

Performance Measurements on Gigabit Ethernet NICs and Server Quality Motherboards

R. Hughes-Jones, S. Dallison, G. Fairey

*Dept. of Physics and Astronomy, University of Manchester, Oxford Rd., Manchester
M13 9PL, UK*

P. Clarke, I. Bridge

University College London, Gower St., London, WC1E 6BT

Abstract

The behaviour of various Gigabit Ethernet NIC's and server quality motherboards has been investigated. The latency, throughput, and the activity on the PCI buses and gigabit Ethernet links were chosen as performance indicators. The tests were performed using two PCs connected back-to-back and sending UDP/IP frames from one to the other. This paper shows the importance of the NIC-PCI memory bus chipset combination, CPU power, good driver / operating system design and configuration to achieve and sustain Gigabit transfer rates. With these considerations taken into account, and suitably designed hardware, transfers can operate at Gigabit speeds. Some recommendations are given for potential high performance data servers.

1 Introduction

With the increased availability and decrease in cost of Gigabit Ethernet using Cat-5 twisted pair cabling, system suppliers and integrators are offering Gigabit Ethernet and associated switches as the preferred interconnect between disk servers and PC compute farms as well as the most common campus or departmental backbone. With the excellent publicity that 'Gigabit Ethernet is just Ethernet', users and administrators are now expecting Gigabit performance just by purchasing Gigabit components.

In order to be able to perform sustained data transfers over the network at Gigabit speeds, it is essential to study the behaviour and performance of the end-system compute platform and the network interface cards (NIC). In general, data must be transferred between *system memory* and the *interface* and then placed on the network. For this reason, the operation and interactions of the *memory*, *CPU*, *memory-bus*, the *bridge to the input-output bus* (often referred to as the "chipset"), the *network interface card* and the *input-output bus* itself (in this case PCI / PCI-X) are of great importance. The design and implementation of the software drivers, protocol stack, and operating system are also vital to good performance. For most data transfers, the information must be read from and stored on permanent storage, thus the performance of the storages sub-systems and this interaction with the computer buses is equally important.

The work reported here forms part of an ongoing evaluation programme for systems and Gigabit Ethernet components for the Trigger/DAQ in the ATLAS experiment at the Large Hadron Collider being built in Geneva, as well for high performance networking for Grid computing.

The report first describes the equipment used and gives details and methodology of the tests performed. The results, analysis and discussion of the tests performed are then presented in sections one for each motherboard used. Within each of these sections, sub-sections describe the results of each Network Interface Card, NIC, which was tested.

2 Hardware and Operating Systems

Four motherboards were tested:

- ? SuperMicro 370DLE
 - o Chipset: ServerWorks III LE Chipset
 - o CPU: PIII 800 MHz
 - o PCI: 32/64 bit 33/66 MHz
- ? IBM das motherboards from the das compute farm
 - o Chipset: ServerWorks CNB20LE
 - o CPU: Dual PIII 1GHz
 - o PCI: 64 bit 33 MHz
- ? SuperMicro P4DP6
 - o Chipset: Intel E7500 (Plumas)
 - o CPU: Dual Xeon Prestonia (2cpu/die)¹ 2.2 GHz
 - o PCI: 64 bit, 66 MHz
- ? SuperMicro P4DP8-G2 with dual Gigabit Ethernet onboard
 - o Chipset: Intel E7500 (Plumas)
 - o CPU: Dual Xeon Prestonia (2cpu/die) 2.2 GHz
 - o PCI: 64 bit, 66 MHz

The SuperMicro [1] P4DP6 motherboard has a 400 MHz front-side bus and 4 independent 64bit PCI / PCI-X buses selectable speeds of 66, 100 or 133 MHz as shown in the block diagram of Figure 2.1; also the “slow” devices and the EIDE controllers are connected via another bus. The P4DP8-G2 is similar with the on-board 10/100 LAN controller replaced by the Intel 82546EB dual port Gigabit Ethernet controller connected to PCI Slot 5.

Both Copper and Fibre NICs were tested and are listed in Figure 2.2 together with the chipset and the Linux driver version.

As most of the systems used in the Grid projects such as the EU DataGrid and DataTAG are currently based on Linux PCs, this platform was selected for making the evaluations. RedHat Linux v 7.1 and v 7.2 were used with the 2.4.14 and 2.4.19 kernels. However, the results are also applicable to other platforms. Care was taken to obtain the latest drivers for the Linux kernel in use. No modifications were made to the IP stacks nor the drivers. Drivers were loaded with the default parameters except were stated.

¹ The Xeon processor has 2 “logical” processors each with it own independent machine states, data registers, control registers and interrupt controller (APIC). However they share the core resources including execution engine, system bus interface and level 2 cache. This hyper-threading technology allows the core to execute two or more separate code streams concurrently, using out-of-order instruction scheduling to maximise the use of the execution units[8].

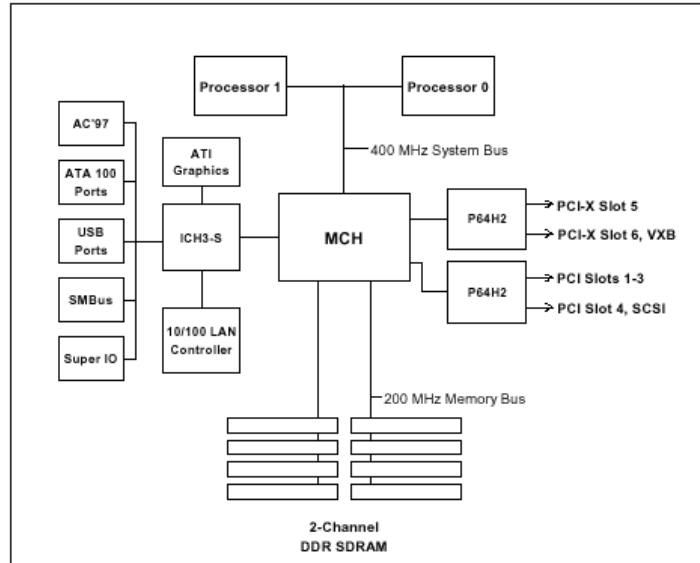


Figure 2.1 Block diagram of the SuperMicro P4DP6 / P4DP8 motherboard.

Manufacturer	Model	Chipset	Linux	Driver version
Alteon	ACENIC	Tigon II rev 6	2.4.14	acenic v0.83 Firmware 12.4.11
NetGear	GA-620	Tigon II	2.4.14	acenic v0.83 Firmware 12.4.11
SysKonnct	SK9843		2.4.14	sk98lin v4.06
			2.4.19-SMP	sk98lin v6.0 β 04
Intel	PRO/1000 XT	82544	2.4.14	e1000
			2.4.19-SMP	e1000 4.4.12
Intel	On board	82546 EB	2.4.19-SMP	e1000 4.4.12

Figure 2.2 A table of the NICs used in the tests.

3 Methodology and Tests Performed

For each combination of motherboard, NIC and Linux Kernel, three sets of measurements were made using two PCs with the NICs directly connected together with suitable fibre or copper crossover cables. In most cases, identical system configurations were used for both PCs – any variations to this are noted in the test results. UDP/IP frames were chosen for the tests as they are processed in a similar manner to TCP/IP frames, but are not subject to the flow control and congestion avoidance algorithms defined in the TCP protocol and thus do not distort the base-level performance. The standard IP stack that came with the relevant Linux kernel was used for the tests. The packet lengths given are those of the user payload.

Recently other tests on Gigabit Ethernet NICs have been reported [8], [9]. These have used TCP/IP and long message sizes up to 1 Gbytes for the memory to memory transfers. It is clear that these results will include the effects of the TCP protocol, its implementation as well as the performance of the NICs and Motherboards.

The following measurements were made:

3.1 Latency

Round trip times were measured using Request-Response UDP frames, one system sent a UDP packet requesting that a Response of the required length be sent back by the remote end. The un-used portions of the UDP packets were filled with random values to prevent any data compression. Each test involved measuring many (~1000) Request-Response singletons. The individual Request-Response times were measured by using the CPU cycle counter on the Pentium [2], and the minimum, average and maximum times were computed. This approach is in agreement with the recommendations of the IPPM and the GGF [3].

The round-trip latency was measured and plotted as a function of the frame size of the response. The slope of this graph is given by the sum of the inverse data transfer rates for each step of the end-to-end path [4]. For example, for two back-to-back PCs with the NICs operating as store and forward devices, this would include:

Memory copy + PCI transfer + Gig Ethernet + PCI transfer + memory copy

The following table gives the slopes expected for PCI – Ethernet – PCI transfer given the different PCI bus widths and speeds:

Transfer Element	Inverse data transfer rate $\mu\text{s}/\text{byte}$	Expected slope $\mu\text{s}/\text{byte}$
32 bit 33 MHz PCI	0.0075	
64 bit 33 MHz PCI	0.00375	
64 bit 64 MHz PCI	0.00188	
Gigabit Ethernet	0.008	
32 bit 33 MHz PCI with Gigabit Ethernet		0.023
64 bit 64 MHz PCI with Gigabit Ethernet		0.0155
64 bit 64 MHz PCI with Gigabit Ethernet		0.0118

The intercept gives the sum of the propagation delays in the hardware components and the end system processing times.

Histograms were also made of the singleton Request-Response measurements. These histograms will show any variations the round-trip latencies, some of which may be caused by other activity in the PCs. In general for the Gigabit Ethernet systems under test, the Full Width Half Minimum of the latency distribution was $\sim 2 \mu\text{s}$ with very few events in the tail, see Figure 6.4, indicating that the method used to measure the times was precise.

These latency tests also indicate any interrupt coalescence in the NICs and provide information on protocol stack performance and buffer management.

3.2 UDP Throughput

The UDPmon [5] tool was used to transmit streams of UDP packets at regular, carefully controlled intervals and the throughput was measured at the receiver. Figure 3.1 shows the messages and data exchanged by UDPmon and Figure 3.2 shows a

network view of the stream of UDP packets. On an unloaded network UDPmon will measure the Capacity of the link with the smallest bandwidth on the path between the two end systems. On a loaded network the tool gives an estimate of the Available bandwidth [3], these being indicated by the flat portions of the curves.

In these tests a series of user payloads from 64 to 1472 bytes were selected and for each packet size, the frame transmit spacing was varied. For each point the following information was recorded:

- ? The time to send and the time to receive the frames
- ? The number packets received, the number lost, number out of order
- ? The distribution of the lost packets
- ? The received inter-packet spacing
- ? CPU load and Number of interrupts for both transmitting and receiving system

The “wire” throughput rates include an extra 60 bytes of overhead² and were plotted as a function of the frame transmit spacing. On the right hand side of the plots, the curves show a 1/t behaviour, where the delay between sending successive packets is most important. When the frame transmit spacing is such that the data rate would be greater than the available bandwidth, one would expect the curves to be flat (often observed to be the case). As the packet size is reduced, processing and transfer overheads become more important and this decreases the achievable data transfer rate.

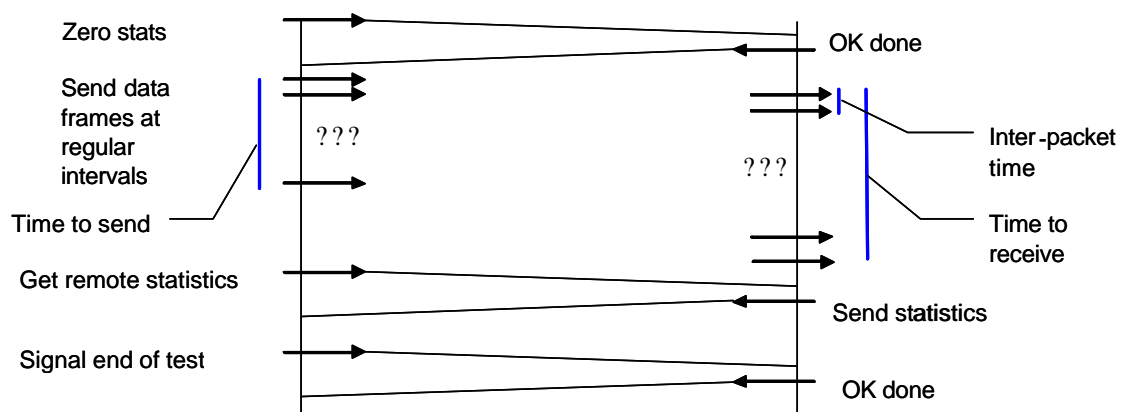


Figure 3.1 Top: The messages exchanged by UDPmon

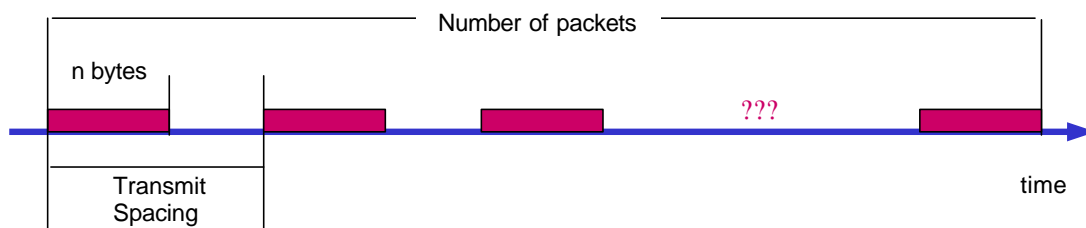


Figure 3.2 The network view of the spaced UDP frames

² The 60 “wire” overhead bytes include: 8 bytes for inter-packet gap, 6 bytes for the preamble, 18 bytes for Ethernet frame header and CRC and 28 bytes of IP and UDP headers.

3.3 Activity on the PCI Buses and the Gigabit Ethernet Link

The activity on the PCI bus of sender and receiver nodes and the passage of the Gigabit Ethernet frames on the fibre was measured using a Logic analyser and a specialized Gigabit Ethernet Probe as shown in Figure 3.3.

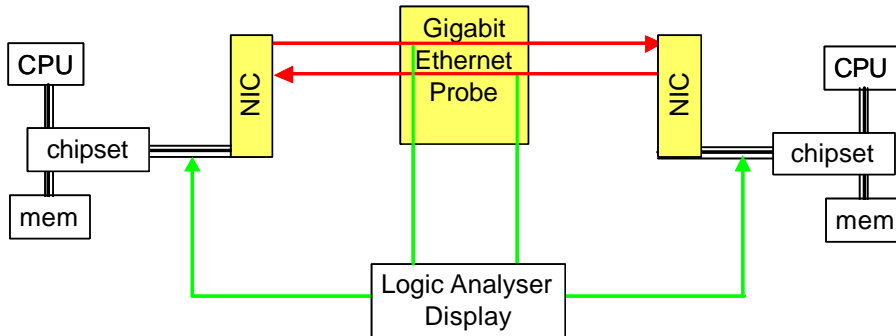


Figure 3.3 Test arrangement for examining the PCI buses and the Gigabit Ethernet link.

4 Measurements made on the SuperMicro 370DLE

The SuperMicro 370DLE motherboard is somewhat old now, but has the advantage of having both 32 bit and 64 bit PCI slots with 33 / 66 MHz jumper selectable. The board uses the ServerWorks III LE Chipset and one PIII 800 MHz CPU. RedHat v7.1 Linux was used with the 2.4.14 kernel for the tests.

4.1 SysKonnnect

4.1.1 UDP Request-Response Latency

The round trip latency shown in Figure 4.1 as a function of the packet size is a smooth function for both 33 MHz and 66 MHz PCI buses, indicating that the driver-NIC buffer management works well. The clear step increase in latency at 1500 bytes is due to the need to send a second partially filled packet, and the smaller slope for the second packet is due to the overlapping of the incoming second frame and the data from the first frame being transferred over the PCI bus. These two effects are common to all the Gigabit NICs tested.

The slope observed for the 32 bit PCI bus, 0.0286 is in reasonable agreement with that expected, the increase being consistent with some memory-memory copies at both ends. However the slope of 0.023 $\mu\text{s}/\text{byte}$ measured for the 64 bit PCI bus is much larger than the 0.0118 $\mu\text{s}/\text{byte}$ expected even allowing for memory copies. Measurements with the logic analyser, discussed in Section 4.1.3, confirmed that the PCI data transfers *did* operate at 64 bit 66 MHz with no wait states. It is possible that the extra time per byte is due to some movement of data within the hardware interface itself.

The small intercept of 62 or 56 μs , depending on bus speed, suggests that the NIC interrupts the CPU for each packet sent and received. This was confirmed by the throughput and PCI measurements.

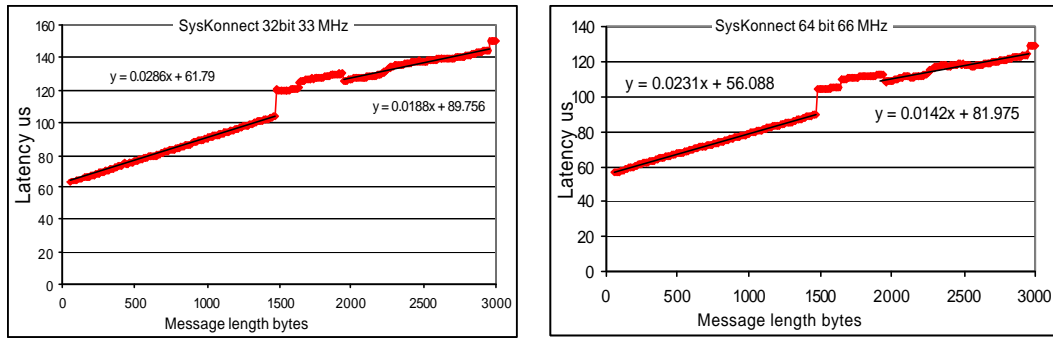


Figure 4.1 UDP Request-Response Latency as a function of the packet size for the SysKonnnect SK9843 NIC on 32 bit 33 MHz and 64 bit 66 MHz PCI buses of the Supermicro 370DLE.

4.1.2 UDP Throughput

The SysKonnnect NIC managed 584Mbit/s throughput for full 1500 byte frames on the 32 bit PCI bus (at a spacing of 20-21 μ s) and 720 Mbit/s (at a spacing of 17 μ s) on the 64 bit 66 MHz PCI bus (Figure 4.2). At inter-packet spacings less than these values there is packet loss, traced to loss in the receiving IP layer when no buffers for the higher layers are available. However, the receiving CPU was loaded at between 65 - 100% when the frames were spaced at 13 μ s or less so lack of CPU power could explain the packet loss.

4.1.3 PCI Activity

Figure 4.3 shows the signals on the sending and receiving PCI buses when frames with 1400 bytes of user data were sent. At time t the dark portions on the send PCI shows the setup of the control & status registers, CSRs, on the NIC card, which then moves the data over the PCI bus – indicated by the assertion of the PCI signals for a long period ($\sim 3 \mu$ s). The Ethernet frame is transferred over the Gigabit Ethernet as shown by the lower 2 signals at time X , to the receive PCI bus at time O , some 27 μ s later. The frame exists on the Ethernet medium for 11.6 μ s. The activity seen on the sending and receiving PCI buses after the data transfers is due to the driver updating the CSRs on the NIC after the frame has been sent or received. These actions are performed after the NIC has generated an interrupt and for the SysKonnnect card this happens for each frame.

Figure 4.3 also shows that the propagation delay of the frame from the 1st word on the sending PCI bus to the 1st word on the receiving PCI was 23 μ s. Including the times to setup the sending CSRs and the interrupt processing on the receiver, the total delay from the software sending the packet to receiving it was 36 μ s. Using the intercept of 56 μ s from the latency measurements one can estimate the IP stack and application processing times to be $\sim 10 \mu$ s on each 800 MHz CPU.

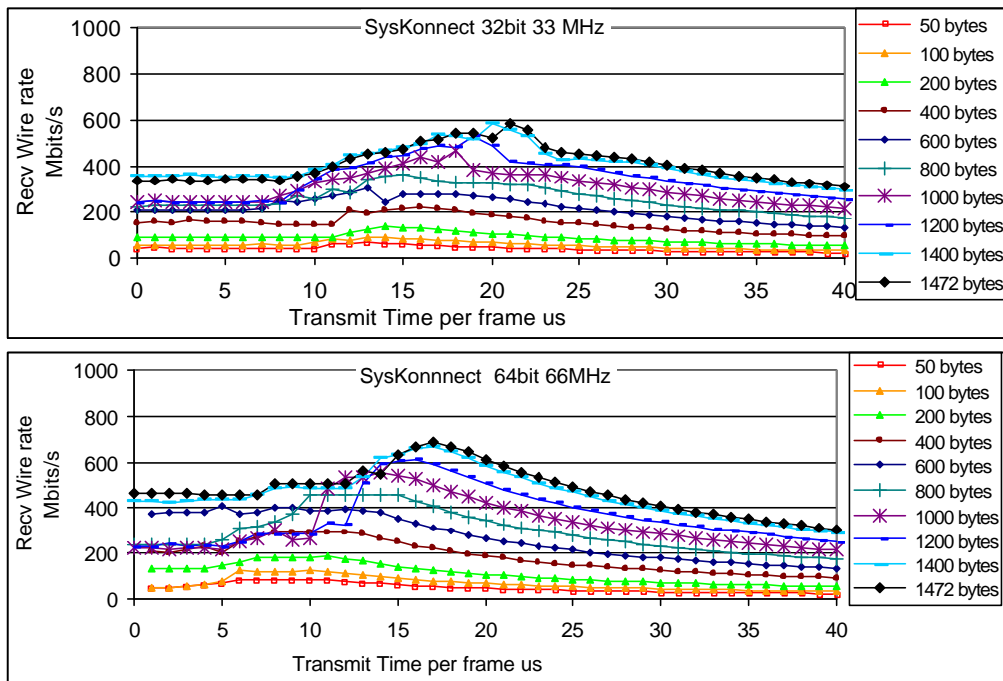


Figure 4.2 UDP Throughput as a function of the packet size for the SysKonnct SK9843 NIC on 32 bit 33 MHz and 64 bit 66 MHz PCI buses of the Supermicro 370DLE.

The upper plot in Figure 4.4 shows the same signals corresponding to packets being generated at a transmit spacing of 20 μ s and the lower plot with the transmit spacing set to 10 μ s. In this case, the packets are transmitted back-to-back on the wire with a spacing of 11.7 μ s, i.e. full wire speed. This shows that a 800MHz CPU is capable of transmitting large frames at Gigabit wire speed.

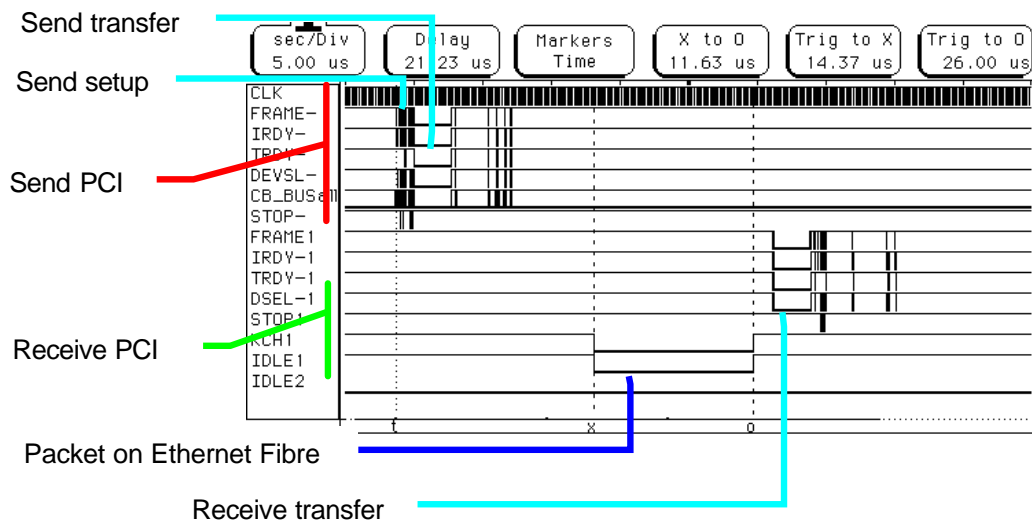


Figure 4.3 The traces of the signals on the send and receive PCI buses for the SysKonnct NIC on the 64 bit 66 MHz bus of the Supermicro 370DLE motherboard. The bottom 3 signals are from the Gigabit Ethernet Probe card and show the presence of the frame on the Gigabit Ethernet fibre.

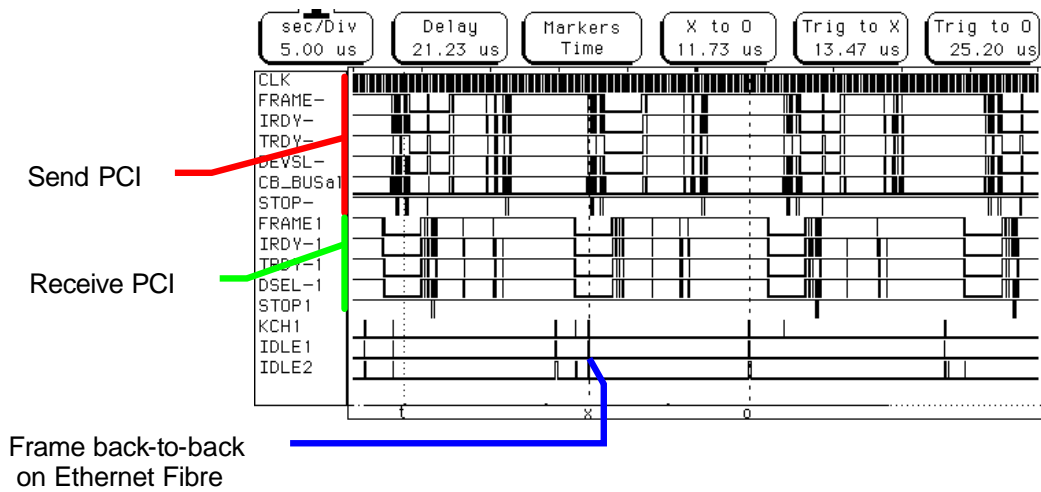
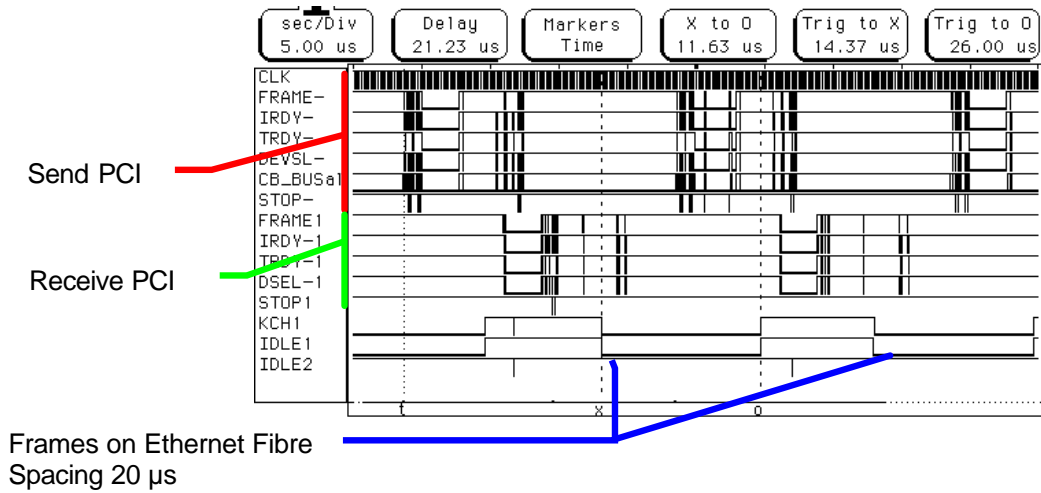


Figure 4.4 The traces of the signals on the send and receive PCI buses and the Gigabit Ethernet Probe card. Upper: signals corresponding to packets being generated at a transmit spacing of 20 μ s. Lower: plots with the transmit spacing set to 10 μ s.

4.2 Intel PRO/1000 XT

4.2.1 UDP Request-Response Latency

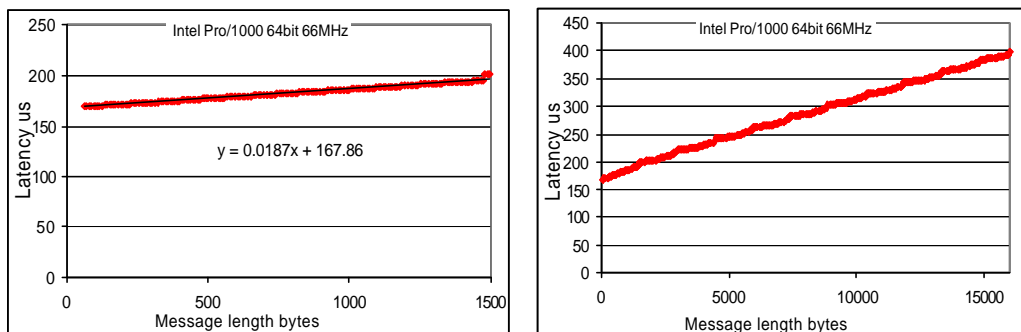


Figure 4.5 UDP Request-Response Latency as a function of the packet size for the Intel PRO/1000 XT Server NIC on the Supermicro 370DLE 64 bit 66 MHz PCI bus.

Figure 4.5 shows the round trip latency as a function of the packet size for the Intel PRO/1000 XT Server NIC connected to the 64 bit 64 MHz PCI bus of the Supermicro

370DLE motherboard. The left hand graph shows smooth behaviour as a function of packet size. The observed slope of $0.018 \mu\text{s}/\text{byte}$ is in reasonable agreement with the $0.0118 \mu\text{s}/\text{byte}$ expected. The right hand graph shows the behaviour for longer messages; it continues to be well behaved, indicating no buffer management problems.

The large intercept of $\sim 168 \mu\text{s}$ suggests that the driver enables interrupt coalescence, which was confirmed by the throughput tests that showed one interrupt for approximately every 33 ($\sim 400 \mu\text{s}$) packets sent and one interrupt for every 10 packets received ($\sim 120 \mu\text{s}$).

4.2.2 UDP Throughput

Figure 4.6 shows that the maximum throughput for full 1500 byte MTU frames on the 64 bit 66 MHz PCI bus was 910 Mbit/s. The lower graph shows the corresponding packet loss, this behaviour of loosing frames at wire rate is typical of most NIC-motherboard combinations. It was traced to “indiscards”, this is loss in the receiving IP layer when no buffers for the higher layers were available. Again, the receiving CPU was loaded at between 65 - 100% when the frames were spaced at $13 \mu\text{s}$ or less.

4.2.3 PCI Activity

Figure 4.7 shows the PCI signals for a 64 byte request from the sending PC followed by a 1400 byte response. Prior to sending the request frame, there is a CSR setup time of $1.75 \mu\text{s}$ followed by $0.25 \mu\text{s}$ of DMA data transfer. The default interrupt coalescence of 64 units was enabled. This gave $\sim 70 \mu\text{s}$ delay between the PCI transfer and the update of the CSRs and this delay is seen on both send and receive actions.

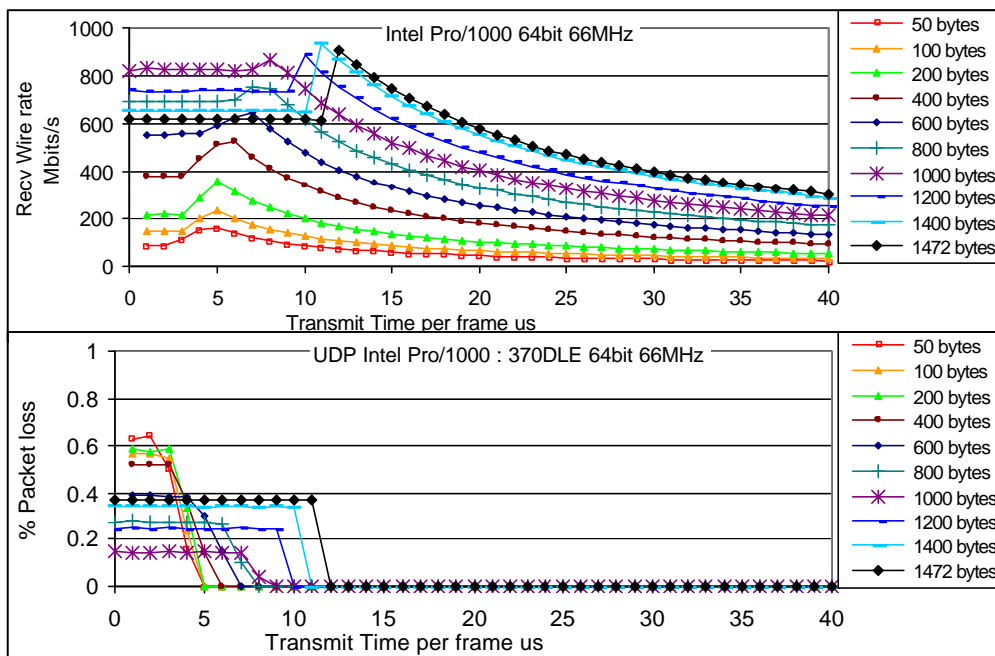


Figure 4.6 UDP Throughput as a function of the packet size for the Intel PRO/1000 XT Server NIC on the Supermicro 370DLE 64 bit 66 MHz PCI bus. The lower graph shows the corresponding packet loss.

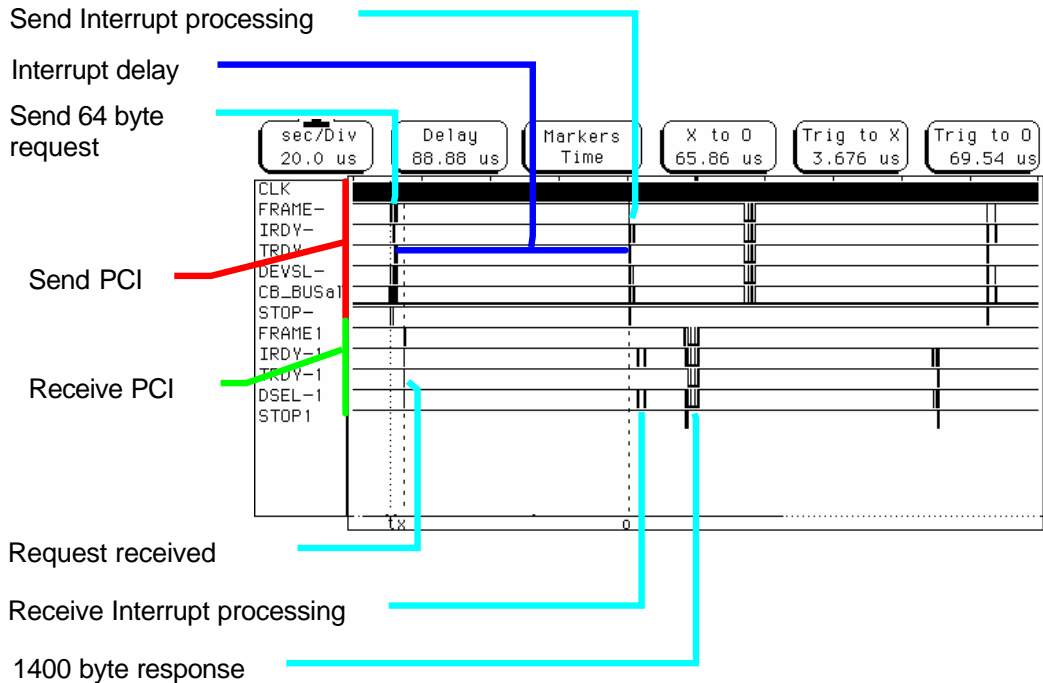


Figure 4.7 The traces of the signals on the send and receive PCI buses for the Intel PRO/1000 XT NIC and the Supermicro 370DLE motherboard for a 64 byte request followed by a 1400 byte response.

Figure 4.8 shows the PCI signals for 1446 byte frames sent every 11 μs, the PCI bus is occupied for ~4.7 μs on sending, which corresponds to ~ 43% usage, while for receiving it takes only ~ 3.25 μs to transfer the frame ~ 30% usage.

Figure 4.9 shows the transfers on a longer time scale. There are regular gaps in the frame transmission approximately every 900 μs. Transfers on the sending PCI stop first followed by those on the receiving PCI. As symmetric flow control was enabled, this suggests the use of Ethernet pause frames to prevent buffer overflow and subsequent frame loss. This behaviour is not unreasonable as one of these frames would occupy the Gigabit Ethernet for 11.7 μs.

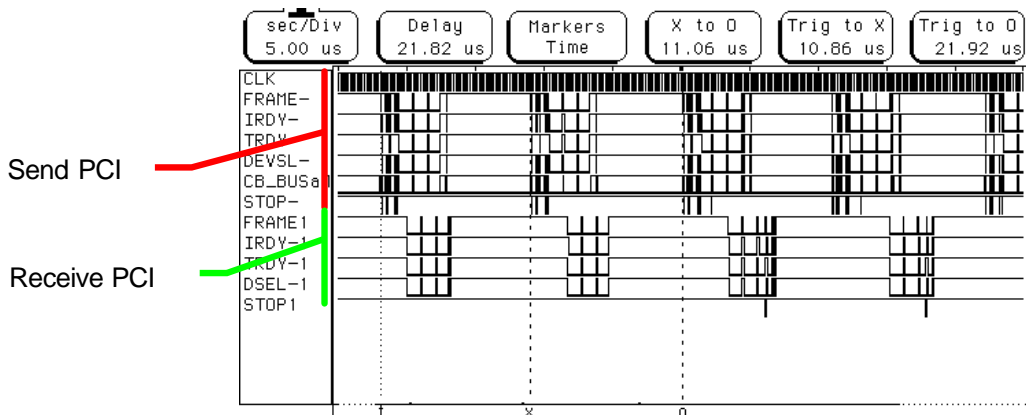


Figure 4.8 The traces of the signals on the send and receive PCI buses for the Intel PRO/1000 XT NIC and the Supermicro 370DLE motherboard for a stream of 1400 byte packets transmitted with a separation of 11 μs.

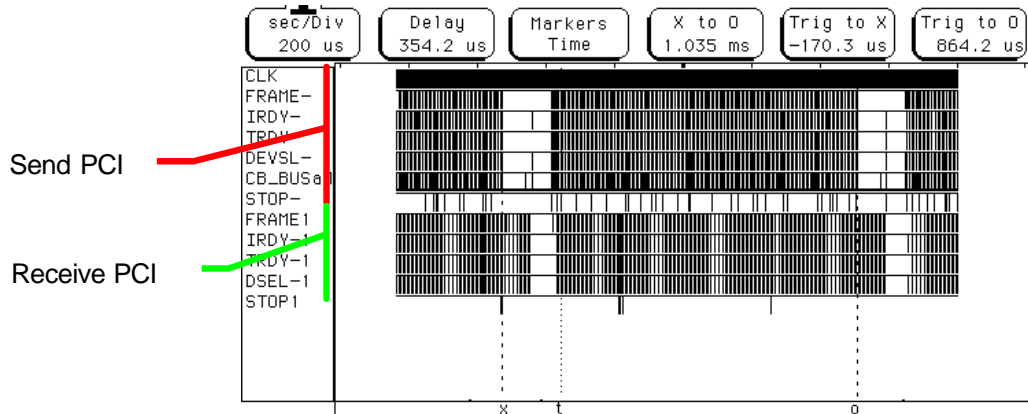


Figure 4.9 The traces of the signals on the send and receive PCI buses for the Intel PRO/1000 XT NIC and the Supermicro 370DLE motherboard for a stream of 1400 byte packets transmitted with a separation of 11 μ s. showing pauses in the packet transmission.

4.3 Alteon ACENIC Performance

4.3.1 UDP Request-Response Latency

Figure 4.10 shows the round trip latency as a function of the packet size with the interface connected to a 32 bit 33 MHz PCI bus on the left and a 64 bit 64 MHz PCI bus on the right. The curves are not as smooth as expected. There is somewhat more variation for message lengths spanning two packets, but for longer messages, the curves are again smoother. The observed slopes are in very good agreement with those expected – see section 3.1.

PCI bus	Observed slope μ s/byte	Expected slope μ s/byte
32 bit 33 MHz	0.023	0.023
64 bit 64 MHz	0.014	0.0118

The large intercept of $\sim 230 \mu$ s suggests that the driver enables interrupt coalescence, which was confirmed by the throughput tests that showed one interrupt for approximately every 33 ($\sim 400 \mu$ s) packets sent and one interrupt for every 10 packets received ($\sim 120 \mu$ s).

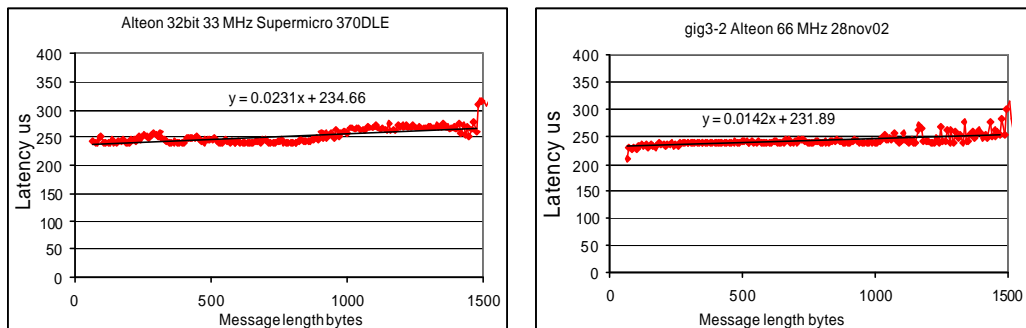


Figure 4.10 UDP Request-Response Latency as a function of the packet size for the Alteon NIC on 32 bit 33 MHz and 64 bit 66 MHz PCI buses of the Supermicro 370DLE.

4.3.2 UDP Throughput

As shown in Figure 4.11 the 674 Mbit/s throughput of the Alteon card for full 1500 MTU frames is good on the 32 bit PCI bus but very impressive at 930 Mbit/s on the 64 bit 66 MHz PCI bus. In both cases packet loss was only observed for packets less than 100 bytes long at spacing of 10 μ s or less. However, the receiving CPU was loaded at between 65 - 100% when the frames were spaced at 13 μ s or less so lack of CPU power could explain the packet loss.

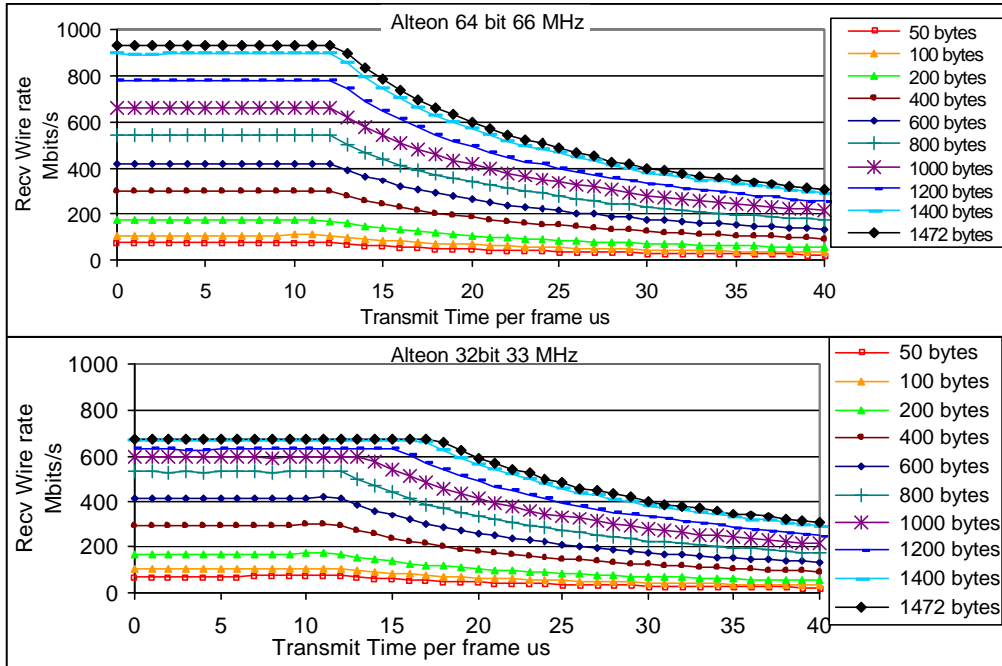


Figure 4.11 UDP Throughput as a function of the packet size for the Alteon NIC on 32 bit 33 MHz and 64 bit 66 MHz PCI buses of the Supermicro 370DLE.

4.3.3 PCI Activity

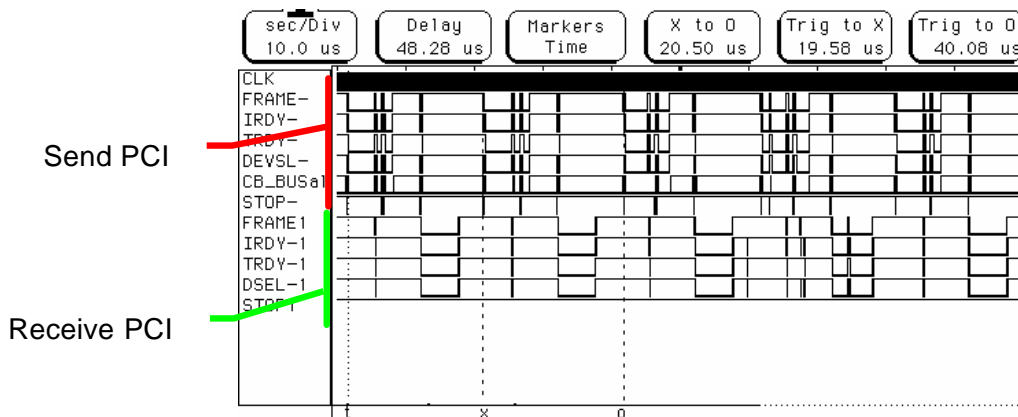


Figure 4.12 The traces of the signals on the send and receive PCI buses for the Alteon NIC on 64 bit 33 MHz PCI bus of the Supermicro 370DLE motherboard for a stream of 1400 byte packets transmitted with a separation of 20 μ s.

Figure 4.12 shows 1400 byte transfers between two Supermicro 370DLE systems using the 64 bit 33 MHz PCI buses and the Alteon ACENIC. The sending DMA transfers have a few suspensions, but the receiving transfers are continuous.

Figure 4.13 shows traces of the PCI signals when the bus speed is 66 MHz and the 1400 byte packets are transmitted at 16µs separation. The PCI DMA transfers from memory to the NIC on transmission look reasonable, but from the pausing of all the PCI signals when moving data from the NIC to memory, it appears that the NIC cannot sustain the 66 MHz transfer rate. However the transfer is still completed within the time to receive packets at line speed which supports the excellent throughput.

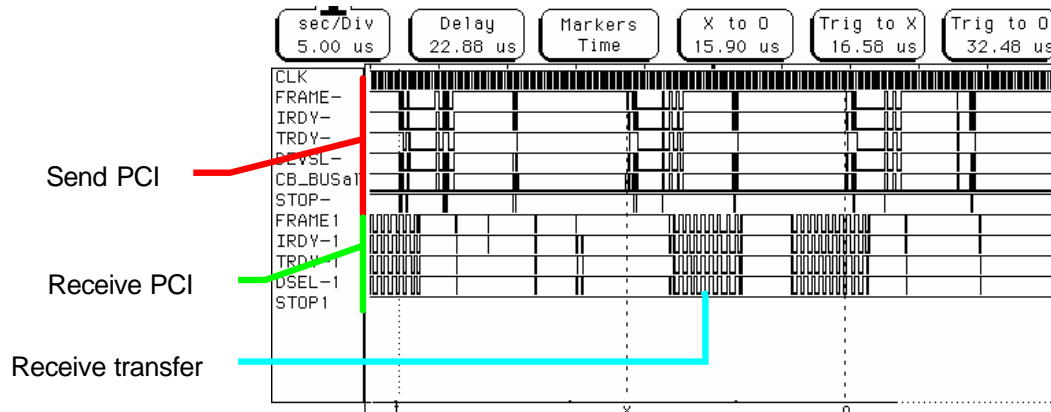


Figure 4.13 The traces of the signals on the send and receive PCI buses for the Alteon NIC on 64 bit 66 MHz PCI bus of the Supermicro 370DLE motherboard for a stream of 1400 byte packets transmitted with a separation of 16 µs.

5 Measurements made on the IBM das board

The motherboards in the IBM das compute server were tested, they had Dual 1GHz Pentium III with the ServerWorks CNB20LE Chipset and the PCI bus was 64 bit 33 MHz. Linux RedHat 7.1 with the 2.4.14 kernel was used for the tests.

5.1 SysKconnect

5.1.1 UDP Throughput

Figure 5.1 shows that the SysKconnect NIC performed very well on the IBM das system giving a throughput of 790 Mbit/s, which may be compared to only 620 Mbit/s on the Supermicro 370DLE. However the IBM had much more processing power with dual 1 MHz CPUs.

5.1.2 PCI Activity

The PCI signals for a single transfer of a 1400 byte packet are shown Figure 5.2. Like those for the Supermicro 370DLE, the DMAs are continuous and each is associated with its CSR updates.

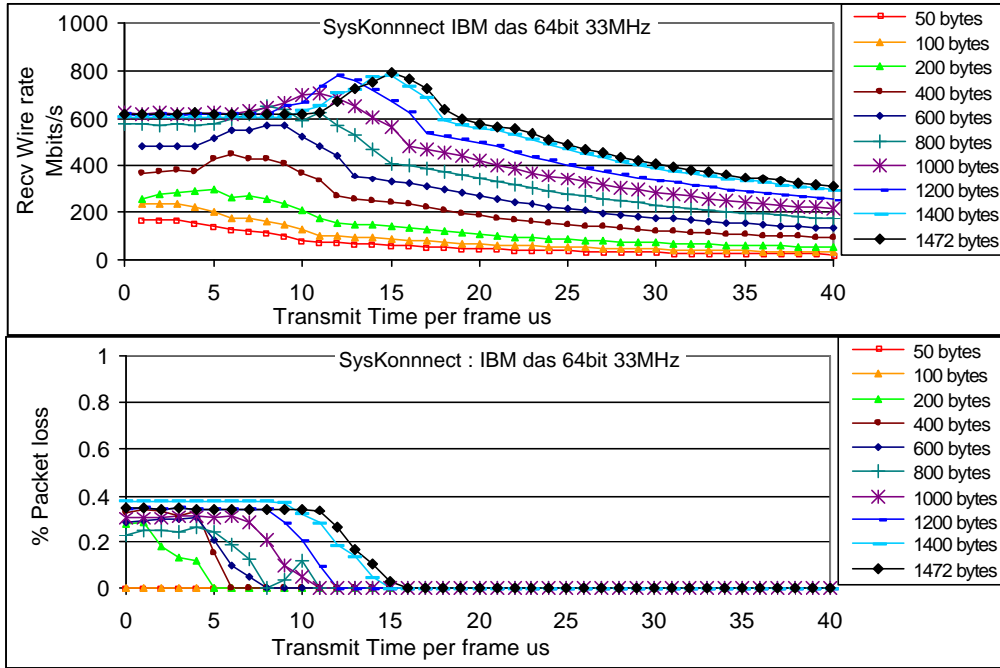


Figure 5.1 UDP Throughput as a function of the packet size for the SysKonnnect SK9843 NIC on the 64 bit 33 MHz PCI bus of the IBM das.

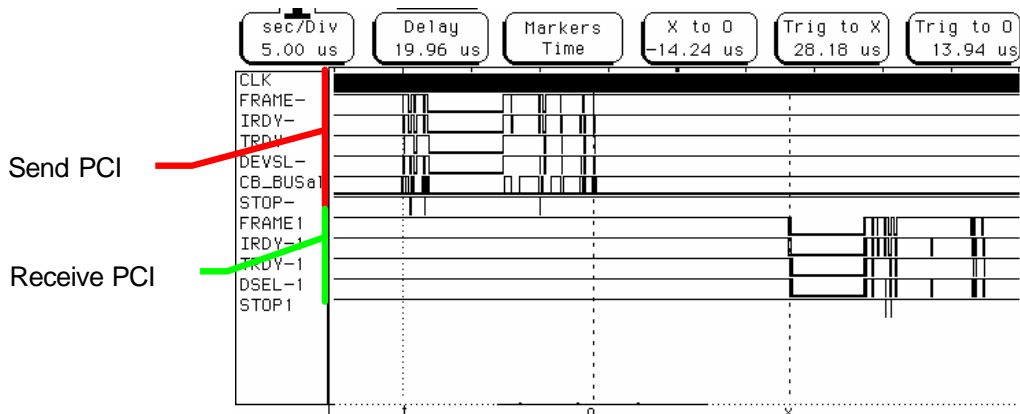


Figure 5.2 The traces of the signals on the send and receive PCI buses SysKonnnect SK9843 NIC for the transfer of a 1400 byte packet on the 64 bit 33 MHz PCI bus of the IBM das.

5.2 Intel PRO/1000 XT

5.2.1 UDP Request-Response Latency

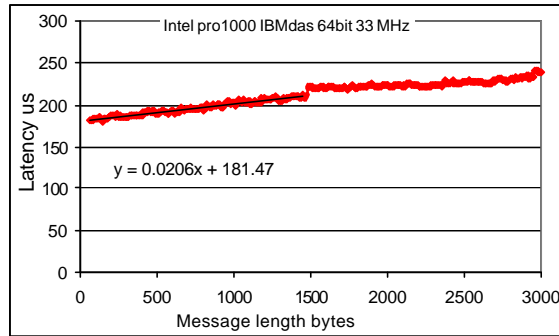


Figure 5.3 UDP Request-Response Latency as a function of the packet size for the Intel PRO/1000 XT Server NIC on the 64 bit 33 MHz PCI bus of the IBM das.

The round trip latency as a function of the packet size shown in Figure 5.3 is not as smooth as that shown for the Intel NIC on the Supermicro 370DLE motherboard. However the observed slope of $0.0206 \mu\text{s}/\text{byte}$ is in good agreement with the $0.0155 \mu\text{s}/\text{byte}$ expected when allowance is made for the memory copies.

The large intercept of $\sim 180 \mu\text{s}$ confirms that the driver had enabled interrupt coalescence.

5.2.2 UDP Throughput

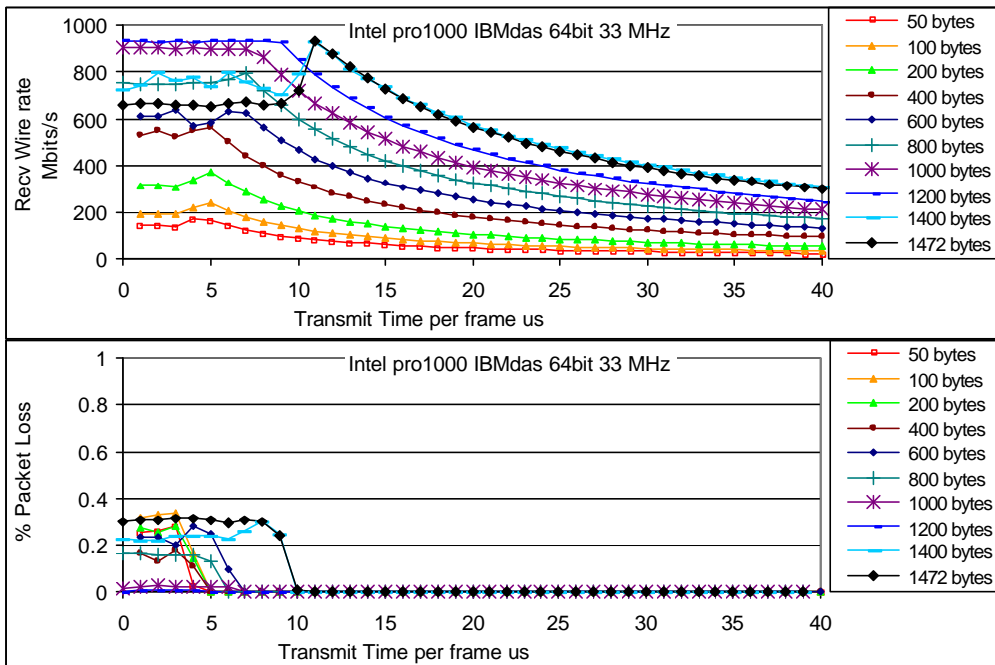


Figure 5.4 UDP Throughput as a function of the packet size and the corresponding packet loss for the Intel PRO/1000 XT NIC on the 64 bit 33 MHz PCI bus of the IBM das.

Figure 5.4 shows that the Intel NIC performed also very well on the IBM das system giving a throughput of 930 Mbit/s, which may be compared to the 910 Mbit/s on the

Supermicro 370DLE. This probably reflects the extra processing power on the IBM board and the different chipset. Again cases of high packet loss in the receiving host software correspond to lower throughput.

5.2.3 PCI Activity

Figure 5.5 shows the PCI signals for 1400 byte frames sent every 11 μ s, the PCI bus was occupied for $\sim 9.3 \mu$ s on sending, which corresponds to $\sim 82\%$ usage, while for receiving the transfers took only $\sim 5.9 \mu$ s to transfer the frame $\sim 50\%$ usage. With this level of bus usage, other concurrent i/o activities, such as disk access, could not be sustained.

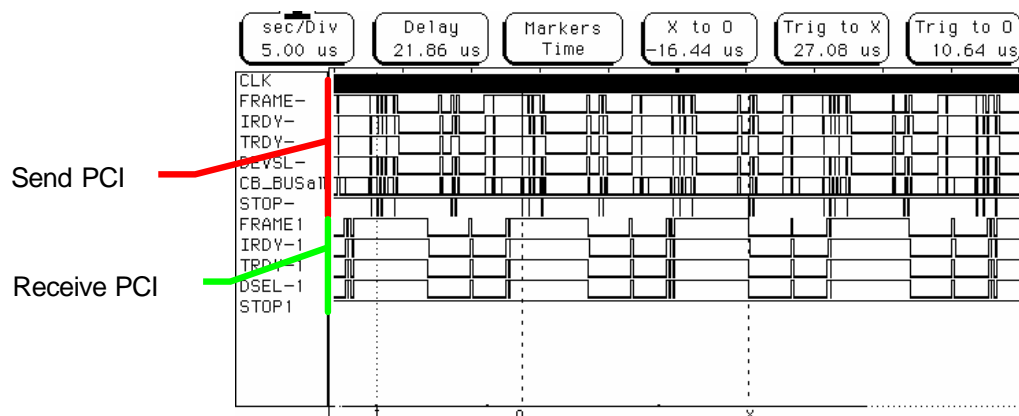


Figure 5.5 The traces of the signals on the send and receive PCI buses for the Intel PRO/1000 XT NIC and the of the IBM das motherboard for a stream of 1400 byte packets transmitted with a separation of 11 μ s.

6 Measurements made on the SuperMicro P4DP6 board

SuperMicro P4DP6 motherboard used the Intel E7500 (Plumas) Chipset and had Dual Xeon Prestonia 2.2 GHz CPUs; the arrangement of the 64 bit PCI and PCI-X buses was discussed in Section 2.

6.1 SysKconnect

6.1.1 UDP Request-Response Latency

The round trip latency as a function of the packet size shown in Figure 6.1 is a smooth function, indicating that the driver-NIC buffer management works well. The slope for the 64 bit PCI bus of 0.0199 μ s/byte is in reasonable agreement with the 0.0117 μ s/byte expected. The intercept of 62 μ s suggests that there is no interrupt coalescence which is confirmed by the data from the UDPmon throughput tests that record one interrupt per packet sent and received.

6.1.2 UDP Throughput

As shown in Figure 6.2 the SysKconnect NIC manages 876 Mbit/s throughput for full 1500 byte frames on the 64 bit 66 MHz PCI bus (at a spacing of 20-21 μ s). At inter-packet spacings less than these values there is packet loss. High packet loss corresponded to low throughput. These losses were traced to “indiscards”, this is loss in the receiving IP layer when no buffers for the higher layers were available.

The corresponding average CPU utilisation on the receiving PC was ~ 20 % for packets greater than 1000 bytes and 30- 40 % for smaller packets. When the inter-packet spacing of < 10 less than μ s, in some cases it appears that the receiving CPU is idle. However, these points correspond to the measurements with very large packet loss.

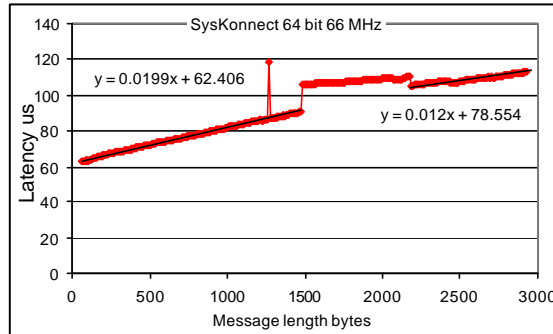


Figure 6.1 UDP Request-Response Latency as a function of the packet size for the SysKconnect NIC using the 64 bit 66 MHz PCI bus on the Supermicro P4DP6 board.

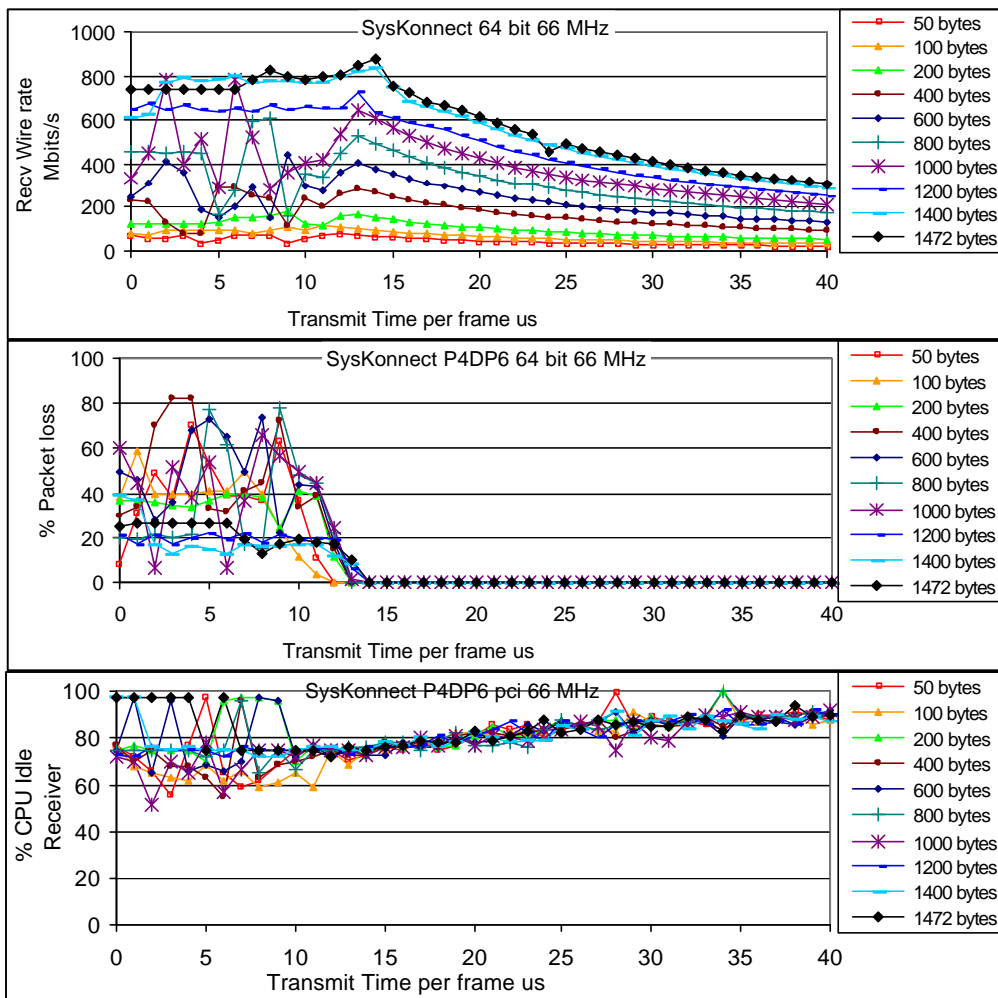


Figure 6.2 UDP Throughput as a function of the packet size and the corresponding packet loss and % of time the receiving CPU was idle for the SysKconnect NIC using the 64 bit 66 MHz PCI bus on the Supermicro P4DP6 board.

6.1.3 PCI Activity

Figure 6.3 shows the PCI activity when sending a 1400 byte packet. The DMA transfers are clean but there are many PCI STOP signals when accessing the NIC CSRs. It appears that the CPU/ motherboard can present the next command to the NIC faster than the NIC can accept them. It is usual to have well defined access and settling times for register access on peripheral devices. However this effect does not prolong the PCI usage by much as confirmed by the 876 Mbit/s throughput obtained.

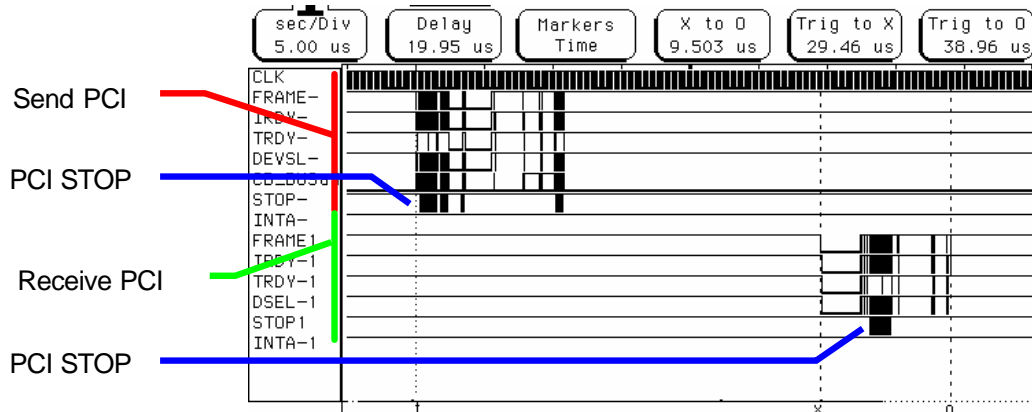


Figure 6.3 The traces of the signals on the send and receive PCI buses for the SysKconnect SK9843 NIC on the 64 bit 66 MHz PCI bus on the Supermicro P4DP6 board for the transfer of a 1400 byte packet.

6.2 Intel PRO/1000

Figure 6.4 shows the round trip latency as a function of the packet size. The behaviour shows some distinct steps. The observed slope of 0.009 $\mu\text{s}/\text{byte}$ is smaller than the 0.0118 $\mu\text{s}/\text{byte}$ expected, but making rough allowance for the steps, agreement is good.

Figure 6.5 shows histograms of the round trip times for various packet sizes, there is no variation with packet size, all having a FWHM of 1.5 μs and no significant tail. This confirms that the timing method of using the CPU cycle counter gives good precision and does not introduce bias due to other activity in the end systems.

6.2.1 UDP Request-Response Latency

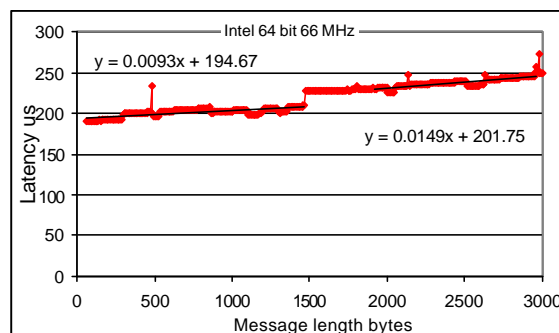


Figure 6.4 UDP Request-Response Latency as a function of the packet size for the Intel PRO/1000 XT NIC on the 64 bit 66 MHz PCI bus of the Supermicro P4DP6 motherboard.

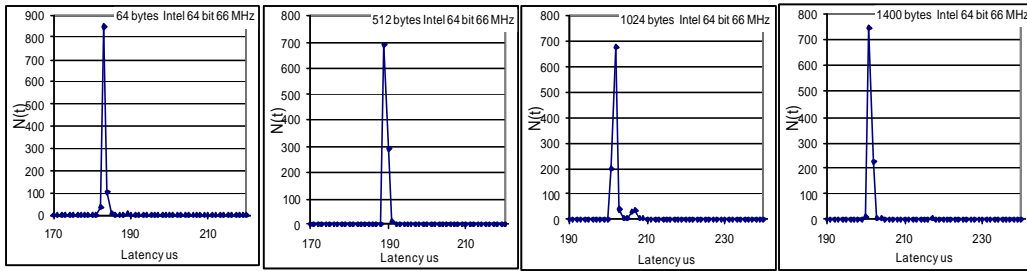


Figure 6.4 Histograms of the UDP Request-Response Latency for 64, 512, 1024 and 1400 byte packet sizes.

6.2.2 UDP Throughput

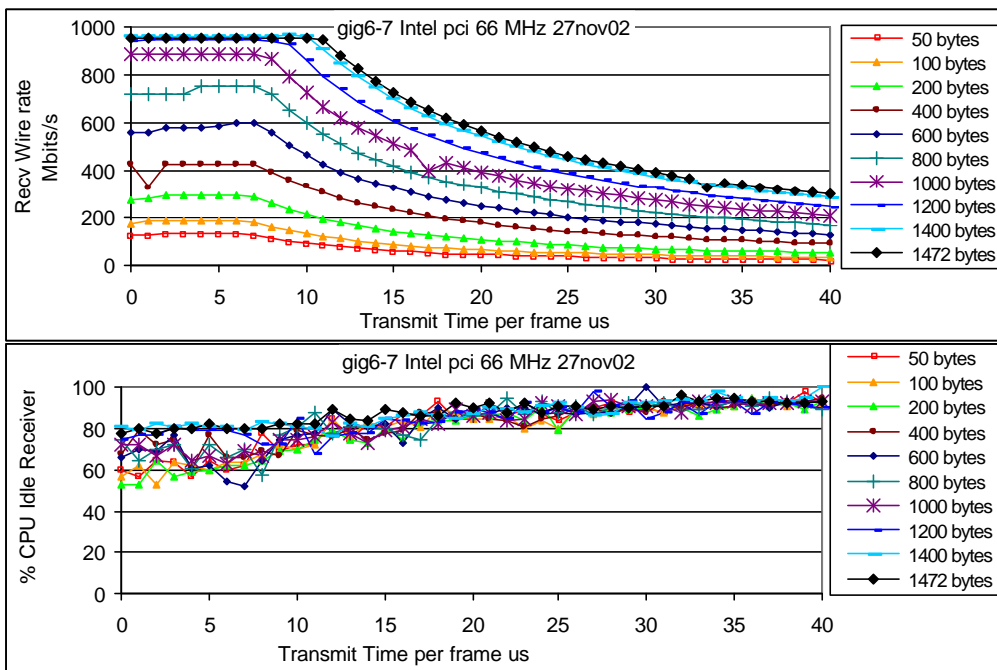


Figure 6.5 UDP Throughput as a function of the packet size and the corresponding % of time the receiving CPU was idle for the Intel PRO/1000 XT NIC on the Supermicro P4DP6 64 bit 64 MHz PCI bus. Note: during these tests no packet loss was observed.

Figure 6.5 shows that the Intel NIC performed very well, giving a throughput of 950 Mbit/s, with no packet loss. The corresponding average CPU utilisation on the receiving PC was ~ 20 % for packets greater than 1000 bytes and 30- 40 % for smaller packets.

6.2.3 PCI Activity

Figure 6.6 shows the PCI activity when sending a 1400 byte packet with the Intel PRO/1000 XT NICs. The DMA transfers are clean but there are several PCI STOP signals that occur when accessing the NIC CSRs. The assertion of STOP delays the completion of the PCI transaction as shown in more detail in the lower picture. The PCI bus is occupied for ~ 2 μ s on setting up the CSRs for sending, while for receiving it takes only ~ 2.9 μ s to transfer the frame over the PCI bus. For 1400 byte packets at wire speed the PCI bus is occupied for ~ 6.2 μ s on sending, which corresponds to

~ 50% usage, while for receiving it takes only ~ 3.0 μ s to transfer the frame, ~ 25% usage.

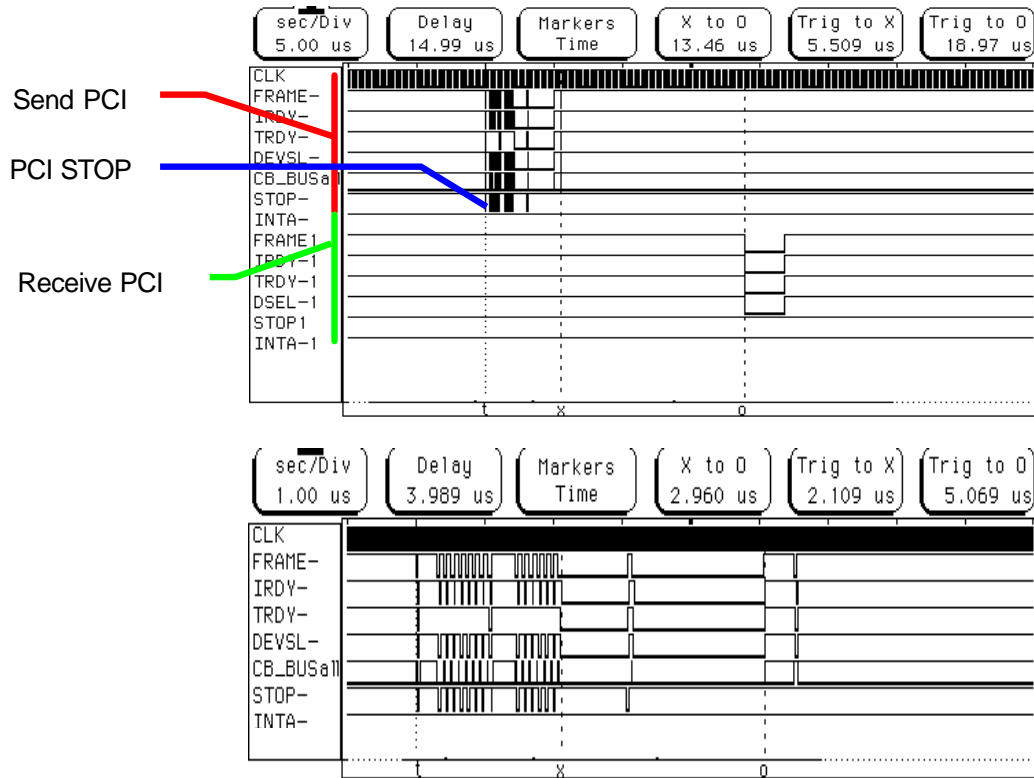


Figure 6.6 The traces of the signals on the send and receive PCI buses for the Intel Pro 1000 NIC and the Supermicro P4DP6 motherboard. The lower picture shows the PCI transactions to send the frame send in more detail.

7 Measurements made on the SuperMicro P4DP8-G2 board

SuperMicro P4DP8-G2 motherboard used the Intel E7500 (Plumas) Chipset and had Dual Xeon Prestonia 2.2 GHz CPUs. It had both 64 bit PCI and PCI-X buses and an onboard Intel 82546EB dual port Gigabit Ethernet controller.

7.1 SysKconnect

7.1.1 UDP Request-Response Latency

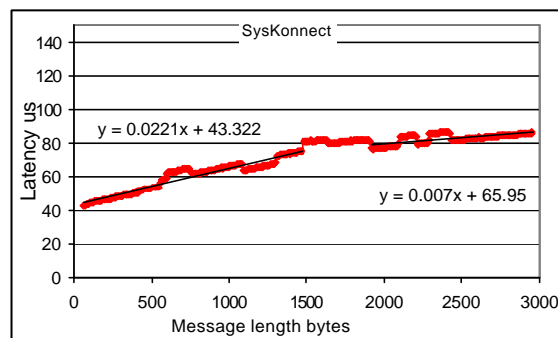


Figure 7.1 UDP Request-Response Latency as a function of the packet size for the SysKconnect NIC using the 64 bit 66 MHz PCI bus on the Supermicro P4DP8-G2 board.

Figure 7.1 shows the Latency as a function of the packet size for the SysKconnect NIC, there are several steps in the curve, and the slope of $0.022 \mu\text{s}/\text{byte}$ is larger than that for the P4DP6 board and the expected value of $0.0118 \mu\text{s}/\text{byte}$.

7.1.2 UDP Throughput

The throughput shown in Figure 7.2 reaches 990 Mbit/s for packets over 1200 bytes, which may be because these SysKconnect NICs are much later revisions than those used in all previous tests.

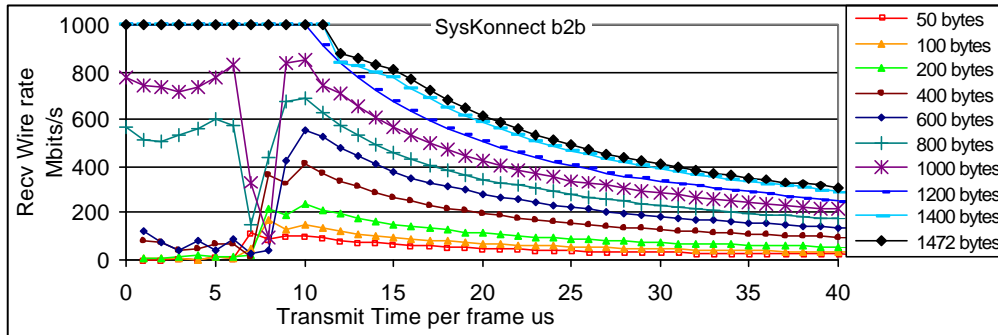


Figure 7.2 UDP Throughput as a function of the packet size for the SysKconnect NIC using the 64 bit 66 MHz PCI bus on the Supermicro P4DP8-G2 board.

7.2 Intel Pro/1000

7.2.1 UDP Throughput

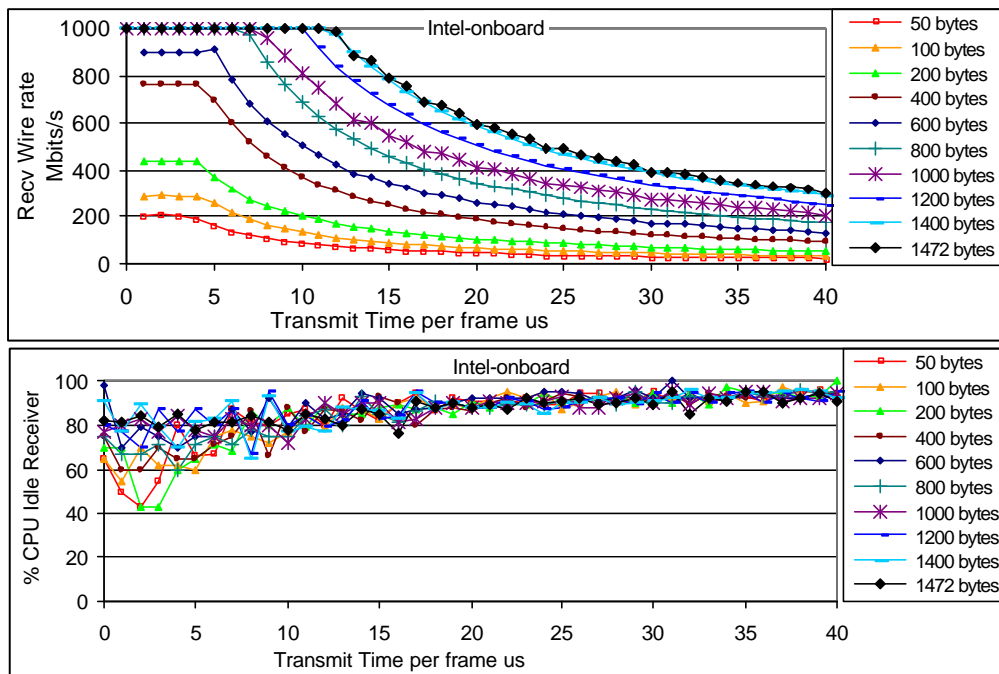


Figure 7.3 UDP Throughput as a function of the packet size and the corresponding % of time the receiving CPU was idle for the onboard Intel Gigabit controller on the Supermicro P4DP8-G2.

Figure 7.3 shows that the onboard Intel controller performed very well, giving a throughput of 995 Mbit/s, with no packet loss. The corresponding average CPU

utilisation on the receiving PC was ~ 20 % for packets greater than 1000 bytes and ~ 30 % for smaller packets.

8 Tests of Interrupt Coalescence on Intel PRO/1000 XT

Tests were performed to determine the effect of the interrupt coalescence feature available for the Intel PRO/1000 XT NIC. The interrupt coalescence (IC) can be set for receive (RxInt) and transmit (TxInt) interrupts, and can be delayed in units of 1.024 us (the default value is 64). Interrupt reduction can improve CPU efficiency if properly tuned for specific network traffic. As Ethernet frames arrive, the NIC places them in memory but the NIC will wait the RxInt time before generating an interrupt to indicate that one or more frames have been received. Thus increasing IC reduces the number of context switches made by the kernel to service the interrupts, but adds extra latency to frame reception. The tests described here were performed by sending UDP packets between two back-to-back PC's. The sending PC (PC1) contained an 800 MHz CPU, whilst the receiving PC (PC2) contained a 2.0 GHz CPU. Both PC's were running under the Linux kernel 2.4.18-SMP. When increasing the ICs care was taken that there were sufficient numbers of descriptors in the ring-buffers associated with the interface to hold the number of packets expected during that IC time.

Figure 8.1 shows the effect of varying the IC's on the round trip latency (RTT). The average RTT is plotted against packet length for values of RxInt on PC2 of 0, 40, 64 and 100 μ s. The value of the latency increases in a predictable way, with the difference in latency being approximately equal to the increased length of the interrupt coalescence.

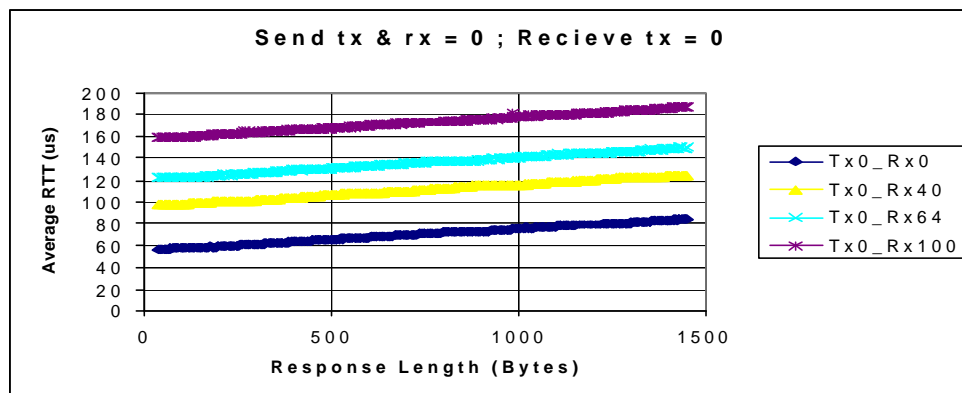


Figure 8.1 The average latency (RTT) plotted against the response packet length for back-to-back PC's and with different values for RxInt.

Figure 8.2 shows throughput plotted as a function of the inter-packet transmit time for a range of values of RxInt on the receiving PC. TxInt was set to the default value of 64 on the sending PC. Maximum throughput is achieved for RxInt values of 20 and above. Figure 8.3 shows the corresponding throughputs with TxInt set to 0 on the sending PC. Here the throughput is significantly affected for all values of RxInt due to increased PCI activity and insufficient power to cope with the context switching in the sending PC. These results show that if TxInt or RxInt are set too low then the additional interrupt processing may have a detrimental effect on the throughput.

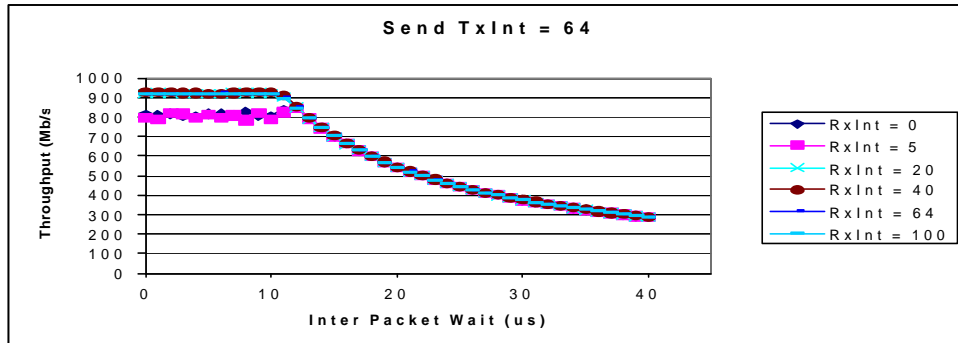


Figure 8.2 The average throughput plotted against the inter-packet transmit time for Txint of 64 μ s on the transmit PC and various values for Rxint on the receive PC.

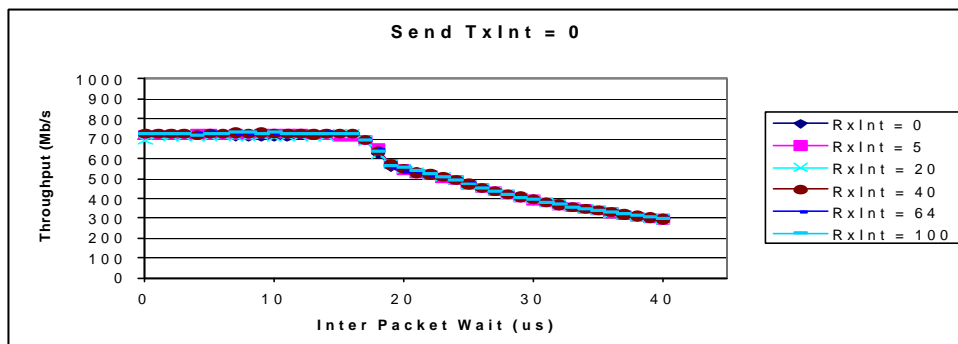


Figure 8.3 The average throughput plotted against the inter-packet transmit time for Txint of 0 μ s on the transmit PC and various values for Rxint on the receive PC.

9 Summary and Conclusions

A variety of high performance server type motherboards with PCI or PCI-X I/O buses have been tested with NICs from different manufacturers. Figure 9.1 gives a table summarising the maximum throughput for 1472 byte UDP packets and the transmit-receive interrupt coalescence times in micro-seconds used in the tests.

The Alteon card performed extremely well on a 64 bit 66MHz bus. The Intel PRO/1000 XT card gave a consistently high throughput between 910 – 950 Mbit/s, improving when used on the more powerful modern motherboards, and the on-board Intel controller operated at Gigabit line speed when one port was used. The early SysKonnnect card gave throughputs between 720 – 876 Mbit/s again improving with more modern motherboards. The SysKonnnect card used on the Supermicro P4DP8-G2 operated at line speed. In these tests all cards were stable in operation with several 100s Gbytes of data being transferred.

The inspection of the signals on the PCI buses and the Gigabit Ethernet media has shown that a PC with a 800 MHz CPU can generate Gigabit Ethernet frames back-to-back at line speed provided that the frames are > 1000 bytes. However, much more processing power is required for the receiving system to prevent packet loss. Network studies at SLAC [7] also indicate that a processor of at least 1 GHz/ Gbit is required. The loss of frames in the IP stack was found to be caused by lack of available buffers between the IP layer and the UDP layer of the IP stack. It is clear that there must be sufficient compute power to allow the UDP and application to complete and ensure free buffers remain in the pool.

NIC	Alteon AceNIC	SysKonnnect SK-9843	IntelPro1000
Motherboard			
SuperMicro 370DLE; Chipset: ServerWorks III LE PCI 32bit 33 MHz RedHat 7.1 Kernel 2.4.14	674 Mbit/s	584 Mbit/s 0-0 μ s	
SuperMicro 370DLE Chipset: ServerWorks III LE PCI 64bit 64 MHz RedHat 7.1 Kernel 2.4.14	930 Mbit/s	720 Mbit/s 0-0 μ s	910 Mbit/s 400-120 μ s
IBM das Chipset: ServerWorks CNB20LE; PCI 64bit 32 MHz RedHat 7.1 Kernel 2.4.14		790 Mbit/s 0-0 μ s	930 Mbit/s 400-120 μ s
SuperMicro P4DP6 Chipset: Intel E7500; PCI 64bit 64 MHz RedHat 7.2 Kernel 2.4.19-SMP		876 Mbit/s 0-0 μ s	950 Mbit/s 70-70 μ s
SuperMicro P4DP8-G2 Chipset: Intel E7500; PCI 64bit 64 MHz RedHat 7.2 Kernel 2.4.19-SMP		990 Mbit/s 0-0 μ s	995 Mbit/s 70-70 μ s

Figure 9.1 Table of the maximum throughput measured for 1472 byte UDP packets and the transmit-receive interrupt coalescence times in micro-seconds for the combinations of motherboard and NIC studied.

Timing information derived from the PCI signals and the round trip latency, allowed the processing time required for the IP stack and test application to be estimated as 10 μ s per node for send and receive of a packet.

The time required for the DMA transfer over the PCI scales with PCI bus width and speed as expected, but the time required to setup the CSRs of the NIC is almost independent of these parameters. It is dependent on how quickly the NIC can deal with the CSR accesses internally. Another issue is the number of CSR accesses that a NIC requires to transmit data, receive data, determine the error status, and update the CSRs when a packet has been sent or received. Clearly for high throughput the number of CSR accesses should be minimised.

A 33bit 32 MHz PCI bus has almost 100% occupancy and the tests indicate a maximum throughput of ~ 670 Mbit/s. A 64bit 32 MHz PCI bus shows 82% usage on sending when operating with interrupt coalescence and delivering 930 Mbit/s. In both these cases, involving a disk sub-system operating on the same PCI bus would seriously impact performance – the data has to traverse the bus twice and there will be extra control information for the disk controller.

To enable and operate data transfers at Gigabit speeds, the results indicate that a 64bit 66 MHz PCI or PCI-X bus be used. Preferably the design of the motherboard should allow storage and network devices to be on separate PCI buses, so that the data does not have to cross the same bus twice. For example, the SuperMicro P4DP6 / P4DP8

Motherboards have 4 independent 64bit 66 MHz PCI / PCI-X buses, allowing suitable separation of the bus traffic.

Driver and operating system design and interaction are most important to achieving high performance. For example, the way the driver interacts with the NIC hardware and the way it manages the internal buffers and data flow can have dramatic impact on the throughput. Some NICs, usually the older ones, require the driver to access many CSR registers to initiate the DMA transfer over the PCI bus and check the status of the operations. The operating system should be configured with sufficient buffer space to allow a continuous flow of data at Gigabit rates.

10 Acknowledgements

We would like to thank members of the EU DataGrid, EU DataTAG and the UK e-science MB-NG projects for their support and help in making test systems available, especially those colleagues at CERN, Universiteit van Amsterdam and University College London.

11 References

[1] SuperMicro motherboard reference material

www.supermicro.com and
www.supermicro.com/PRODUCT/MotherBoards/E7500/P4DP6.htm

[2] For information on using the CPU cycle counter see
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

And

F. Saka "The measurement software and clock synchronisation"

<http://home.cern.ch/~fsaka/>

[3] For definition of the terms used here, see GGF NM-WG draft L. Cottrell, R. Hughes-Jones, T. Kielmann, B. Lowekamp, M. Swany, B. Tierney "A Hierarchy of Network Measurements for Grid Applications and Services"

www-didc.lbl.gov/NMWG/measurements.pdf

[4] R. E. Hughes-Jones, F. Saka "Investigation of the Performance of 100Mbit and Gigabit Ethernet Components Using Raw Ethernet Frames" ATL-COM-DAQ-2000-014 http://www.hep.man.ac.uk/~rich/atlas/atlas_net_note_draft5.pdf

[5] UDPmon R. Hughes-Jones A tool for investigating network performance. Writeup and tool available from www.hep.man.ac.uk/~rich/net.

?6? Details of the Gigabit Ethernet Probe Card:

<http://www.hep.man.ac.uk/~scott/projects/atlas/probe/probehome.html>

[7] SLAC Network tests on bulk throughput site:

www-iepm.slac.stanford.edu/monitoring/bulk/

SC2001 & high throughput measurements

www-iepm.slac.stanford.edu/monitoring/bulk/sc2001

[8] See Intel White paper “Hyper-threading Technology on the Intel Xenon Processor Family for Servers”

<http://www.intel.com/eBusiness/pdf/prod/server/xeon/wp020901.pdf>

Intel “IA-32 Intel Architecture Software Developer’s Manual Vol 1. Basic Architecture” <http://www.intel.com/design/pentium4/manuals/245471.htm>

[9] Internet 2 Presentation “Gigabit Ethernet NICs Performances”.

[10] P. Gray, A.Bets University of Northern Iowa “Performance Evaluation of Copper-based Gigabit Ethernet Interfaes”