



Investigation of the Performance of 100Mbit and Gigabit Ethernet Components Using Raw Ethernet Frames

ATL-COM-DAQ-2000-014

R. E. Hughes-Jones¹, F. Saka²

1. Introduction

This paper describes tests carried out as part of the ATLAS Trigger/DAQ Pilot Project to investigate and measure the performance of 100 Mbit and Gigabit Ethernet components using raw Ethernet frames. Results are presented for Ethernet interfaces, links and 100 Mbit and Gigabit switches from Alteon [1] and BATM [2]. This work complements previous investigations with the 3com 3300 100 Mbit switch [3], the Cisco Catalyst 6500 Gigabit switch [3] and that done on the Netwiz TurboSwitches [4].

The aims of the work were to:

- provide understanding of the operation of current networking components.
- make baseline network measurements that can be used as input to modelling that includes simulation of:
 - low level firmware/hardware performance
 - the ATLAS Pilot Project testbed
 - the full ATLAS Level2 trigger
- gain experience in evaluating network components and their interaction with computing systems to be able to specify the performance required for the experiment.

The document first describes the equipment used and the test environment and then gives details of the architecture of the switches used. Section 3 outlines the Latency measurements made, presenting equations giving the propagation delay through the network as well as histograms of the latency times. Section 4 discusses the data streaming measurements, presenting information on latency, data throughput and frame loss. Section 5 describes the results obtained when multicast frames were used.

To be able to view the plots and diagrams clearly, it is advisable to print this document in colour on a printer with Postscript capabilities.

¹ The University of Manchester

² Royal Holloway & Bedford New College, currently at CERN

2. Description of the Tests

2.1. Specification of the Test Equipment

For the tests involving Gigabit Ethernet, up to 8 PCs were connected to the Ethernet switches using fibre optic cables. Two of the systems had 400 MHz processors, the rest were equipped with 350 MHz Pentium II processors but all used the Alteon ACEnic network interface. Some tests were also made with two PCs directly connected with the fibre optic cable.

The same PCs were used in the tests involving 100 Mbit Ethernet but with Intel Ether-Express pro 100 network interfaces, with the 82558b chip or later revision, and twisted pair cable. A different set of PCs equipped with 200 MHz Pentium II processors were used for some of the Fast Ethernet tests. Some of the tests required NFS access, this was provided by a separate Ethernet interface connected to the CERN LAN at 10 Mbit. The switches and interfaces used for the tests were not connected to the CERN LAN. When required, a HP 1663AS Logic analyser was connected to the PCI buses of the PCs. Full details of the hardware and software are given in Table 2.1.

Hardware			
CPU	350 MHz Pentium II 1 processor installed	400 MHz Pentium II 1 processor installed	200 MHz Pentium II 1 processor installed
Motherboard	GigaByte Mother board GA-686BX Single processor Intel 440BX AGP chipset 32-bit 33 MHz PCI bus	GigaByte Mother board GA-6BXD Dual processor Intel 440BX AGP chipset	32-bit 33 MHz PCI bus
Memory	128 Mbytes of Memory 512 Kbytes Cache		32 Mbytes of Memory 512 Kbytes Cache
Graphics	Matrox Millennium Graphics card		
Ethernet Card for IP	3com PCI Fast EtherLink XL NIC 3C905B-TX 10/100 TP		
Ethernet Card for tests	Intel Ether-Express pro 100 network interfaces, with the 82558b chip or later revision		
Gigabit Card	Alteon ACEnic Tigon-2 32/64 bit, 33/66MHz PCI Firmware 12.3.9		
Software			
Operating system	Redhat Linux 5.1		
Kernel	2.0.36 with BigPhysarea patch		

Table 2.1 Specification of the Hardware and Software components

2.2. Ethernet Frame Format and Data Transfers

Raw Ethernet frames using the IEEE 802.3 MAC format [5] were used for all the tests described in this note; they consisted of the 14 byte header followed by variable length of user data. The header was made up of the 6 byte destination address, the 6 byte source address and the 2 byte data length indicating the number of bytes following, as shown in Table 2.1.

When a frame was moved over the PCI bus, the total data transfer was the length of the user data plus the 14 byte header. When placing a frame on the Ethernet cable, the hardware first generates an 8 byte preamble, then outputs the 14 byte header followed by the user data and then appends a 4 byte CRC. Figure 2.1 shows a timing diagram of a frame flowing through the system assuming each stage operates a "store and forward" policy, that is, the whole frame is received by a stage before any action is taken by that stage.

Byte1	Byte2	Byte3	Byte4
Destination address		Source address	
Length of User Data		User Data	

Table 2.1 Format of the Ethernet frame.

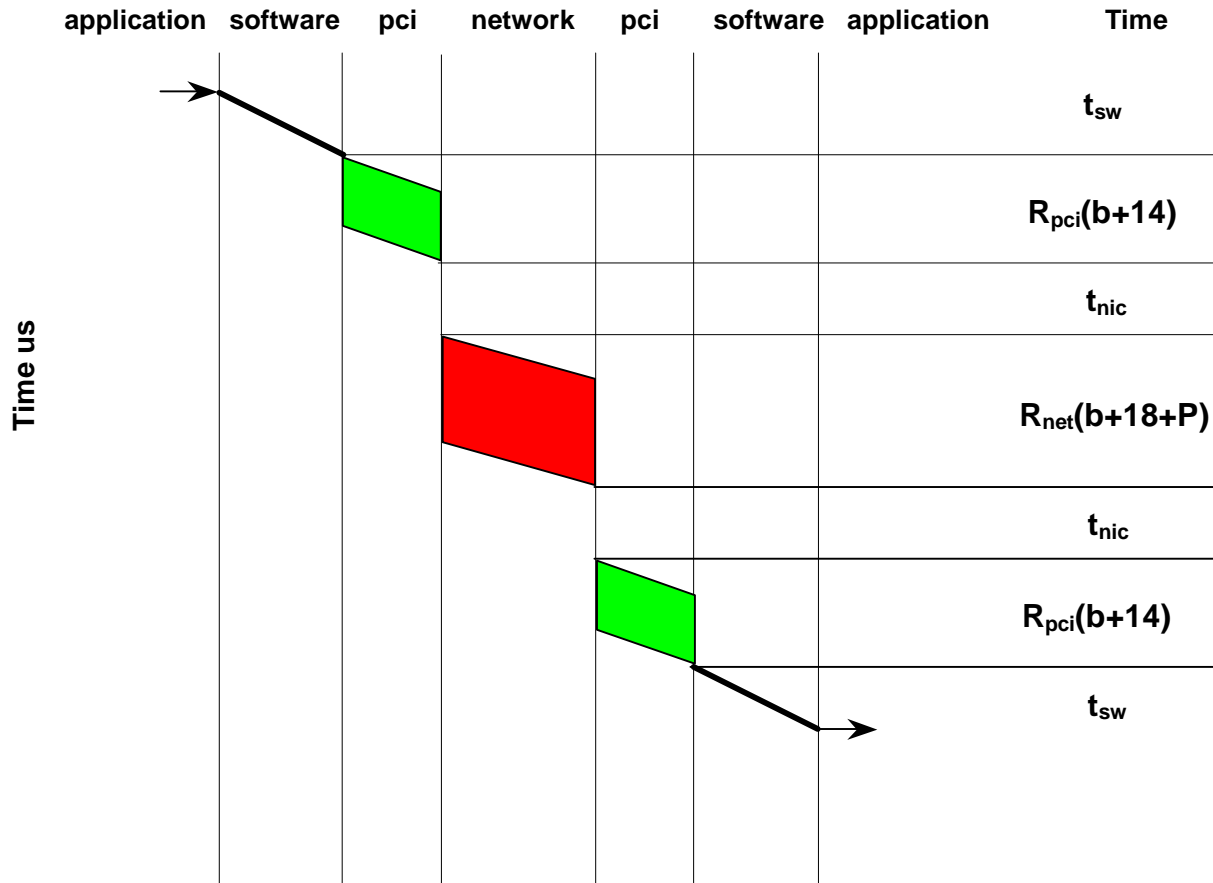


Figure 2.1 Plot to illustrate the time flow of a frame moving through the system. The shaded areas indicate when the time taken is dependent on the frame length.

Following the discussion in [2], the time to transfer a frame of data from one system to another is given by the equation:

$$T = 2t_{sw} + 2t_{nic} + R_{net}(b + 18 + P) + 2R_{PCI}(b + 14)$$

Where

- b the number of bytes transferred
 - R the transfer time per byte on the media concerned
 - t the transit time for a particular subsystem
 - P is the 8 byte preamble sent by the hardware prior to each Ethernet packet
- The Subsystems are: *sw* software; *nic* network interface card; *PCI* the PCI bus; *net* Ethernet

Re-arranging the equation to group the constant and message length dependency terms we have:

$$T = (2t_{sw} + 2t_{nic} + 26R_{net} + 28R_{PCI}) + b(R_{net} + 2R_{PCI}) \quad \text{Equation 2.1}$$

2.3. Discussion of the Switch Architecture

2.3.1. The Alteon 180 Gigabit Switch

The Alteon 180 Gigabit Ethernet switch [1] has 8 gigabit ports and one gigabit uplink port. Separate twisted pair connectors on the 8 ports allow these ports to be used as 10/100 Mbit links, but this was not used in these tests. Each port has an associated Layer-2 switching engine contained in a custom ASIC, which also contains dual 32-bit RISC processors that can be used for session switching e.g. WEB session load balancing. There is also memory associated with each port, and it is interesting to note that the up-link is mounted on a card that appears very similar to the Alteon 32/64 bit PCI interface card. The total throughput claimed is 5 million 64 byte frames/s. This is equivalent to a “backplane” throughput of 2.56 Gbit/s.

2.3.2. The BATM Titan 4 Switch

Title:
rhj_t4_topology
Creator:
Tgif-3.0-p10 by William Chia-Wei Cheng (william@cs.UCLA.edu)
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 2.3 Block diagram of the BATM switch.

The BATM Titan 4 switch [2] has as a chassis that can hold up to four plug in network modules. A module has either one Gigabit Ethernet port using the Galileo 48320 MAC and switching chip or eight 100 Mbit ports using two quad transceivers connected to a Galileo GT48310 8 port Fast Ethernet MAC and Switch Controller, as shown in the block diagram of Figure 2.3. The Galileo chips on each module are attached via proprietary full duplex G.Link interconnect, operating at 1.2 Gbit/s in each direction, to a Galileo 48300 4-port crossbar switch chip mounted on the backplane. The reported throughput of this crossbar chip is 12 Gigabit/s [7]. This means up to four Fast Ethernet modules (a maximum of 3.2 Gbit/s) or four Gigabit Ethernet Modules (a maximum of 4 Gbit/s) can be attached to the backplane. There is also an Intel i960 processor inside the switch, which is responsible for handling the SNMP based switch management software.

From the layout, we infer that when switching between Gigabit Ethernet ports, frames always go via the backplane and the crossbar, thus the latencies between different pair combinations are expected to be the same. For a Fast Ethernet module, the GT48310 contains a switching engine that can route frames locally on the module or over the backplane depending on the destination address. Thus we expect different latencies when switching between ports of the same Fast Ethernet module and when switching between ports on different Fast Ethernet modules.

3. The Request-Response and Ping-Pong Measurements

The Request-Response and Ping-Pong tests were functionally similar in that they used the initiating PC to measure the round trip times as a function of the frame size. A linear equation, as discussed in Section 2.2, was fitted to this data to describe the time taken to send a packet from one system to another. The results of these fits are presented in the following sections.

The test programs used a user-mode library, developed in the MESH [6] project at CERN to drive the 100 Mbit and Gigabit interfaces. It memory maps the CSRs (Control and Status Registers) of the Intel and ACEnic interfaces into user space thus avoiding the operating system context switches associated with using a normal driver to send Ethernet frames. The library handles four queues of frame descriptors. To send a frame, a descriptor of the frame was placed on the Send queue and returned on the Sent queue after the data had been transmitted by the interface. When data was received from Ethernet, a frame descriptor from the Receive queue was filled in and returned to the program on the Received queue.

The Request-Response tests measured the time taken to send a "Request" to a remote host and obtain a "Response" as a function of the size of the "Response" message. The length of the "Request" message could be varied, but for the measurements reported here, the request was fixed at 64 bytes long. The time for a series of Request-Responses, typically 10000, was measured using the real time clock of the sender, and the latency was calculated as the time to send the Response message in one direction. The Request-Response plots show the *total round trip* latency as a function of the user data length.

For the Ping-Pong tests, the latency was measured as a function of the message size by sending a Ping message to the remote host and waiting for the Pong message to return. Both messages were the same size. The time for a series of Ping-Pongs, typically 10000, was measured using the real time clock of the sender, and the latency was calculated as the time to send the message in one direction. The Ping-Pong plots show the *one-way* latency as a function of the user data length.

For the tests described in this paper, the network interfaces, and switch ports were set in full duplex mode and IEEE 802.3x flow control was enabled. Any special conditions are noted in the text.

When a Request-Response or Ping-Pong test was made to investigate the fundamental latency of a switch no other traffic was sent through the switch during the test. This ensures that there were no frame queues within the switch.

3.1. Latency as a function of Frame size using Gigabit links

3.1.1. Two Directly Connected PCs

These tests were made using the 400 MHz PCs and the Alteon interface cards. The middle curve on Figure 3.2 shows the Request-Response latency as a function of the response frame size for two directly connected PCs. The curve is smooth and linear. Allowing for the 64 byte request, the equation describing the one way propagation delay T as a function of the frame size b bytes is:

$$T(b) = 26.32 + 0.0235 * b \mu\text{s}$$

Equation 2.1 demonstrated that the slope should be made up of contributions for the transfer of the frame over the network and PCI bus. Allowing for the transmission of the packet at 1 Gigabit line speed (0.008 $\mu\text{s}/\text{byte}$), this indicates an effective PCI transfer rate of 0.00775 $\mu\text{s}/\text{byte}$ for the PCI bus on each PC. This is in good agreement with data being bursted at full rate over the 33 MHz PCI that would run at 0.00758 $\mu\text{s}/\text{byte}$.

Logic analyser traces of the PCI bus showing the Request-Response shown in Figure 3.1 confirm this interpretation. The upper group of traces shows the signals on the PCI bus of the requesting PC and the lower group shows the signals on the PCI bus of the responding PC. The request is 64 bytes long and the response is 1500 bytes long. The traces show that the Alteon card sends data on Ethernet by first reading it from memory in one continuous PCI burst at 0.00745 $\mu\text{s}/\text{byte}$, as expected from a 32 bit 33

MHz PCI bus (0.00758 μ s/byte). However when data is received from Ethernet it is written to memory using a series of PCI bursts with a maximum length of 380 bytes.

The time between successive requests in Figure 3.1 was 90.9 μ s, which agrees with the time measured in the software rests. The red bars in Figure 3.1 show the actual data transfer times over PCI, and the green bars show the total time associated with the transfer, which includes accessing the control registers (CSRs) on the interface. When sending a frame (top set of traces left hand side) it takes $\sim 1 \mu$ s to inform the CSRs to initiate the transfer, which takes place 5 μ s later. 6 μ s after this the final CSR update takes place, giving a total time of $\sim 12 \mu$ s to send 600 ns of data. After a frame has been received and transferred to memory over PCI (top set of traces right hand side), CSR access is complete after 5.3 μ s and the application software is made aware of the frame. After another 6.6 μ s, the receive frame queue is updated. Note that for the responding PC, the received frame queue update overlaps the sending of the response.

The times between the end of the PCI transfer of data into the sending interface and the start of the PCI transfer from the receiving interface were 9.4 μ s for 64 bytes and 20.56 μ s for 1500 bytes. Allowing for the transfer time of the frame over Gigabit Ethernet these times indicate that the transit time of the Alteon interface, t_{nic} , was $\sim 4.25 \mu$ s.

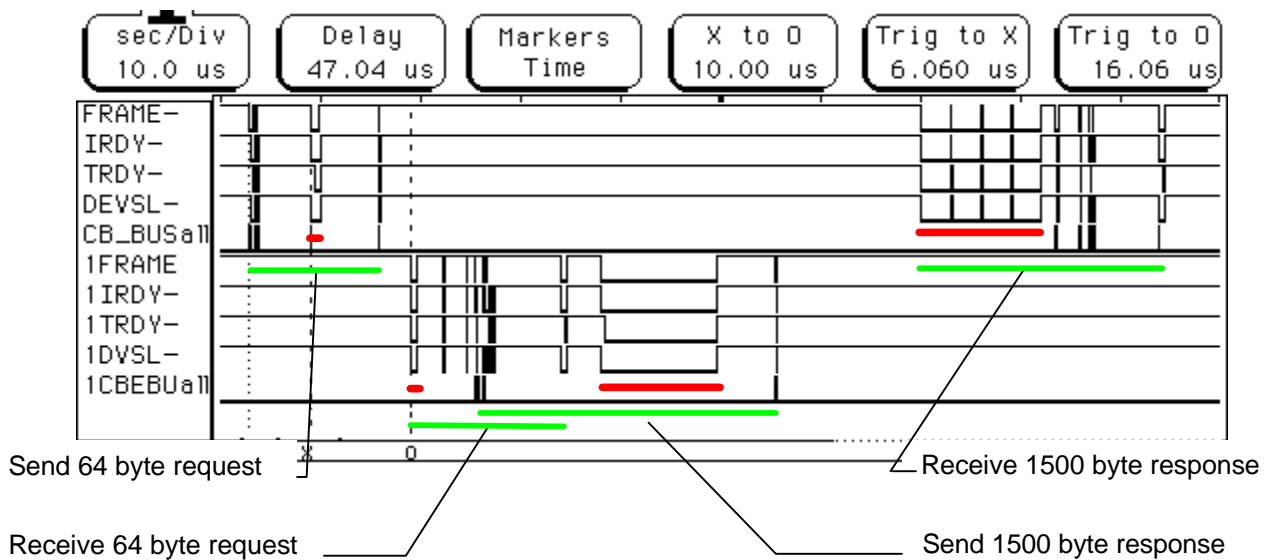


Figure 3.1. The upper group of Logic analyser traces show the signals on PCI bus of the requesting PC and the lower group of signals are from the responding PC. Data transfers on the PCI are shown with a red bar. The green bars show the total data and control access times for each transfer. A 64 byte request was send from the requesting PC, after it was received, a 1500 byte response was transmitted by the responding PC. This frame was then received by the requesting PC.

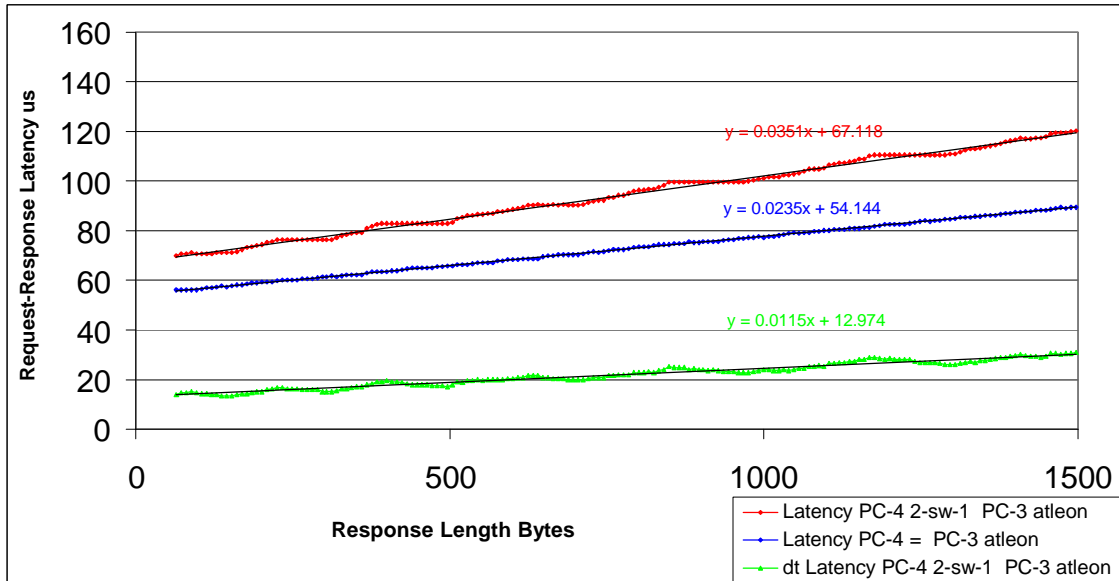


Figure 3.2. The Request-Response Latency as a function of Response Length for two PCs directly connected is shown in the middle curve; the top curve was measured when the PCs were connected using the Alteon Gigabit Ethernet switch. The bottom curve shows the contribution from the switch.

3.1.2. Two PCs Connected with the Alteon Switch

The top curve in Figure 3.2 shows the Request-Response latency as a function of the response frame size for the same two PCs connected via the Alteon gigabit switch. The latency introduced by the switch is shown in the bottom curve. This was calculated by subtracting the times measured when the PCs were directly connected from those when the PCs were connected using the switch.

Allowing for the 64 byte request, the equation describing the one way propagation delay T from one PC to the other through the switch as a function of the frame size b in bytes is:

$$T(b) = 32.436 + 0.0351 * b \text{ } \mu\text{s}$$

and the one way latency though the switch alone T_s is:

$$T_s(b) = 6.119 + 0.0115 * b \text{ } \mu\text{s}$$

The slope of $0.0115 \text{ } \mu\text{s}/\text{byte}$ is consistent with storing the incoming packet at 1 Gigabit Ethernet line speed (i.e. at $0.008 \text{ } \mu\text{s}/\text{byte}$) and then transferring the frame over the backplane of the switch at an effective 2.29 Gbit/s ($0.0035 \text{ } \mu\text{s}/\text{byte}$). This may be compared with the value of 2.56 Gbit/s deduced in Section 2.3.1 from the Alteon Switch data sheet.

Figure 3.3 shows histograms of the Request-Response latency for various response lengths for the PCs connected via the switch. The dark blue points in the histograms represent the data with just the latency traffic was sent through the switch, these distributions show a single peak with a Full Width Half Maximum, FWHM, of $2.6 \text{ } \mu\text{s}$ (or $\sigma = 1.1 \mu\text{s}$). The red points in the graphs show the latencies when a stream of 1500 byte “background” frames were sent every $28 \text{ } \mu\text{s}$ between two other ports to give an additional load on the switch. The distributions are displaced to longer latencies by 3 to $12 \mu\text{s}$ and show considerable structure. This suggests that there is a shared resource, such as a backplane bus or buffer controller within the switch which causes the frames to queue. In this case one could expect to observe three peaks caused by:

- no delay to request or response,
- one frame is delayed by the background traffic,
- both frames are delayed.

If just one frame were delayed, then on average the delay would be half the time taken to process the 1500 byte “background” frame and its header, this would be about $2.6 \text{ } \mu\text{s}$ using the measured transfer rate for the backplane of the switch.

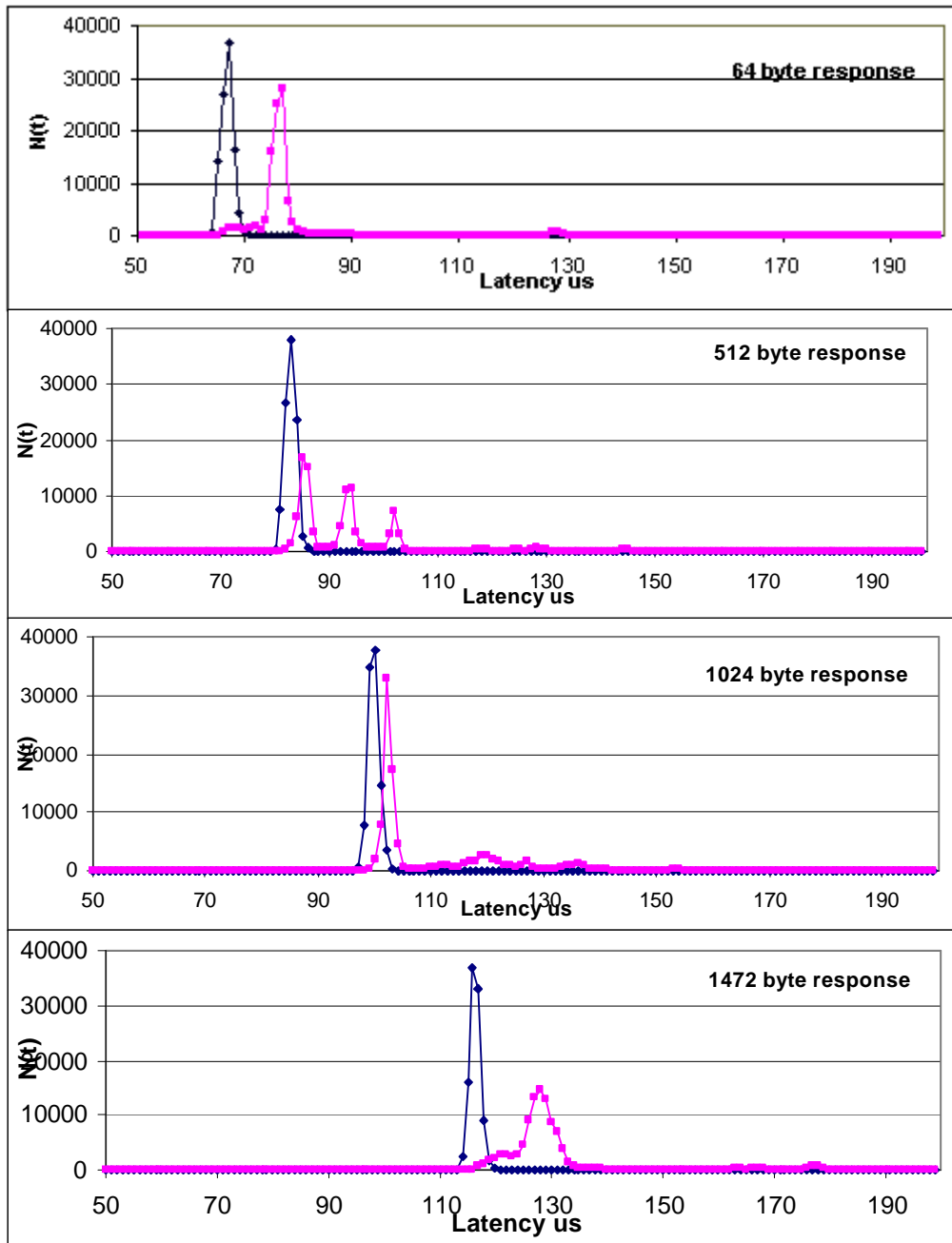


Figure 3.3 Histograms of the Request-Response Latency as a function of Response Length for PCs connected using the Alteon switch. Red points show the Request-Response Latency when 1500 byte “background traffic” frames are sent every 28 ms between two other ports of the switch.

3.1.3. Two PCs Connected with the BATM Switch

The top curve in Figure 3.4 shows the Request-Response latency as a function of the response frame size for the same two PCs connected via the BATM switch with gigabit modules. The latency introduced by the switch is shown in the bottom curve. This was calculated by subtracting the times measured when the two PCs were directly connected.

Allowing for the 64 byte request, the equation describing the one way propagation delay T from one PC to the other through the switch as a function of the frame size b in bytes is:

$$T(b) = 32.341 + 0.0416 * b \text{ } \mu\text{s}$$

and the one way latency though the switch alone T_s is:

$$T_s(b) = 6.04 + 0.0179 * b \text{ } \mu\text{s}$$

The slope of $0.0179 \text{ } \mu\text{s}/\text{byte}$ is consistent with storing the incoming packet at 1 Gigabit Ethernet line speed ($0.008 \text{ } \mu\text{s}/\text{byte}$) and then transferring the frame over the crossbar and backplane of the switch at an effective 808 Mbit/s. This rate is somewhat lower than the maximum specification for the BATM switch's internal interconnects of 1.2 Gbit/s, as discussed in Section 2.3.2.

The switch also introduces a “ramp and step” structure which occur about every 200 bytes and has flat portions ~ 88 bytes long.

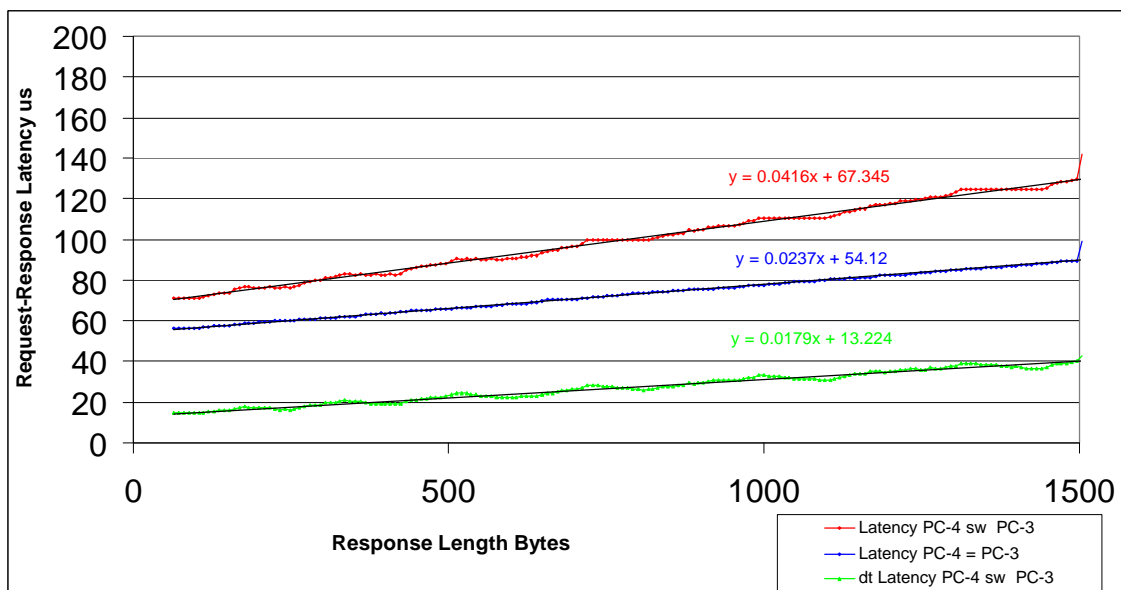


Figure 3.4. Request-Response Latency as a function of Response Length for two PCs directly connected and connected using the BATM Gigabit Ethernet switch. The lower trace shows the contribution from the switch.

Figure 3.5 shows histograms, plotted on a logarithmic scale, of the Request-Response latency for various response lengths for PCs directly connected and when connected via the switch. The test procedure waits for the response for each request, so there were no frames queuing in either PC. The histogram of the latency for directly connected PCs shows a shoulder at $8 \text{ } \mu\text{s}$ from the main peak for about 0.1 % of the frames. When the switch is used, it introduces an extra delay of between 35 to 40 μs for about 4% of the frames. The reason for this extra latency is unclear but it may be connected with the switch management functions such as gathering statistics from the MAC sub-systems.

The dark blue points in the right hand graphs show the latencies when a stream of 1500 byte packets is sent between two other ports on the switch. There is little difference in the position of the main peak, but the tails are displaced to higher latencies by a few μs . The fact that the position of the main peak does not move when background frames are sent through the switch, confirms the non-blocking

behaviour of the internal crossbar. Non-blocking in this paper, is used to mean that traffic passing between one pair of ports on the switch is independent of traffic passing between other ports and has no effect on the performance of those ports.

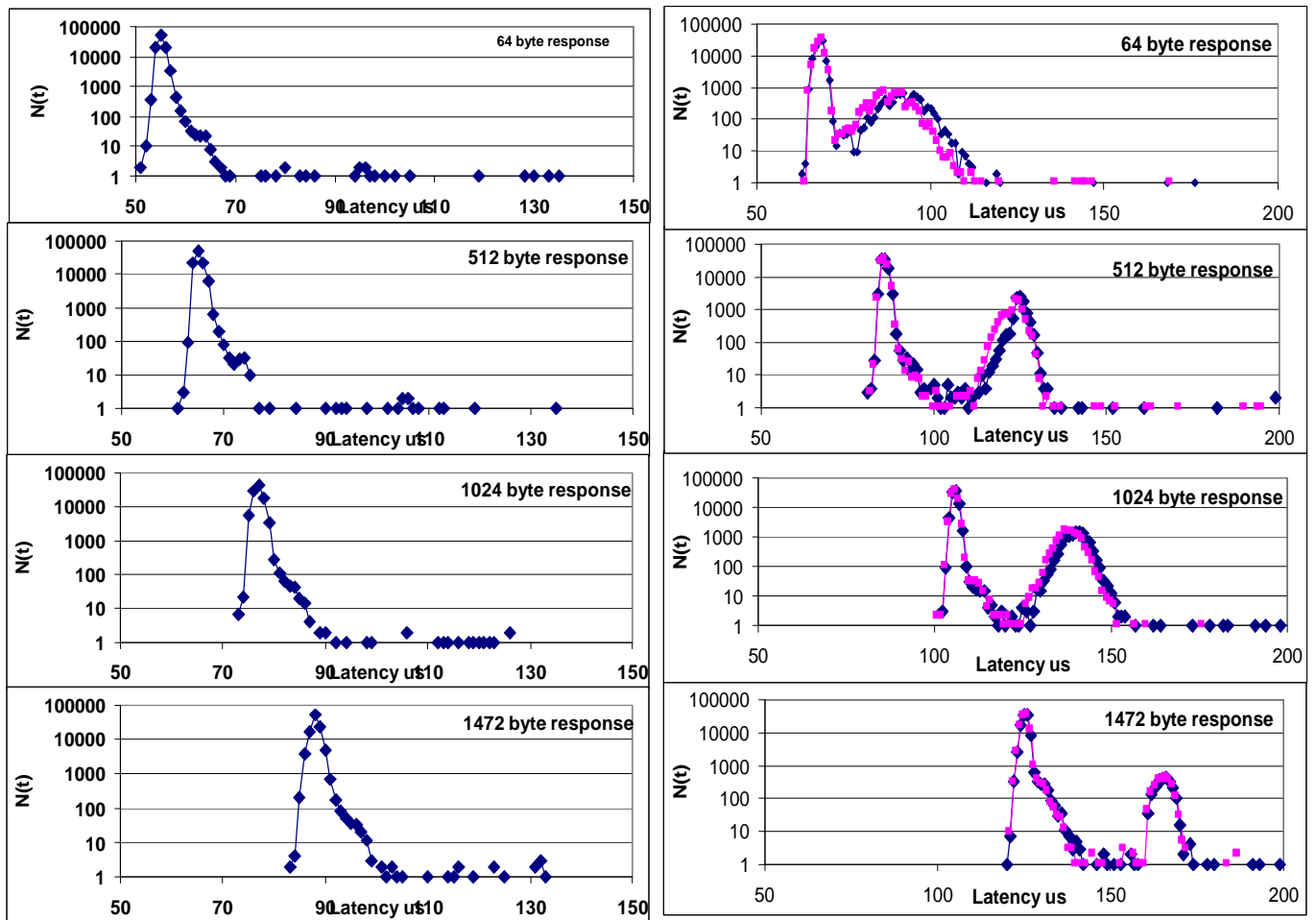


Figure 3.5. Histograms of the Request-Response Latency as a function of Response Length.

Left: two PCs directly connected.

Right: PCs connected using the switch. The dark blue points show the Request-Response Latency with background traffic consisting of 1500 byte frames sent every 28 ms between two other ports of the switch.

3.2. Latency as a function of packet size using 100 Mbit links

3.2.1. Two Directly Connected PCs

Ping-Pong tests were made to measure the one way latency as a function of the frame size for two 350 MHz PCs directly connected with a 100 Mbit Ethernet link, the data is shown in bottom curve of Figure 3.6. The equation describing the one way propagation delay between the two PCs, T , as a function of the frame size b bytes is:

$$T(b) = 33.4 + 0.0814 * b \text{ } \mu\text{s}$$

Equation 2.1 demonstrated that the slope should be made up of contributions for the transfer of the frame over the network and PCI assuming that each stage operated on a store and forward basis. The Intel interface does not wait for completion of the data transfer from memory, but starts to transmit the frame over Ethernet soon after the DMA has been started. This does not cause wait-state problems for the frame as the time to transmit the frame over the 100 Mbit Ethernet is ~10 times ($0.08 \mu\text{s}/\text{byte}$) that required to move the data over PCI. When a frame is received from Ethernet, the interface stores several bytes and then moves that data into memory in a series of short PCI bursts. Thus both PCI transfer times are effectively overlapped with the time to send the data over the Ethernet, as the slope of $0.0814 \mu\text{s}/\text{byte}$ indicates.

3.2.2. Two PCs Connected with the BATM Switch

Figure 3.6 also shows the one way latency as a function of the response frame size for the same two PCs connected via the BATM switch using 100Mbit modules. The top curve shows the result when both links are connected to different 100Mbit modules, while the middle curve is for links going to the same module. Tests were also made with two switches interconnected with a 100 Mbit link or with a Gigabit link, as shown in Figure 3.7. The results of these measurements are summarised in Table 3.1

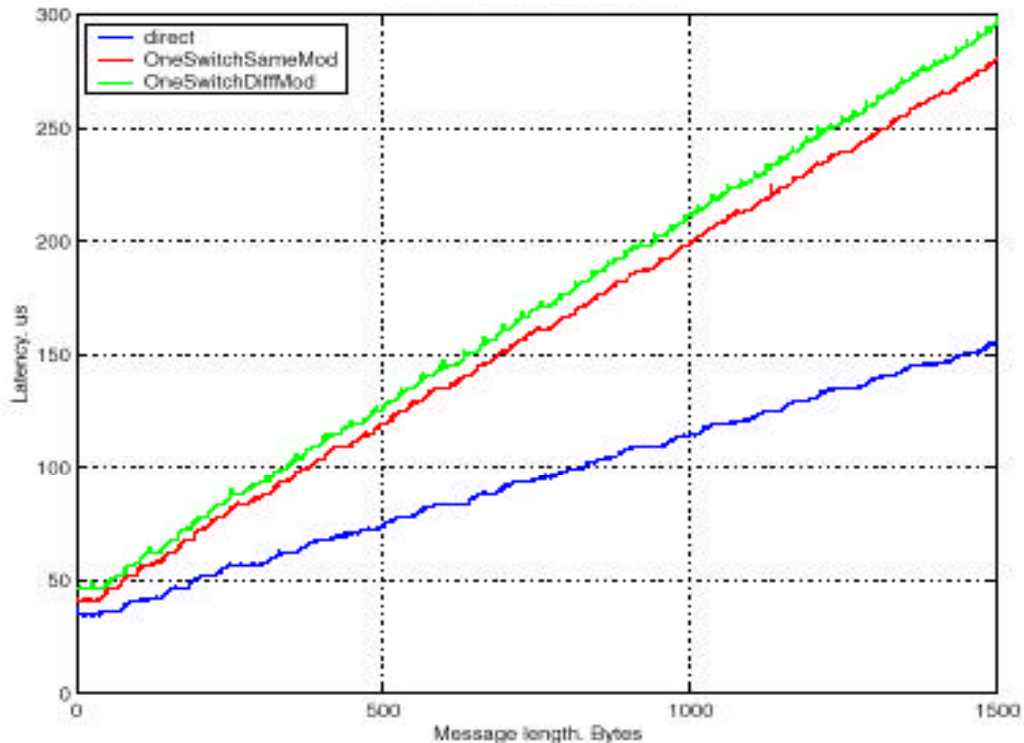


Figure 3.6 One way latency for 100Mbit Ethernet between 2 PCs directly connected and connected via the BATM switch.

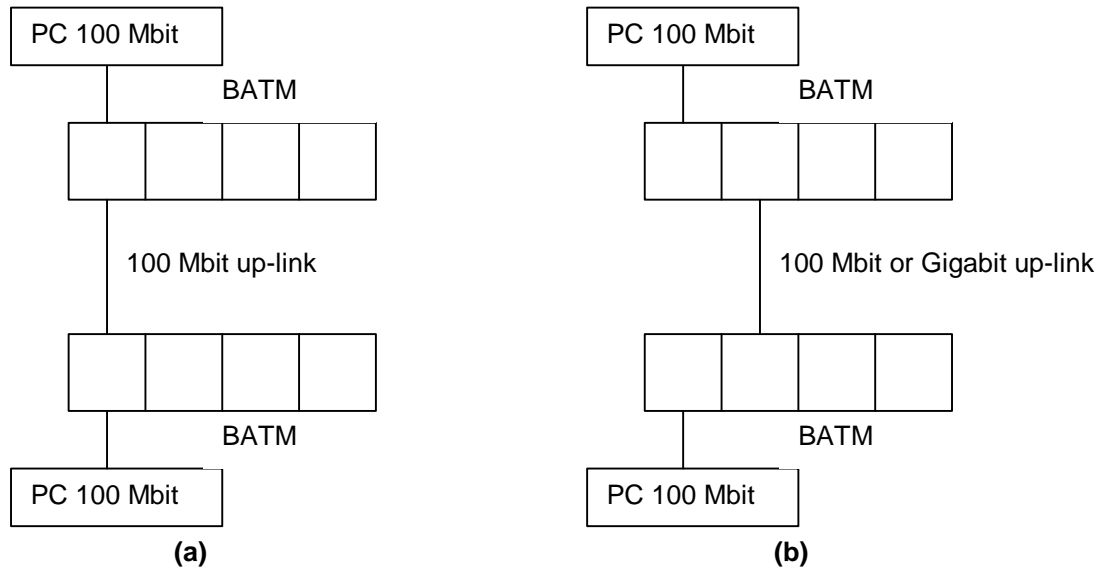


Figure 3.7 Connection of the PCs and BATM switches for the tests, (a) has the up-link connecting the switches using the same modules while in (b) the up-link uses different modules.

	PC to PC direct	One Switch		Two Switches 100 Mbit link		Two Switches Gigabit link
		Same 100 Mbit module	Different 100 Mbit modules	Same 100 Mbit module	Different 100 Mbit modules	
Zero length latency μs	33.4	38.3	41.8	43.4	50.0	47.3
Gradient $\mu\text{s}/\text{byte}$	0.0814	0.1614	0.1695	0.2414	0.2576	0.1867
Zero length latency μs of switch		4.9	8.4	10.0	16.6	13.9
Gradient $\mu\text{s}/\text{byte}$ of switch		0.08	0.0881	0.16	0.1762	0.1053
Contribution To Gradient		100 Mbit in	100 Mbit in backplane	100 Mbit in 100 Mbit uplink	100 Mbit in backplane 100 Mbit uplink backplane	100 Mbit in backplane Gigabit uplink backplane

Table 3.1 Results for 100 Mbit tests for different configurations of the BATM switch.

From these results, the equation describing the one way propagation delay, T_s , through the 100 Mbit switch as a function of the frame size b bytes for both links on the same module is:

$$T_s(b) = 4.9 + 0.08 * b \mu\text{s}$$

and for links on different modules is:

$$T_s(b) = 8.4 + 0.0881 * b \mu\text{s}$$

The slope of $0.0881 \mu\text{s}/\text{byte}$ is consistent with storing the incoming packet at the 100 Mbit/s line speed ($0.08 \mu\text{s}/\text{byte}$) and then transferring the frame over the crossbar and backplane of the switch at an effective 987 Mbit/s ($0.0081 \mu\text{s}/\text{byte}$). Table 3.1 shows that when two switches are connected in cascade, the latency and transfer rates double as expected. Using the data for the case when the two switches are joined by a Gigabit up-link, the effective transfer rate over the switch backplane was found to be 925 Mbit/s ($0.00865 \mu\text{s}/\text{byte}$). The difference is not understood, but the G.Link transfer rates between the different types of modules in the BATM switch and the crossbar chip on the backplane could be different.

3.3. Latency as a function of frame size 100 Mbit to Gigabit

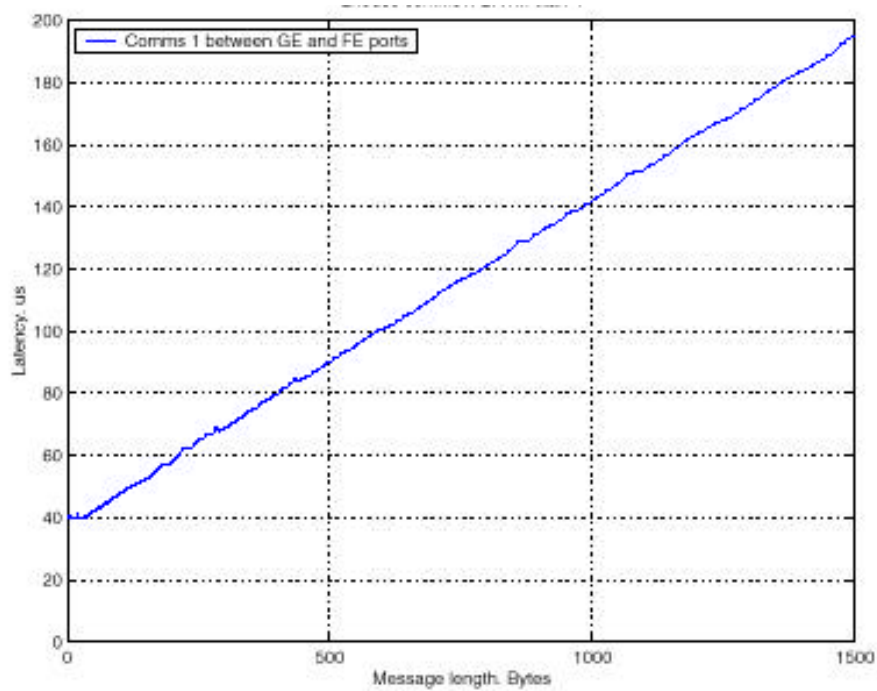


Figure 3.8 One way latency between two PCs one connected using 100 Mbit Ethernet, the other using a Gigabit link via the BATM switch.

Figure 3.8 shows the one way latency as a function of the frame size for the same two PCs connected via the BATM switch one connection using a 100Mbit module, the other a Gigabit module. The equation describing the one way propagation delay through the switch T_s as a function of the frame size b bytes is:

$$T_s(b) = 37.4 + 0.105 * b \text{ } \mu\text{s}$$

The slope should be made up of the following contributions :

	$\mu\text{s}/\text{byte}$
the transfer of the frame over the PCI bus of the sending PC	
transmission of the packet at 100 Mbit line speed to the switch	0.08
the transfer of the frame over the Switch backplane	0.0099
transmission of the packet at 1 Gigabit line speed from switch to receiving PC	0.008
the transfer of the frame over the PCI bus of the receiving PC	0.00775
Total	0.10565

The transfer of the frame over the PCI bus of the sending PC is not included as the Intel Ether-Express pro 100 interface overlaps transmission on the Ethernet with obtaining the data over PCI from local memory, see Section 3.2.1. There is good agreement between the measured value of the slope and that computed from the components.

4. The Data Streaming Measurements

A network wide global clock was required in order to measure the one-way latencies. This was achieved by synchronising the on-chip cycle counters in the Pentium II CPUs and using this register to timestamp the frames when they were created and received by the test programs [8]. This approach made the assumption that the clocks of each node differed linearly. The ping-pong exercise was used to synchronise the clocks as follows: firstly, a transmitting node reads its local time and sends a "ping" message to a receiving node. The receiving node puts a time stamp on the message and sends it back as the "pong" message. Upon receiving the "pong" message, the original transmitting node reads its local time again and the time in the "pong" message. The original transmitting node can then deduce that the time in the pong message corresponds to its own time half way during the ping-pong process, or $(\text{start_time} + \text{end_time})/2$. By making a linear fit to a series of these measurements, the offset between the two clocks was determined.

There was a tendency for clocks on the PCs to drift primarily due to the effects of temperature on the crystals providing the clocks. As a result, we took a number of precautions in ensuring that our results were accurate to $\pm 2.5 \mu\text{s}$:

- 1) Before beginning a set of measurements, the PCs were put into warm up phase for 25mins.
- 2) Before each measurement, a resynchronisation is performed.
- 3) Each measurement took less than 3 minutes; measurements indicated that the typical drift in time between two CPU clocks was $\sim 0.15 \mu\text{s}/\text{min}$.

To perform a measurement, the clocks of the set of PCs in use were first synchronised as described above. Each PC then read a file telling it the size of Ethernet frames to send, the time to wait between each subsequent frames, and the destination of the frame. Before sending each frame, the transmitter put a time stamp into the frame and also a sequence number. Upon reception, the receiving PC read its clock, and using the time offsets, it was possible to record how long the frames took to traverse the network one-way. If the received sequence numbers were not consecutive, the number of lost frames could be calculated. Frames could be generated regularly spaced in time or with the frame spacing following a negative exponential distribution about a given mean (the Poisson distribution).

The code used in these tests operated in the full MESH light-weight thread switching environment [6]. To send a frame the user-mode library wrote the location of the frame descriptor to the network interface, the interface then moved the frame over PCI from memory using onboard DMA, and then transmitted the frame onto the Ethernet.

The queuing of the frames in the nodes and network was investigated by continuously sending frames from one node to another and recording the one-way latency for a range of time intervals between generating the frames.

The maximum throughput of the network was measured by streaming frames from one node to another with a queue of frames in the transmitting node. This ensured that the transmission of a frame never had to wait for the sending software to generate the frame. The number of frames sent in a given time gave the transmit throughput, and the number of frames observed by the receiver gave the receive throughput. The one way latency depended on the time the frame spent in the queue of frames to be sent on the sending PC as well as propagation delays and queuing times at other points of the network.

4.1. One way Latency as a function of the average time between frames

4.1.1. Two Directly Connected PCs

Figure 4.1 shows a plot of the latency observed as a function of the average time between generating packets for two PCs directly connected. Curves for three frame sizes are plotted for frames generated at regular intervals and with a random Poisson distribution. As the average time is decreased, queues of frames can build up in the transmitting PC due to the exponential nature of the Poisson generating distribution. When queues form, even for brief periods, the time to send the packet from transmitting program to receiving program increases. If the average frame generation rate exceeds the maximum possible rate that frames could be transported through the system, "infinite" length queues would

develop, the latency would rapidly increase (!) and the throughput would limit. This was observed for times smaller than the lowest latency point plotted in Figure 4.1. For the regularly spaced frames, shown as the the open points, there was a step function rise at the last point. Otherwise these curves were flat, as expected, and agreed with the latency of the randomly generated frames at large time intervals between the frames.

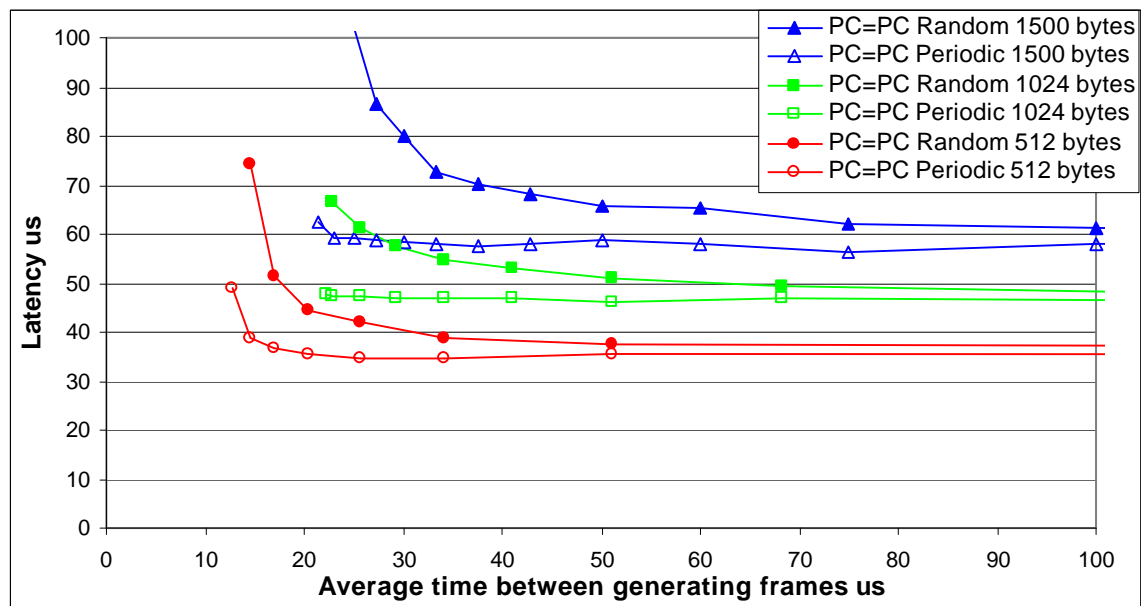


Figure 4.1 One way latency as a function of the average time between generating frames for directly connected PCs. Frames were generated at regular intervals and with a random Poisson distribution.

Figure 4.2a shows PCI signals for the sending PC when it is streaming 1500 byte frames regularly spaced at 22 μ s by the software. As expected, the control registers in the Alteon card are written then the card DMA's the data over PCI, with a time between successive frames of 22.4 μ s. Referring to the traces showing the 64 byte frame sent in Figure 3.1, one notes that the CSR access to initiate a new frame occurs before the update of the sent frame queue for the previous frame. If the time between frames is reduced to 20 μ s, then the regular CSR – DMA pattern is replaced by that shown in Figure 4.2b. It appears that the CSR registers are updated in rapid sequences lasting about 40 μ s, there is then a period of 60-80 μ s with no activity on the bus, and then the interface makes a set of data transfers, presumably overlapping the PCI data transfers with transmitting the data on Ethernet.

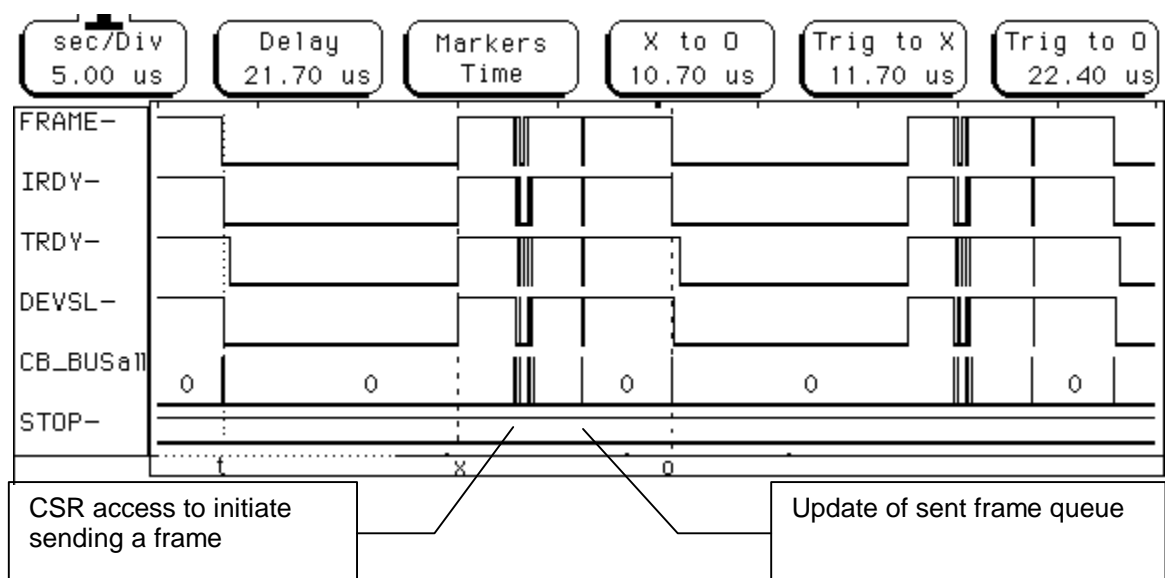


Figure 4.2 (a) PCI signals for the PC sending 1500 byte frames generated with a regular spacing of 22 ms.

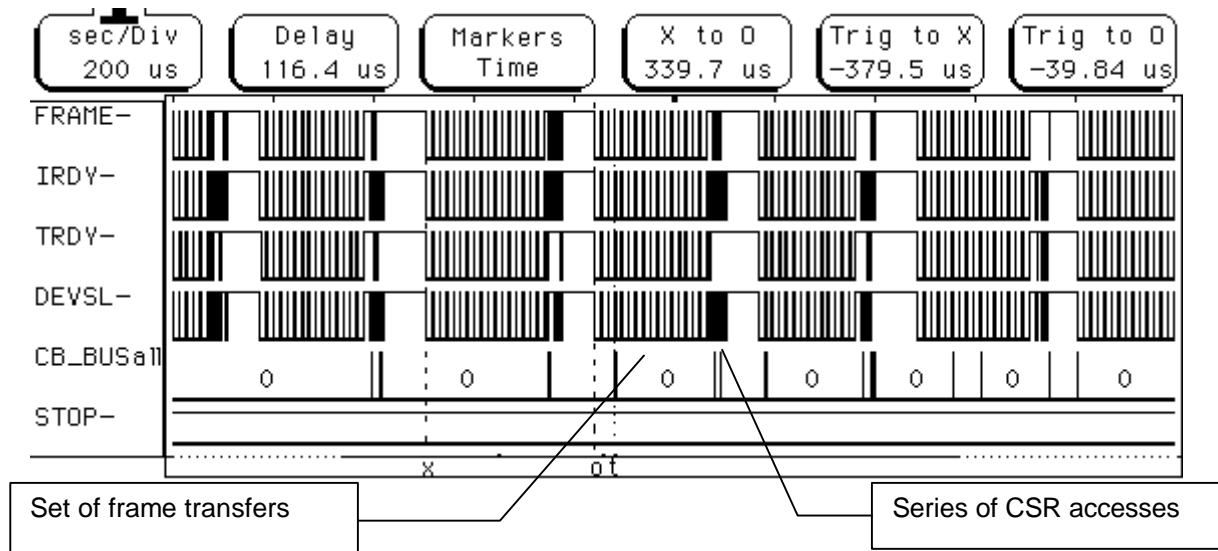


Figure 4.2 (b) PCI signals for the PC sending 1500 byte frames generated with a regular spacing of 20 ms. About 25 frames were sent in each group of data transfers.

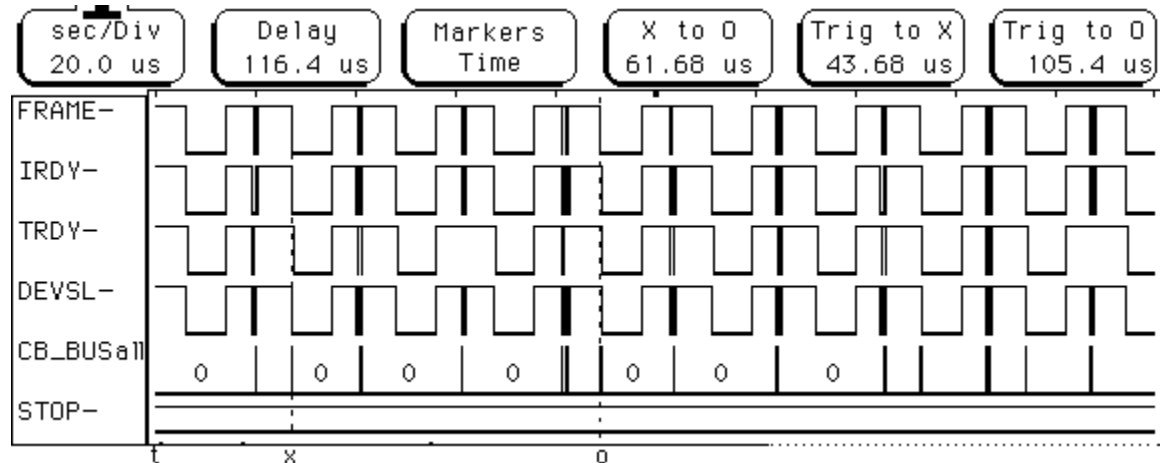


Figure 4.2(c) PCI signals for the PC sending 1000 byte frames generated with a regular spacing of 21 ms.



Figure 4.2 (d) PCI signals for the PC sending 1000 byte frames generated with a regular spacing of 20 ms. Note the CSR - data overlap during the first two sets of transfers and the long idle times.

Figure 4.2c shows the expected behaviour on the PCI bus for 1000 bytes frames spaced at 20.56 μs (nominal 21 μs) and Figure 4.2d shows the grouping of frames when the nominal spacing was reduced to 20 μs . Again the CSR registers were updated in rapid sequences lasting about 40 μs , followed by a period of $\sim 80 \mu\text{s}$ with no activity on the bus before the interface transferred the data. However, much larger periods $\sim 635 \mu\text{s}$ of no bus activity are also observed, but these were at irregular intervals. In this case, there is evidence that the CSR updates can be interleaved with the data transfers, see for example the first two data transfer periods in Figure 4.2(d). The exact mechanism that causes these effects is still unclear, but if the receiver was not able to process the frames in time, the interface could invoke the flow-control mechanism to suspend transmission. Further investigation is in progress.

Table 4.1 compares the calculated minimum times to send a frame based on the PCI and Gigabit transfer rates with those measured for regularly spaced packets and those generated with a Poisson distribution, as shown in Figure 4.1. The calculated times used the PCI transfer rate, the speed on the Ethernet and included the IPG. They assumed store and forward but did not take the time required to update the CSRs $\sim 5 \mu\text{s}$ into account. For 512 and 1024 byte messages, it appears that the store and forward model is appropriate, but for 1500 bytes, measured times of $\sim 29 \mu\text{s}$ would have been expected. The discrepancy is possibly due to optimisation in the Alteon card overlapping parts of the transfers.

Frame data size	Calculated minimum time	Regular spaced frames	Frames with a Poisson Distribution
512	8.32	12.8	14.36
1024	16.3	22.26	22.76
1500	23.73	21.43	23.08

Table 4.1 Comparison of Calculated and measured minimum times between sending frames, all times in ms.

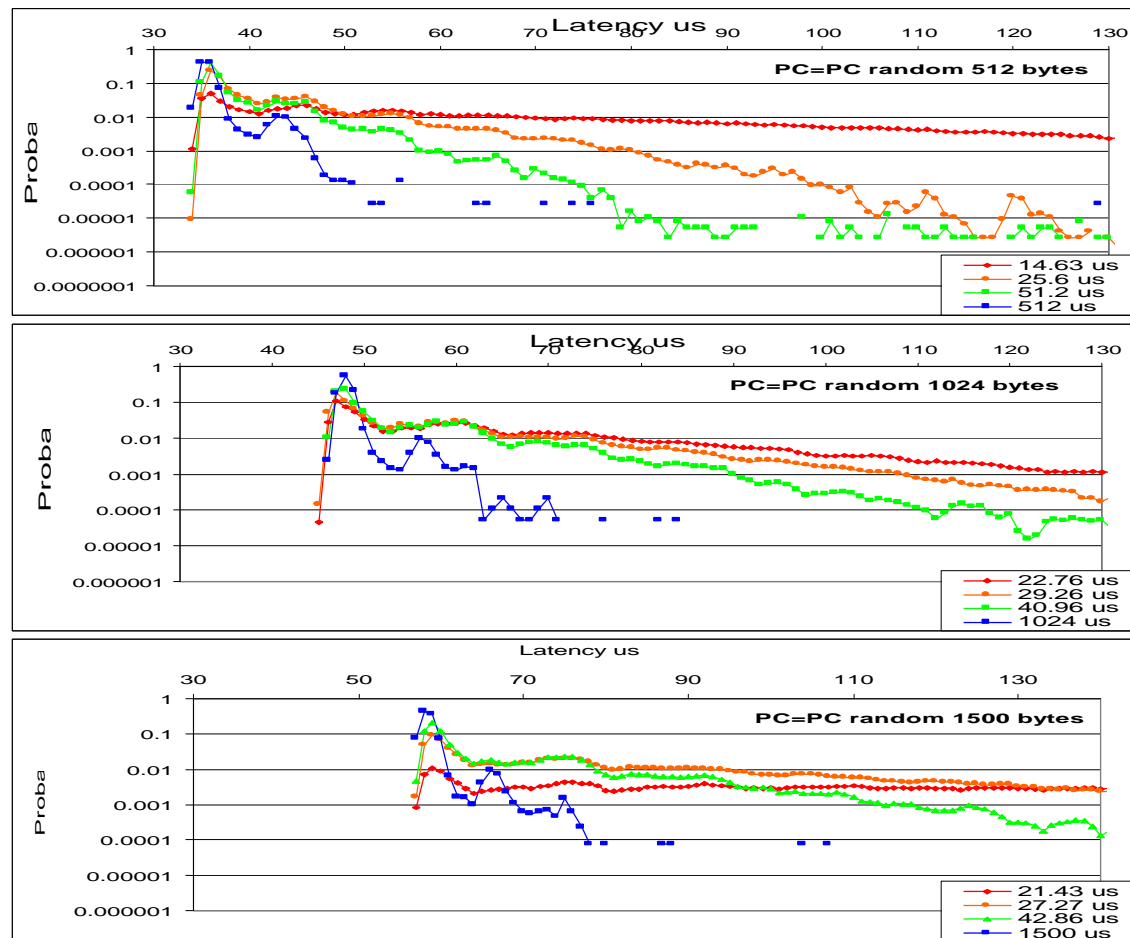


Figure 4.3 Probability of having a given one-way latency as a function of the average time between generating frames for directly connected PCs.

Histograms of the probability of having a given one-way latency as a function of the average time between generating frames are shown in Figure 4.3 for three frame sizes. The latencies corresponding to the positions of the main peaks agree with that calculated from the request-response and ping-pong measurements of Section 3.1.1, but are $\sim 2.5 \mu\text{s}$ less in each case. These differences arise because the streaming code times stamps the frames as they are placed on the send queue of the Alteon interface, and does not include other overheads such as creating the frame which was included in the Request-Response measurements. The sloping steps in the curves to the right of the main peak reflect the probability of frames queuing for transmission in the PC, and have been reproduced by the ATLAS network simulations using Ptolemy[9].

4.1.2. Two PCs Connected with the Alteon Switch

Figure 4.4 shows a plot of the latency observed as a function of the average time between generating packets with the Poisson distribution for two PCs connected using the Alteon switch with curves. The data is shown for three frame sizes. The curves are similar to those shown in Figure 4.1, which is reasonable given that the dominant queuing of frames was in the transmitting node. The increase in the latency over Figure 4.1 for an average time of $100 \mu\text{s}$ between the frames agrees well with that expected from the Alteon switch measurements in Section 3.1.2

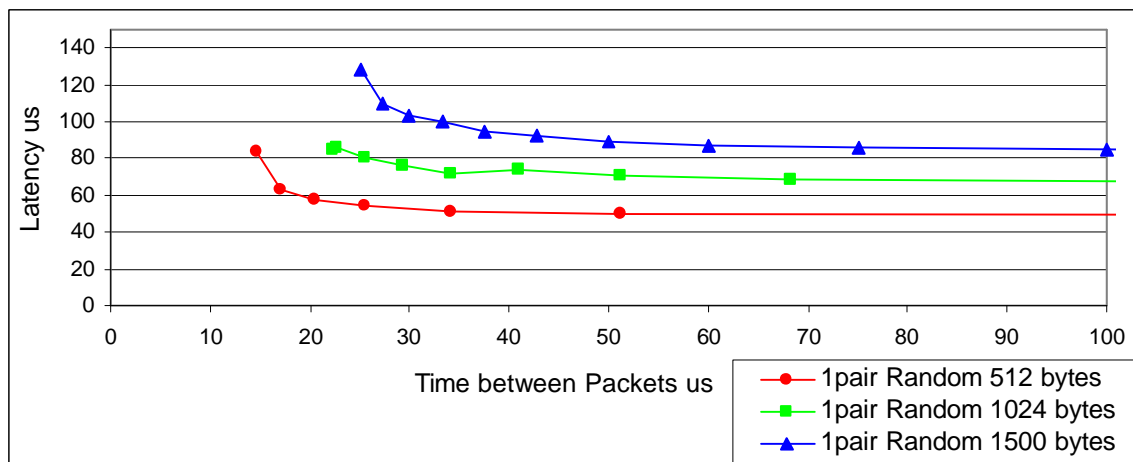


Figure 4.4 One-way latency as a function of the average time between generating frames with a random Poisson distribution. The PCs were connected using the Alteon Gigabit switch.

Figure 4.5 shows histograms of the probability of having a given one-way latency as a function of the average time between generating frames when the data passed through the Alteon switch for the three frame sizes. The latency corresponding to the position of the main peaks agree with the peak positions for directly connected PCs, shown in Figure 4.3, with the addition of the switch transit time T_s for the Alteon switch as calculated in Section 3.1.2. The shapes of the distributions are similar to those for directly connected PCs, but the tails to longer latencies appear to fall off less quickly than for the case of directly connected PCs.

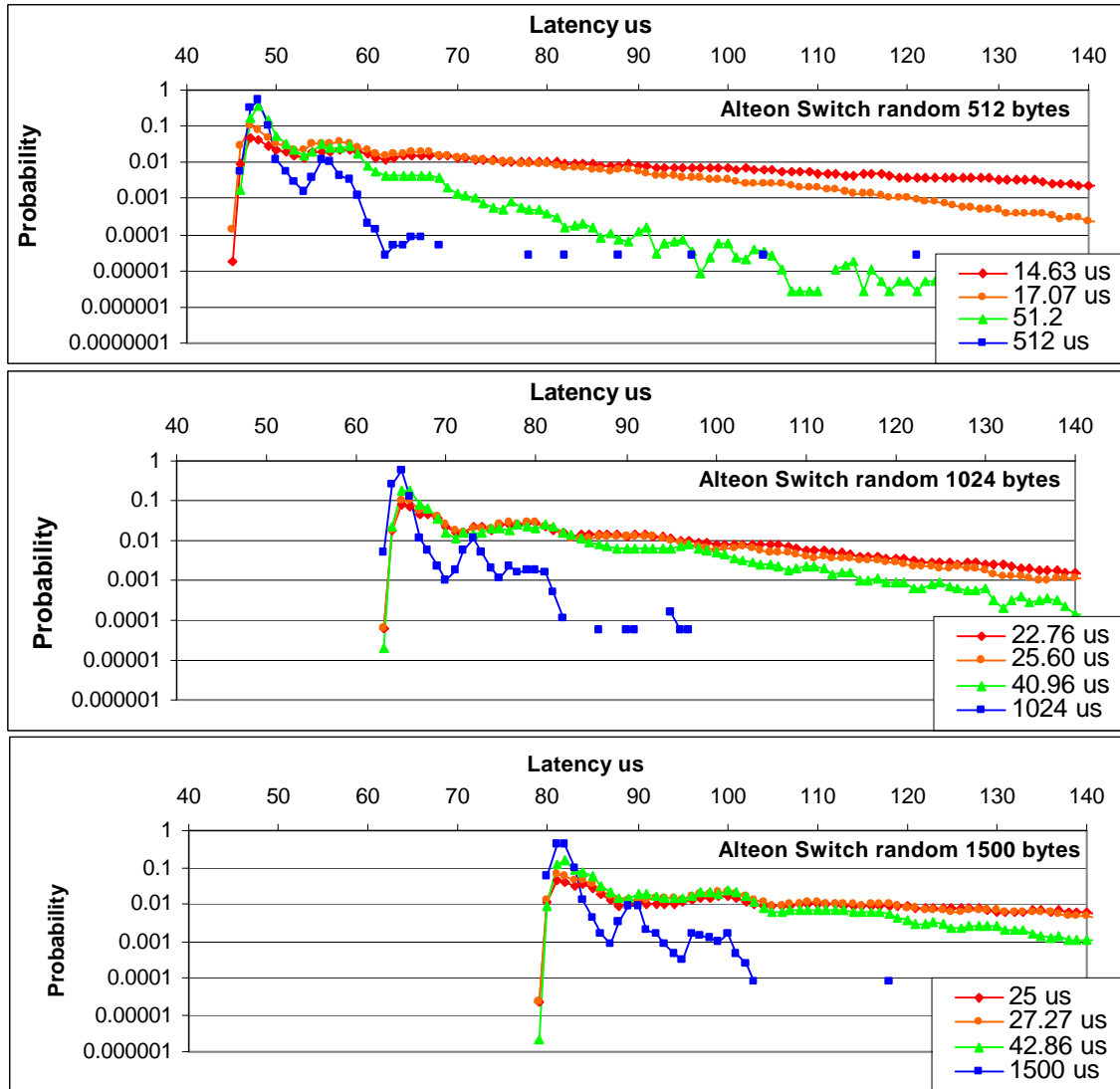


Figure 4.5 Plots of probability of the observed latency vs the average time between generating packets for two PCs connected using the Alteon Gigabit switch. Data for three frame sizes are shown.

4.1.3. Two PCs Connected with the BATM Switch

Figure 4.6 shows a plot of the latency observed as a function of the average time between generating packets with the Poisson distribution for two PCs connected using the BATM switch with curves for the three frame sizes. The curves are similar to those shown in Figure 4.1, which is reasonable given that the dominant queuing of frames was again in the transmitting node. The increase in the latency over Figure 4.1 for 100μs average time between the frames is about 10% smaller than that expected from the BATM switch measurements in Section 3.1.3, the reason is unknown.

Figure 4.7 shows histograms of the probability of having a given one-way latency as a function of the average time between generating frames when the data passed through the BATM switch for the three frame sizes. The latency corresponding to the position of the main peaks agree to within a few percent with the peak positions for directly connected PCs, shown in Figure 4.3, with the addition of the switch transit time T_s for the BATM switch as calculated in Section 3.1.3

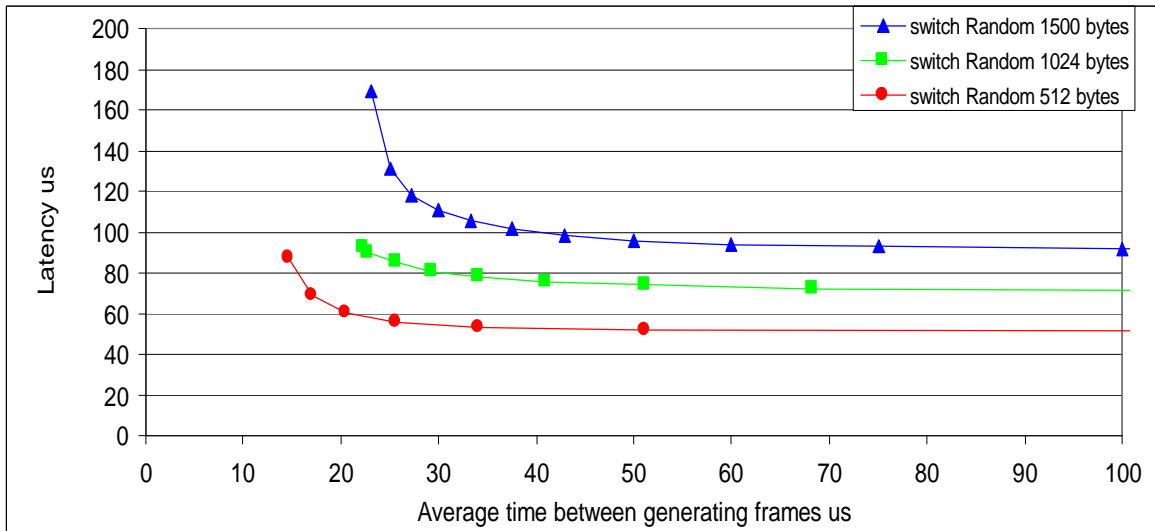


Figure 4.6 One-way latency as a function of the average time between generating frames using a random Poisson distribution. The PCs were connected using the BATM Gigabit switch.

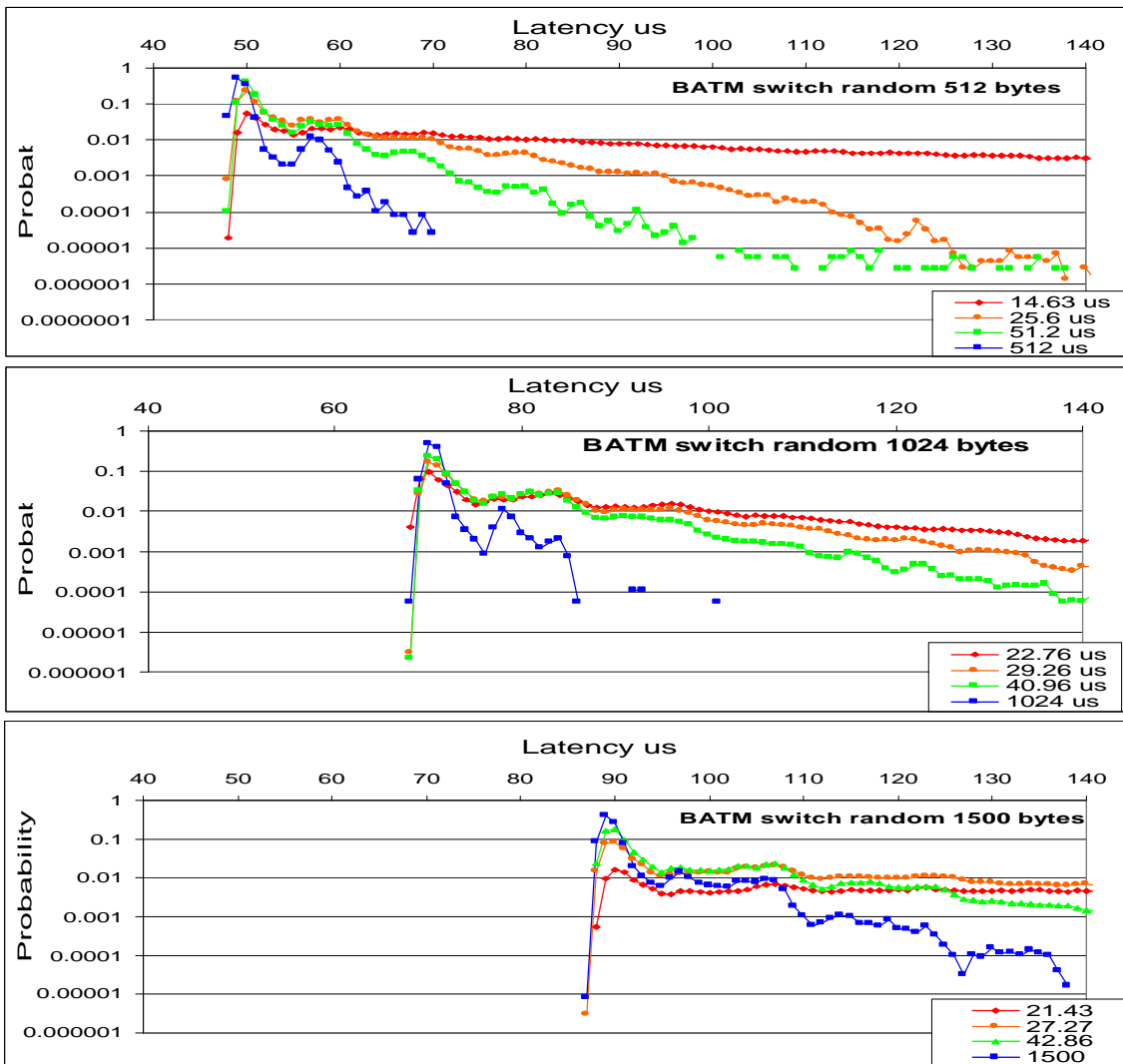


Figure 4.7 Plots of probability of the observed latency vs the average time between generating packets for two PCs connected using the BATM Gigabit switch Data for three frame sizes are shown.

4.1.4. Atleon Switch with streaming between two pairs of nodes

Figure 4.8 shows a comparison of the one-way latency when one and two pairs of node are streaming 1500 byte frames sent regularly every 21.53 μs through Gigabit Ethernet links to the Alteon switch. The mean of the peak moves to larger latencies by $\sim 1.1 \mu\text{s}$ when two pairs are sending data. The Request-response times presented in Section 3.1.2 also showed an increase in latency when background traffic was sent though the switch, however this was $\sim 12 \mu\text{s}$ for 1472 packets sent at random times.

Considering the data shown in Figure 4.8, a stream of 1500 byte frames would occupy the backplane for 5.3 μs every 22.3 μs . If the frames were sent randomly, then $\sim 24\%$ of them would have to wait for 2.65 μs . So overall, on average the frames would wait $\sim 0.64 \mu\text{s}$, or about half that observed. The fact that both streams were sent at regular, but unsynchronised, 22.3 μs intervals would clearly modify this simple view, depending on the difference in phase between the two streams of frames.

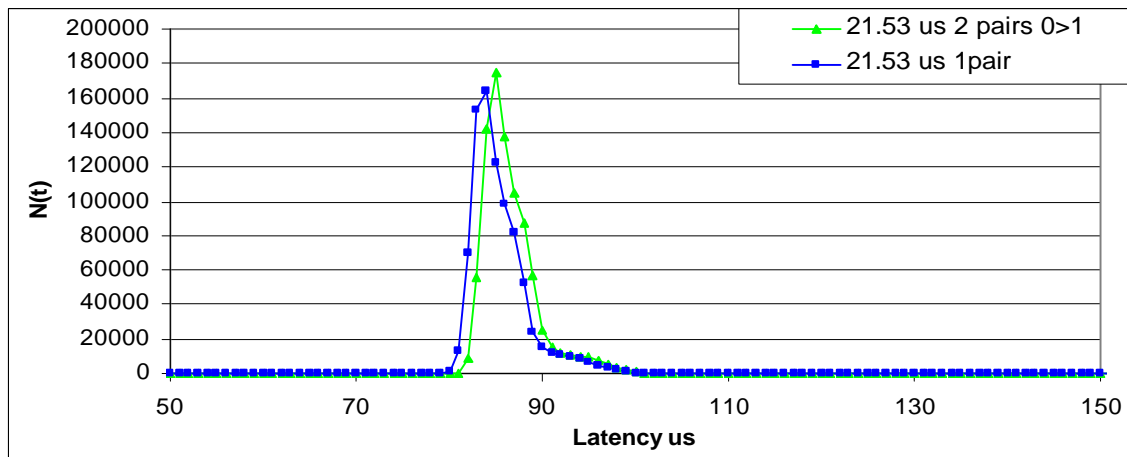


Figure 4.8 One-way Latency for data streaming using one and two pairs of ports of the Alteon switch.

4.1.5. BATM Switch with streaming between multiple pairs of nodes

Measurements of the one-way latency were also made with one, two, and three nodes streaming to the receiving node via the BATM Gigabit switch. Unicast frames were used and all the frames were sent at regular intervals. The intervals were increased in line with the number of sending nodes to enable the receiving node to process the frames and prevent infinite queues from building up. The latency was measured between the same pair of nodes for all the tests. Figure 4.9 shows the probability of obtaining a given latency for three different frame sizes. There is a clear increase in the width of the latency distributions as the number of nodes sending data increases. The widths scale with frame size as shown in Figure 4.10. These data confirm the expected queuing of frames trying to enter the destination node.

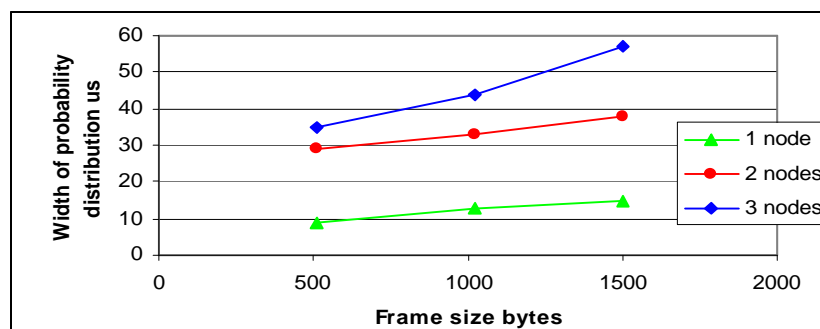


Figure 4.10 Comparison of widths of the Probability distributions for multiple streaming nodes.

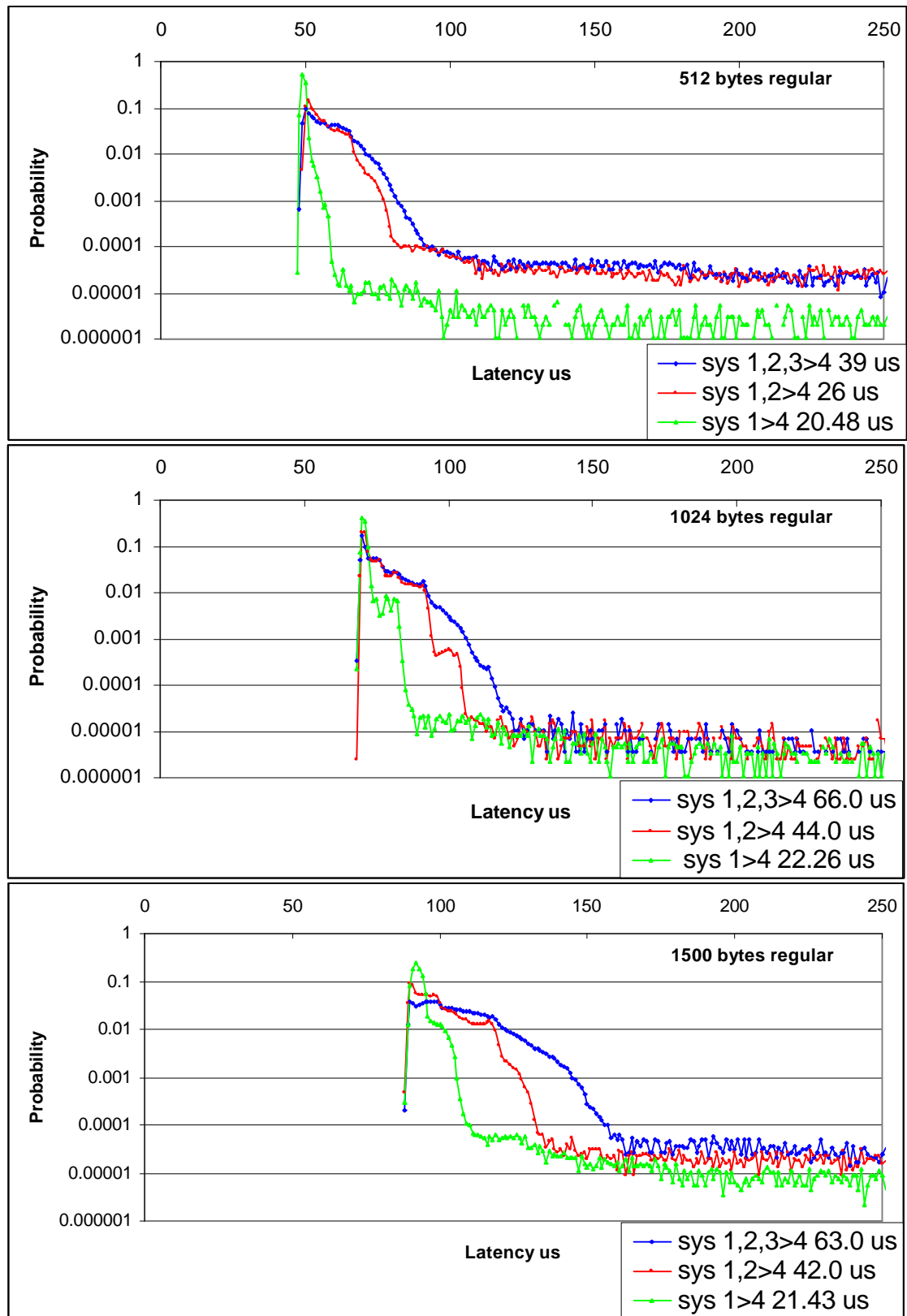


Figure 4.9 Plots of probability of the observed latency for 1, 2 and 3 nodes sending to the receiving node. The average time between generating packets is shown in the key for each case.

4.2. Throughput as a Function of Frame Size for Gigabit Ethernet

Two PCs were connected together with Gigabit Ethernet, one of the PCs was set to receive and the other to transmit. The transmitting PC sent frames as fast as it was capable and the receiving PC tried to receive all the transmitted frames. Measurements were made with the PC directly connected and connected with the Alteon or BATM switch. In all the plots shown, the Ethernet flow control (802.3x) was enabled therefore the rate at which the transmitter was able to send depended on the receiving rate of the receiver, assuming that the flow control worked perfectly throughout the system.

The transmitter also inserted a sequence number into the frame making it possible for frame losses to be detected on the receiver side. The transmit throughput was calculated locally by the transmitter and the receive throughput by the receiver. Calculating the number of frames per second observed during the time of the test and multiplying this by the message size gave the throughputs. The inter-frame time was calculated by dividing the time taken for the test by the number of frames recorded. Each point on the graphs was measured by transmitting at that message size for one second

Figure 4.11 shows the measured average inter-frame separation time as a function of the frame size. The results for direct connection were the same as going through the switch, showing that the switch had no effect on this measurement. In general, throughput is insensitive to the time frames queue in the system, provided no frames are lost and queues do not grow large enough to invoke any flow control to pause the flow of frames. The frame overheads remain constant but more data was transmitted with each frame as the frame size was increased. Thus, if there were no limitations, the curve should increase linearly as suggested by the open points which were calculated using the PCI and Gigabit Ethernet transfer rates and the IPG; the calculations were normalised to the measurements using the 575 byte point.

What was actually observed was that the average inter-frame time remained flat at 12.5 μ s until about 575 bytes, it then rose unevenly up to a message size of 1000 bytes, then finally curved at somewhat shorter times than expected to 1500 bytes frame size. The first part (0 to 575 bytes message size) is due to the fact that the combination of software, PCI and NIC in the PC cannot receive at a faster rate than 12.5 μ s. For message sizes from 575 to over 1000 bytes, the unusual curve is due to the firmware provided by Alteon. On previous versions of the firmware, the curve was a different shape. Above 1000 bytes, it appears that the interface provides some form of overlapped transfer designed to improve throughput.

Figure 4.12 shows the data re-plotted to show the throughput as a function of frame size. One would expect to see a smooth rising curve given by the frame size divided by the average inter-frame separation shown in Figure 4.11. The measured throughputs were 48.4 Mbytes/s with 1024 byte frames and 70 Mbytes/s with 1500 byte frames.

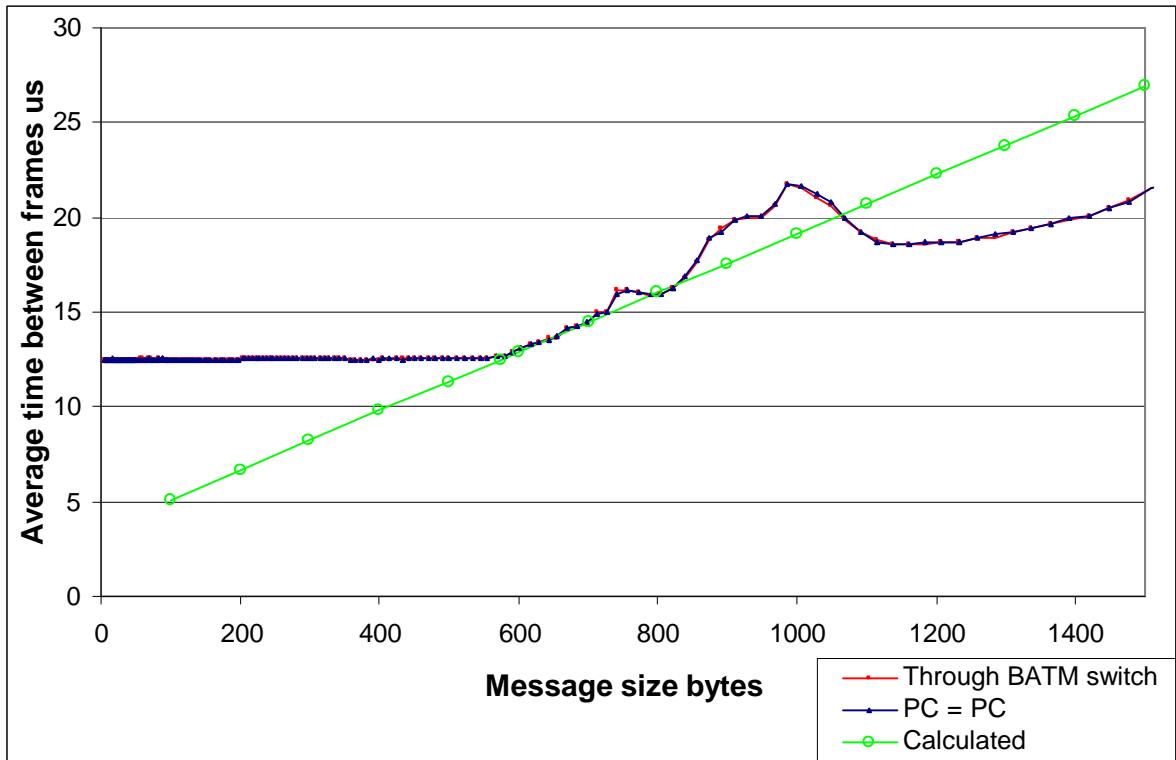


Figure 4.11 The average time between receiving frames obtained from unidirectional streaming using Gigabit Ethernet for two PCs directly connected and connected through the BATM switch.

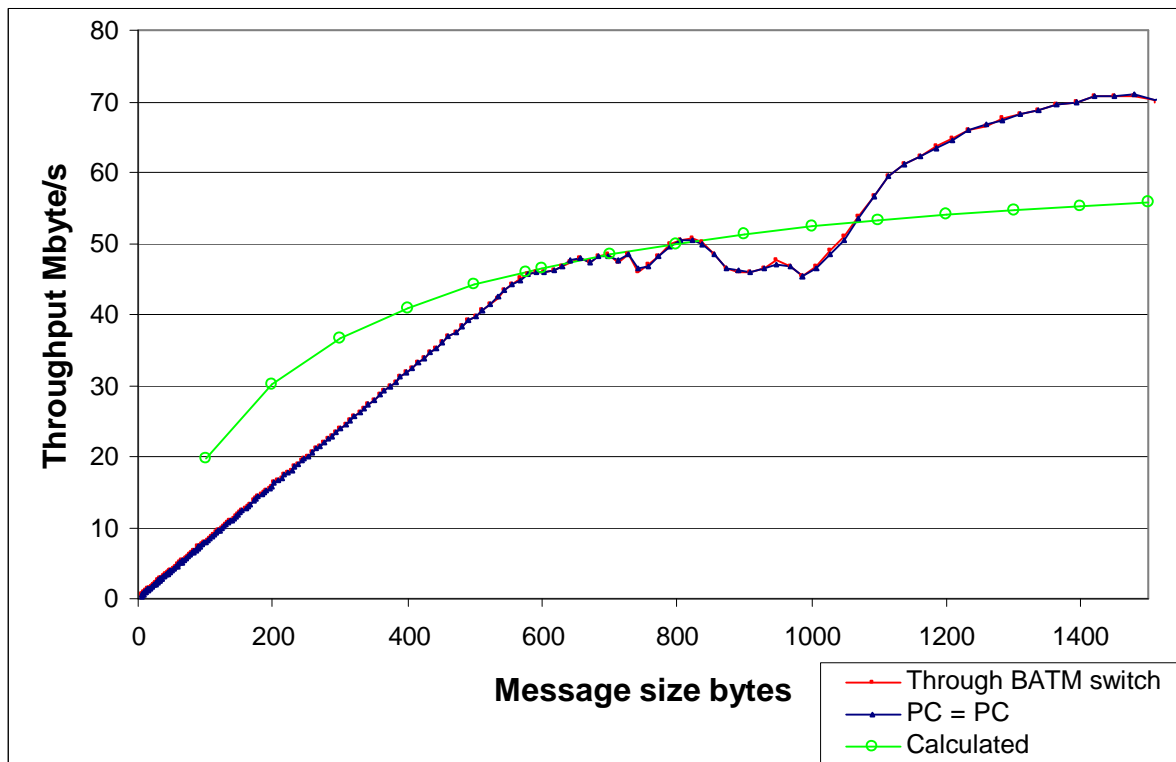


Figure 4.12 Throughput obtained from unidirectional streaming using Gigabit Ethernet for two PCs directly connected and connected through the BATM switch.

4.3. Throughput as a Function of Frame Size for 100Mbit Ethernet

Streaming measurements were also made using the PCs connected with 100 Mbit Ethernet. The curves in Figure 4.13 show the measured average inter-frame separation time as a function of the frame size for received frames. Data are shown for the two PCs connected via the same BATM switch module, via different modules in the same switch, and passing via two switches connected in cascade using a 100 Mbit Ethernet link. Ethernet flow control (802.3x) was enabled for all the interfaces involved in the tests. The results are equivalent, and data for a direct connection were the same as those presented, showing that the average inter-frame time is insensitive to switch topology, provided there is no frame loss and the rates are such that long queues of frames do not invoke the Ethernet flow control.

After a frame size of ~80 bytes, all the curves increase linearly, as expected from the discussion in Section 4.2, and the equation of the curve from 80 to 1500 bytes is

$$T = 4.088 + 0.079 * b \text{ } \mu\text{s}$$

The slope of 0.079 bytes/ μs is consistent with the transfer rate of 100 Mbit Ethernet; the PCI transfer rate is not included as the Intel 100 Mbit Ethernet interfaces overlap the PCI and Ethernet transfers (see Section 3.2.1). The corresponding throughput plots are shown in Figure 4.14 and show a total throughput of 12 Mbytes/s for frames over 900 bytes long, effectively this is 100 % utilisation of the 100 Mbit Ethernet.

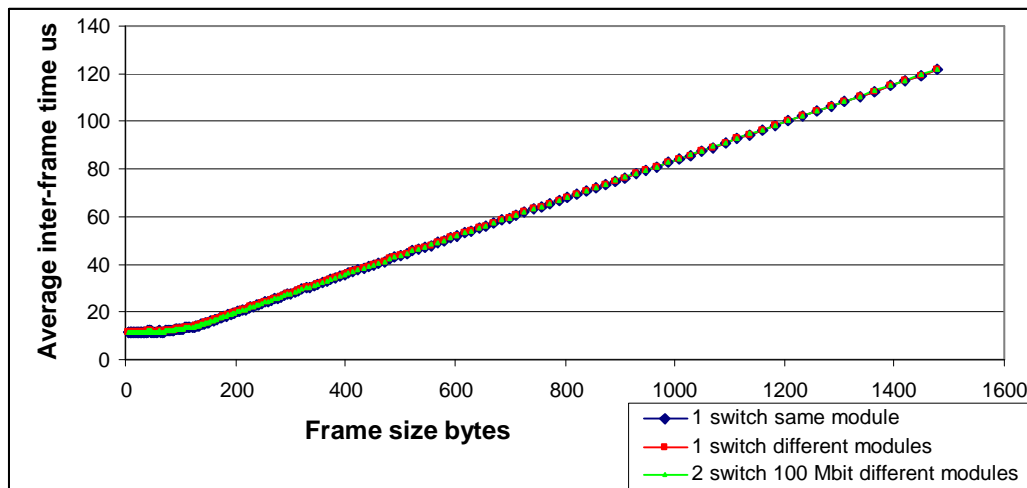


Figure 4.13 Average inter-frame time as a function of message size for frames streaming between two nodes connected with 100 Mbit Ethernet via the BATM switch.

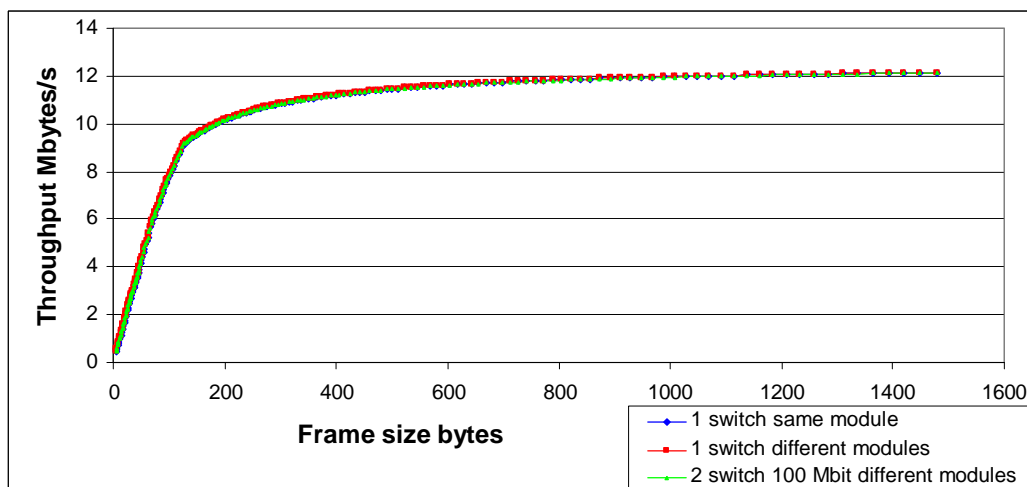


Figure 4.14 Throughput obtained by streaming frames between two PCs connected with 100 Mbit Ethernet using the BATM switch.

4.4. Throughput as a Function of Frame Size for data streaming from 100Mbit to Gigabit Ethernet

4.4.1. Streaming on One 100Mbit Link

These test used two PCs connected via the BATM switch, one PC used a 100 Mbit Ethernet connection while the second PC used a Gigabit connection. Frames were streamed from the 100 Mbit Ethernet PC to the Gigabit Ethernet PC and then vice versa to see if there were any differences. Both sets of data are plotted in Figure 4.15. As can be seen from these results, the throughput obtained is the same whether we are going from Gigabit to 100 Mbit Ethernet or vice versa. In fact, this result is the same when streaming from 100 Mbit Ethernet port to 100 Mbit Ethernet port, as is expected because the 100 Mbit Ethernet port is the slowest component in the test.

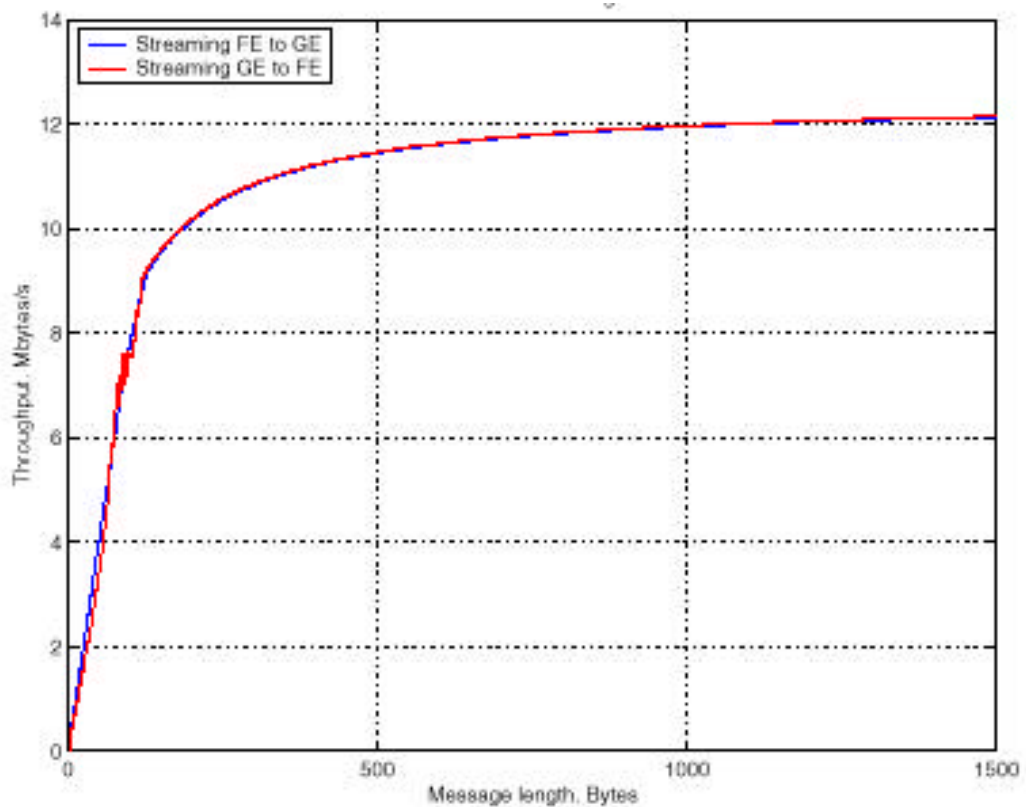


Figure 4.15 Throughput achieved when streaming from 100Mbit to Gigabit Ethernet via the BATM switch.

4.5. Throughput for streaming multiple 100Mbit to Gigabit Ethernet

Table 4.2 shows the total data throughputs measured when 1, 6 and 8 Fast Ethernet nodes were set to stream into a single Gigabit node via the BATM switch. When 6 nodes were used, the throughput was limited by the ability of the Gigabit link and interface to receive the data. The maximum throughputs measured as a function of the frame size, are in agreement with those discussed in Section 4.2 for a Gigabit Ethernet link between two PCs. When 1500 byte frames were used, a throughput of 12.16 Mbytes/s was measured from each of the 100 Mbit nodes, making up the total 72.93 Mbytes/s. 12.16 Mbytes/s is the maximum rate for a 100 Mbit link, and in this case, the Gigabit link was able to sink all the data.

When 8 nodes were used, there was a small drop in throughput for larger frames, which could be due to the flow control mechanism. For 1500 byte frames, the throughput fell to 8.95 Mbytes/s for each 100 Mbit node, and clearly the Gigabit link was saturated.

Frame size bytes	Total throughput Mbytes/s		
	1 fe->gigabit	6 fe->gigabit	8 fe->gigabit
512	11.5	40.706	40.709
1024	12	47.125	47.05
1500	12.1	72.93	71.632

Table 4.2 Measured throughput for 1, 6 and 8 nodes using 100 Mbit Ethernet to stream data to one node connected to the switch with Gigabit Ethernet.

4.6. Frame loss on mixed 100Mbit and Gigabit Ethernet Networks

No frame loss was observed when streaming from a node on Fast Ethernet to a node on Gigabit Ethernet. This was also the case in the opposite direction unless the switch had not learned the MAC address of the destination node. This occurred if the destination node had not sent any frames. The switch would be unaware of the location of the MAC address of the destination node and frames addressed to this node were broadcast to all ports of the switch. Since the mechanism for doing this is different than for switching unicast traffic to known ports, it is possible for frames to get lost in one case and not the other.

Figure 4.13 shows the received throughput compared to the send throughput in the case where the switch had not learned the MAC address of the destination node. The loss throughput is the difference between the transmit and receive throughput.

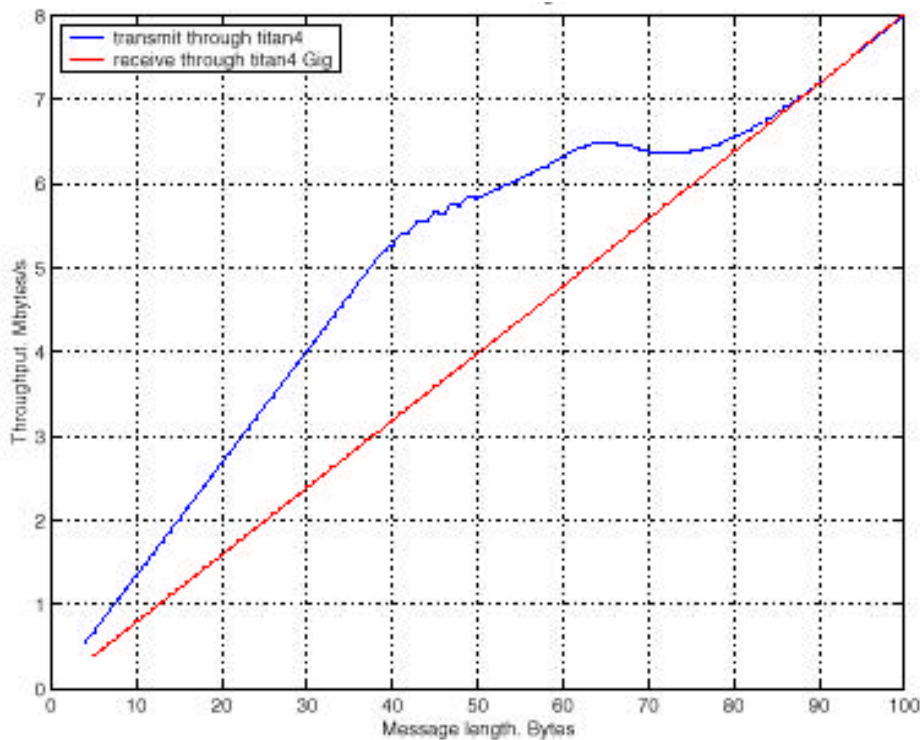


Figure 4.13 Frame transmit rate and receive rate as a function of the frame size obtained from unidirectional streaming using Gigabit Ethernet for two PCs connected through the BATM switch. The excess of the transmitted frames over those received indicates frame loss.

5. Investigations with Multicast Frames

This investigation examined the behaviour of the interfaces and switches when multicast Ethernet frames were generated. It was not concerned with IP multicast traffic that exchanges data between sets of nodes using a common Class D IP address. Multicast Ethernet frames have the first bit of the destination MAC address asserted, the rest of the MAC address is used to define a set of nodes wishing to receive and process this information. For example 01-xx-xx-xx-xx-xx (hex) would be a multicast frame. Broadcast frames are one instance of multicast frames where the destination address is FF-FF-FF-FF-FF-FF (hex), and by definition, all nodes must receive and process this information.

5.1.1. Tests using Broadcast Frames

Tests were made using the 400 MHz PCs and the Alteon Gigabit Ethernet interface cards with an arrangement similar to that described in Section 3, but using broadcast frames (a destination address of FF-FF-FF-FF-FF-FF) for the Request, the Response and the Ping-Pong. The middle curve on Figure 5.1 shows the Request-Response latency as a function of the response frame size for two directly connected PCs. The curve is linear but not quite as smooth as that obtained with unicast frames, shown in Figure 3.2. Allowing for the 64 byte request, the equation describing the one way propagation delay T as a function of the frame size b bytes is:

$$T(b) = 26.47 + 0.0235 * b \mu s$$

which agrees with that for unicast frames given in Section 3.1.1. The upper curve in Figure 5.1 shows the Request-Response latency when the frames traverse the BATM switch with just two ports connected. The Latency is constant at 90 μs up to a frame length of 600 bytes and then increases at 0.041 μs /byte. The corresponding one-way latencies are 45 μs up to a frame length of 600 bytes and then

$$T(b) = 20.91 + 0.0407 * b \mu s$$

Almost identical results were obtained when the other two ports on the switch were connected to two other PCs set to receive and sink any frames sent to them. In this case the switch had to duplicate or fan out the frames to three outgoing ports on the switch.

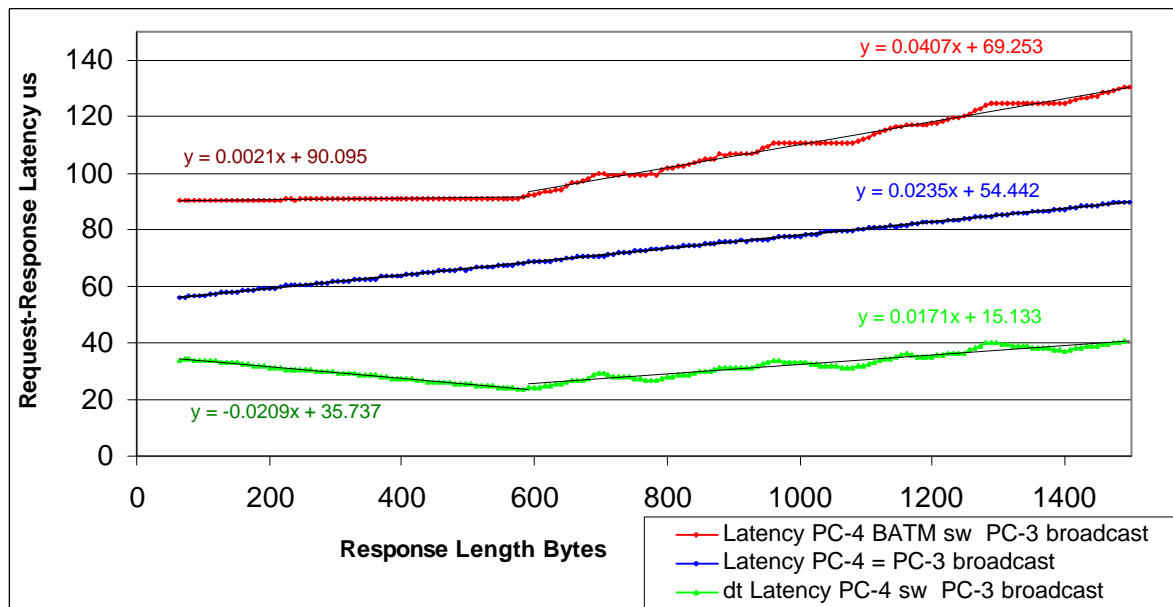


Figure 5.1 The Request-Response Latency as a function of Response Length using Broadcast frames for two PCs directly connected is shown in the middle curve; the top curve was measured when the PCs were connected using the BATM Gigabit Ethernet switch. The bottom curve shows the contribution from the switch.

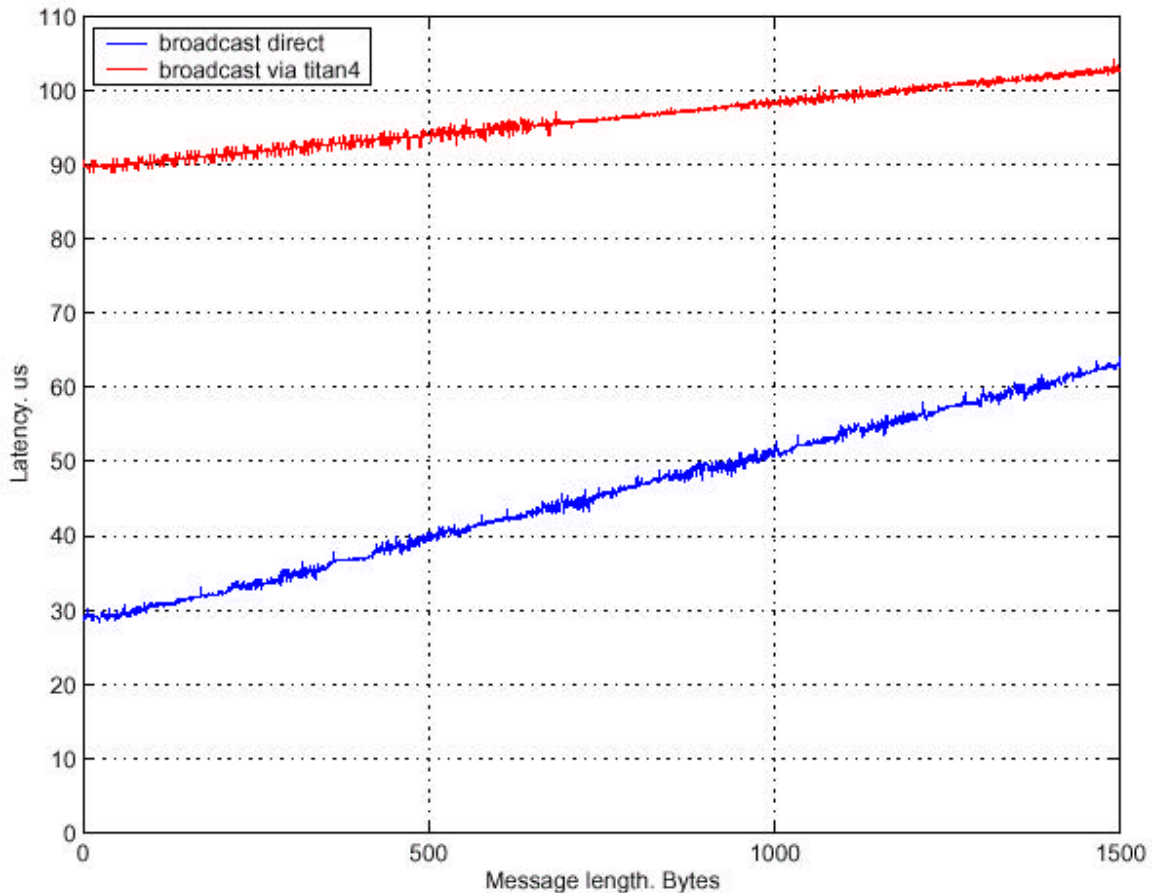


Figure 5.2. One way latency obtained from Ping-Pong measurements using broadcast frames. The lower curve has a slope of 0.024 ms/byte and is for directly connected PCs, The upper curve has a slope of 0.009 ms/byte and is for frames from PC to PC traversing the BATM switch.

Figure 5.2 shown the latencies obtained using Ping-Pong measurements. Both sets of data agree on the behaviour for directly connected PCs, but the Request-Response data indicates that broadcast traffic requires an extra $\sim 10 \mu\text{s}$ when passing via the switch, whereas the Ping-pong data requires $\sim 60 \mu\text{s}$.

The constant latency of Figure 5.1 (and the fact that the slope for the Ping-Pong measurements is less for frames crossing the switch that for directly connected PCs.) suggests that either the switch is moving a minimum frame length of ~ 600 bytes for multicast traffic, or more likely, that processing of the multicast frame within the switch and duplicating or moving it to each of the outputs is overlapped with the time the frame takes to enter the switch.

5.1.2. Tests using Multicast Frames

Request-Response latency measurements were made using multicast frames by programming the Alteon interfaces to accept a multicast address of 01-00-C0-A0-F0-E0 and sending all frames to that destination. The middle curve on Figure 5.3 shows the Request-Response latency as a function of the response frame size for two directly connected PCs. Allowing for the 64 byte request, the equation describing the one way propagation delay T as a function of the frame size b bytes is:

$$T(b) = 27.94 + 0.0235 * b \mu\text{s}$$

This, and the results when the switch was included are very similar to those discussed in the previous section where broadcast frames were used.

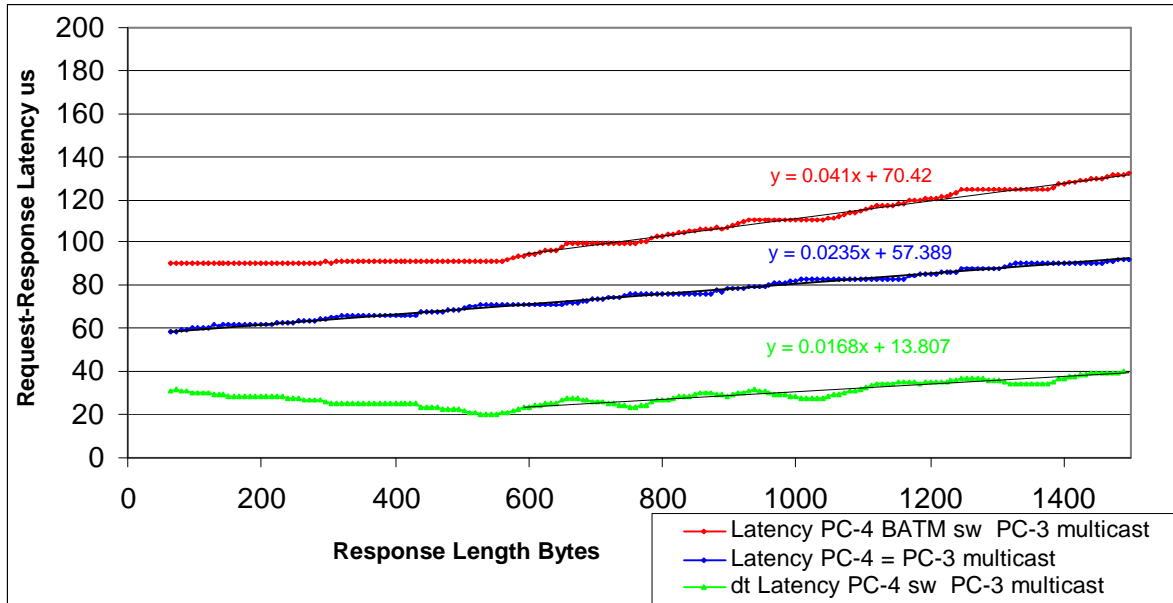


Figure 5.3 Request-Response Latency as a function of Response length using multicast frames. The middle curve shows the case for two PCs directly connected; the top curve was measured when the PCs were connected using the BATM Gigabit Ethernet switch. The bottom curve shows the contribution from the switch.

However, when two other PCs were connected to the switch and set to receive the multicast traffic, the tests did not run smoothly and very long maximum round trip times, ~2.5 s were periodically recorded, which distorts the mean as shown in Figure 5.4. This, and the discrepancy between the Latency measurements mentioned in Section 5.1.1 indicate that further work is required.

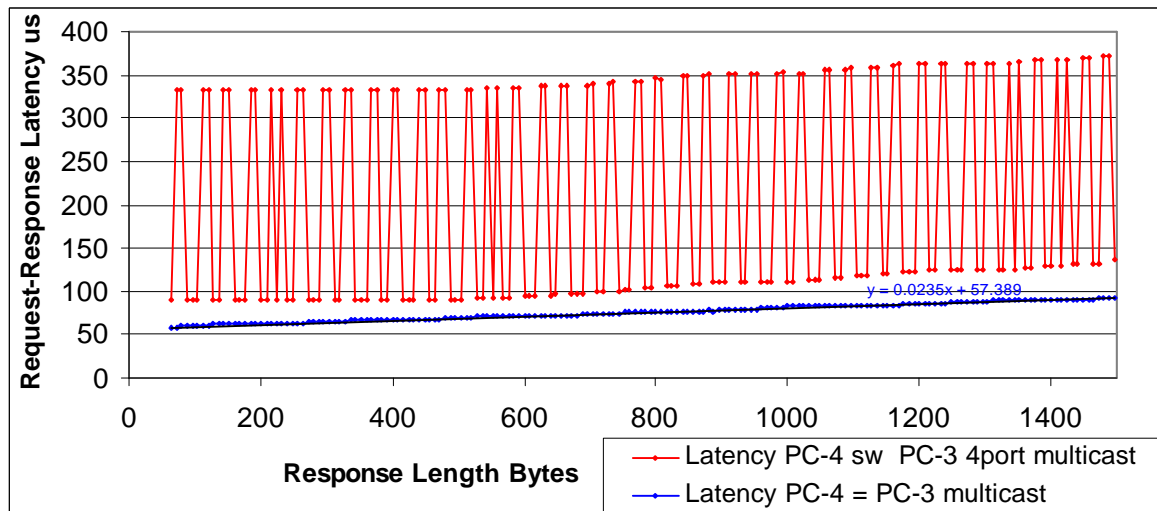


Figure 5.4 Request-Response Latency as a function of Response length using multicast frames with all 4 ports of the switch connected.

5.1.3. Tests using Multicast Request with Unicast Response

The four PC were connected to the BATM switch with multicast frame reception enabled. One PC was set to send a Request to the multicast destination 01-00-C0-A0-F0-E0, upon reception of the request, the three other PCs sent unicast replies tagged with their node number. The Request-Response latency was histogrammed separately for each of the three nodes. Histograms of these latencies are shown in Figure 5.5 for 64, 1024 and 1472 byte responses. The plots show that PC ID1 tends to reply first, followed by ID2 and then ID3. The time difference between the peaks changes only slowly with response length as indicated in Table 5.1 These differences in the round-trip times for the three PCs have contributions from the following:

- time to fan out the multicast request in the switch
- the time for the individual PCs to respond
- the time taken by any queuing of the response frames entering the requesting node.

The difference in the time the PCs take to respond to the request as one operated at 400 MHz and the other two (ID2 and ID3) at 350 MHz is estimated to be $< 1\mu\text{s}$, and is not thought to be important here. In all cases the minimum allowed time between packets on Gigabit Ethernet is less than the spacing of the peaks – especially for 64 bytes. The time for interface to place frame in memory is based on the information from the PCI traces of Figure 3.1 and includes the time to bust over PCI and the time to update the CSRs to inform the software that there is a new frame present. Inspection of Table 5.1 suggests that the time to fan-out a multicast frame in the switch is $\sim 14\mu\text{s}$, but we note that this is excessive compared to the measured backplane transfer rate of $0.0099\mu\text{s}/\text{byte}$ which would transfer a 64 byte frame in $\sim 0.7\mu\text{s}$. For the 1500 byte responses, one would expect that the second and third response would each have to queue to enter the requesting PC.

Frame length	Min. separation time on Gigabit Ethernet	Time for interface to place frame in memory	Separation of peaks (average)
64	0.753	6	14.5
1024	8.43	13.7	15.5
1500	12.016	17.3	19

Table 5.1 Time to queue at different points of the system. All times inms.

4 Apr 00

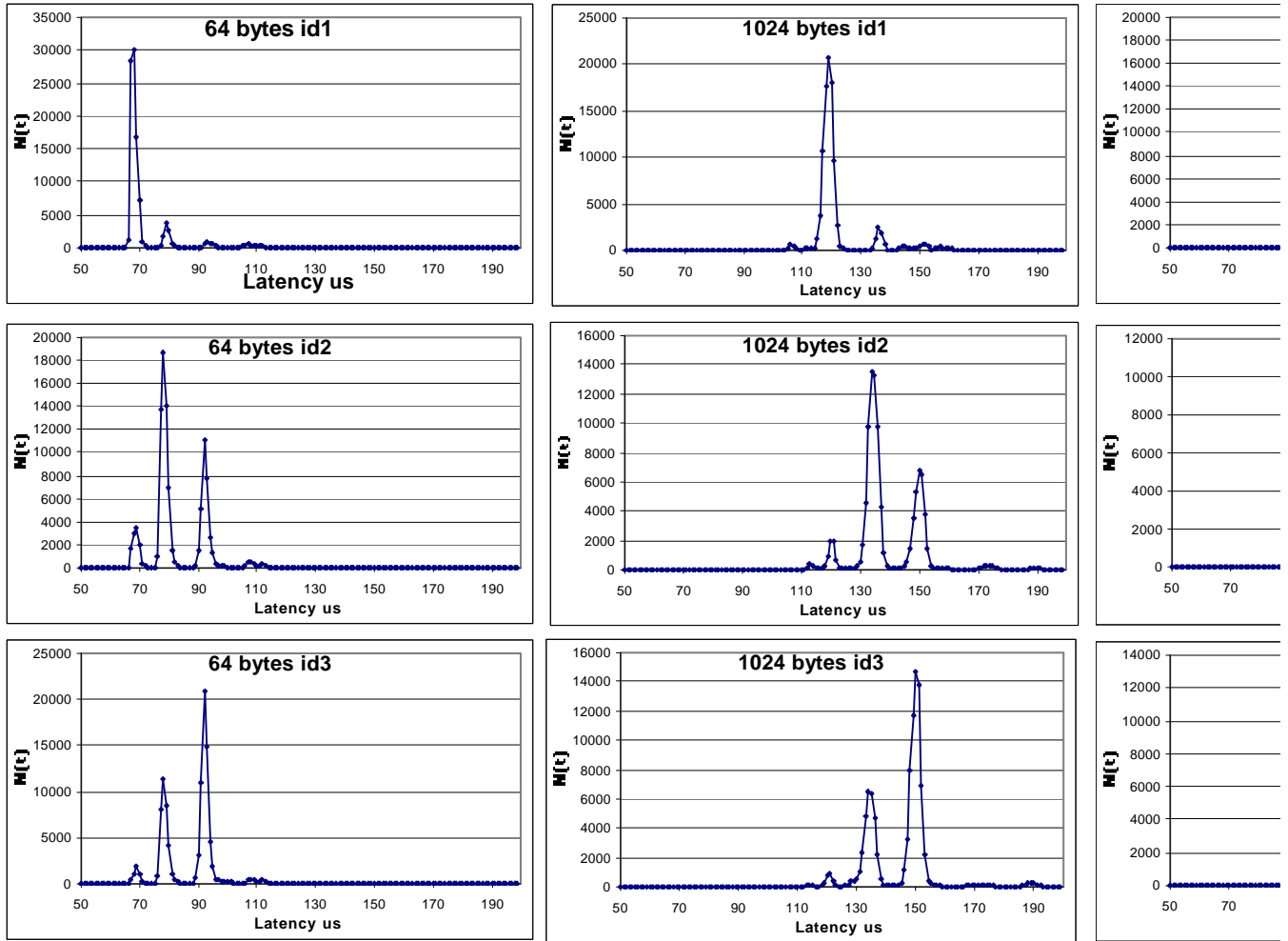


Figure 5.5 Histograms of the Request-Response latency from the different PCs repnding to a multicast request.

6. References

- [1] For information on the Alteon 180 Gigabit switch see the Alteon web site <http://www.alteon.com> and support at <http://www.alteon.com/support/index.phtml/>
- [2] For information on the BATM Titon 4 Ethernet switch see the BATM web site <http://www.batm.com>
- [3] R. Hughes-Jones Testing 100 Mbit Ethernet Network Interfaces and 3com 3300 Switches MAN/HEP/99-2 Jun 99
R. Hughes-Jones Gigabit Ethernet Tests using the Cisco 6500 Switch MAN/HEP/99-4 Aug 99
- [4] F. Saka "TCP/IP Measurements on the Turboswitch 2000" CERN Group Report see <http://home.cern.ch/~fsaka/>
- [5] IEEE 802 standard on Ethernet frame format
- [6] M.Boosten et al, "MESH Messaging and Scheduling for Fine Grain Parallel Processing on Commodity Platforms" Architectures, Languages and Techniques. 1999. IOS Press. p263-276. Edited by B.M. Cook.
M. Boosten, R.W. Dobinson, P.D.V. van der Stok "High Bandwidth Concurrent Processing on Commodity Platforms" June 1999. IEEE Real-Time 99, Santa Fe, U.S.A.
And also <http://home.cern.ch/~mboosten/>
- [7] For specifications and details of the Galileo chips see the Galileo web page www.galileot.com
- [8] For a full description of the clock time synchronisation and measurement software see .
F. Saka "The measurement software and clock synchronisation" <http://home.cern.ch/~fsaka/>
- [9] Specification of Ptolemy simulation of Gigabit Ethernet
http://hep.man.ac.uk/~rich/ptolemy/network_simulation_v06.pdf