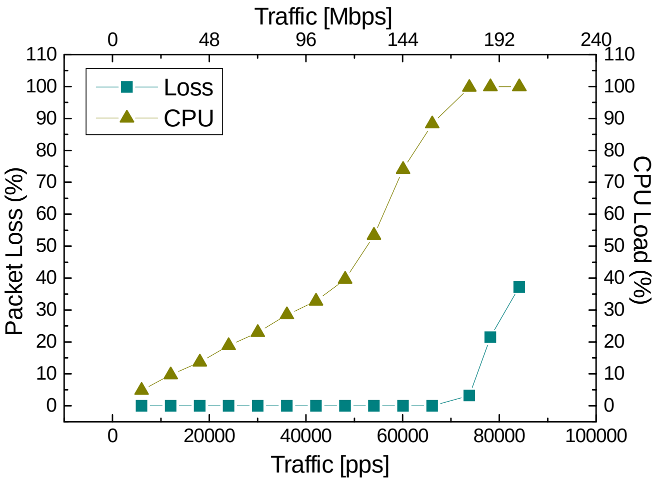


Figure 2-8: PIV 2.8 GHz CPU Performance Results

As expected, the performance with this architecture is the highest and the maximum traffic rate is around 70.000 pps, which is equivalent to 168 Mbps for frames of 300 Bytes (top X axis on figure 2-8).

The absolute maximum traffic rate (frames of 1.500 bytes) in bits per second for this CPU is:

$$Tr_{\max}^{PIV-2,8\,GHz} = 840\ Mbps$$

Regarding the CPU load behavior (figure 2-8 right) there is a little difference compared to the previous CPUs. In this case, although the shape of the CPU load graph is similar to the previous ones, we can observe that the saturation point is higher than the previous CPUs, around 80% CPU load, which indicates that this CPU can do several tasks in parallel with the task of capturing packets while doing measurements.

2.1.4 CPU load evolution versus Traffic Rate

It has been considered interesting to evaluate more in depth somehow the CPU load evolutions meanwhile the traffic raises, in order to know how the CPU load influences on the limit rate within the probes can work without packet loss. For this reason this section focuses on the CPU evolution analysis of each architecture.

A separately representation has already done in the previous figures with each alone platform. However for better analysis, the figure 2-9 represents together the results for all the analyzed architectures:

1) AMD-K6, 450 MHz, Cache 64 KB, RAM 64 MB, NIC 100 Mbps.
2) PIII, 750 MHz, Cache 256 KB, RAM 128 MB, NIC 100 Mbps.
3) PIV, 2,8 GHz, Cache 512 KB, RAM 1 GB, NIC 100 Mbps.

The CPU behavior is depicted along with packet loss ratio for each platform. The bottom X axis represents the traffic on the network (packet per second), while the top X axis represents the traffic rate in bits per second (considering mean Internet packets of 300 Bytes).

It is interesting to note that all the architectures show a similar behavior: the CPU load shows three different parts: the first one represents a linear growth; the second one represents also a linear grown but with a grown constant higher; and the last one represents an exponential grown.

Also the packet loss ratio evolves in a similar manner for all the architectures. Firstly the packet loss increases linearly when the CPU load is within the linear grown zone. When the CPU load is near of the exponential grown zone then the packet loss ratio also grows exponentially and asymptotically towards 100 %.

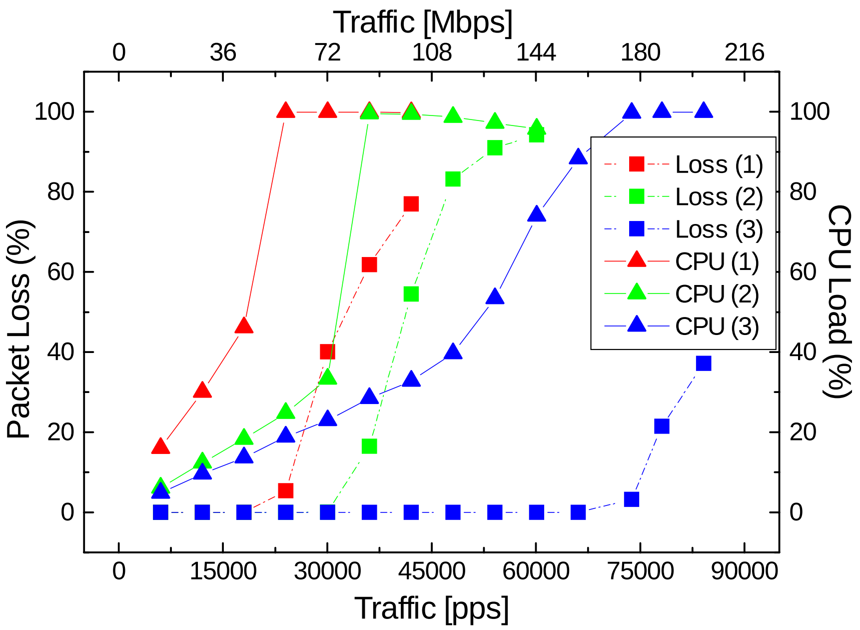


Figure 2-9: CPU Load and Packet Loss versus Delivered Traffic

Obviously the numeric values depend on each type of architecture being the best result the ones belonging to the PIV 2,8 GHz platform, as expected.

2.1.5 Conclusions

To extract conclusions from the point of view of the influence on the performance of the type of CPU is more interesting to have a new graph that represents together the main results shown in the previous figures.

Figure 2-10 represents the captured packets by the three architectures analyzed previously:

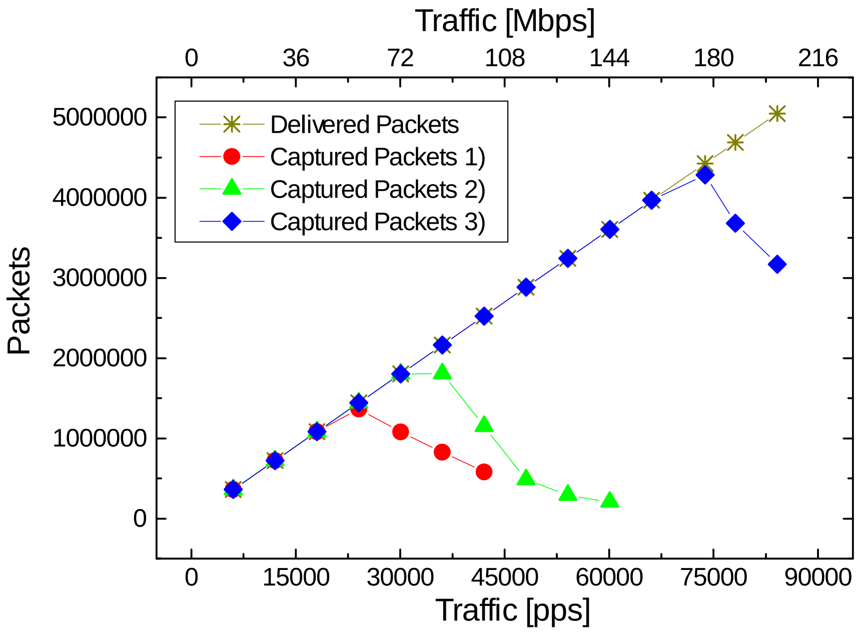


Figure 2-10: Performance Comparison Results

The number of each platform expressed on the legend of figure 2-10 corresponds at:

1) AMD-K6, 450 MHz, Cache 64 KB, RAM 64 MB, NIC 100 Mbps.
2) PIII, 750 MHz, Cache 256 KB, RAM 128 MB, NIC 100 Mbps.
3) PIV, 2.8 GHz, Cache 512 KB, RAM 1 GB, NIC 100 Mbps.

There is a remarkable saturation point, different to each CPU type, in which packets start being dropped, and the line representing captured packets splits from the line representing delivered packets. Obviously while each CPU is working at rates bellow its saturation point no packet loss will be produced and the results show by the probe can be considered as reliable.

On the other hand, the saturation point is related to the CPU performance, so the higher CPU performance, the higher saturation point. For the tests analyzed previously, the highest saturation point corresponds to the PIV CPU at 2.8 GHz, being the limit rate without packet loss around 70.00 PPS (packets per second), which represents about 168 Mbps (for frames of 300 bytes) with the conditions that the test has been performed.

Finally, it is also shown that while each CPU is working below the saturation point depicted in figure 2-10, the CPU load increases almost linearly and no packet loss is experimented as shown in figure 2-9. Once the CPU reaches the saturation point, the CPU load starts to exponentially increase to 100% and packet loss is highly produced.

The previous results and analysis allow us to model both the CPU and Packet Loss behavior as shown in figure 2-11.

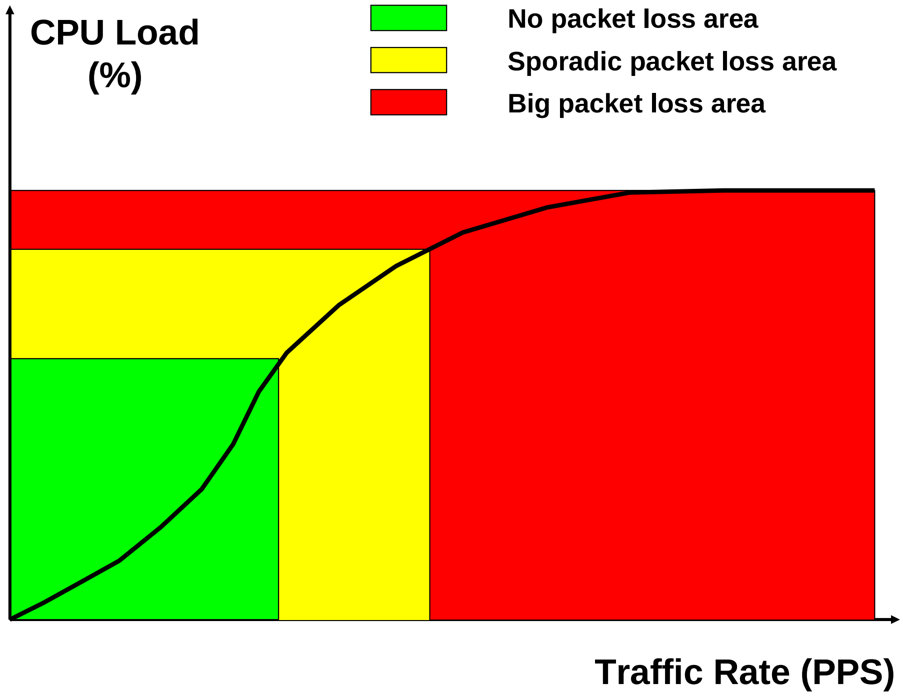


Figure 2-11: CPU and Packet Loss Behavior Model

2.2 Influence of RAM memory

Usually the amount of RAM on the PC is a key factor that improves the general performance. For this reason this factor has been considered necessary being checked in order to know if the limit traffic rate without packet loss depends of the amount of RAM and somehow, just in case.

According to this approach, the following platforms have been tested according the topology shown in figure 2-1.

1) AMD-K6, 450 MHz, Cache 64 KB, RAM 64 MB, NIC 100 Mbps.
2) AMD-K6, 450 MHz, Cache 64 KB, RAM 256 MB, NIC 100 Mbps.
3) PIV, 2.8 GHz, Cache 512 KB, RAM 256 MB, NIC 100 Mbps.
4) PIV, 2.8 GHz, Cache 512 KB, RAM 512 MB, NIC 100 Mbps.
5) PIV, 2.8 GHz, Cache 512 KB, RAM 1GB, NIC 100 Mbps.

The procedure followed to get the results of these tests is the one explained on section 2. Both the numerical and graphical results are shown bellow.

2.2.1 AMD-K6, 450 MHz and 64 MB

Results corresponding to this configuration were already shown on figures 2-3 and 2-4.

2.2.2 AMD-K6, 450 MHz and 256 MB

The configuration of the probe and results were as follows:

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 256 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Both numeric and graphic results are presented bellow.

Network Load (%)	Traffic Rate (PPS)	Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
5	6.010	360.576	360.576	100,00%	0,00%	16,69%
10	12.019	721.153	721.153	100,00%	0,00%	33,06%
15	18.029	1.081.730	1.081.730	100,00%	0,00%	49,32%
20	24.038	1.442.307	1.261.453	87,46%	12,54%	99,90%
25	30.048	1.802.884	964.724	53,51%	46,49%	99,88%
45	54.087	3.245.192	135.770	4,18%	95,82%	97,73%

Figure 2-12: AMD-K6 450 MHz & 256 MB Performance Numeric Results

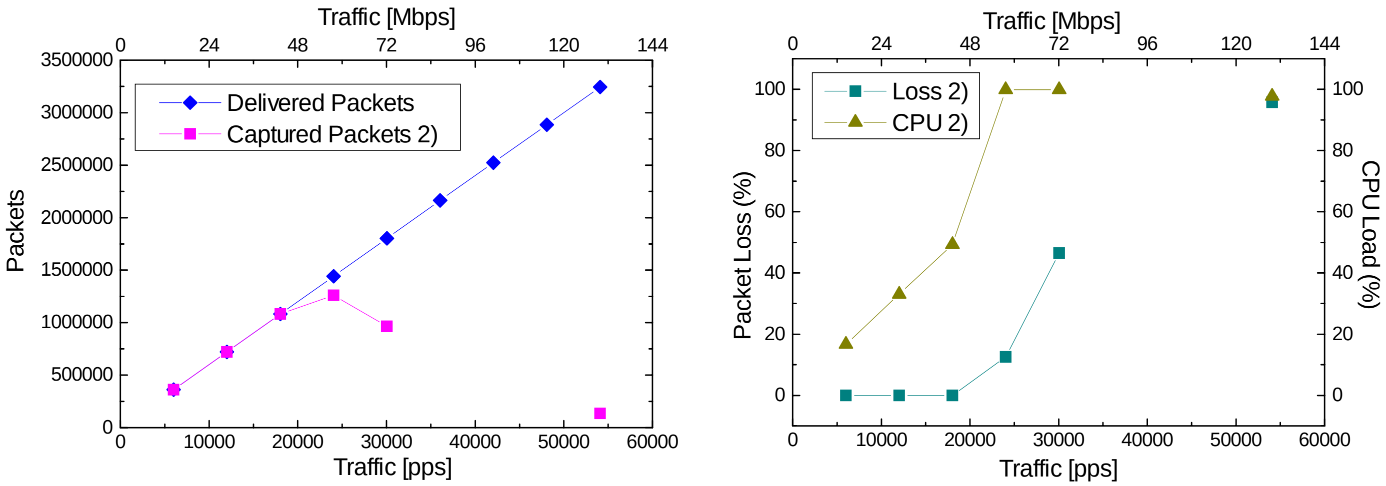


Figure 2-13: AMD-K6 450 MHz & 256 MB Performance Results

After a slight observation of pictures of figure 2-13 it can be concluded that the shape of the graphs representing the captured packets, the CPU load and the packet loss follows the main structure of the ones obtained with the previous tests done to characterize different CPUs. That is a characteristic common to the rest of measurements done with different RAM amount, as will be shown bellow.

On the other hand, the numeric value for the maximum traffic rate is similar to the one obtained for the same CPU with 64 MB of RAM.

2.2.3 PIV, 2.8 GHz and 256 MB

The configuration of the probe for this test and results were as follows:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 256 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Network Load (%)	Traffic Rate (PPS)	Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
45	54.087	3.245.192	3.245.192	100,00%	0,00%	43,79%
50	60.096	3.605.769	3.605.769	100,00%	0,00%	47,54%
55	66.106	3.966.346	3.966.346	100,00%	0,00%	73,64%
60	72.115	4.326.923	4.326.923	100,00%	0,00%	97,83%
65	78.125	4.687.500	4.168.792	88,93%	11,07%	99,90%
70	84.135	5.048.076	3.674.284	72,79%	27,21%	99,90%

Figure 2-14: PIV 2.8 GHz & 256 MB Performance Numeric Results

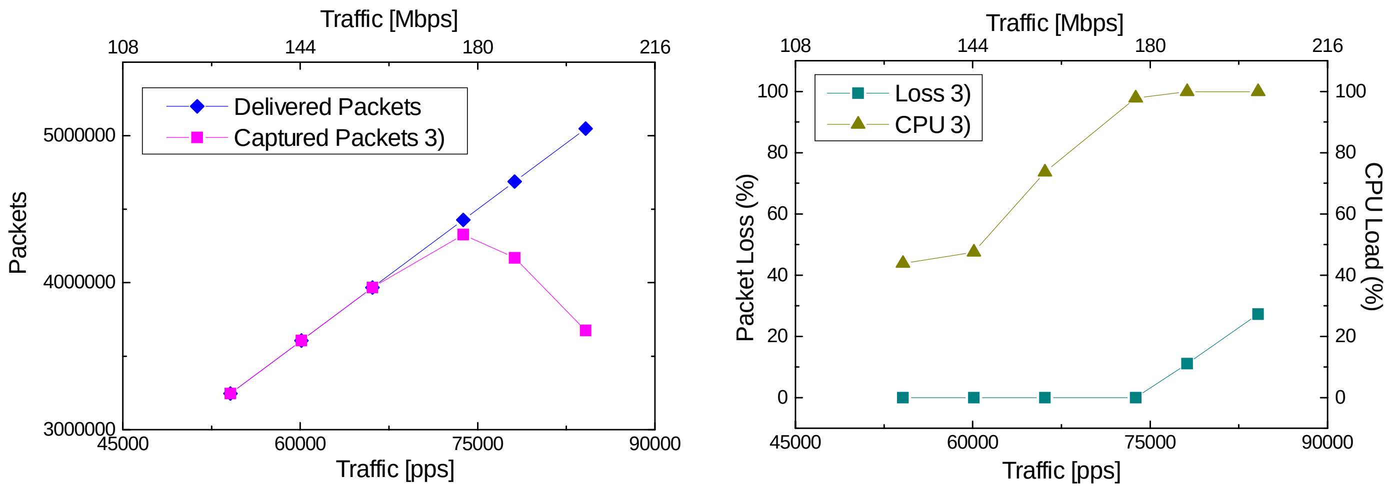


Figure 2-15: PIV 2.8 GHz & 256 MB Performance Numeric Results

The previous figures shown the behaviour of PIV 2,8 CPU with 256 MB of RAM, which is similar to the one obtained for the same CPU with 1 GB shown in figure 2-8. The maximum traffic rate has also a similar value than the one detailed on section 2.1.3.

2.2.4 PIV, 2.8 GHz and 512 MB

The configuration of the probe for this test and results were as follows:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Network Load (%)	Traffic Rate (PPS)	Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
45	54.087	3.245.192	3.245.192	100,00%	0,00%	42,67%
50	60.096	3.605.769	3.605.769	100,00%	0,00%	49,21%
55	66.106	3.966.346	3.966.346	100,00%	0,00%	73,83%
60	72.115	4.326.923	4.326.923	100,00%	0,00%	97,33%
65	78.125	4.687.500	4.160.381	88,75%	11,25%	99,90%
70	84.135	5.048.076	3.670.771	72,72%	27,28%	99,85%

Figure 2-16: PIV 2.8 GHz & 512 MB Performance Numeric Results

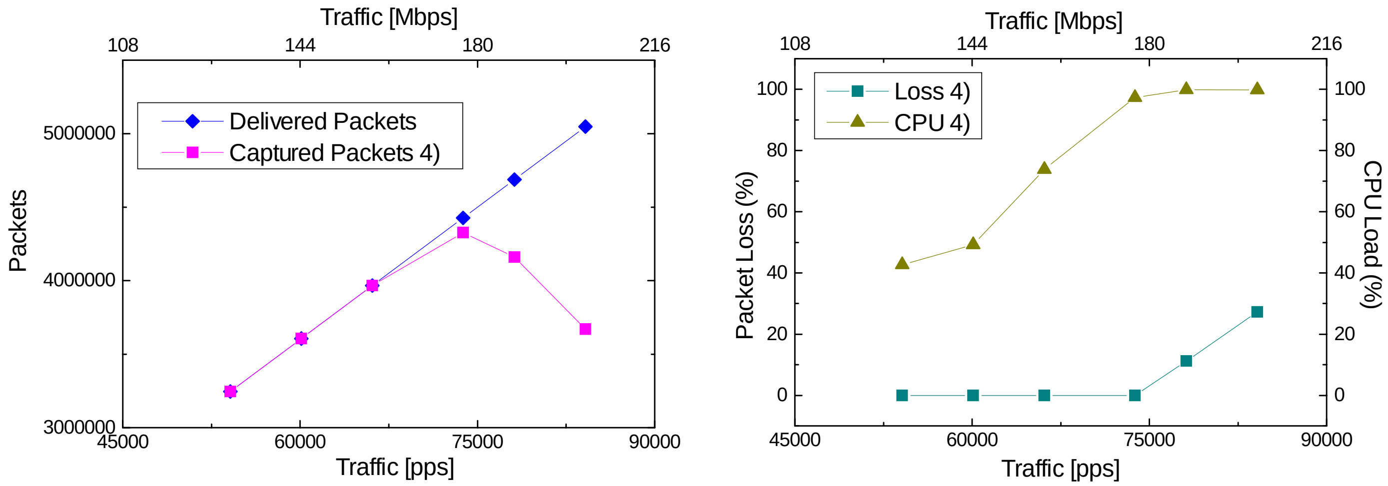


Figure 2-17: **PIV 2.8 GHz & 512 MB Performance Numeric Results**

Values for this test are similar to the ones gotten for the same CPU with 1 GB of RAM.

2.2.5 PIV, 2.8 GHz and 1 GB

Results corresponding to this configuration were already shown on figures 2-7 and 2-8.

2.2.6 Conclusions

In this case it was also stated interesting to check the influence of the amount of RAM on the CPU load, as in the CPU type case. Hence, on the previous results the CPU load evolution has been presented for each alone configuration. However, for comparison purposes all of them have been grouped in one figure to better understand the influence of RAM amount. This behavior can be seen on figure 2-18.

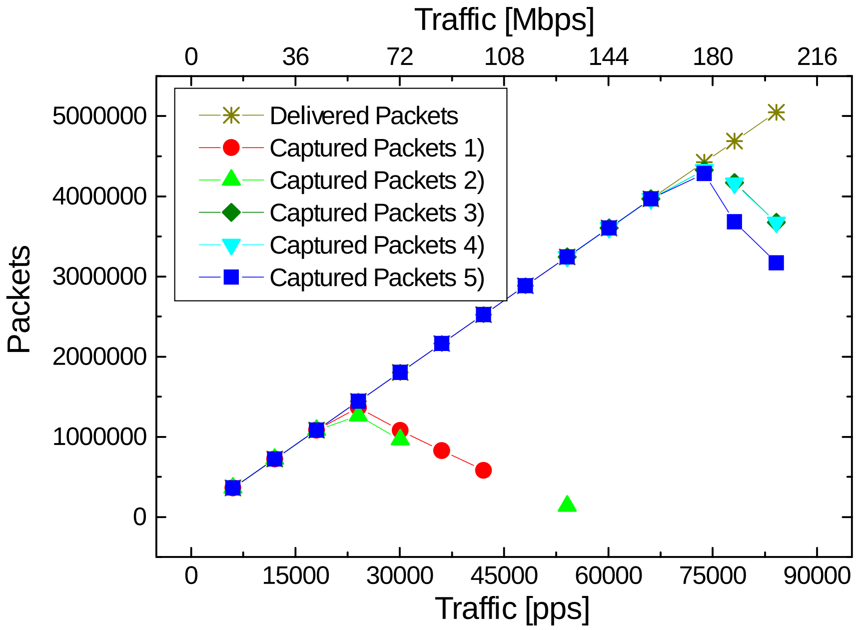


Figure 2-18: **Captured Traffic versus Traffic Rate with different RAM amount**

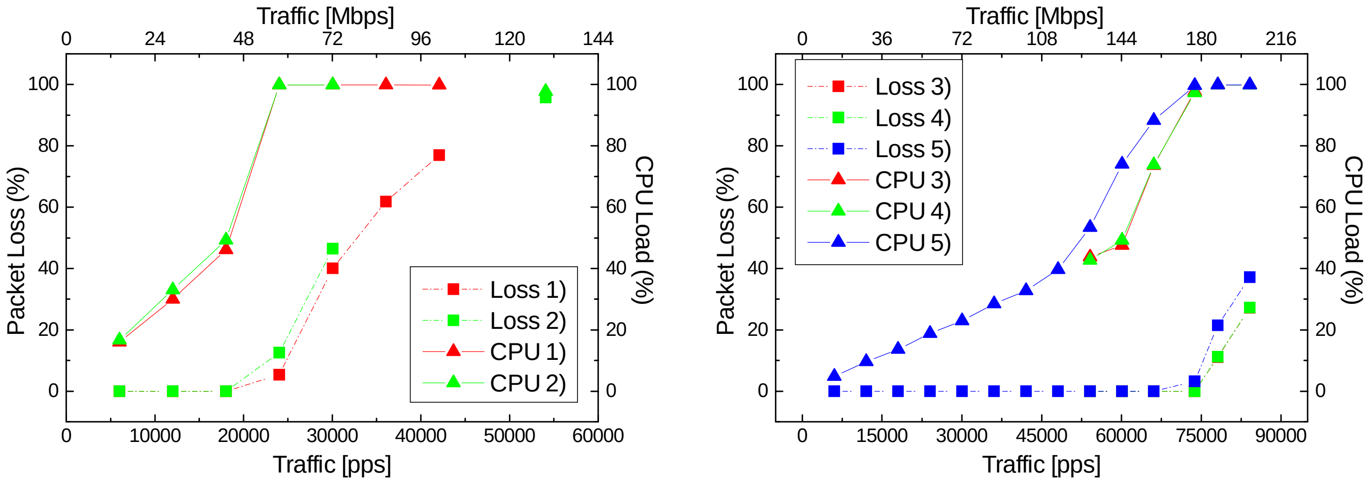


Figure 2-19: CPU Load and Packet Loss versus Traffic Rate with different RAM amount

On the other hand, it can be observed that there is not any important different behavior for each CPU despite the RAM amount used. Also is interesting to analyze figures 2-18 and 2-19 where it can be observed also almost the same behavior for all the architectures. So, we can conclude that the amount of RAM on the system does not influence on the limit traffic rates obtained in section 2.1 where the probes can work without packet loss.

2.3 Influence of NIC rate

The Network Interface Card, usually called simply NIC, is other hardware element that can strongly influence on the limit rate on which the probe can work without packet loss. Not only because of the rate it is specifically designed to work, that is 10/100/1000 Mbps, but also because of the way it works when a packet comes to the NIC, that is either informing immediately to the CPU via interruption or grouping various packet before activating such interruption.

With this test we have tried to characterize the behavior of two common NICs as well as the influence on the limit rate on the probes.

According to this approach, the following platforms have been tested according the topology shown in figure 2-1.

1) AMD-K6, 450 MHz, Cache 64 KB, RAM 64 MB, NIC 100 Mbps
2) AMD-K6, 450 MHz, Cache 64 KB, RAM 64 MB, NIC 10 Mbps

The procedure followed to get the results of these tests is the one explained on section 2. Both the numerical and graphical results are shown bellow.

2.3.1 NIC 100 Mbps

Results corresponding to this configuration were already shown on figures 2-3 and 2-4. The complete information about the used NIC is below:

Ovislink N8139D based on RTL 8139C chip (Linux driver used: 8139too.c, version 0.9.26).

2.3.2 NIC 10 Mbps

The configuration of the probe and results were as follows:

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Network Load (%)	Traffic Rate (PPS)	Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
5	6.011	360.660	360.660	100,00%	0,00%	18,13%
10	12.019	721.153	721.082	99,99%	0,01%	40,22%
11	13.221	793.269	721.362	90,94%	9,06%	39,35%
12	14.423	865.384	721.603	83,39%	16,61%	38,43%
15	18.029	1.081.730	721.505	66,70%	33,30%	40,12%

Figure 2-20: AMD-K6 450 MHz & NIC 10 Mbps Performance Numeric Results

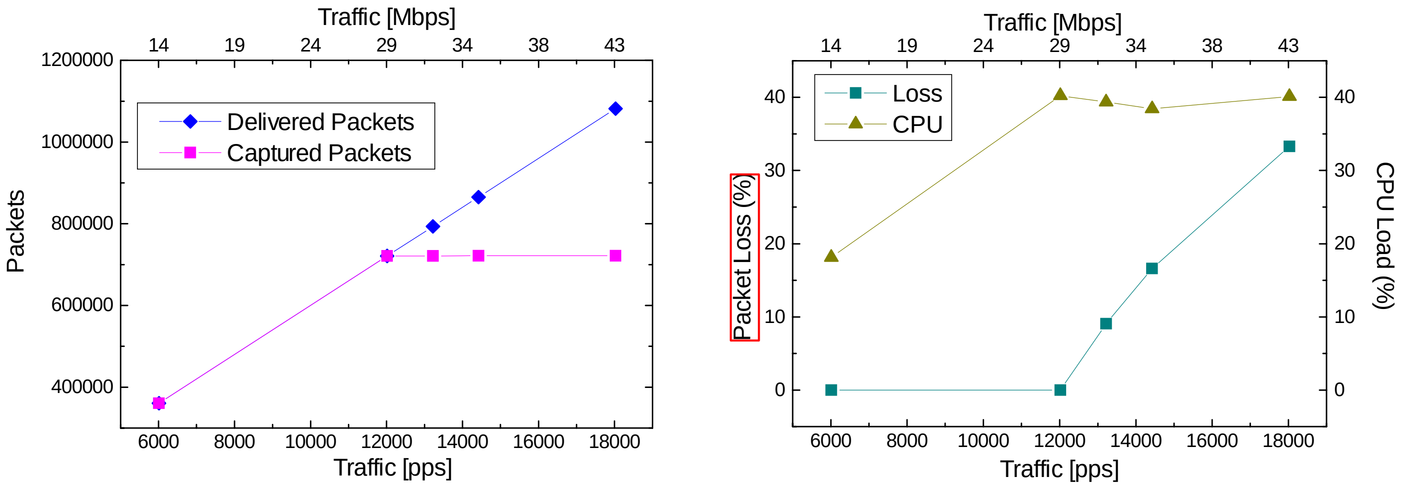


Figure 2-21: AMD-K6 450 MHz & NIC 10 Mbps Performance Numeric Results

2.3.3 Conclusions

Figure 2-22 shows that there is no significant difference on the CPU load for both types of NIC tested, so the only reason why the Packet Loss behavior is different is because of the lower limit rate of the 10 Mbps NIC. When the traffic rate is near of the NIC's limit rate then the packet loss rate rapidly increases.

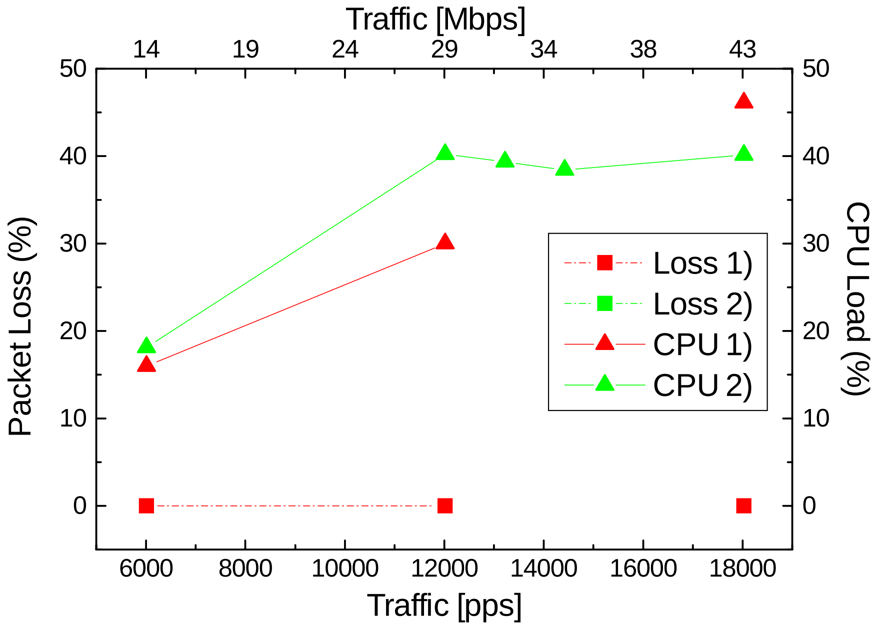


Figure 2-22: **CPU Load and Packet Loss versus Delivered Traffic with different NIC**

Figure 2-23 shows together the behavior of the two different tested platforms for a better comparison in order to extract some conclusions.

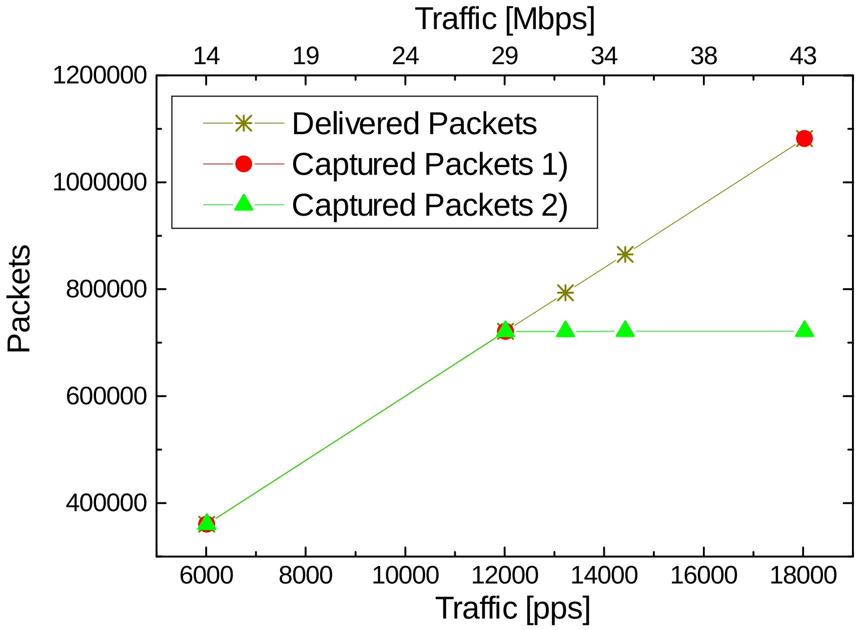


Figure 2-23: **Performance Comparison Results**

The influence of NIC operation rate on the number of captured packets is clear: while the NIC limit rate is below the CPU saturation point, the captured packets is the same than delivered packets, at least until the NIC limit rate is reached, and then there is packets loss. Otherwise, when the NIC limit rate is above the CPU saturation point, the captured packets rapidly decrease since this point.

2.4 Influence of the filtering rules in the meter

The purpose of these tests is to know how the number of filtering rules configured in the meter can influence in its performance. Following the topology proposed in figure 2-1, the limit rate of the meter was found for a given configuration system working with minimum CPU load – i.e. single rule. However, the performance of the meter can be influenced by the complexity of the filter, so it is necessary to make different tests with different filtering rules to see how the limit rate defined in the previous section varies.

Following this philosophy, three different tests, i.e., no filter, medium complexity filter and high complexity filter, have been made, but keeping the same hardware configuration. The following sections describe de main issues and results of these tests.

The following configuration was applied to all the tests described on the following sections and it is provided for a better understanding and further comparative tasks:

- CPU: AMD-K6, 450 MHz.
- Cache: 64 KB.
- RAM: 64 MB.
- NIC: 100 Mbps:
- Operating System: Linux version 2.4.20-8.

The specific rules for filtering traffic that were used are:

1) No filter: capture all the traffic.
2) Medium complexity filter: 'src 2001:DB8:400:800::13'. This rule captures only traffic coming from the source IPv6 address 2001:DB8:400:800::13.
3) High complexity filter: 'src 2001:DB8:400:800::13 and dst 2001:DB8:400:800::14 and tcp and src port 1500'. This rule captures only traffic matching all the parameters of the rule.

Below both the numeric and graphic results are shown.

2.4.1 No filter

Results corresponding to this configuration were already shown on figures 2-3 and 2-4.

2.4.2 Medium complexity filter

Results corresponding to this test are presented below.

Network Load (%)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Packet Loss (%)	CPU Load (%)
5	6.048	362.898	362.898	0,00%	18,28%
10	12.097	725.802	725.802	0,00%	33,18%
15	18.145	1.088.706	1.088.706	0,00%	50,46%
20	24.194	1.451.610	1.238.676	14,67%	99,90%
25	30.242	1.814.514	1.055.826	41,81%	99,90%
30	36.290	2.177.418	790.632	63,69%	99,82%
35	42.339	2.540.322	418.338	83,53%	99,80%

Figure 2-24: Medium Complexity Filter Performance Numeric Results

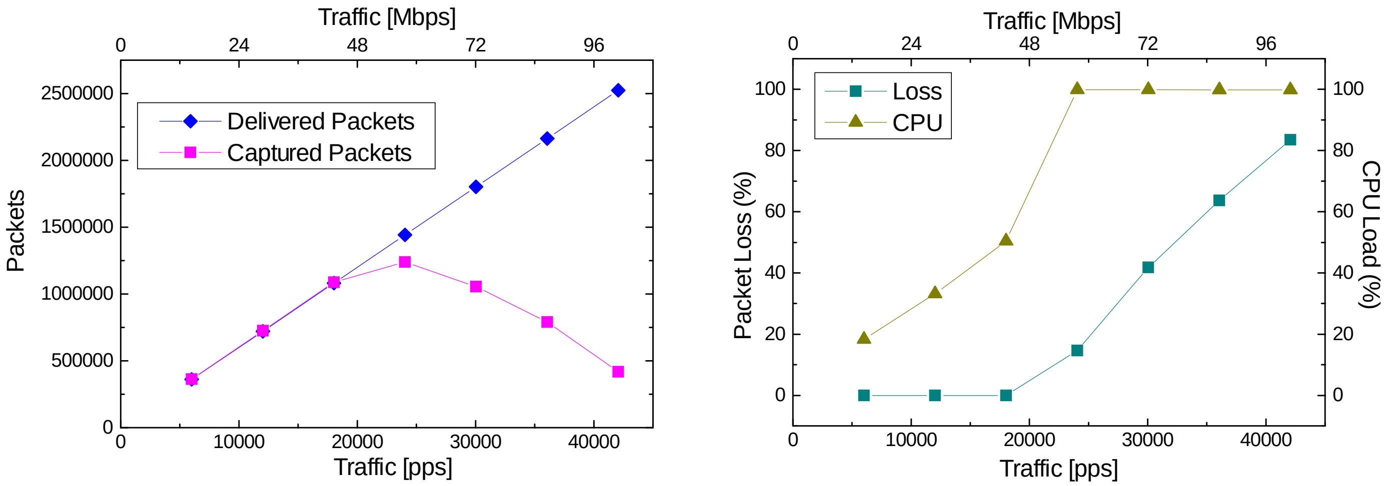


Figure 2-25: **Medium Complexity Filter Performance Graphic Results**

2.4.3 High complexity filter

Results corresponding to this test are presented below.

Network Load (%)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Packet Loss (%)	CPU Load (%)
5	6.048	362.898	362.898	0,00%	15,98
15	18.145	1.088.706	1.088.706	0,00%	49,35
20	24.194	1.451.610	1.242.900	14,38%	99,90
25	30.242	1.814.514	1.058.190	41,68%	99,88
30	36.290	2.177.418	763.674	64,93%	99,85
35	42.339	2.540.322	353.766	86,07%	99,85

Figure 2-26: **High Complexity Filter Performance Numeric Results**

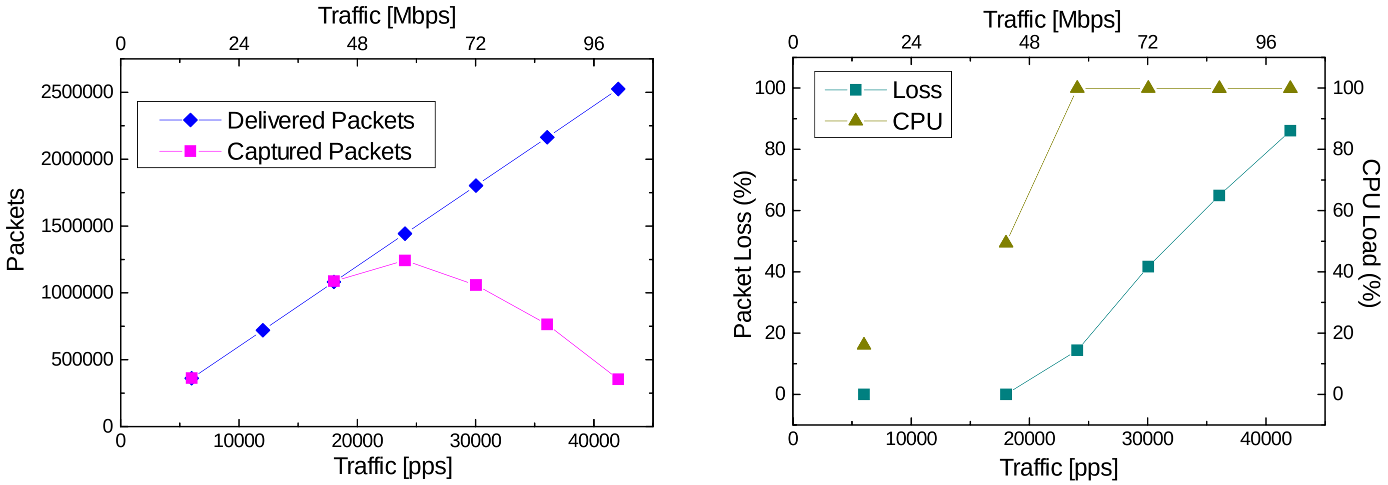


Figure 2-27: **High Complexity Filter Performance Graphic Results**

2.4.4 Conclusions

To better understand the influence of the filtering rule complexity on the CPU load, the above results are presented together in the following picture.

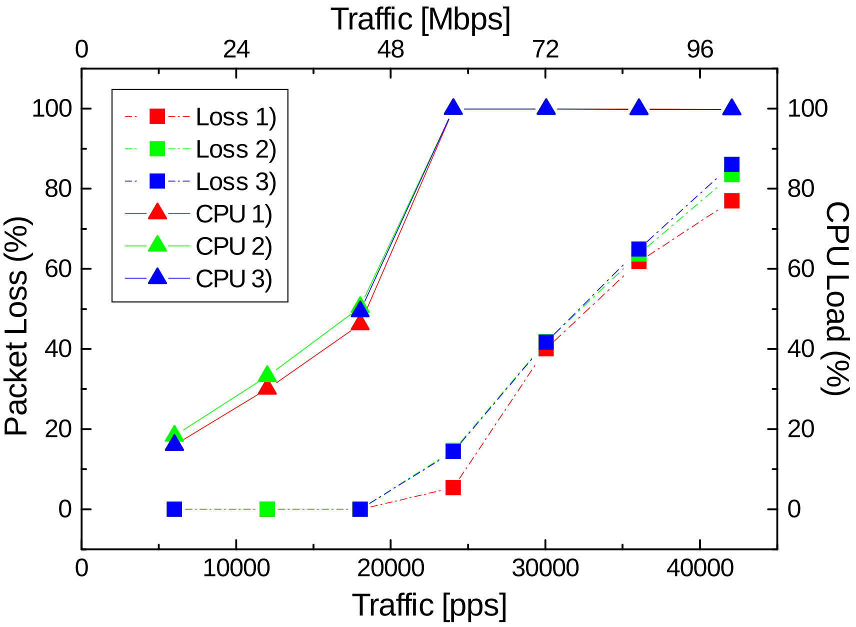


Figure 2-28: Packet Loss versus Delivered Traffic with different filtering rules

The next figure helps to obtain conclusions about the influence on the probe's performance of the filtering rule complexity.

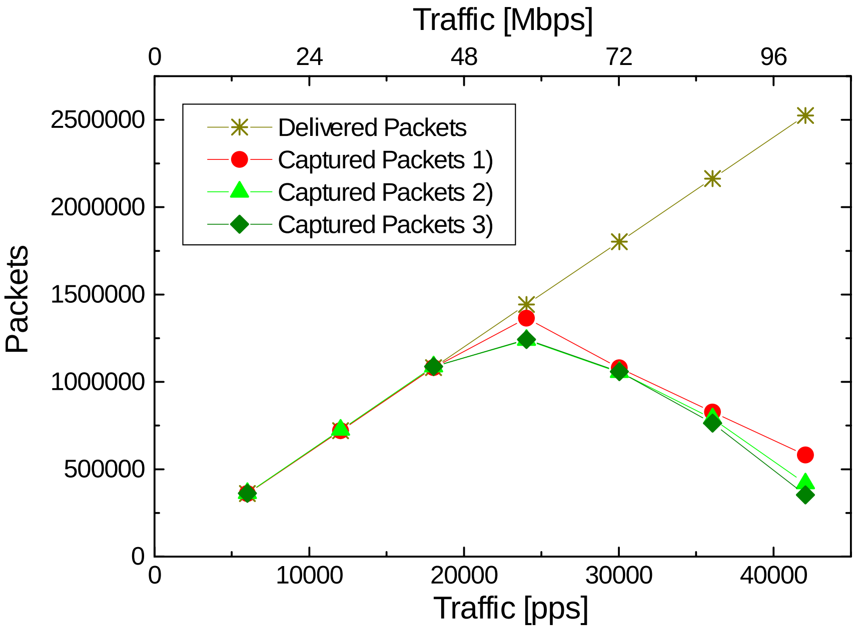


Figure 2-29: Captured packets for different filtering rules

The graphics show similar results for all the filters and as it was stated for the RAM case, due to all the experimental results belong to a trusted interval and that the observed differences are very small, we can conclude that the complexity of rule used does not influence the global performance.

2.5 Influence of the number of measurement tasks running on the probe

Until now all the tests done were realized by running only one measurement task on the probe at a time, that is, the probe was only configured with the '*volume measurement*', so it has only to capture packets. In a real scenario, the probe could have to make several measurements, so several measurement tasks running at a time on the probe could exist.

This test tries to find out if the number of tasks running on the probe can influence on the performance. To do that, the testbed network topology shown in figure 2-1 was implemented and the following tasks were configured on the probe:

1) Volume Measurement. This is the measurement usually configured to capture packets and it was the used to do all the performance tests.
2) Passive One Way Delay Measurement. The measurement to know the one way delay of the existing traffic on the network.
3) Active One Way Delay. An UDPv6 flow was configured with packets of 300 bytes (payload) and a rate of 1.000 packets per second. The total bandwidth of the flow on the LAN was about 3 Mbps.
4) Traffic Measurement. The measurement to know the total bandwidth on the network.

The configuration of the probe with this CPU was as follows:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 1 GB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

The traffic generated by the SmartBits was configured at 100% of network load to stress the probe and the two frame sizes were chosen: 300 and 1.500 bytes in order to take an idea of the behavior of the probe with low-medium frames and with highest frames.

Both numeric and graphics results are presented below.

Number of Tasks	Frame Size (Bytes)	Network Load (%)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Packet Loss (%)	CPU Load (%)
1	300	100	39.063	2.343.750	2.343.109	0,03%	33,43%
1	1.500	100	8.224	493.421	493.371	0,01%	7,17%
4	300	100	39.063	2.343.750	2.340.427	0,14%	53,58%
4	1.500	100	8.224	493.421	492.971	0,09%	10,77%

Figure 2-30: Several Tasks Performance Numeric Results

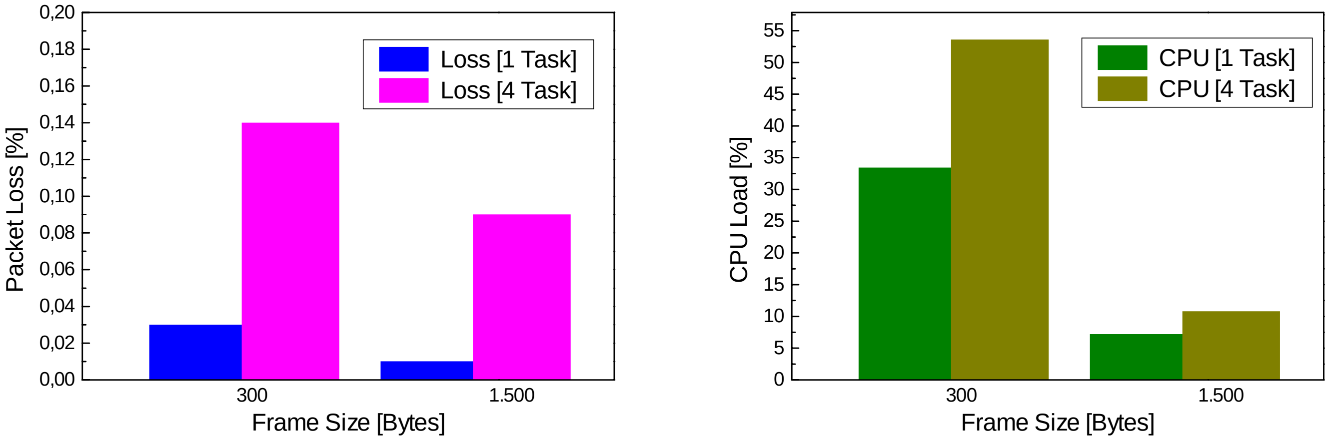


Figure 2-31: Several Tasks Performance Graphic Results

2.5.1 Conclusions

As expected the number of measurements tasks running on the probe influences on the performance of the probe. This can be shown in figure 2-31 right where the CPU load is always higher on the tests done with 4 measurement tasks. The increase on CPU load depends on the number and type of measurements tasks, being very hard to evaluate numeric percentages for all of them. However, the frame size is an objective parameter that can be used to evaluate how the number of measurement tasks influences and this general rule seems to be applied: the lower the frame size, the higher the increase of CPU load.

However, although the increase on the CPU load is very visible, that seems to affect negligibly on the packet loss as can be observed on the figure 2-31 left. With packets of 300 bytes the increase on packet loss is less than 0,12% over more than 2.300.000 delivered packets. With packets of 1.500 bytes, the increase is even lower.

So taking into account that results the following can be stated: the number of measurement tasks running on the probe at a time could affect on the limit traffic rate that the probe works without packet loss. For powerful architectures, like the one used in this test, the influence is negligibly for frames on the range 300 to 1.500 bytes (layer II, including CRC), and the traffic limit rate is approximately the same that the one calculated on section 2.1. However for frames lower than 300 bytes, the CPU load presents greater values (even for one measurement task), so the influence of the number of measurement tasks might be clearly greater because the increase of CPU load required to take care of more than one measurement task might decrease the limit traffic rate.

2.6 Influence of frame size

As it was above stated, the tests done to characterize diverse CPU types were all done by using frames of 84 bytes in order to reach the maximum traffic rate in packets per second. One could think that the frame size might influence on the performance of probes. If so, packet loss might appear when increasing the frame sizes at a given traffic rate. The objective of this test is to check the influence (if any) of the frame size on the performance and to evaluate such influence.

To do so the network testbed topology implemented was the one shown in figure 2-1. The configuration of the probe with this CPU was as follows:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 1 GB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

According to figure 2-2 the interval time between packets (Tbp) chosen for this test was 1 µs, which was very close to the minimum on an Ethernet network (0.96 µs). The objective was to start the test at a traffic rate where is known that packet loss is going to appear and then by modifying the frame size to observe the evolution of the packet loss rate. The following frame sizes (all layer II, including CRC) were used:

1) 84 bytes
2) 300 bytes
3) 600 bytes
4) 900 bytes
5) 1200 bytes
6) 1500 bytes

Bellow both numeric and graphics results are provided for each frame size

Frame Size (B)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
84	96.525	5.791.514	4.670.368	80,64%	19,358%	94,82%
300	39.002	2.340.093	2.339.640	99,98%	0,019%	33,89%
600	20.145	1.208.709	1.208.559	99,99%	0,013%	17,69%
900	13.580	814.777	814.702	99,99%	0,009%	11,83%
1.200	10.242	614.502	614.442	99,99%	0,010%	8,56%
1.500	8.221	493.258	493.213	99,99%	0,010%	7,17%

Figure 2-32: Several Frame Size Performance Results

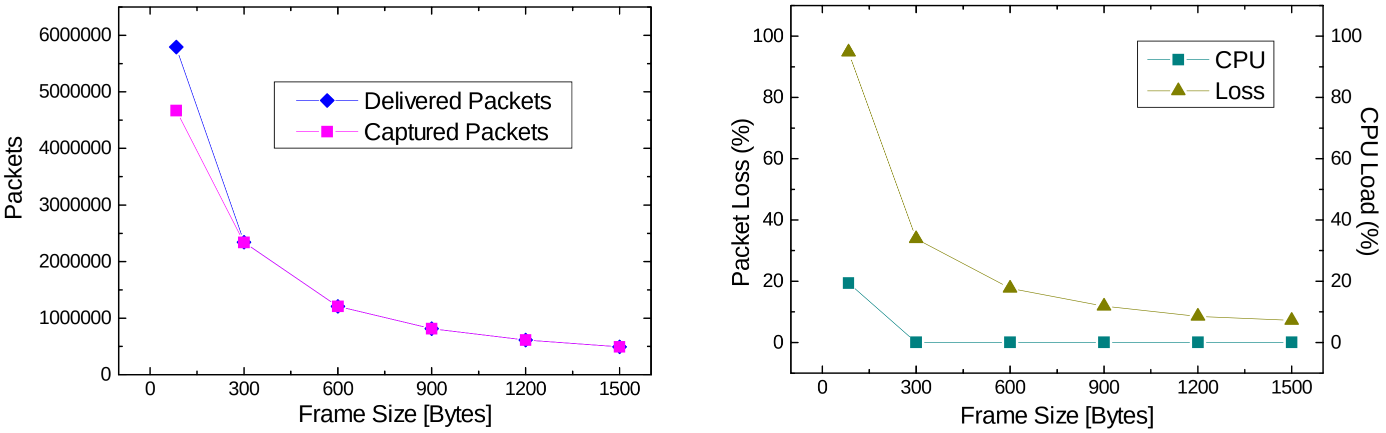


Figure 2-33: Several Frame Size Graphics Results

2.6.1 Conclusions

By taking a look on figure 2-33 right, one can see that the CPU load decreases exponentially when the frame size increases. This fact shows that the limit traffic rate is reached only with low frame sizes which stress the CPU more intensively than higher ones. As consequence, packet loss only might appear for low frame sizes. If even there is no packet loss for such frames, for higher frames there is unlikely that loss arises. This is the reason why the packet loss line has the zero constant value on the right figure.

As conclusion, it can be stated that once the probes is working bellow its limit traffic rate, if the frame size increases, the limit traffic rate does not change and packet loss rate will decrease.

2.7 Influence of interval time between packets

The previous tests showed that once the probe is working without producing packet loss at a traffic rate, if the frame size increases it does not influence on the packet loss because the CPU works less stressed. Such tests were done sending frames with the same interval time between frames (Tbp), but one could ask how the different values for the Tbp can affect to the performance of the probe. That is the target of this new test.

To do that, the network testbed topology shown in figure 2-1 was implemented, as usual and now all the packets sent had the same size of 300 bytes. That frame size was chosen because it is near of the Internet average packet size (even it is slightly below, so the test stresses a little more the CPU) and it gives us an idea of the behavior of probe with real traffic.

The configuration of the probe with this CPU was as follows:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 1 GB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Different Tbp was chosen to evaluate the behaviour of the probe while capturing packets. Such Tbp were:

1) 1 µs
2) 3 µs
3) 5 µs
4) 7 µs
5) 9 µs
6) 15 µs
7) 25 µs

And the results were as follows.

Tbp (µs)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
1	39.002	2.340.093	2.339.640	99,98%	0,019%	33,89%
3	36.179	2.170.757	2.170.431	99,98%	0,015%	31,81%
5	33.738	2.024.296	2.024.001	99,99%	0,015%	29,23%
7	31.605	1.896.328	1.896.025	99,98%	0,016%	27,09%
9	29.727	1.783.593	1.783.323	99,98%	0,015%	25,49%
15	25.227	1.513.617	1.513.450	99,99%	0,011%	22,40%
25	20.145	1.208.695	1.208.614	99,99%	0,007%	17,94%

Figure 2-34: Time Between Packets Performance Results

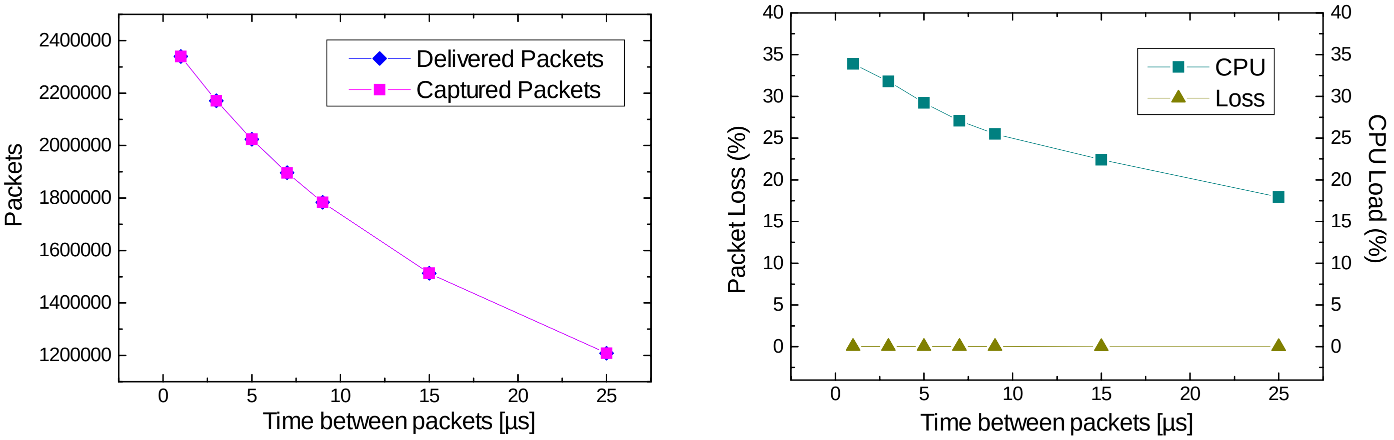


Figure 2-35: Time Between Packets Performance Graphics Results

2.7.1 Conclusions

It is interesting to observe in figure 2-35 right how the CPU load of the probe decreases almost linearly when the interval time between packets increases. In fact this is an expected behavior because the number of packets sent also decreases, as shown in table 2-34, and hence the CPU is less stressed.

In this case the produced packet loss is negligible (less than 0,02% over more than 1 million of sent packets) and it is almost constant when increasing Tbp.

As main conclusion it can be stated that the decrease of Tbp stresses the CPU and the CPU load increases linearly, although the packet loss does not increase if the traffic rate is below the limit traffic rate, so the Tbp variation do not influence on the packet loss rate.

2.8 Influence of the duration of the measurement

The final test about performance tries to evaluate if the measurement duration has any influence on the CPU load and hence on the lost packets by the probes. As usually, the testbed network topology implemented was the one shown in figure 2-1. To do these tests the probe was configured with the following platform:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 1 GB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

All the frames in these tests had 84 bytes of length (layer II, including CRC) and the network load chosen were:

1) 45 %
2) 50 %

Both the numeric and the graphics results are presented bellow.

Network Load (%)	Measur. Duration (s)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Captured pkts (%)	Packet Loss (%)	CPU Load (%)
45	60	54.087	3.245.192	3.244.916	99,99%	0,01%	48,98%
45	180	54.087	9.735.576	9.734.795	99,99%	0,01%	50,31%
45	300	54.087	16.225.961	14.780.411	91,09%	8,91%	50,45%
45	600	54.087	32.451.922	31.320.898	96,51%	3,49%	50,29%
50	60	60.096	3.605.769	3.605.401	99,99%	0,01%	37,93%
50	180	60.096	10.817.307	10.816.220	99,99%	0,01%	39,13%
50	300	60.096	18.028.846	15.921.162	88,31%	11,69%	38,72%
50	600	60.096	36.057.690	32.626.796	90,48%	9,52%	77,87%

Figure 2-36: Measurement Duration Performance Results

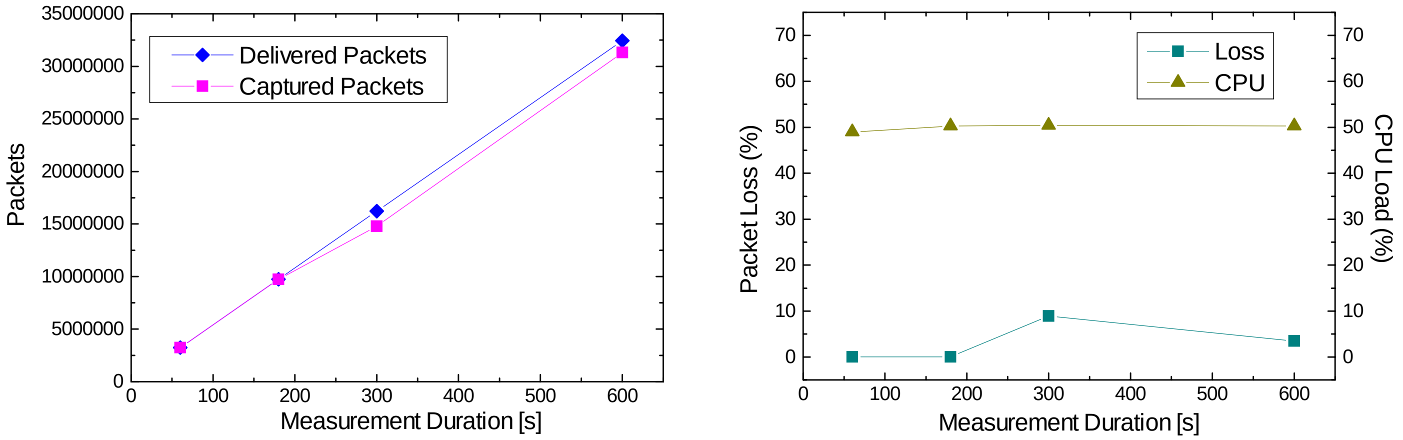


Figure 2-37: Measurement Duration Performance Graphics Results (1)

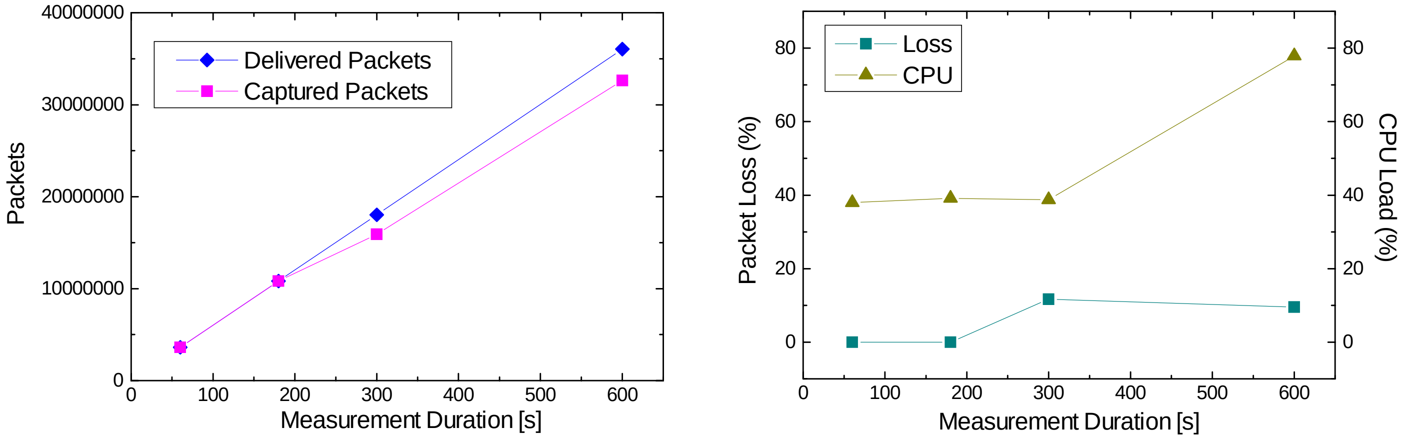
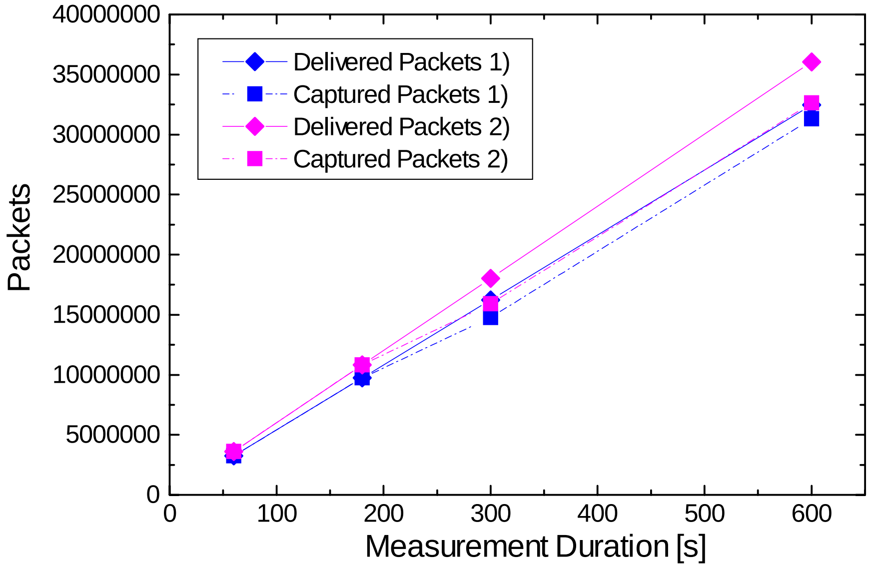


Figure 2-38: Measurement Duration Performance Graphics Results (2)

2.8.1 Conclusions

To better understand the influence of the measurement duration on both the CPU load and packet loss, the above results are presented together in the following pictures.



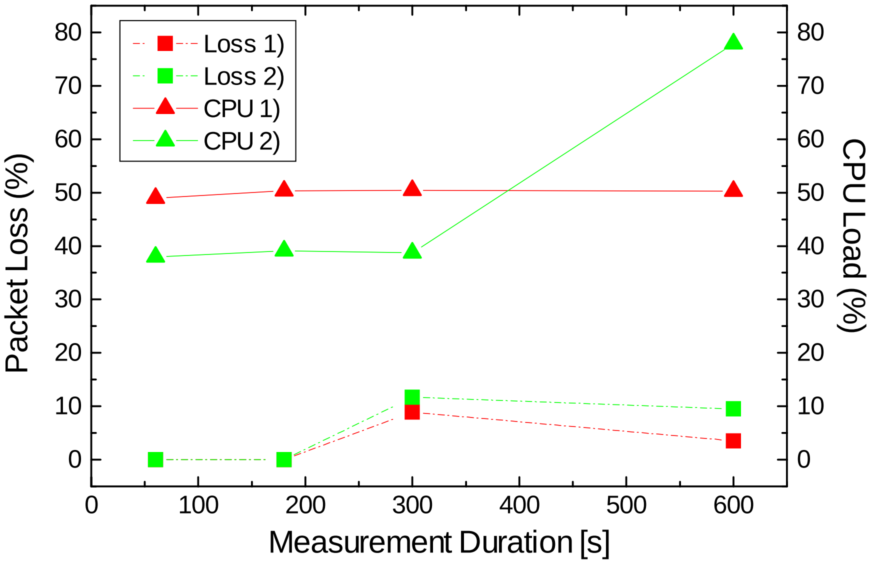


Figure 2-39: Result summary with different measurement duration

The clearest evidence that one could get after analyzing the results is that the packet loss rate seems to increase when the measurement duration also increases. That behavior seems to appear when the measurement duration is between 180 and 300 seconds where the packet loss rate reaches the value around the 10% for both network load of 45 and 50%. After that point the packet loss percentage seems to be stabilized.

The reasons for that behavior are not clear because the main motivation of packet loss is usually the CPU load values too high. However that is not the reason in this case because the CPU load response is almost plain for the whole range of the measurement durations. At least this is true for the case of network load of 45% (case 1). In case 2 (network load of 50%) there is a peak on the CPU load value gotten for a measurement of 600 seconds which is unusual compared with the previous values for the same case. The strange value can be due to irregularities in the process of measurement of the CPU load (`top` command). This argument can be supported for the no-correlation between the CPU load and packet loss for that measurement duration. If really the CPU load for 600 seconds were around 80%, the packet loss could not be lower than the measurement duration of 300 seconds, as it is the case, so the CPU load value for 600 seconds seems to be an spurious due to irregularities with the `top` command.

3. TIME PRECISION TESTS

Time precision tests are perhaps the most important since they give us an idea of how precise the system is when performing time-based measurements. These tests inform about the precision of the results provided by the prototype and how different aspects similar to the ones listed above influence on it. As one of the most important measurements that the 6QM System can do is the passive one-way-delay (POWD), then the accuracy tests are focused on it, although evaluations about the precision of active one-way-delay (AOWD) are also done.

There are at least two factors that strongly influence on the precision of time-based measurements performed by any QoS measurement system: synchronization of probes and the timestamp jitter.

The former can be minimized by using proper synchronization systems. The preferred one is a GPS antenna directly attached to the probe that sends a synchronization signal once per second. The probes must to have the proper software that adjusts the internal clock according to the GPS signal. This is the best method because it has the best time precision, near of nanoseconds. Other alternative is that each probe uses a NTP server to adjust its internal clock by using the NTP protocol. When two or more probes use the same NTP server attached to the same segment LAN that the probes, the synchronization of probes can be good enough, less than 0.5 ms, according to our experience on the laboratory.

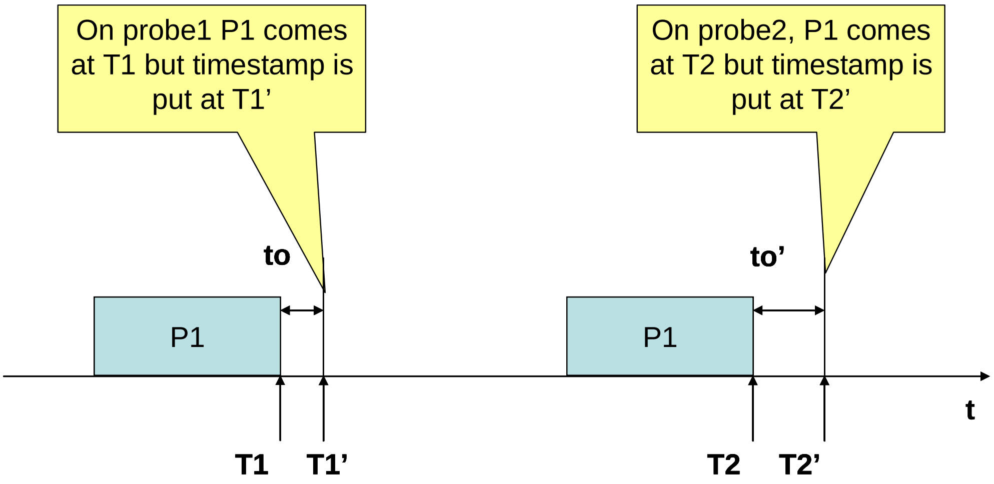
The method chosen to synchronize the probes involved on the precision of OWD tests were the second one, as will be detailed on section 3.1.2. So it can be stated that the synchronization of probes was not an issue on the tests performed so it did not influence negatively.

The second factor that strongly influence on the precision of the measured OWD is the timestamp that each probe put on each captured packet. This is the factor that is more difficult to control because it depends on the tasks that both the operating system and the measurement task must do. At least the probe hast to do the following tasks for each captured packet (not necessarily sorted):

- The NIC send an interruption to the CPU once a packet is captured
- The measurement task asks the Operating System about the time
- The Operating System extracts the time from the internal clock and it sends it to the measurement task
- The measurement task calculates the packet ID
- The measurement associates the timestamp to the packet ID

The time that the probe takes to do each task can randomly vary from time to time and it depends of factors like the Operating System used, the type of CPU, the CPU load, etc. As consequence, the interval of time from the packet is captured until the timestamp is calculated and put on the packet (so called offset timestamp) might be no constant for each captured packet, so a timestamp jitter is produced that might directly influence on the precision of the calculated OWD.

Figure 3-1 illustrates the effect of having timestamp offset on the calculated OWD.



$$\text{Delay} = T2' - T1' = (T2 + to') - (T1 + to) = (T2 - T1) + (to' - to) =$$

$$\text{OWD} + \text{Toffset}$$

Figure 3-1: Timestamp Offset

As can be observed, having large timestamp offset makes the system little reliable. Even worse is having a random large timestamp offset, that is, large timestamp jitter. So the evaluation of the timestamp jitter is one of the tasks done within the time precision tests.

Taking these considerations into account, the time precision tests done to estimate the reliability of the OWD results provided by the 6QM prototype were the following:

- Timestamp jitter. As explained before this tests evaluates the timestamp jitter to find out how influences on the global quality results
- Global precision of POWD. This test evaluates the precision of OWD provided by the 6QM prototype when real traffic is captured from the network.
- Global precision of active AOWD. This test evaluates the precision of the OWD provided by the 6QM prototype by using artificial traffic generated by the probes.

The description of both the methodology and network testbed implemented in all the three kinds of tests is presented bellow.

3.1 Description of tests methodology

3.1.1 Methodology to obtain the timestamp jitter

To characterize the timestamp jitter on the probe, the SmatBits 600 (SMB-600) device was used to deliver traffic. As is shown in figure 3-2 the probe was directly attached to the SMB-600 so all the traffic was injected on the probe without middle device. This is important because the SMB-600 was configured to deliver an IPv6-UDP flow at a traffic rate constant.

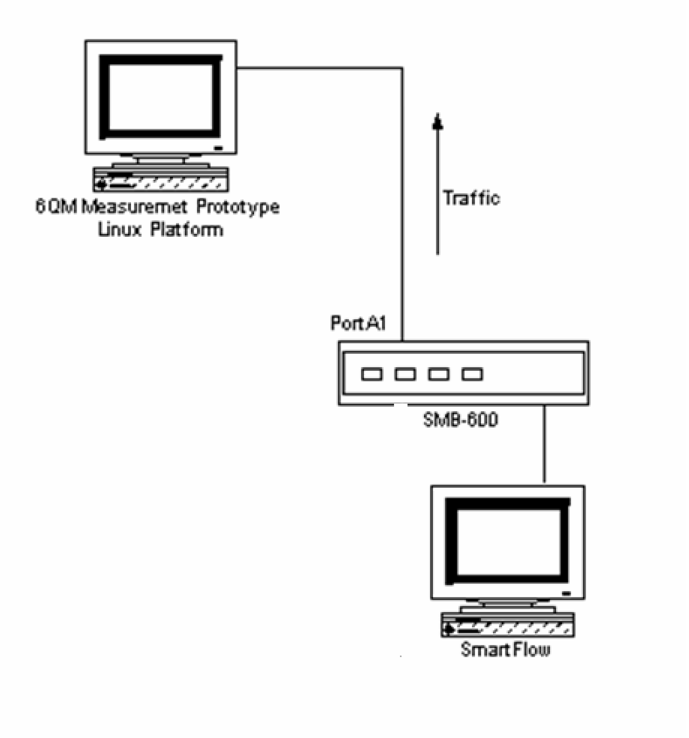


Figure 3-2: Test Topology to Calculate the Timestamp Jitter

All the packets belonging to the flow had the same size, although all of them had different payload that the SMB-600 randomly generated. The key element in this test is that the SMB send the packets at a constant traffic rate which means that the interval time between two consecutive packets (Tbp according to figure 2-2) was always constant for all the packets belonging to the flow.

Obviously it was not possible to obtain OWD results because only one probe was used on this test but it is not the target of this measurement. The main objective was to analyze the timestamp of each packet captured by the probe to extract statistics about Tbp on the probe. By comparing the Tpb calculated from the probe's data with the Tbp configured on the SMB-600 the timestamp jitter can be estimated.

This kind of test was done with the following types of platforms:

Probe 1

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Furthermore, different measurements with different traffic rates and frame sizes were also performed to know the influence of those factors on the timestamp jitter.

Results and conclusions are presented in section 3.2.

3.1.2 Methodology to obtain the global precision of the measured POWD

To evaluate the precision of the time-based results provided by the 6QM prototype the SMB-600 device was also used because its versatility. The topology used for these tests is depicted in the figure 3-3.

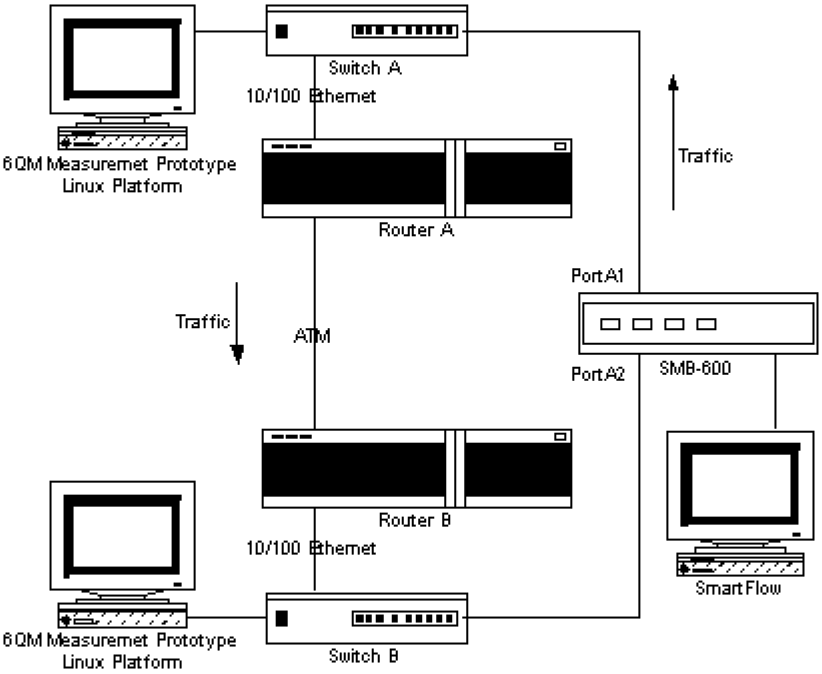


Figure 3-3: Test Topology to Calculate the Global POWD Precision

The SMB-600 system sent one IPv6-UDP flows from the Ethernet PortA1 to the Ethernet PortA2 during 60 seconds. The flow consisted of several IPv6 packets which were sent with a constant packet rate, which means that the interval time between two consecutive packets were always constant. All the packets had the same size ant their payload was randomly changed by the SMB-600.

Both intermediate routers were configured in order to simulate “real” network conditions with random delays and packet loss. This was done by configuring on router B a maximum outgoing bandwidth lower than the incoming bandwidth on router A.

Both OpenIMP probes were configured to capture only IPv6-UDP packets. Each probe put a timestamp to each captured packet. The calculator (not shown in the picture) then calculated the OWD based on those timestamps.

Each probe had two Ethernet cards. One of them was used to communicate with the 6QM controller (not shown in the picture) and to be synchronized with the GPS-NTP server (not shown in the pictured). The 6QM Controller/Calculator, the NTP server and the NIC of probes used to communicate with them were all attached to the same segment LAN, so only one hop between probes and the NTP server was needed. This allowed that the synchronization between the probes was very good. The worse measured offset between them was 0.47 ms.

The second Ethernet card on the probes was attached to the network shown in figure 3-3 and they was used for capturing the traffic delivered by the SMB-600.

The SMB-600 is also able to perform calculation about the OWD of packets delivered by itself. The precision of the SmartBits OWD or Latency reports is 0.10 µs, so it was a good reference in order to compare to the results obtained by mean of the probes.

In order to avoid the probes working near of their limit rate (and so with the risk of generate artificial packet loss), the traffic rate chosen to deliver the packets from the SMB was bellow the limit traffic rate calculated on the previous section for the probe attached to the SMB's PortA1. The same criterion was applied to the maximum bandwidth allowed on the output of Router B where the second probe was attached.

The performed tests have also evaluated the influence the different factors on the global precision of POWD. Such factors were:

- Frame size
- Several measurement tasks
- Time duration

The configuration for each host involved in all these tests was as follows:

Probe 1 (attached to Switch B)

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2 (attached to Switch B)

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Calculator

- CPU: Pentium II (Deschutes), 450 MHz
- CACHE: 512 KB
- Memory: 128 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.22 (gcc version 3.2 20020903 (Red Hat Linux 8.0 3.2-7))

3.1.3 Methodology to obtain the global precision of the measured AOWD

In order to know the precision of the measured AOWD, the network testbed implemented was the one shown in figure 3-4.

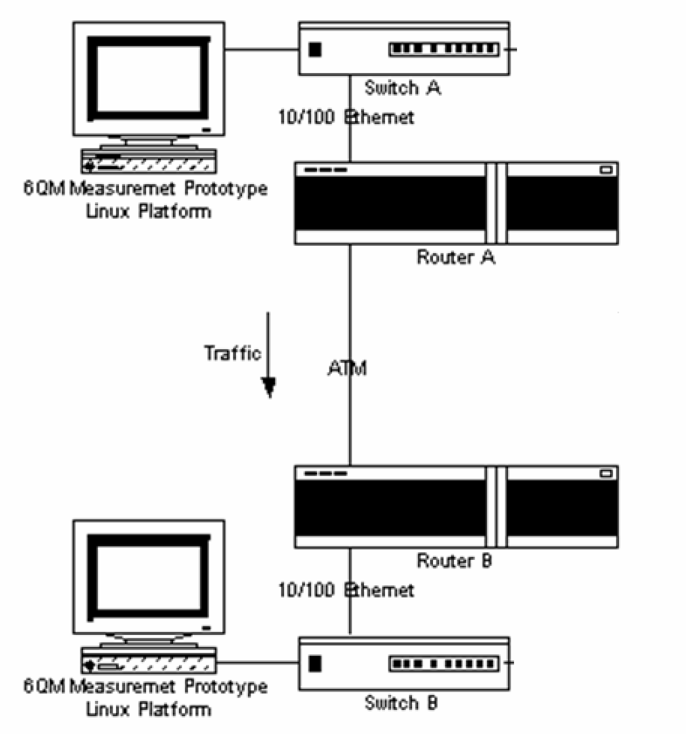


Figure 3-4: Test Topology to Calculate the Global AOWD Precision

Each probe was configured with two measurements tasks. One of them was the calculation of AOWD and the other one the calculation of POWD. The configuration of the later one was to capture only the flow generated by the active probe (AOWD task), so the evaluated packets of each task were the same.

The active flow consisted of IPv6-UDP packets of 600 bytes (layer II, including CRC) being delivered by the probe1 at 4.167 packets per second, so the bandwidth used on the network was 20 Mbps.

Given the fact that the precision of POWD was already calculated with the previous tests, the difference between active time statistics and passive ones is useful to calculate the precision of the AOWD.

The platforms used on this test were:

Probe 1

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

3.2 Timestamp jitter

According to the methodology explained in section 3.1.1, the precision of the timestamp has been evaluated for two platforms, as follows.

3.2.1 AMD-K6, 450 MHz and 64 MB

The complete data about this probe are the following:

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

All the packets had the same size (300 bytes, layer II, including CRC), the test duration was 60 seconds and the difference among these tests was the interval time between packets, Tbp according to figure 2-2.

To evaluate the timestamp jitter, it has been compared the sum of Tb and Tbp (according to figure 2-2) obtained by mean of SMB-600 and 6QM probe.

Both the numeric and graphic results are presented bellow.

Packet Size (B)	Tbp (µs)	Tp+Tbp (µs)	Network Load (%)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Captured pkts (%)

300	0,96	24,96	100,00	39.063	2.343.750	225.135	9,61%
300	13,44	37,44	67,23	26.261	1.575.632	462.243	29,34%
300	39,36	63,36	40,00	15.625	937.500	701.944	74,87%
300	183,36	207,36	12,31	4.808	288.468	288.453	99,99%
300	500	524,00	4,88	1.906	114.375	114.375	100,00%
300	1.000	1024,00	2,50	976	58.546	58.546	100,00%
300	5.000	5024,00	0,51	199	11.929	11.929	99,98%

Probe Measured Results						
Tp+Tbp						
Max. (µs)	**Min. (µs)**	**Mean (µs)**	**Stdvt (µs)**	**Error for Mean Tp+Tbp (µs)**	**Packet Loss (%)**	**CPU Load (%)**
385.281	10	266	881,83	**241,04**	90,39%	97,81
95.202	12	129	207,94	**91,56**	70,66%	98,68
218.747	36	85	297,95	**21,64**	25,13%	98,39
3.347	10	207	9,04	**0,36**	0,01%	28,03
739	356	524	5,81	**0,00**	0,00%	10,92
1.257	860	1.024	5,94	**0,00**	0,00%	5,74
5.279	4.785	5.029	8,45	**5,00**	0,02%	1,15

Figure 3-5: Timestamp Jitter Results for AMD K6 450 MHz

To summarize these results, the following statistics are represented:

- Maximum Tp+Tbp error. It means the difference between the maximum Tp+Tbp measured on the probe and the Tp+Tbp configure on the SMB-600. If no timestamp jitter were on the probe, there would not be maximum values for Tp+Tbp because of the values would be equal and consequently the error maximum Tp+Tbp would be cero.
- Minimum Tp+Tbp error. It means the difference between the minimum Tp+Tbp measured on the probe and the Tp+Tbp configure on the SMB-600. If no timestamp jitter were on the probe, there would not be minimum values for Tp+Tbp because of the values would be equal and consequently the error maximum Tp+Tbp would be cero.
- Mean Tp+Tbp error. It means the difference between the mean Tp+Tbp measured for each capture packet on the probe and the Tp+Tbp configure on the SMB-600. If no timestamp jitter were on the probe, there would not be maximum nor minimum values for Tp+Tbp because of the values would be equal and consequently the error mean Tp+Tbp would be cero. It can be considered as timestamp offset.
- Standard deviation of mean Tb+Tbp. It is not enough to characterize the timestamp jitter because it does not give an idea of how different are the obtained values. To better characterize the probe, the standard variation of the mean Tb+Tbp is also calculated. The standard deviation of mean Tb+Tbp can be considered as timestamp jitter.
- Packet Loss. The number of packets not captured.
- CPU Load.

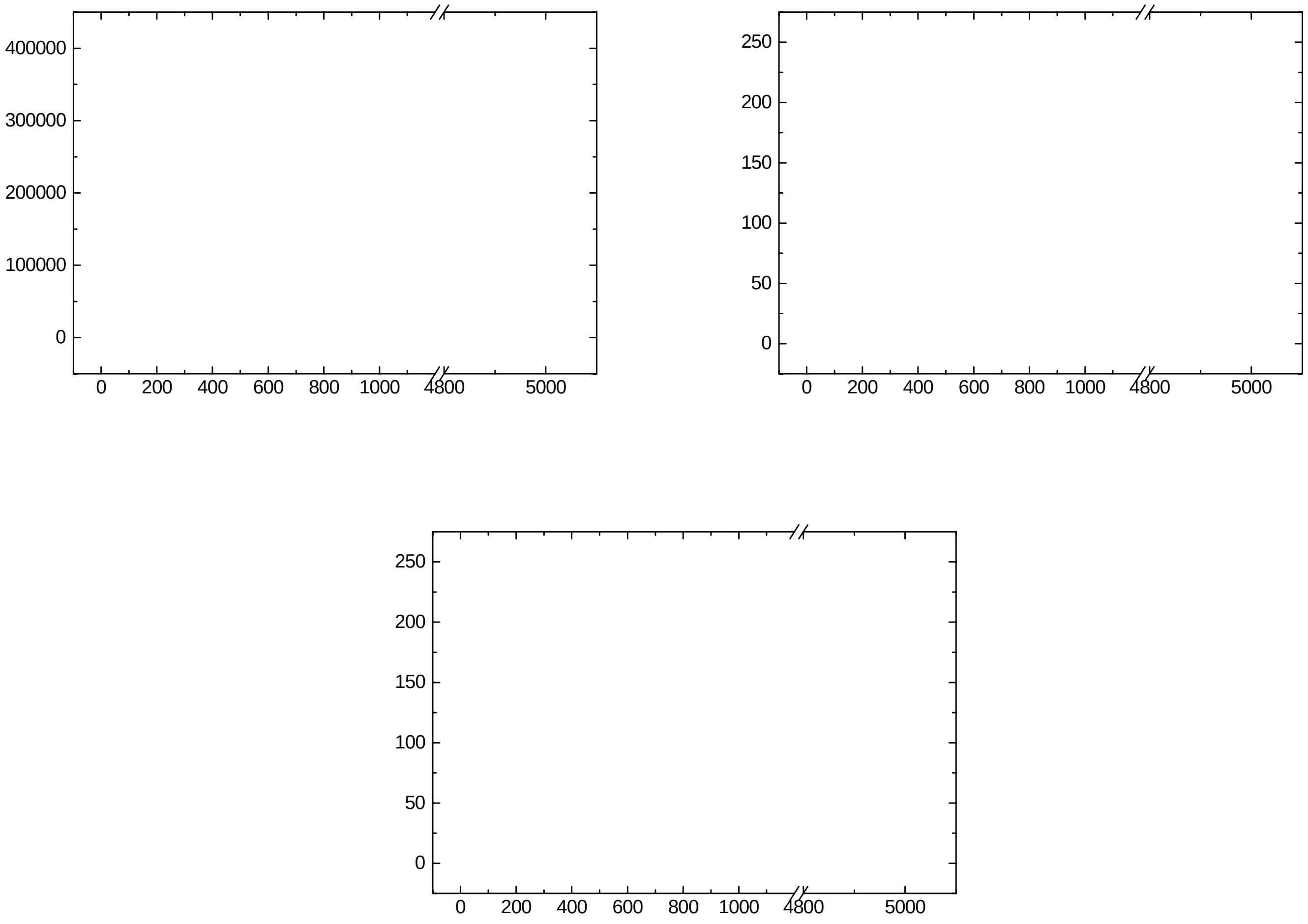


Figure 3-6: Error values for Tb+Tbp for AMD K6 450 MHz

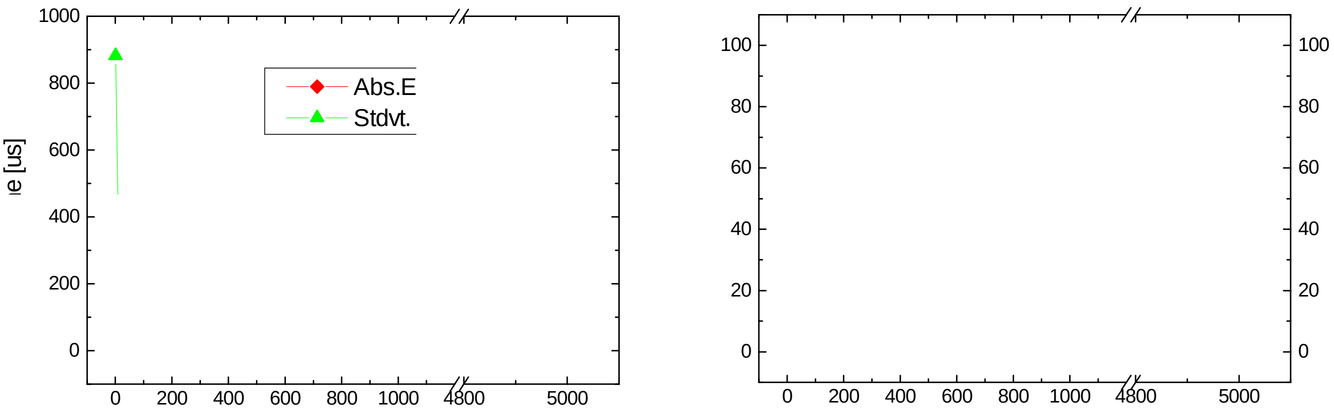


Figure 3-7: Tb+Tbp error and CPU-Packet Loss for AMD K6 450 MHz

3.2.1.1 Conclusions

The first conclusion is that the maximum and minimum error gotten for Tb+Tbp (timestamp offset) can vary from a wide range, as can be seen on figure 3-6. For the maximum values it seems that the error tends to be low when the gap between packets (Tbp) is great, whereas the opposite can be seen for minimum values.

The reason for the bad results of the maximum Tb+Tbp error can be the big packet loss produced for low Tb. As pictured on figure 3-7 the packet loss is very big for low Tb because the traffic rate is elevated. For example, values of 0,96 µs and 13,44 µs are equivalent to 39.063 packets per second (pps) and 26.261 pps respectively, which are beyond the limit traffic rate calculated for this platform on section 2.

When loss is produced, the lost packets are usually consecutive, so there is a big interval greater than Tp+Tbp that leads to the maximum Tb+Tbp error is also too big. So it can be concluded that there is some correlation between maximum Tb+Tbp error and packet loss.

On the other hand, mean Tb+Tbp error is very low, nearly zero when packet loss is not produced, while its value is greater when packet loss rate also increases. The reason for that is again the high values of maximum Tb+Tbp error that pulls up the mean values. Also, it can be observed some correlation between the standard deviation of the mean Tb+Tbp and the packet loss.

Finally, it can be stated that both the timestamp offset and the timestamp jitter are negligible when the probe is working at a traffic rate bellow its limit traffic rate because low or even no packet loss is produced. Furthermore, when the probe is working bellow its limit traffic rate, the gap between packets (Tbp) does not influence on the timestamp.

3.2.2 PIV, 2.8 GHz and 512 MB

The complete data about this probe are the following:

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

For this platform, different test were performed with different packets size (84, 300, 600, 900, 1200 and 1500 bytes, layer II, including CRC), the test duration was 60 seconds and the difference among these tests was the interval time between packets, Tbp according to figure 2-2.

To evaluate the timestamp jitter, it has been compared the sum of Tb and Tbp (according to figure 2-2) obtained by mean of SMB-600 and 6QM probe.

Both the numeric and graphic results are presented bellow.

Packet Size (B)	Tbp (μs)	Tp+Tbp (μs)	Network Load (%)	Traffic Rate (PPS)	Total Delivered pkts	Captured pkts	Captured pkts (%)
84	0,96	7,68	100,00	120.192	7.211.538	2.241.874	31,09%
84	13,44	20,16	40,00	48.077	2.884.615	2.883.306	99,95%
84	500	506,72	1,64	1.971	118.269	118.269	100,00%
84	1.000	1006,72	0,83	993	59.567	59.567	100,00%
84	5.000	5006,72	0,17	200	11.971	11.971	100,00%
300	0,96	24,96	100,00	39.063	2.343.750	2.340.472	99,86%
300	39,36	63,36	40,00	15.625	937.500	937.456	100,00%
300	500	524,00	4,88	1.906	114.375	114.375	100,00%
300	1.000	1024,00	2,50	976	58.546	58.546	100,00%
300	5.000	5024,00	0,51	199	11.929	11.929	100,00%
600	0,96	48,96	100,00	20.161	1.209.677	1.208.618	99,91%
600	75,36	123,36	40,00	8.065	483.870	483.870	100,00%
600	500	548,00	9,04	1.823	109.366	109.366	100,00%
600	1.000	1048,00	4,73	954	57.217	57.217	100,00%
600	5.000	5048,00	0,98	198	11.879	11.879	100,00%
900	0,96	72,96	100,00	13.587	815.217	814.434	99,90%
900	111,36	183,36	40,00	5.435	326.086	326.086	100,00%
900	500	572,00	12,85	1.746	104.779	104.779	100,00%
900	1.000	1072,00	6,86	932	55.940	55.939	100,00%
900	5.000	5072,00	1,45	197	11.828	11.828	100,00%
1.200	0,96	96,96	100,00	10.246	614.754	614.430	99,95%
1.200	147,36	243,36	40,00	4.098	245.901	245.901	100,00%
1.200	500	596,00	16,36	1.676	100.561	100.561	100,00%
1.200	1.000	1096,00	8,90	912	54.713	54.670	99,92%
1.200	5.000	5096,00	1,91	196	11.772	11.772	100,00%
1.500	0,96	120,96	100,00	8.224	493.421	493.202	99,96%
1.500	183,36	303,36	40,00	3.289	197.368	197.368	100,00%
1.500	500	620,00	19,59	1.611	96.675	96.675	100,00%
1.500	1.000	1120,00	10,85	892	53.541	53.506	99,93%
1.500	5.000	5120,00	2,37	195	11.718	11.718	100,00%

Figure 3-8: Timestamp Jitter Results for PIV 2.8 GHz (1)

Probe Measured Results						
Tp+Tbp						
Max. (µs)	**Min. (µs)**	**Mean (µs)**	**Stdvt (µs)**	**Error for Mean Tp+Tbp (µs)**	**Packet Loss (%)**	**CPU Load (%)**
61.934	1	26	62,72	**18,32**	68,91%	99,46
9.441	2	20	6,67	**0,16**	0,05%	58,00
520	496	507	0,71	**0,28**	0,00%	2,39
1.017	998	1.007	0,68	**0,28**	0,00%	1,09
5.021	5.004	5.012	0,87	**5,28**	0,00%	0,23
21.276	2	25	17,46	**0,04**	0,14%	46,85
2.944	39	64	3,03	**0,64**	0,00%	18,40
542	507	524	0,87	**0,00**	0,00%	2,34
1.036	1.015	1.024	1,02	**0,00**	0,00%	1,16
5.039	5.024	5.029	0,89	**5,00**	0,00%	0,33
6.051	20	49	13,41	**0,04**	0,09%	24,25
150	98	124	0,74	**0,64**	0,00%	9,63
557	540	548	0,89	**0,00**	0,00%	2,00
1.058	1.039	1.048	0,93	**0,00**	0,00%	1,12
5.060	5.043	5.050	1,26	**2,00**	0,00%	0,26
24.952	50	73	33,85	**0,04**	0,10%	16,38
203	165	183	1,23	**0,36**	0,00%	6,43
590	555	572	0,94	**0,00**	0,00%	1,94
2.145	1.063	1.072	4,63	**0,00**	0,00%	1,19
5.085	5.064	5.072	1,02	**0,00**	0,00%	0,33
21.278	74	97	28,63	**0,04**	0,05%	12,36
259	230	243	1,24	**0,36**	0,00%	4,70
611	584	596	1,00	**0,00**	0,00%	2,04
48.250	1.088	1.097	201,67	**1,00**	0,08%	1,03
5.110	5.088	5.096	1,07	**0,00**	0,00%	0,29
20.794	96	121	30,39	**0,04**	0,04%	9,76
319	289	303	1,21	**0,36**	0,00%	3,64
633	609	620	0,95	**0,00**	0,00%	1,83
40.343	1.112	1.121	169,56	**1,00**	0,07%	1,00
5.132	5.112	5.119	1,36	**1,00**	0,00%	0,25

Figure 3-9: Timestamp Jitter Results for PIV 2.8 GHz (2)

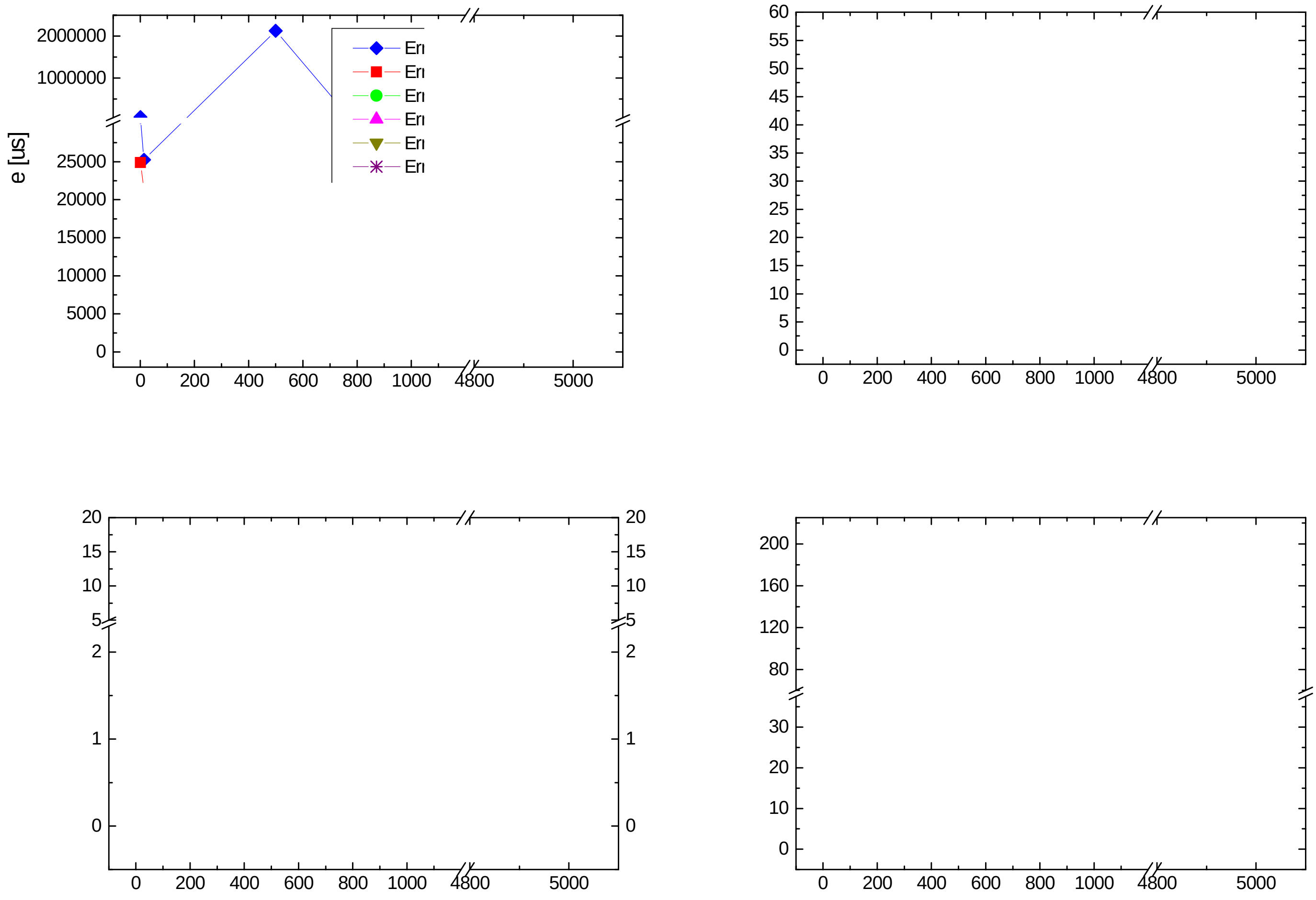


Figure 3-10: Error values for Tb+Tbp for PIV 2.8 GHz

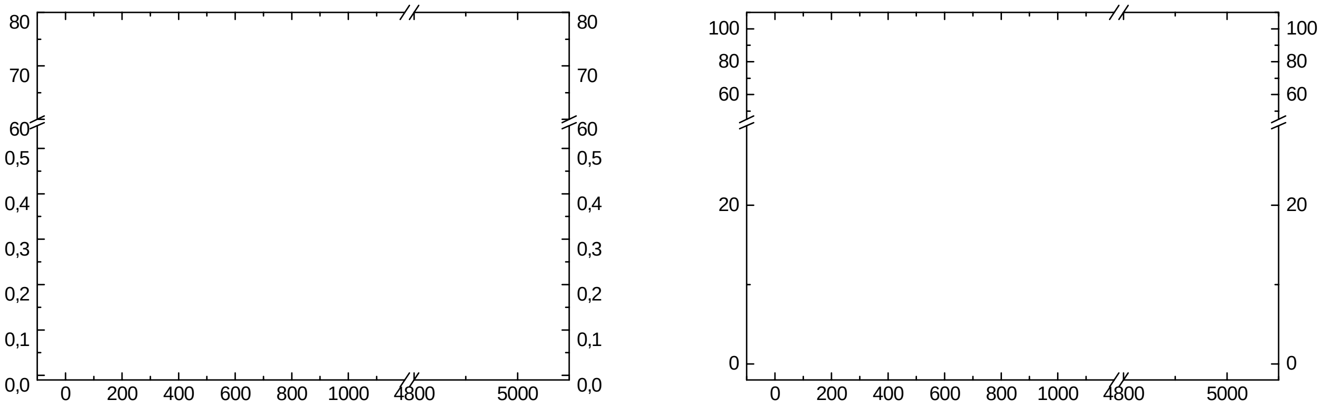


Figure 3-11: Packet Loss and CPU for PIV 2.8 GHz

3.2.2.1 Conclusions

Given the fact that this platform is more powerful, less packet loss is produced, even for high traffic rates (low Tbp). As consequence it cannot be visualized the same correlation between packet loss and maximum timestamp offset error than the previous case. However, in spite of the low packet loss, they continue causing random maximum timestamp offsets that can be very high.

With regarding the packet size, it seems that there is some correlation with the maximum timestamp offset: the higher the packet size, the lower the maximum timestamp offset error, which reinforces the conclusion of the correlation between packet loss and timestamp offset error because very low packet loss is produced for high packet sizes.

Regarding the minimum timestamp offset, it seems that for low packet sizes (upto 900 bytes) the higher the gap between packets (Tbp) the higher the minimum timestamp offset error and the opposite for high packet sizes (900, 1200 and 1500 bytes).

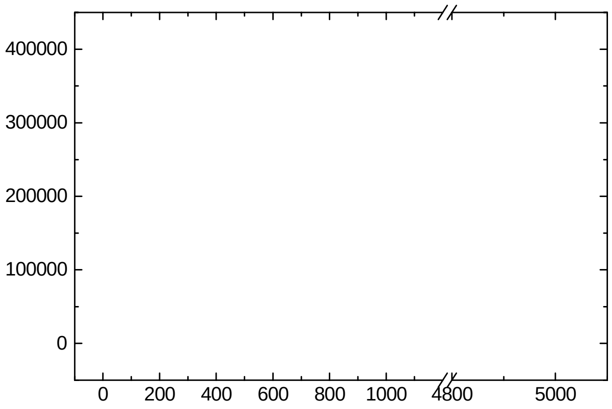
Related to the mean timestamp offset error, as it can be seen in figure 3-10 all the obtained vales are bellow 5 µs (except the first value for frames of 84 bytes), which indicates that the timestamp in this platform is exceptionally good because the CPU is rarely stressed.

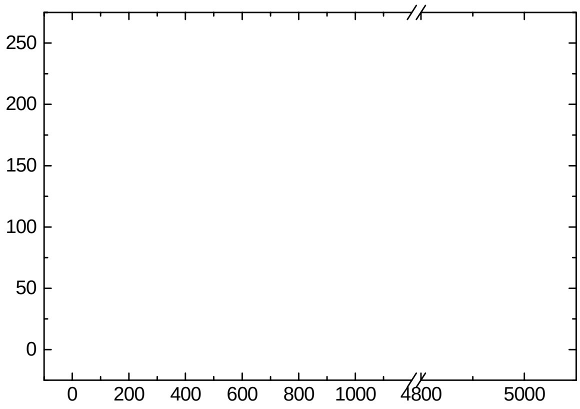
Finally, as expected, the timestamp jitter (standard deviation mean Tb+Tbp) has low values for all the packet sizes with two exceptions: when the CPU is very stressed, that is, very high traffic rate (very low Tb) and low packet size, the timestamp jitter can increase upto 80 µs. There is also a peak for Tb=1000 µs and packet size of 1200 and 1500 bytes. It seems that the only reason for that is the packet loss peak produced at those rates, which are over the average packet loss. In spite of that peak, the absolute value of timestamp jitter is bellow Tb+Tbp, so it can also be considered negligible.

Hence it can be concluded that both timestamp offset and jitter are negligible for traffic rates bellow the limit traffic rate and they do not depends on the packet size.

3.2.3 Comparison between platforms

Bellow a comparison between statistics for each platform is presented to extract some conclusions. The figures show the results obtained in the previous two sections for packets size of 300 bytes.





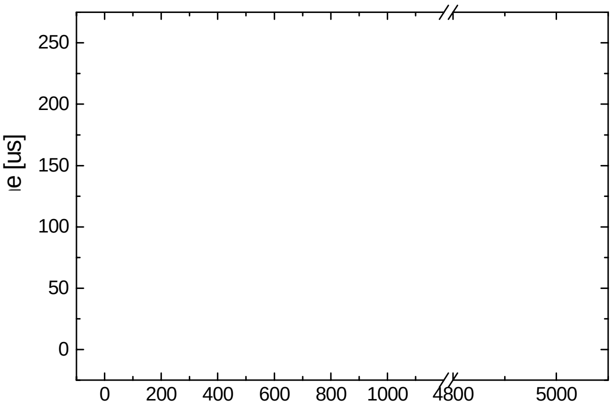


Figure 3-12: Error values for Tb+Tbp comparison

In figure 3-12 can be seen that the CPU power influences strongly on the timestamp errors: the more powerful, the better the timestamp offset statistics for low Tbp (high traffic rates). However, from an specific point, the results for both platforms are similar (except for minimum timestamp offset error). Such a point can be established around 500 µs.

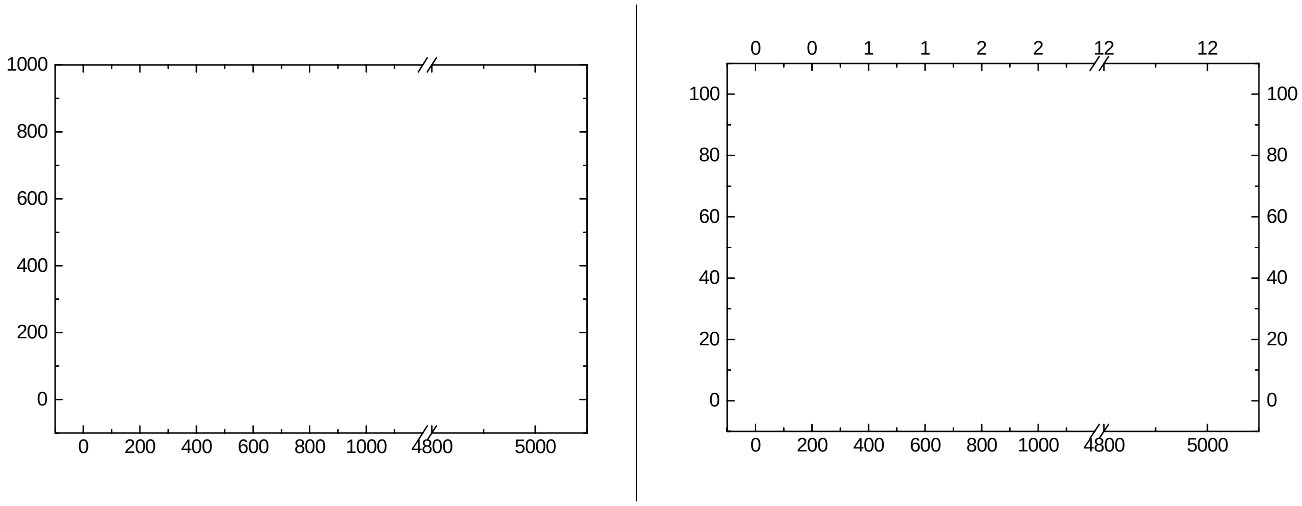


Figure 3-13: Tb+Tbp error and CPU-Packet Loss comparison

On the other hand, the same conclusion can be stated for the timestamp jitter, as shown in figure 3-13 left. It seems that the reason for that is because the packet loss because from Tb=500 µs, both CPU have similar values for packet loss. So the correct timestamp can be guarantied only if the traffic rate is bellow the limit traffic rate for each platform.

3.3 Global precision of measured POWD

Once the timestamp offset and the timestamp jitter have been evaluated, the next step is to evaluate the precision of the POWD information measured by the OpenIMP system under real traffic. To do so, the network topology shown in figure 3-3 was implemented and the procedure explained in section 3.1.2 was followed.

The platforms used for each probe were:

Probe 1

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Bellow the information presented by the SMB-600 and the OpenIMP system is presented. It is necessary to highlight that all the tests performed in this section (five in total) had the same characteristics in terms of frame size, traffic rate, etc.

			SmartBits Info							
Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Delivered Packets**	**Received Packets**	**Packets Loss**	**Packet Loss (%)**	**OWD min (us)**	**OWD max (ms)**	**OWD mean (ms)**
2,5	84	3.005	180.288	91.680	88.608	49,15%	496,9	327,08	136,07
2,5	84	3.005	180.288	87.799	92.489	51,30%	458,4	327,09	129,64
2,5	84	3.005	180.288	91.387	88.901	49,31%	504,5	327,09	139,48
2,5	84	3.005	180.288	68.363	111.925	62,08%	411,3	143,88	98,43
2,5	84	3.005	180.288	65.977	114.311	63,40%	478,5	149,57	108,25

Figure 3-14: Global precision POWD Results – SmartBits Information

			OpenIMP System Measurements							
Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Total Received Packets (Probe1)**	**Packet Loss (%) (Probe1)**	**Total Received Packets (Probe2)**	**Packet Loss (%) (Probe2)**	**Packet Loss (%) (Probe1-Probe2)**	**OWD min (us)**	**OWD max (ms)**	**OWD mean (ms)**
2,5	84	3.005	166.579	7,60%	91.676	0,00%	0,45	884,0	327,46	136,45
2,5	84	3.005	167.191	7,26%	87.792	0,01%	0,47	507,0	327,13	129,69
2,5	84	3.005	166.619	7,58%	91.380	0,01%	0,45	528,0	327,11	139,50
2,5	84	3.005	167.819	6,92%	68.362	0,00%	0,59	511,0	143,91	98,50
2,5	84	3.005	168.023	6,80%	65.974	0,00%	0,61	480,0	149,54	108,24

Figure 3-15: Global Precision POWD Results – OpenIMP System Measurements

Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Time offset (Probe1) (us)**	**Time offset (Probe2) (us)**	**ABS. Relative offset Probe1-Probe2 (us)**
2,5	84	3.005	-249	-477	228
2,5	84	3.005	-197	-333	136

2,5	84	3.005	34	57	23
2,5	84	3.005	3.863	3.847	16
2,5	84	3.005	-135	329	464

Figure 3-16: Global Precision POWD Results – Probes Synchronization

Bandwidth			Results							
Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	ABS. OWD min Error (us)	ABS. OWD min Error (%)	ABS. OWD max Error (us)	ABS. OWD max Error (%)	ABS. OWD mean Error (us)	ABS. OWD mean Error (%)	Packet Loss Error	Packet Loss Error (%)
2,5	84	3.005	387,1	77,90%	377,30	0,12%	376,31	0,28%	13.705	7,61%
2,5	84	3.005	48,6	10,60%	46,90	0,01%	46,67	0,04%	13.090	7,27%
2,5	84	3.005	23,5	4,66%	22,00	0,01%	13,03	0,01%	13.662	7,59%
2,5	84	3.005	99,7	24,24%	34,40	0,02%	70,57	0,07%	12.468	6,92%
2,5	84	3.005	1,5	0,31%	31,90	0,02%	16,91	0,02%	12.262	6,80%

Figure 3-17: Global Precision POWD Results – Numeric Results

To extract conclusions, the most relevant statistics form the previous tables are presented in the next figures. It includes:

- Minimun OWD. The minimum measured OWD values by the SMB-600 and the OpenIMP system are presented along with the relative error between them.
- Maximun OWD. The maximum measured OWD values by the SMB-600 and the OpenIMP system are presented along with the relative error between them.
- Mean OWD. The mean measured OWD values by the SMB-600 and the OpenIMP system are presented along with the relative error between them. It is also presented the synchronization offset between the probes used on the tests in order to provide an idea about the influence of the probes synchronization in the mean OWD error.
- Packet Loss. The packet loss rate in both probes is presented along with the mean OWD error inn order to provide an idea about te influence of the packet loss in the mean OWD error.

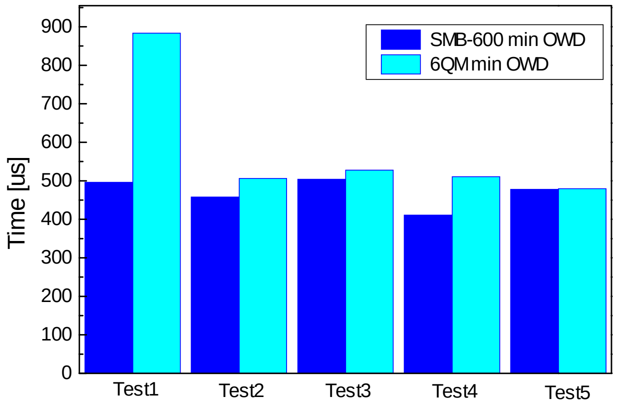


Figure 3-18: Global Precision POWD Graphic Results – Minimum OWD

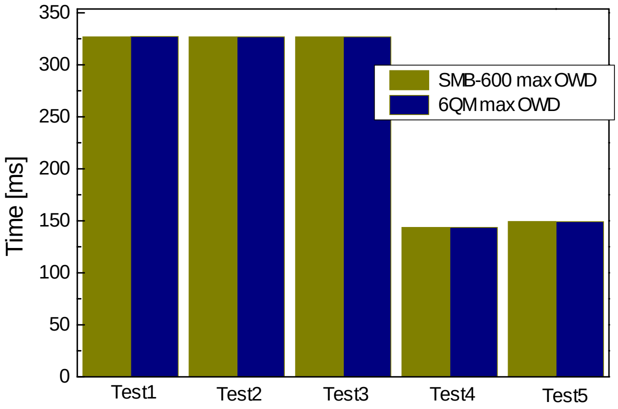


Figure 3-19: Global Precision POWD Graphic Results – Maximum OWD

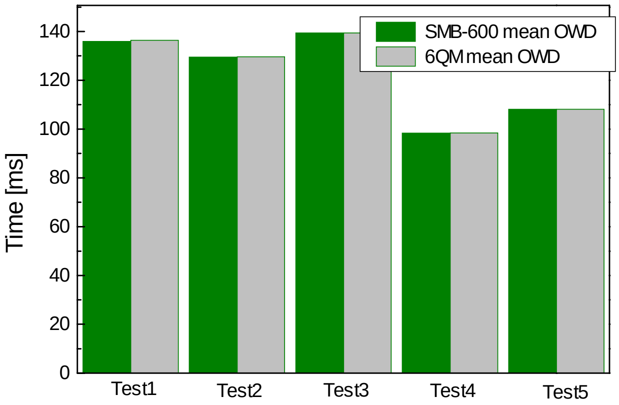


Figure 3-20: Global Precision POWD Graphic Results – Mean OWD

Figure 3-21: Global Precision POWD Graphic Results – Packet Loss on the probes

3.3.1 Conclusions

The first conclusion is that the limit rate on which the probes experiment packet loss for measuring the OWD seems to be lower that the one calculated in previous sections for traffic capture measures. This point can be checked on figure 3-21 where the picture shows the packet loss measured on each probe. It can be shown that the packet loss for Probe1 (which is the less powerful) is around 7% for all the measurements. However, the packet loss do not influence on the POWD precision as shown bellow.

On the other hand, the OWD results show that the system has different accuracy depending on which parameter is analyzed. Regarding the results about Minimum OWD, it can be obtained big difference between the values measured by the SmartBits and the OpenIMP systems. The maximum error in minimum OWD obtained is around 387 µs, which means about 78 %. It is really a very high error that has likely as main reason the wrong timestamp as shown in the previous section.

However, it can be appreciated a very low difference on the mean OWD measured and calculated by both systems. It seems that the method used by the OpenIMP system is enough accurate because of the similar OWD measured by both the system and the calibrated reference (SmartBits). The worst measured error is 0.28 % (376 µs of error for a real POWD of 136,07 ms) which represents an excellent result, and the average measured error is even better, 0.08 %. On the other hand, it seems to have some correlation between the mean OWD error and the packet loss produced on the probes.

Similar results can be enounced for errors on the maximum OWD measured on the system.

The only main constraint here is the synchronization between probes, the better synchronization, the better results. However with a common NTP synchronization, it is shown that good results can be obtained, at least to measure traffic going through networks with more than one router where the processing time is higher than microseconds.

As final conclusion, it can be stated that the precision of the delay-related values measured by the OpenIMP system are extraordinarily good for maximum and mean POWD. For these parameters, with a synchronization between probes better than 465 µs, the worse mean POWD precision obtained was 376 µs (0,28 % of error) and the worse maximum POWD precision was 377,3 µs (0,12 % of error). However the precision of minimum POWD is worse: 387,1 µs (77,9 % of error).

3.4 Influence of frame size on the POWD precision

The tests performed on the previous section shown that the precision of the OpenIMP is very good when the synchronization of the probes is good. All those tests where performed with an IPv6-UPD flow where the size of the used packets were 84 B. One could think that the packet size might influence on the precision of the system, and this is the target of the following test.

To find out if there is influence of the packet size on the precision of the probes, measurements with different frame sizes have been performed with two different networks loads, according to figure 2-2:

1) Network Load 40%
2) Network Load 100%

The index of each network load is used on the figures of this section.

The network topology shown in figure 3-3 was implemented and the platforms used for each probe were:

Probe 1

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

The obtained results are presented bellow[2].

			SmartBits Info						
Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Delivered Packets**	**Received Packets**	**Packets Loss**	**Packet Loss (%)**	**OWD min (us)**	**OWD max (ms)**	**OWD mean (ms)**
40	84	48.077	2.884.615	721.674	2.162.941	74,98%	-------	43,18	43,09
40	300	15.625	937.500	87.975	849.525	90,62%	-------	132,78	131,95
40	600	8.065	483.870	44.349	439.521	90,83%	-------	256,98	254,40
40	900	5.435	326.086	82.038	244.048	74,84%	-------	381,30	378,87
40	1.200	4.098	245.901	61.737	184.164	74,89%	-------	256,68	255,01
40	1.500	3.289	197.368	49.604	147.764	74,87%	-------	319,74	317,38
100	84	120.192	7.211.538	721.693	6.489.845	89,99%	-------	44,70	43,86
100	300	39.063	2.343.750	234.898	2.108.852	89,98%	-------	134,12	132,96
100	600	20.161	1.209.677	121.488	1.088.189	89,96%	-------	258,48	256,53
100	900	13.587	815.217	82.041	733.176	89,94%	-------	382,72	379,66
100	1.200	10.246	614.754	61.739	553.015	89,96%	-------	258,14	255,80
100	1.500	8.224	493.421	49.605	443.816	89,95%	-------	321,19	318,21

Figure 3-22: Influence of Frame Size on POWD – SmartBits Information

			OpenIMP System Measurements							
Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Total Received Packets (Probe1)**	**Packet Loss (%) (Probe1)**	**Total Received Packets (Probe2)**	**Packet Loss (%) (Probe2)**	**Packet Loss (%) (Probe1-Probe2)**	**OWD min (us)**	**OWD max (ms)**	**OWD mean (ms)**
40	84	48.077	2.883.277	0,05%	713.628	1,11%	75,25%	0,17	43,60	43,13
40	300	15.625	937.411	0,01%	87.975	0,00%	90,62%	0,36	133,62	131,86

---

[2] It was not possible to extract the minimum OWD values measured by the SMB-600 in this test, so such information is not provided.

40	600	8.065	483.831	0,01%	44.349	0,00%	90,83%	0,65	256,95	254,36
40	900	5.435	326.086	0,00%	82.038	0,00%	74,84%	0,84	381,19	378,75
40	1.200	4.098	245.901	0,00%	61.737	0,00%	74,89%	1,06	256,56	254,88
40	1.500	3.289	197.344	0,01%	49.604	0,00%	74,86%	1,30	319,64	317,27
100	84	120.192	3.014.486	58,20%	713.405	1,15%	63,35%	0,17	44,14	43,05
100	300	39.063	2.335.875	0,34%	234.730	0,07%	89,95%	0,38	133,09	132,42
100	600	20.161	1.209.388	0,02%	121.488	0,00%	89,95%	0,62	257,02	255,66
100	900	13.587	813.480	0,21%	82.041	0,00%	89,91%	0,83	381,41	378,88
100	1.200	10.246	613.004	0,28%	61.739	0,00%	89,93%	1,36	256,88	255,15
100	1.500	8.224	490.493	0,59%	49.599	0,01%	89,89%	1,18	319,56	317,22

Figure 3-23: Influence of Frame Size on POWD – OpenIMP System Measurements

Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	CPU Load Probe1 (%)	CPU Load Probe2 (%)	Time offset (Probe1) (ms)	Time offset (Probe2) (ms)	ABS. Relative offset Probe1-Probe2 (us)
40	84	48.077	58,76	72,18	0,091	0,063	28,0
40	300	15.625	18,73	22,38	0,196	0,064	132,0
40	600	8.065	9,63	11,7	0,212	0,094	118,0
40	900	5.435	6,37	7,81	0,188	0,075	113,0
40	1.200	4.098	5	6,16	0,136	0,083	53,0
40	1.500	3.289	3,7	5,04	0,102	0,085	17,0
100	84	120.192	99,81	72,37	0,264	0,073	191,00
100	300	39.063	46,38	22,79	0,054	0,084	29,67
100	600	20.161	24,53	11,79	0,284	0,156	128,50
100	900	13.587	16,02	7,72	0,165	0,172	6,50
100	1.200	10.246	11,98	6,3	0,235	0,128	107,50
100	1.500	8.224	9,72	7,42	-0,018	0,043	61,00

Figure 3-24: Influence of Frame Size on POWD – Probes Synchronization

Bandwidth			Results							
Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	ABS. OWD min Error (us)	ABS. OWD min Error (%)	ABS. OWD max Error (us)	ABS. OWD max Error (%)	ABS. OWD mean Error (us)	ABS. OWD mean Error (%)	Packet Loss Error	Packet Loss Error (%)
40	84	48.077	-------	-------	425,00	0,98%	44,75	0,10%	4.332.590,00	0,33%
40	300	15.625	-------	-------	831,00	0,63%	82,94	0,06%	1.698.961,00	0,01%
40	600	8.065	-------	-------	32,00	0,01%	39,57	0,02%	879.003,00	0,01%
40	900	5.435	-------	-------	110,00	0,03%	117,14	0,03%	488.096,00	0,00%
40	1.200	4.098	-------	-------	122,00	0,05%	128,01	0,05%	368.328,00	0,00%
40	1.500	3.289	-------	-------	101,00	0,03%	110,86	0,03%	295.504,00	0,01%
100	84	120.192	-------	-------	564,15	1,26%	802,18	1,83%	8.790.925,50	58,31%
100	300	39.063	-------	-------	1.033,30	0,77%	533,12	0,40%	4.209.996,00	0,34%
100	600	20.161	-------	-------	1.463,75	0,57%	863,07	0,34%	2.176.088,50	0,02%
100	900	13.587	-------	-------	1.313,55	0,34%	779,84	0,21%	1.464.615,00	0,21%
100	1.200	10.246	-------	-------	1.262,25	0,49%	651,19	0,25%	1.104.279,50	0,28%
100	1.500	8.224	-------	-------	1.621,90	0,50%	981,87	0,31%	884.709,50	0,59%

Figure 3-25: Influence of Frame Size on POWD – Numeric Results

The most relevant parameters have been extracted from the previous tables and they are shown in the following pictures, which help to extract the conclusions on the following section.

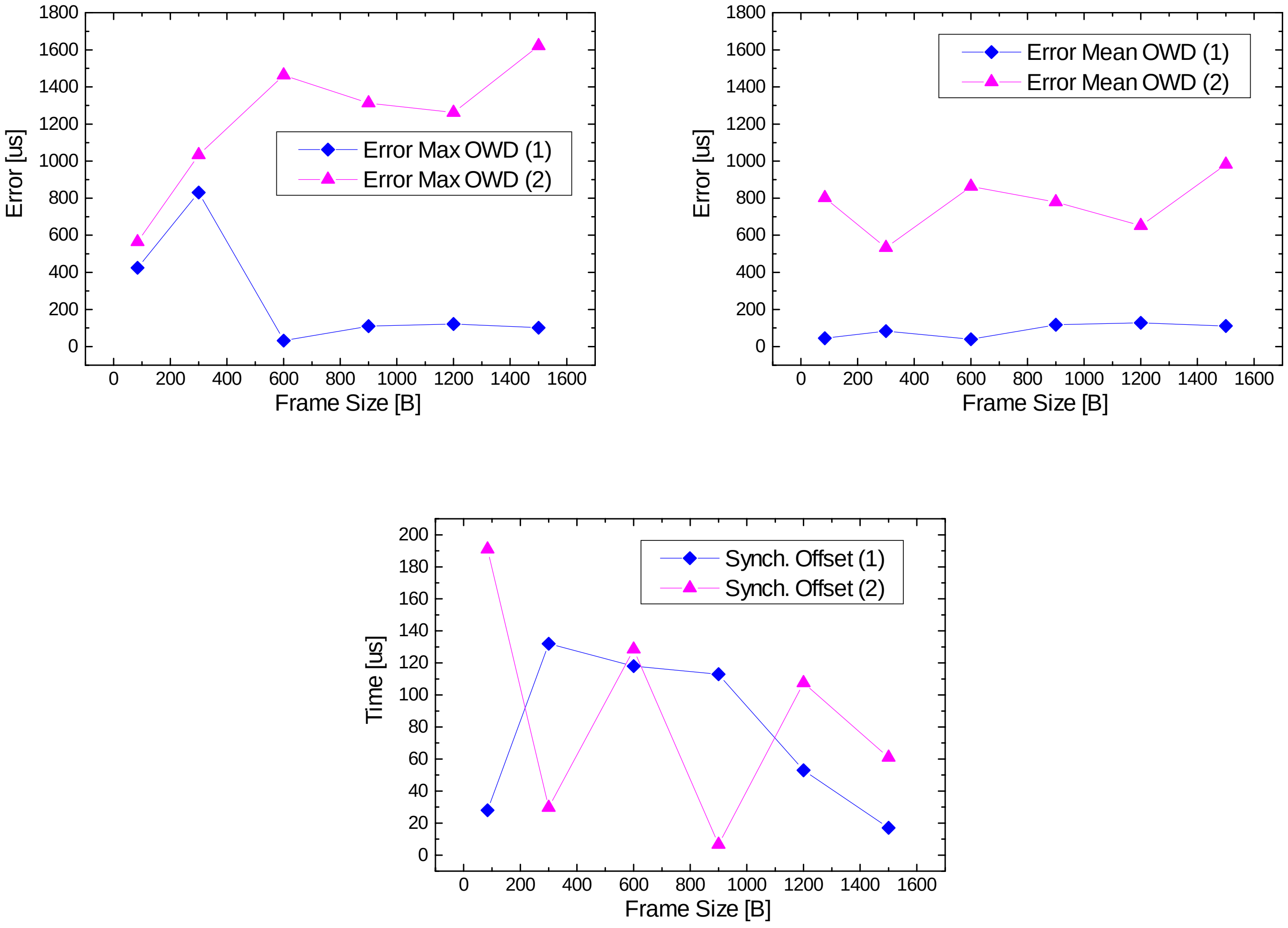


Figure 3-26: Influence of Frame Size on POWD – OWD Error

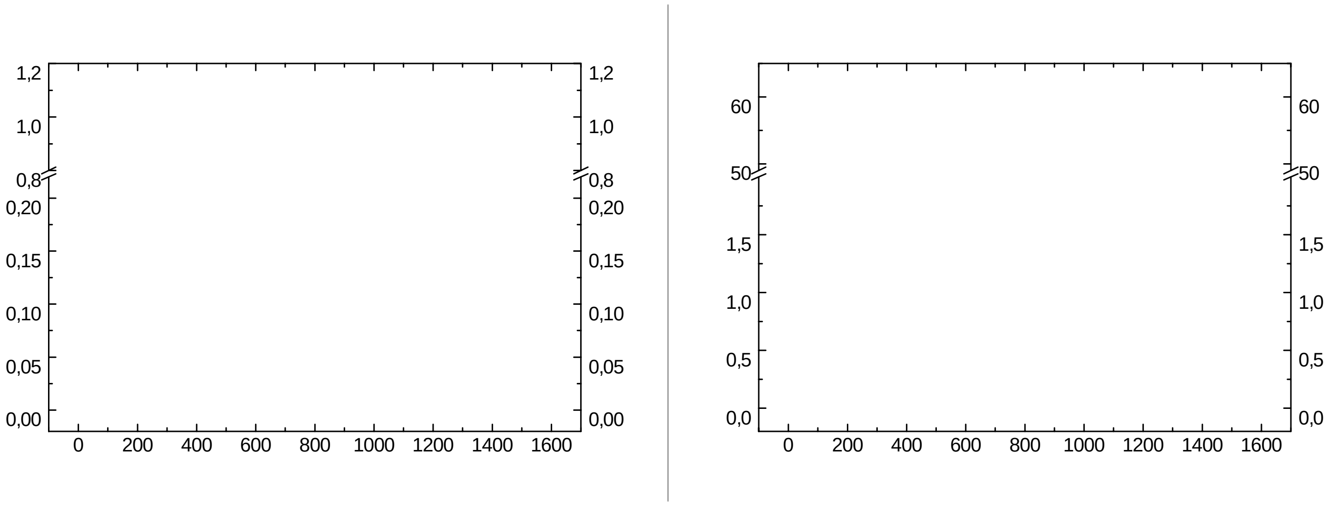


Figure 3-27: Influence of Frame Size on the Global Precision POWD – Packet Loss

3.4.1 Conclusions

By seeing the figure 3-26, the first conclusion that can be extracted is that the network load influences on the precision of the maximum and mean POWD measured by the probes. The higher the network load the poorer the precision. Given the fact that the synchronization between probes had similar values for both set of tests, the reason for the influence of the network load on the precision could be the packet loss produced on the probes, as shown in figure 3-27.

It can be also stated that the frame size influence on the precision: the higher the frame size, the better precision. In figure 3-26 it can be shown that the values for maximum and mean POWD decrease and they tends to be low.

In spite of the influence of the network load, the precision obtained in this test is better than:

- 982 µs for mean POWD
- 1.622 µs for maximum POWD.

3.5 Influence of number of measurement tasks on the POWD precision

The target of this test is to find out if the number of measurements tasks running at a time on the probes can influence on the precision of the POWD calculated by the OpenIMP system. This assumption is logical in principle because many measurements tasks running on the probe means more CPU load, so the precision of the measured values could be affected.

To evaluate such influence the network topology shown in figure 3-3 has been implemented and the following tasks have been configured on each probe:

1) Volume Measurement. This is the measurement usually configured to capture packets.
2) Passive One Way Delay Measurement. The measurement to know the one way delay of the passive traffic sent by the SMB-600.
3) Active One Way Delay. An UDPv6 flow was configured with packets of 300 bytes (payload) and a rate of 1.000 packets per second. The total bandwidth of the flow on the LAN was about 3 Mbps.
4) Passive One Way Delay Measurement. The measurement to know the one way delay of the active traffic sent by probes.
5) Traffic Measurement. The measurement to know the total bandwidth on the network.

The platforms used for each probe were:

Probe 1

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Finally, the results obtained for this test are presented bellow[3].

			SmartBits Info						
Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Delivered Packets**	**Received Packets**	**Packets Loss**	**Packet Loss (%)**	**OWD min (us)**	**OWD max (ms)**	**OWD mean (ms)**
20	84	24.038	1.442.307	438.232	1.004.075	69,62%	-------	99,59	62,12
20	300	7.813	468.750	165.364	303.386	64,72%	-------	169,91	152,60
20	600	4.032	241.935	102.721	139.214	57,54%	-------	257,00	255,33
20	900	2.717	163.043	71.501	91.542	56,15%	-------	381,19	357,83
20	1.200	2.049	122.950	53.043	69.907	56,86%	-------	256,73	254,39
20	1.500	1.645	98.684	43.209	55.475	56,21%	-------	319,30	307,46

Figure 3-28: Influence of Number of Measurements on POWD – SmartBits Information

			OpenIMP System Measurements							
Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Total Received Packets (Probe1)**	**Packet Loss (%) (Probe1)**	**Total Received Packets (Probe2)**	**Packet Loss (%) (Probe2)**	**Packet Loss (%) (Probe1-Probe2)**	**OWD min (us)**	**OWD max (ms)**	**OWD mean (ms)**
20	84	24.038	1.442.093	0,01%	435.985	0,51%	69,77%	1,1	99,56	62,19
20	300	7.813	468.744	0,00%	165.084	0,17%	64,78%	3,5	169,92	152,59
20	600	4.032	238.754	1,31%	102.736	0,01%	56,97%	0,79	257,09	255,22
20	900	2.717	161.707	0,82%	71.387	0,16%	55,85%	0,8	381,12	357,77
20	1.200	2.049	122.950	0,00%	53.036	0,01%	56,86%	1,1	257,14	254,21
20	1.500	1.645	98.684	0,00%	43.218	0,02%	56,21%	7,14	319,17	307,33

Figure 3-29: Influence of Number of Measurements on POWD – OpenIMP Measurements

Network Load (%)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**CPU Load Probe1 (%)**	**CPU Load Probe2 (%)**	**Time offset (Probe1) (ms)**	**Time offset (Probe2) (ms)**	**ABS. Relative offset Probe1-Probe2 (us)**
20	84	24.038	29,13	53,17	0,168	0,066	102,00
20	300	7.813	10,68	25,52	0,095	0,053	42,00
20	600	4.032	7,83	16,14	0,13	0,073	57,00
20	900	2.717	5,49	11,49	0,061	-0,006	67,00
20	1.200	2.049	3,38	10,25	0,149	0,025	124,00
20	1.500	1.645	2,32	8,81	0,105	0,053	52,00

---

[3] *It was not possible to extract the minimum OWD values measured by the SMB-600 in this test, so such information is not provided.*

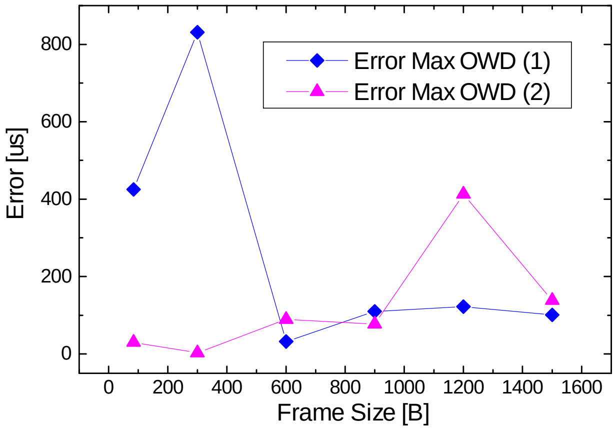
Figure 3-30: **Influence of Number of Measurements on POWD – Probes Synchronization**

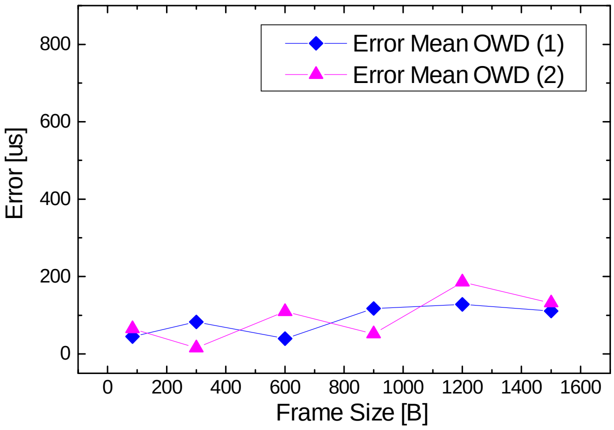
Bandwidth			Results							
Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	ABS. OWD min Error (us)	ABS. OWD min Error (%)	ABS. OWD max Error (us)	ABS. OWD max Error (%)	ABS. OWD mean Error (us)	ABS. OWD mean Error (%)	Packet Loss Error	Packet Loss Error (%)
20	84	24.038	-------	-------	30,00	0,03%	65,20	0,10%	2.010.183	0,17%
20	300	7.813	-------	-------	3,00	0,00%	15,80	0,01%	607.046	0,06%
20	600	4.032	-------	-------	89,00	0,03%	109,50	0,04%	275.232	1,31%
20	900	2.717	-------	-------	77,00	0,02%	51,90	0,01%	181.862	0,89%
20	1.200	2.049	-------	-------	413,00	0,16%	185,50	0,07%	139.821	0,01%
20	1.500	1.645	-------	-------	138,00	0,04%	132,00	0,04%	110.941	0,01%

Figure 3-31: **Influence of Number of Measurements on POWD – Numeric Results**

In order to extract some conclusions the most significant parameters of the previous results are shown in the following pictures, which includes maximum and mean POWD error and packet loss rates on each probe.

Indeed there are two set of results represented on each picture. The results labeled as 1) correspond to the POWD measurements performed on previous section where only one measurement task was running on the probes. On the other hand, the results labeled as 2) represent the results of the POWD measurements when 4 measurement tasks were running at a time on each probe. In this way some comparison between running one or several tasks can be gotten.





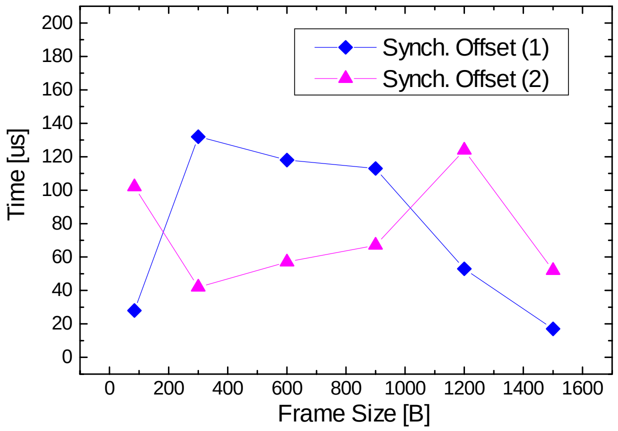


Figure 3-32: Influence of the Number of Tasks on POWD – OWD Error

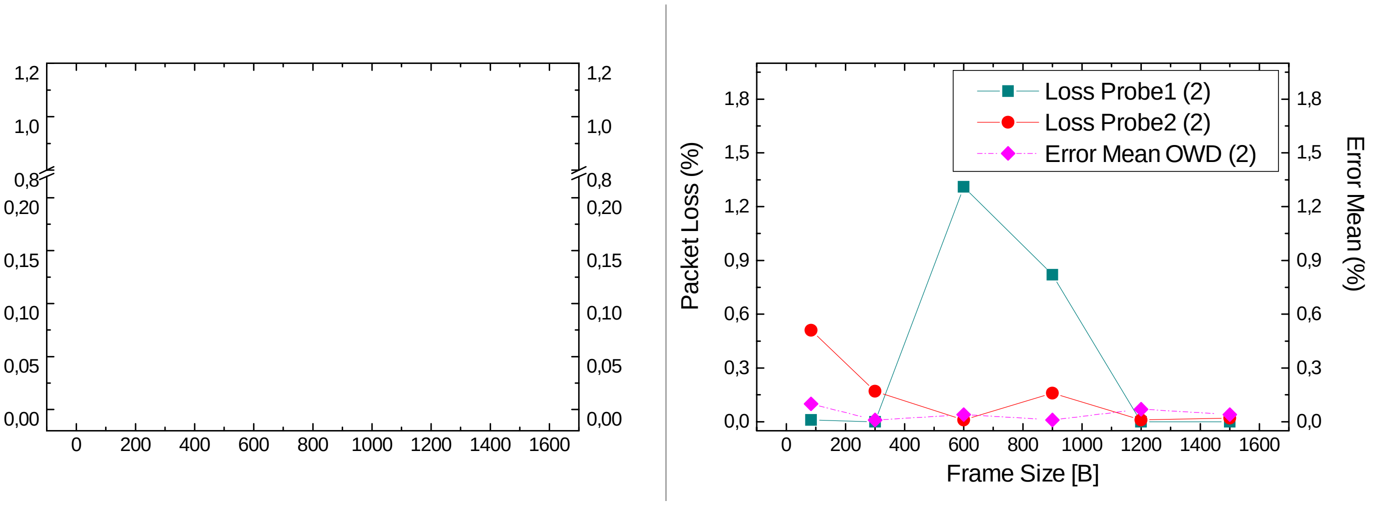


Figure 3-33: Influence of the Number of tasks on POWD – Packet Loss

3.5.1 Conclusions

Figure 3-32 shows that there is no significant difference on the mean POWD error gotten while running one or several measurement tasks. Both sets of results present similar values for all the frame sizes. On the other hand, the synchronization between both probes seems not to have a big influence on the mean POWD error because the synchronization difference between both sets of results were better than 100 µs for all the frame sizes, so it can be assumed that such parameter did not influence on the mean POWD values obtained.

Regarding the maximum POWD error it seems that both set of results present similar values for frame sizes bigger than 600 Bytes whereas for lower packets it seems that the probes running more than one measurement task has better behavior. The explanation for this can be found on the figure 3-33 where we can see that there is a slightly greater packet loss for low frame sizes when only one measurement task were running.

Hence we can conclude that no significant influence on the POWD precision can be found on the number of measurement tasks.

3.6 Influence of the measurement duration on the POWD precision

Finally this test tries to evaluate if the measurement duration can influence on the POWD precision. To do that, the network topology shown in figure 3-3 was implemented, having each probe the following platforms:

Probe 1

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

All the tests had the same configuration in terms of network load and frame size, being the only difference the measurement duration which had the following values:

- 60 seconds
- 180 seconds
- 300 seconds
- 600 seconds

The obtained results are presented in the following tables.

			SmartBits Info						
Network Load (%)	Test Duration (s)	Traffic Rate (PPS)	Delivered Packets	Received Packets	Packets Loss	Packet Loss (%)	OWD min (us)	OWD max (ms)	OWD mean (ms)
40	60	15.625	937.500	87.975	849.525	90,62%	-------	132,78	131,95
40	180	15.625	2.812.500	703.645	2.108.855	74,98%	-------	132,72	132,42
40	300	15.625	4.687.500	1.172.397	3.515.103	74,99%	-------	132,72	132,45
40	600	15.625	9.375.000	2.344.277	7.030.723	74,99%	-------	132,72	132,47

Figure 3-34: Influence of Measurement Duration on POWD – SmartBits Information

			OpenIMP System Measurements							
Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	Total Received Packets (Probe1)	Packet Loss (%) (Probe1)	Total Received Packets (Probe2)	Packet Loss (%) (Probe2)	Packet Loss (%) (Probe1-Probe2)	OWD min (us)	OWD max (ms)	OWD mean (ms)

40	60	15.625	937.411	0,01%	87.975	0,00%	90,62%	0,36	133,62	131,86
40	180	15.625	2.812.385	0,00%	703.590	0,01%	74,98%	0,3	133,01	132,37
40	300	15.625	4.684.822	0,06%	1.172.012	0,03%	74,98%	0,3	132,99	132,51
40	600	15.625	9.366.775	0,09%	2.343.135	0,05%	74,98%	0,3	133,08	132,38

Figure 3-35: Influence of Measurement Duration on POWD – OpenIMP Measurements

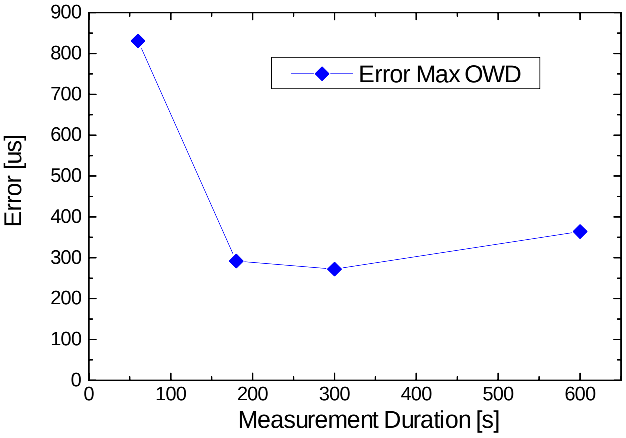
Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	CPU Load Probe1 (%)	CPU Load Probe2 (%)	Time offset (Probe1) (ms)	Time offset (Probe2) (ms)	ABS. Relative offset Probe1-Probe2 (us)
40	60	15.625	18,73	22,38	0,196	0,064	132,0
40	180	15.625	18,25	22,66	0,102	0,011	91
40	300	15.625	18,43	22,73	-0,041	-0,038	3
40	600	15.625	18,55	22,78	0,035	-0,074	109

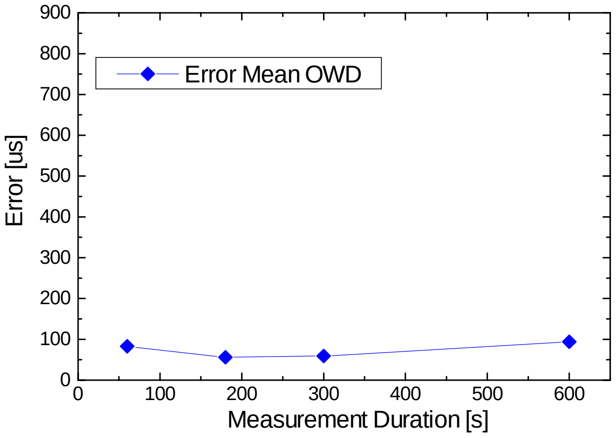
Figure 3-36: Influence of Measurement Duration on POWD – Probes Synchronization

Bandwidth			Results							
Network Load (%)	Frame Size (Bytes)	Traffic Rate (PPS)	ABS. OWD min Error (us)	ABS. OWD min Error (%)	ABS. OWD max Error (us)	ABS. OWD max Error (%)	ABS. OWD mean Error (us)	ABS. OWD mean Error (%)	Packet Loss Error	Packet Loss Error (%)
40	60	15.625	-------	-------	831,00	0,63%	82,94	0,06%	1.698.961,00	0,01%
40	180	15.625	-------	-------	292,0	0,22%	56,10	0,04%	4.217.650	0,01%
40	300	15.625	-------	-------	272,0	0,20%	58,90	0,04%	7.027.913	0,07%
40	600	15.625	-------	-------	364,0	0,27%	94,10	0,07%	14.054.363	0,10%

Figure 3-37: Influence of Measurement Duration on POWD – Numeric Results

As done in the previous sections, the more relevant parameters of the obtained results are presented bellow, which include maximum and mean POWD errors and packet loss on each probe.





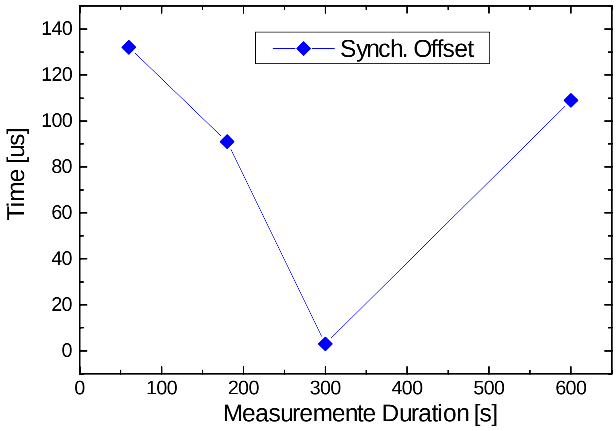


Figure 3-38: Influence of the Measurement Duration on POWD – OWD Error

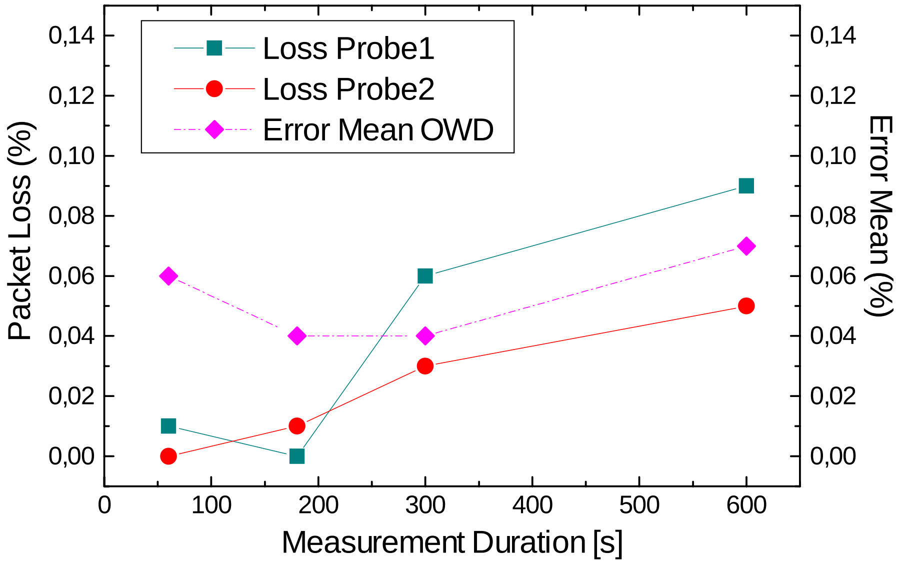


Figure 3-39: Influence of the Measurement Duration on POWD – Packet Loss

3.6.1 Conclusions

After visualizing the figure 3-38 it can be stated that the influence of the measurement duration on the maximum POWD error is very low. The difference between the biggest and the lowest maximum POWD error is lower than 559 µs, being the best results gotten for big measurement durations.

On the other hand, the influence of the duration on the mean POWD error is even lower, and the obtained values have a nearly flat response. The difference between the biggest and lowest mean POWD error is lower than 27 µs.

As conclusion, it seems that there is no significant influence of the measurement duration on the POWD precision.

3.7 Global precision of measured AOWD

This test tries to evaluate the precision of the active one-way-delay AOWD calculated by the OpenIMP system. As mentioned in section 3.1.3, both probes of figure 3-4 were configured with two measurement tasks. One of them was the active OWD measurement which sent an active IPv6-UDP flow with packets of 600 Bytes (layer II, including CRC) at 4.167 packets per second, which represents a bandwidth of 20 Mbps.

The second measurement task configured on both probes was the passive OWD measurement which was configured to capture only packets belonging to the active flow, so the OWD measured on both measurement tasks refers to the same packets.

The methodology to calculate the OWD is different in each type of measurement task, so it has sense to evaluate the difference on the results obtained in each measurement. Furthermore, as shown in previous sections, the precision of the passive measurement is very good, so the values gotten with this type of measurement was taken as reference to evaluate the results obtained with the active measurements.

The platforms used on the probes for these tests have been the following:

Probe 1

- CPU: Intel® Pentium® 4, 2.80 GHz
- CACHE: 512 KB
- Memory: 512 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8smp (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Probe 2

- CPU: AMD-K6(tm) 3D processor, 450 MHz
- CACHE: 64 KB
- Memory: 64 MB
- NIC: 10/100 Mbps
- System: Linux version 2.4.20-8 (gcc version 3.2.2 20030222 (Red Hat Linux 3.2.2-5))

Seven measurements have been performed and the results are presented in the following tables.

			Active Probes Measurements						
Bandwidh (Mbps)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Delivered Packets**	**Received Packets**	**Packets Loss**	**Packet Loss (%)**	**OWD min (ms)**	**OWD max (ms)**	**OWD mean (ms)**
20	600	4.167	250.020	53.622	196.398	78,55%	-112,697	110,83	24,00
20	600	4.167	250.020	45.906	204.114	81,64%	-569,121	179,46	124,00
20	600	4.167	250.020	26.989	223.031	89,21%	-185,797	265,63	144,00
20	600	4.167	250.020	35.147	214.873	85,94%	-259,381	267,66	137,00
20	600	4.167	250.020	23.369	226.651	90,65%	-28,165	367,27	232,00
20	600	4.167	250.020	28.948	221.072	88,42%	-102,471	265,86	148,00
20	600	4.167	250.020	23.910	226.110	90,44%	-12,683	319,11	195,00

Figure 3-40: Precision of AOWD – Active Probes Measurements

			Passive Probes Measurements								
Bandwidh (Mbps)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**Total Received Packets (Probe1)**	**Packet Loss (%) (Probe1)**	**Total Received Packets (Probe2)**	**Packet Loss (%) (Probe2)**	**Packet Loss (%) (Probe1-Probe2)**	**OWD min (ms)**	**OWD max (ms)**	**OWD mean (ms)**
20	600	4.167	249.909	0,04%	53.692	-0,13%	78,52%	0,616	70,38	23,54
20	600	4.167	250.007	0,01%	46.168	-0,57%	81,53%	0,665	125,81	64,66
20	600	4.167	249.952	0,03%	27.137	-0,55%	89,14%	0,482	143,16	124,85
20	600	4.167	249.774	0,10%	35.226	-0,22%	86,23%	0,577	134,50	125,53
20	600	4.167	247.943	0,83%	24.221	-3,65%	90,23%	0,615	222,38	168,63
20	600	4.167	249.996	0,01%	29.272	-1,12%	88,29%	0,542	145,98	123,66
20	600	4.167	245.835	1,67%	23.959	-0,20%	90,25%	0,611	193,41	145,54

Figure 3-41: Precision of AOWD – Passive Probes Measurements

Bandwidh (Mbps)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**CPU Load Probe1 (%)**	**CPU Load Probe2 (%)**	**Time offset (Probe1) (ms)**	**Time offset (Probe2) (ms)**	**ABS. Relative offset Probe1-Probe2 (us)**
20	600	4.167	29,13	53,17	0,168	0,066	102
20	600	4.167	10,68	25,52	0,095	0,053	42
20	600	4.167	------	------	0,267	0,131	136
20	600	4.167	7,83	16,14	0,130	0,073	57
20	600	4.167	5,49	11,49	0,061	-0,006	67
20	600	4.167	3,38	10,25	0,149	0,025	124
20	600	4.167	2,32	8,81	0,105	0,053	52

Figure 3-42: Precision of AOWD – Probes Synchronization

Bandwidth			Results							
Bandwidh (Mbps)	**Frame Size (Bytes)**	**Traffic Rate (PPS)**	**ABS. OWD min Error (ms)**	**ABS. OWD min Error (%)**	**ABS. OWD max Error (ms)**	**ABS. OWD max Error (%)**	**ABS. OWD mean Error (ms)**	**ABS. OWD mean Error (%)**	**Packet Loss Error**	**Packet Loss Error (%)**
20	600	4.167	113,313	18394,97%	11,44	11,51%	46,38	65,90%	392.615	0,02%
20	600	4.167	569,786	85682,11%	9,14	5,36%	1,81	1,44%	407.953	0,10%
20	600	4.167	186,279	38647,10%	8,89	3,46%	0,84	0,59%	445.846	0,03%
20	600	4.167	259,958	45053,38%	10,77	4,19%	2,50	1,86%	435.421	2,33%
20	600	4.167	28,78	4679,67%	9,19	2,57%	9,62	4,32%	450.373	0,49%
20	600	4.167	103,013	19006,09%	9,39	3,66%	2,02	1,39%	441.796	0,12%
20	600	4.167	13,294	2175,78%	0,81	0,25%	1,59	0,82%	447.986	1,65%

Figure 3-43: Precision of AOWD – Numeric Results

By following the same criteria that in previous sections, only the relevant parameters of the previous results haven been represented in figures. Specifically, they are:

- Minimun OWD measured on both active and passive measurements
- Error on the minimum OWD measured, taking the minimum POWD as reference
- Maximun OWD measured on both active and passive measurements
- Error on the maximum OWD measured, taking the maximum POWD as reference
- Mean OWD measured on both active and passive measurements
- Error on the mean OWD measured, taking the mean POWD as reference
- Difference on the packet loss measured on the probe2 which received the flow’s packets.

And the figures are presented bellow.

Figure 3-44: Global Precision AOWD Graphic Results – Minimum OWD

Figure 3-45: Precision on AOWD Graphic Results – Maximum OWD

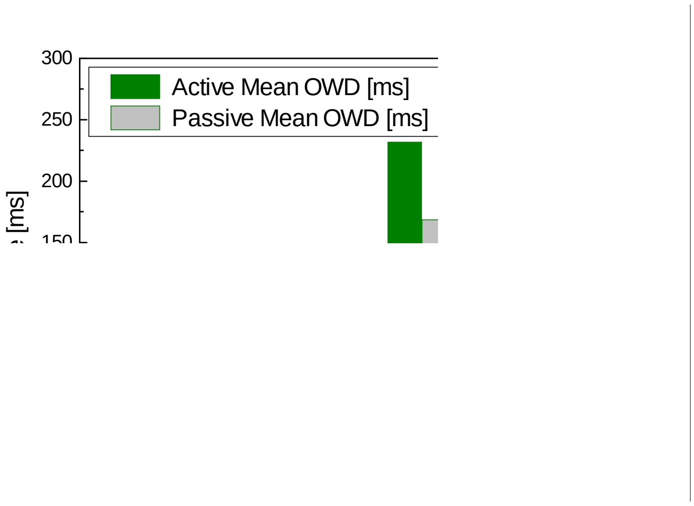


Figure 3-46: **Precision on AOWD Graphic Results – Mean OWD**

Figure 3-47: **Precision on AOWD Graphic Results – Packet Loss and mean OWD error**

3.7.1 Conclusions

The first conclusion is that the minimum OWD measured by the active measurement is very poor, not only because all the measured values are negative (which is not possible), but also because the absolute error between the values measured by the active and passive measurements are enormous, higher than 70 ms (10.000%) most cases, as shown in figure 3-44

Regarding the maximum OWD measured by the active measurement, the results shown on figure 3-45 are not good neither because most of the results shows error greater than 9 ms.

In regard with the mean OWD measured by the active measurement, it can be stated than in general they have lower error than the previous parameters since the error average (without considering the result of test1 of figure 3-46) is around 3 ms. However it can appear spurious errors like the one obtained for the test1 which was 46,38 ms. These spurious decrease the reliability of the mean AOWD measured by the active measurement.

Finally it can be also seen that the active measurement has poorer performance than the passive one. This can be shown by observing in table 3-41 the number of packets captured by the passive measurement on the probe2 (the one that received the flow's packets). In all the tests they are greater than the packets captured by the active measurement on the probe2, which results on negative packet loss for the passive measurement on the probe2. One could think that this difference on the captured packets in both active and passive measurement is the reason for the poor precision of the POWD. However, in figure 3-47 it can be seen that there is no correlation between the difference on the captured packets and the error on the mean OWD.

Neither the synchronization of both probes can be argued as the reason for that poor precision because the same synchronization (good or bad) affects to both active and passive measurements on the same way because both of them are run on the same hosts. So it seems that the real reason for the poorer precision of the AOWD is the way how the delay is calculated, which is very different on passive and active measurements.

4. SUMMARY AND CONCLUSIONS

The main target in this deliverable tries to evaluate the accuracy on the most important parameters of the OpenIMP system, mainly the packet loss and precision of the measured one-way-delay. The evaluation of only those two parameters is enough because all the calculations made by the OpenIMP system are based on them, so no extra tests are required. The tests have been very complete not only to know the reliability of such parameter but also, and maybe the most important, how different aspects like type of CPU, RAM amount on the system, number of measurement tasks running on the probes, etc. can influence on precision of the obtained results.

As secondary objective, it was stated interesting to find out the limit traffic rate that the system can support with reliable information. Particularly, if the OC3 interface could be supported by the probes as it was promised on the Technical Annex of this project.

To do that several tests have been performed, as explained on the previous sections and the following conclusions have been extracted from them.

The performance of the probes[4] is strongly bound to the performance of the architecture used. The power CPU, the lower packet loss produced on the probes meanwhile capturing packets. The work of the measurement task stresses the CPU depending of the traffic rate. It has been found that the evolution of the CPU load on the probes is exponential with regard the traffic rate. It has been shown also that there is a saturation point on the CPU load that packet loss starts to appear. Such saturation point depends on the type of CPU and it ranges from 40% (AMD K6 450 MHz and PIII 700 MHz) of CPU load for less powerful CPUs upto 80% of CPU for the most powerful CPU (PIV 2.8 GHz).

The traffic rate that leads to reach the saturation point on the CPU depends also on the type of CPU. Such traffic rate have been evaluated around 22.000 packets per second (pps) for AMD K6 450 MHz, 32.000 pps for PIII 700 MHz and 70.000 pps for PIV 2.8 GHz. The most adequate way to express the limit rate on the probes is in packets per second since each packet is treated individually on the probes, independently of its size. However traffic rate in terms of bit per second is commonest and those values were translated on section 2 to Mbps in order to know the architecture that supports the OC3 interface. The obtained results were the following:

	Limit Traffic Rate for average Internet packets (300 B)	**Maximum Limit Traffic Rate (1.500 B)**
AMD K6 450 MHz	52,8 Mbps	264 Mbps
PIII 700 MHz	76,8 Mbps	384 Mbps
PIV 2.8 MHz	168 Mbps	840 Mbps

Figure 4-1: Limit Traffic Rate for different architectures

---

[4] *In this deliverable the meaning of "performance of the probe" refers to the capacity of the volume measurement to capture packets without loss.*

Taking into account that OC3 interface supports 156 Mbps, it can be stated that the recommend architecture to works with guarantee on such environment would be the PIV 2.8 MHz. The other ones could also work, but it might produce packet loss that would be counted erroneously by the OpenIMP system if the size of packets were low/medium.

During the tests performed to write this deliverable it has been also shown that there is no influence on such limit traffic rates of different factors like the RAM amount installed on the probe, the gap between consecutive packets (Tbp according to figure 2-2) and the type of NIC or the complexity of the filtering rule configured to captures only the packets that match the rule.

The number of measurement tasks running on the probe at the same time might affect by decreasing the limit traffic rate because the CPU load increases according to the number and type of measurement tasks. This could force the CPU to reach the saturation point with lower traffic rates and starts to appear packet loss. However, the experiments done on the laboratory for PIV 2.8 GHz show that the influence of the number of measurement tasks is negligibly for frame size greater than 300 B which is the most usual on practice. However, for other CPUs the limit traffic rate might be decreased even with greater frame size, although it is very difficult to evaluate because it depends on the number and type of measurement tasks.

The packet size is the last factor evaluated on the probe performance and the obtained results show that the worse behavior of the probe is for low sizes.

The precision on the passive delay-related parameter has been also evaluated and the obtained results show that the timestamp offset on the captured packets (time between the packet is received and the timestamp is calculated and put on each packet) is absolutely negligibly, around tens of nanoseconds, if the limit traffic rate is not overcome. The timestamp jitter (variation of the timestamp offset) is also very low, around tens of microseconds when the traffic rate is bellow the limit.

The good behavior for the timestamp influence on the precision of the passive one-way-delay (POWD) calculated by the system. It has been found that such precision is very good when the limit traffic rate is not overcome and the synchronization between probes is good enough. In this conditions (synchronization between probes was better than 465 µs) the precision on mean POWD and maximum POWD were 376 µs and 377 µs respectively, being the real values of mean and maximum POWD 136 ms and 327 ms respectively. The minimum POWD parameter has poorer precision as show the results gotten on the laboratory tests.

It seems to be a slight correlation between the packet loss produced on the probes and the precision of the POWD. On the other hand, given the fact that the passive OWD measurement stressed the CPU more than the capture measurement, the saturation point on the CPU is reached with lower traffic rate, so the limit traffic rate for POWD measurement is slightly lower. However this fact does not influence seriously on the precision of the POWD because the precision on the mean and maximum POWD obtained for traffic rates greater than the limit (probe synchronization better than 200 µs) were 982 µs and 1.622 µs, being the real mean and maximum POWD 313 ms and 321 ms respectively, which show that they are good values enough.

Furthermore, it has been found that different factors like the number of measurement tasks running on the probes and the measurement duration do not influence on the precision of the calculated POWD.

On the other hand, the precision of the AOWD has been also evaluated, being the poorest result obtained. It has been found that negative values for minimum AOWD can appear and the precision of the maximum and mean AOWD is around 9 ms and 3 ms respectively, which represent lower precision than POWD.

Finally, it can be concluded that in general terms the behavior of the OpenIMP system is good enough and have the expected precision.

5. REFERENCES

[1] Spirent Communications IPv6 Tests Solutions. http://www.spirentcom.com/ipv6/.

[2] K. Thompson, G. Miller, and R. Wilder. "Wide Area Internet Traffic Patterns and Characteristics" IEEE Network, Vol. 11 No. 6, pp. 10-23, Nov/Dec 1997.

[3] Fomenkov, M., Keys, K., Moore, D. and Claffy, K., "Longitudinal study of Internet traffic in 1998-2003" http://www.caida.org/outreach/papers/2003/nlanr/

[4] S. McCreary, kc claffy "Trends in Wide Area IP Traffic Patterns" http://www.caida.org/outreach/papers/2000/AIX0005/AIX0005.pdf

[5] M. Li, M. Claypool, R. Kinicki and J. Nichols "Characteristics of Streaming Media Stored on the Web" Technical Report TR-03-18, CS at Worcester Polythechnic Institute, May 2003. http://www.cs.wpi.edu/~claypool/papers/video-crawler/paper.pdf

[6] T. Kuang and C. Williamson "A measurement study of RealMedia streaming traffic" http://pages.cpsc.ucalgary.ca/~carey/papers/itcom02.pdf

[7] T. Karagiannis, A. Broido, N. Brownlee, kc claffy, and M. Faloutsos. "File-sharing in the Internet: A characterization of P2P traffic in the backbone" Technical report., 2004. http://www.cs.ucr.edu/~tkarag/papers/tech.pdf

[8] Salman A. Baset and Henning Schulzrinne "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol" http://www1.cs.columbia.edu/~salman/publications/cucs-039-04.pdf

[9] Jörn Seger,"Modelling Approach for VoIP Traffic Aggregations for Transferring Tele-traffic Trunks in a QoS enabled IP-Backbone Environment" http://www.ist-intermon.org/workshop/papers/02_01_voip_aggregation090203_udo.pdf