Evaluating Timestamping Accuracy for ASUS P5LD2-VM Motherboard with Intel NICs

Thuy T.T. Nguyen
Centre for Advanced Internet Architectures, Technical Report 070228F
Swinburne University of Technology
Melbourne, Australia
tnguyen@swin.edu.au

***Abstract*—This technical report gives detail on the results obtained from testing of the ASUS P5LD2-VM Motherboard with Intel network interface cards (Intel 10/100 and Onboard Intel PRO/1000 PCI Express) for time stamping accuracy. The hardware is tested for the purposes of being a Flow Meter in our prototype ANGEL QoS architecture. A number of affecting factors have been investigated, including the link speed configuration, the use of an intermediate switch in end-to-end connection, and the employment of the interrupt moderation scheme in the on-board Intel NIC. The trade-offs between time stamping accuracy and CPU performance has been quantitatively evaluated. We also discuss the implications of the issues on measuring game traffic.**

I. Introduction

This technical report outlines the results obtained from testing the SUS P5LD2-VM Motherboard with Intel network interface cards (Intel 10/100 and Onboard Intel PRO/1000 PCI Express) for time stamping accuracy. The hardware device is tested with the purposes of being used as a Flow Meter (FM) for our prototype ANGEL QoS architecture [1].

This technical report uses a similar setup and testing method as previous evaluation of CAIA laboratory computers in [2] and [3] .

The aim of the investigation is to find the optimal configuration for Ethernet interfaces for the most accurate timestamping results. The use of an additional intermediate switch with timestamping accuracy was also investigated. Due to the interrupt moderation scheme employed by the on-board Gigabit Ethernet card, the controller bundles packet events that arrive within a delay period, which is controlled by a number of different timers. This results in significant packet time stamping error (in the order of hundreds of microseconds) for packet rates higher than 8000 packets per second. The trade-offs between maximum interrupt rate per second employed by the interrupt moderation mechanism and time-stamping accuracy, CPU usage and packet loss rate have been quantitatively evaluated. We also discuss the potential implications on measuring game flows.

From February 2007 and July 2010 this report was a confidential deliverable to the Smart Internet Technologies CRC. With permission it has now been released to the wider community.

II. Equipment and Setup

The Flow Meter (FM) box has two different network cards (fxp0 (Intel 10/100Mbps Ethernet controller) and em0 (Intel Pro 1000Mbps Ethernet controller)). Time-stamping accuracy was tested for both cards with two link speeds, 100Mbps (for both fxp0 and em0 Ethernet interfaces with and without an intermediate switch in end-to-end connection) and 1Gbps (for em0 Ethernet interface with an intermediate switch in end-to-end connection).

We use the SmartBits2000 [4] to generate network traffic with timing accuracy of up to 100 nanoseconds. These packets are then captured by the FM where the packet inter-arrival timestamps can be compared with the known values.

Packet inter-arrival time set on SmartBits2000 was 25μsec, 125μsec, 500μsec, 1msec and 25msec for each of the link speed test. Each inter-arrival time was tested using a single SmartBits2000 card flooding packets (92byte + 4byte CRC long) to the FM's Ethernet port. Tcpdump was running at the FM to capture traffic for time-stamping analysis. The packet Inter-Arrival Time (IAT) values of traffic captured were compared against the rate at which the flow was sent from the SmartBits2000 device.

As noted in [3] the inter-arrival time used by SmartBits2000 and the Tcpdump time stamps differ in the way they are measured. SmartBits2000 measures inter-arrival time as the time between the end of one packet and

the start of the next packet. The Tcpdump -ttt option measures the time from the beginning of one packet to the beginning of the next packet. As such, the packet insertion time (96 bytes at 100Mbit/sec plus the preamble of 64 bit times) is included into the time interval. The timestamps reported by tcpdump should be the inter-arrival time of packets as sent by the SmartBits2000 plus approximately 8.3μsec [3] [2].

The test setup is illustrated in Figure 1 and 2:

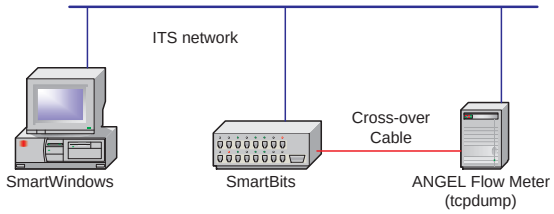

Fig. 1. Test Setup - Direct Connection

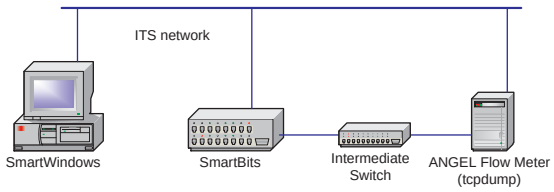

Fig. 2. Test Setup - Connection via an Intermediate Switch

Hardware technical specifications can be found in Table I.

TABLE I
HARDWARE SPECIFICATION

Test Component	Detail Specifications
Flow Meter	Intel Pentium 4 3.00GHz with Hyper-Threading, 1GB (2 x 512MB) DDR2 533 RAM, Seagate ST380817AS 80GB SATA HDD, Asus P5LD2-VM motherboard, Running FreeBSD 6.1
fxp0 NIC	Intel 10/100 PIC NIC
em0 NIC	Onboard Intel PRO/1000 PCI Express NIC
Smartbits	SmartBits-2000 Netcom Systems
Switch	Dell Power Connect 5324

III. TIME STAMPING ACCURACY FOR FXP0 AND EM0 INTERFACES FOR 100MBPS LINK CONFIGURATION

The test is setup with direction connection between the smartbits and the FM (see Figure 1).

Figure 3 shows the packet inter-arrival time reported by tcpdump (-ttt option) for packet IAT of 25μsec measured at fxp0 interface. The interface is explicitly configured with 100Mbps link rate and full-duplex.

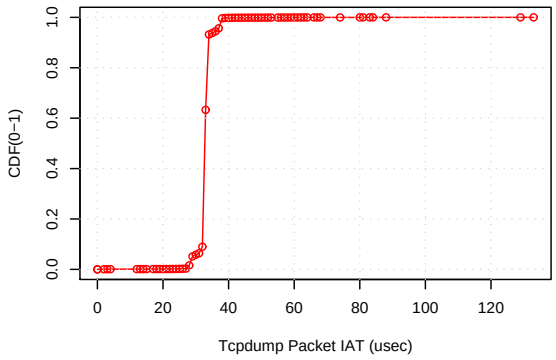

Fig. 3. Packet IAT of 25usec - fxp0 interface

As can be seen in Figure 3, with the Smartbits packet inter-arrival time of 25 μsec, 84.2% of packets (out of 50000 packets collected by tcpdump at fxp0 interface during the test) have the packet IAT times in the range of 25-26 μsec (after removing the 8.3 μsec due to packet insertion time - shown in Figure 4 in the range of 33.3-34.3 μsec). There are 2% and 0.3% of total packets with greater than 5μsec and 10μsec error respectively.

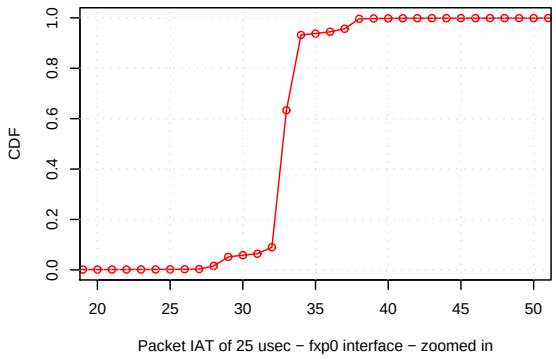

Fig. 4. Packet IAT of 25usec - fxp0 interface - zoom-in graph

Similar results have been seen with packet IAT of 125 μsec, 500 μsec, 1msec and 25msec. Statistics for all these tests are shown in Table II.

As can be seen from the result, fxp0 interface produces quite accurate packet time-stamping with less than 10 μsec absolute error for more than 99.5% of packets for all the packet inter-arrival time tests.

The test is repeated on the em0 interface. The Ethernet interface is configured explicitly with 100Mbps link rate and full-duplex mode.

TABLE II
SUMMARY OF RESULTS - FXP0 INTERFACE

Packet IAT	Mean μsec	Std.Dev. μsec	Error $<$ 1 μsec	Error $<$ 10 μsec	Error $<$ 100 μsec
25	25.02	2.24	84.22%	99.70%	100%
125	125.02	1.44	91.99%	99.93%	100%
500	500.00	2.61	91.11%	99.97%	100%
1,000	999.99	4.61	94.89%	99.97%	100%
25,000	24999.16	495.70	80.20%	99.82%	99.87%

TABLE III
SUMMARY OF RESULTS - EM0 INTERFACE)

Packet IAT	Mean μsec	Std.Dev. μsec	Error $<$ 10 μsec	Error $<$ 100 μsec	Error $<$ 250 μsec
25	25.02	117.18	0%	93.33%	93.33%
125	125.02	217.49	0%	0%	73.32%
500	500.00	157.63	95.20%	97.14%	97.15%
1,000	999.99	198.73	92.87%	98.07%	98.09%
25,000	24999.16	442.30	0%	95.12%	95.13%

Figure 5 shows the packet inter-arrival time reported by tcpdump (-ttt option) for packet IAT of 25μsec measured at em0 interface.

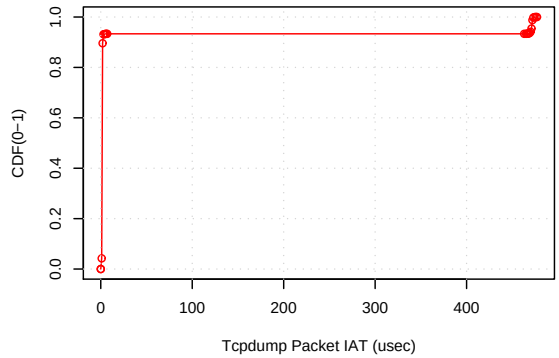

Fig. 5. Packet IAT of 25μsec - em0 interface

As can be seen in Figure 5, with the Smartbits packet inter-arrival time of 25 μsec, 93% of packets (out of 50000 packets collected by tcpdump at em0 interface during the test) have the packet IAT times in the range of 2-3 μsec (less than the 8.3 μsec of packet insertion time - an indication of a possible time-stamping flaw). The rest 7% of total packets have packet IAT in the range of 470-477 μsec. Although the packet IAT distribution results in an accurate mean value of 25.02 μsec (after removing the 8.3 μsec due to packet insertion time), the timestamp error for individual packet is significant.

Similar results were seen with other packet IAT rate. Statistics for test with packet IAT of 125 μsec, 500 μsec, 1msec and 25msec are shown in Table III.

As can be seen from the results, the em0 interface configured explicitly with a 100Mbps link rate produces significant time-stamping error per packet while still maintaining an accurate average packet IAT rate.

The packet time-stamping absolute error fluctuated depending on the packet rate received at the em0 interface. The time-stamping error was in the order of multiple 100μsec (e.g. 75% of total packets with an IAT of 125 μsec having an absolute time-stamping error of greater than 358μsec). Packets appeared to be bundled for interrupt handling with a delay of approximately 500μsec. Deeper investigation of packet inter-arrival time pattern was done with the Ethernet card configured at link rate of 1000Mbps in the next section.

IV. TIME STAMPING ACCURACY FOR EM0 INTERFACE FOR 1GBPS LINK CONFIGURATION

The test is setup with connection via an intermediate switch between the smartbits and the FM (see Figure 2).

Figure 6 shows the packet inter-arrival time reported by tcpdump (-ttt option) for packet IAT of 25μsec measured at em0 interface. The interface is explicitly configured with 1000Mbps (1Gbps) link rate and full-duplex.

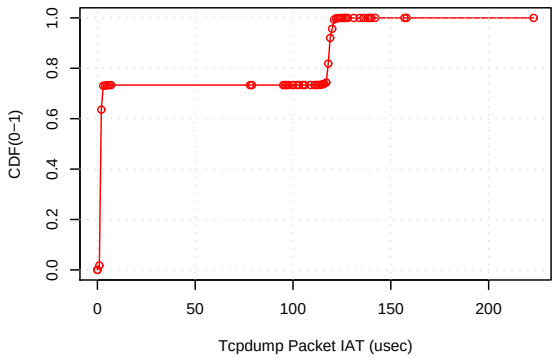

Fig. 6. Packet IAT of 25μsec - 1000Mbps em0 interface

As can be seen in Figure 6, with the Smartbits packet inter-arrival time set at 25 μsec, 73.4% of packets (out of 50000 packets collected by tcpdump at em0 interface during the test) have packet IAT times in the range of 2-5 μsec (less than the 8.3 μsec of packet insertion

time - an indication of a possible time-stamping flaw). Approximately 26.5% of total packets have packet IAT in the range of 75-124 μsec, and less than 0.1% of packets have packet IAT in the range of 125-223μsec. However, the packet IAT distribution results in an accurate mean value of 25.02 μsec (after removing the 8.3 μsec due to packet insertion time).

Similar results have been seen with other packet IAT rates. Statistics for test with packet IAT of 125 μsec, 500 μsec, 1msec and 25msec are shown in Table IV.

TABLE IV
SUMMARY OF RESULTS - EM0 INTERFACE)

Packet IAT	Mean μsec	Std.Dev. μsec	Error $<$ 10 μsec	Error $<$ 100 μsec	Error $<$ 250 μsec
25	25.02	51.75	0%	99.97%	100%
125	125.02	39.29	88.02%	89.52%	100%
500	500.00	136.23	90.92%	92.98%	99.88%
1,000	999.99	186.55	87.55%	92.72%	99.88%
25,000	24999.16	519.00	0.02%	81.27%	99.87%

Comparing with results shown in Table III, explicitly configuring the em0 interface in full-duplex mode with a link rate of 1000Mbps improves the time-stamping for individual packets. Packet time-stamping errors are less than 250μsec for more than 98% of total packets for all the packet IAT tests.

With the aim of testing the time-stamping accuracy for high packet rate, we take a closer look at individual packets timestamp for packet IAT of 25μsec. The packet inter-arrival time has an interesting pattern of groups of every 3 to 4 packets (with $\sim$2 or 3 packets with small inter-arrival times of 2-3 μsec and one with a large IAT of greater than 100 μsec. The total packet IAT of each group is $\sim$125μsec). Figure 7 illustrates the packet inter-arrival time pattern.

A similar pattern has been seen for packet IATs of 9.6μsec. However, the group pattern contains more packets ($\sim$6 or 7 packets) with small inter-arrival times of 2-3 μsec and one with a large IAT of greater than 100 μsec. The total packet IAT of each group is also $\sim$125μsec. Figure 8 illustrates the packet inter-arrival time pattern for packet IAT of 9.6μsec.

For packet IAT of $>=$ 125 μsec, the pattern no longer exists as can be seen in Figure 9.

This packet IAT pattern can be explained by the interrupt moderation mechanism used by Intel Gigabit

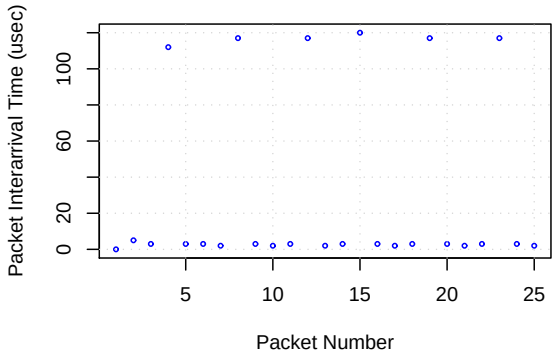

Fig. 7. Packet IAT of 25μsec - 1000Mbps em0 interface

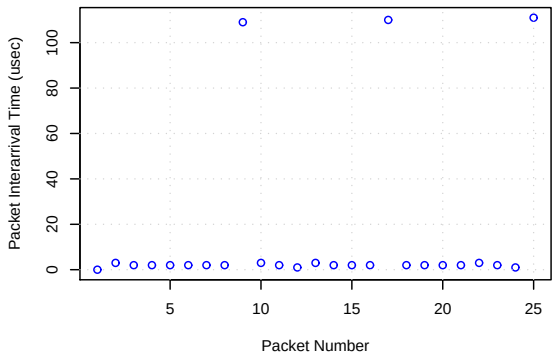

Fig. 8. Packet IAT of 9.6μsec - 1000Mbps em0 interface

Ethernet Controllers [5]. In the absence of any interrupt moderation, the controller will interrupt the CPU after every packet event. With the interrupt moderation scheme employment, rather than interrupting immediately upon transmitting or receiving a new packet, the controller is configured to delay the generation of this interrupt. The controller keeps bundling packet events that arrive within the delay period. When the timer expires, the controller delivers the bundled events to software all at once. The interrupt moderation scheme is claimed to improve CPU performance in the case of

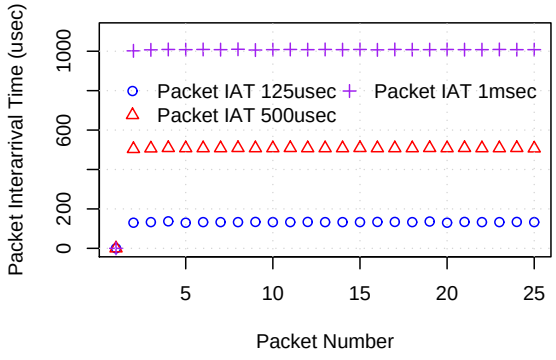

Fig. 9. Packet IAT of $>=$ 125μsec - 1000Mbps em0 interface

high data rate arrives at the Ethernet controller [5].

As described in [5], there are different timers for managing the interrupt rate, including: absolute timers, packet timers and one master timer for throttling all interrupt sources. The absolute timer allows the controller to transmit or receive multiple packets prior to generating an interrupt. Its timer starts to count down upon receipt of the first packet (after software has enabled interrupts), and is not altered by subsequent packets. Once the timer reaches zero, the controller generates a new interrupt. The absolute timers do not perform well in low traffic network, as a single packet has to wait for the full timer delay even when the network is idle. To overcome the problem, they are used in combination with packet timers. The packet timer starts to count down upon receipt/transmission of a new packet. If the controller receives/transmits another packet before the timer expires, it resets the timer to its original value and restarts the countdown. The controller generates a new interrupt when the timer reaches zero. Packet timers specify the maximum delay a single packet has to wait in the queue for the generation of interrupt in low data rate situations. To cope with the challenge of determining the optimal setting for transmit and receive timers, the interrupt throttling mechanism is introduced for placing an upper bound on the controller's interrupt rate. The throttling mechanism operates independently of any interrupt source.

With the em0 card in our FM box, the interrupt throttling rate is set at maximum of 8000 interrupts per second [6](which translates to 125 μsec timer), the absolute receive timer at 66 μsec and the receive packet timer at 0 μsec by default (sysctl configurable). These values explain the packet IAT pattern that we have seen above. For example, with Smartbits packet IAT set at 25 μsec and 9.6 μsec, the packet arrival rate at the Ethernet controller will be 33.3μsec and 17.9μsec respectively (including the 8.3 μsec packet insertion time). The default throttle timer value of 125 μsec allows 3.75 and 6.98 packets per interrupt respectively. It also causes the total delay of 125 μsec per interrupt. Since the absolute timer is smaller than the throttle timer, it does not have a significant effect on packet events bundling and interrupt delay.

Tuning the timer parameters can help to reduce the absolute individual packet time-stamping error. For example, increasing the interrupt throttling rate would help to reduce the maximum extra delay that the packet has to wait in the queue for bundling interrupt. However this comes with the cost of the CPU's performance.

The effects of different interrupt throttling rate configurations and absolute timer values were investigated. Three throttling rates of 8K, 16K and 32K and receiving absolute timers of 0μsec and 66μsec (default value) were tested with the Smartbits packet IAT set at 9.6μsec with a packet length of 92 bytes. The CPU clock rate was set to the default of 1000HZ for FreeBSD6.1. Results were collected for four repetitive tests at each configuration. Each test collected a million packets and CPU usage was monitored during the tests at 0.5 sec intervals.

Figure 10 shows the packet inter-arrival time reported by tcpdump for a packet IAT of 9.6μsec measured at em0 interface. The interface was explicitly configured in full-duplex mode at 1000Mbps (1Gbps) link rate. The kernel was compiled with 8K, 16K and 32K maximum interrupt throttling rate per second. The absolute receive timer was fixed at default value of 66 μsec.

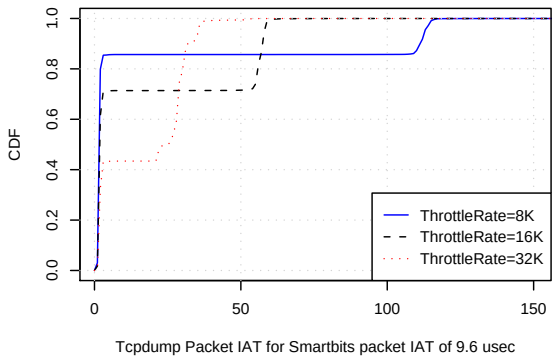

Fig. 10. Packet IAT of 9.6μsec - 1000Mbps em0 interface - abs.recv. timer 66 μsec

As can be seen in Figure 10, measured packet IAT error decreases as the interrupt throttling rate is increased. Most of the packet IATs are capped by the throttle delay values, which were 31.25μsec, 62.5μsec and 125 μsec for maximum rates of 32K, 16K and 8K interrupts per second respectively.

Slightly better results have been seen with the absolute receive timer of 0μsec. Figure 11 shows the packet inter-arrival time reported by tcpdump for packet IAT set to 9.6μsec measured at the em0 interface. The absolute receive timer set at 0 μsec.

Statistics for these tests are shown in Table V and Table VI.

Figure 12 shows the CPU usage for the tests with 8K, 16K and 32K maximum interrupt throttling rate and the absolute receive timer set at the default value of 66 μsec. Increasing the throttling rate from 8000 interrupts per second to 16000 interrupts per seconds increases the average CPU usage by 0.78%. However, in both

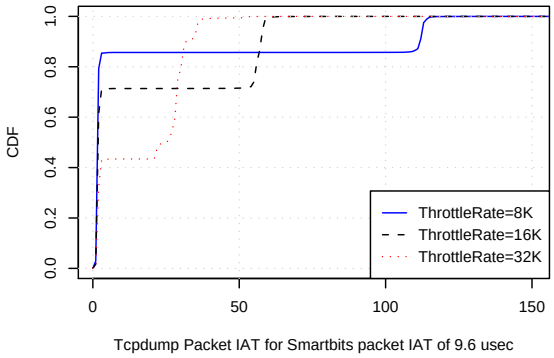

Fig. 11. Packet IAT of 9.6μsec - 1000Mbps em0 interface - abs.recv. timer 0 μsec

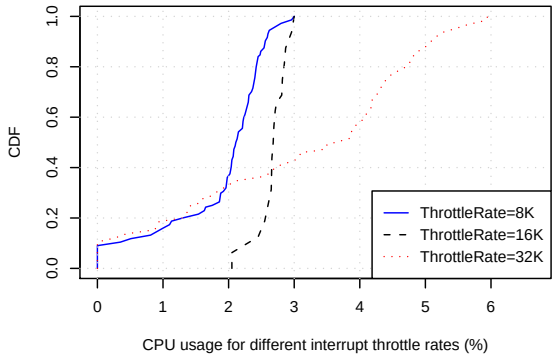

Fig. 12. CPU usage for different interrupt throttling rates - abs timer = 66 μsec

TABLE V
SUMMARY OF RESULTS - PACKET IAT FOR DIFFERENT THROTTLE RATES - DEFAULT ABS. RECV. TIMER = 66 μSEC)

Packet IAT	Mean μsec	Std.Dev. μsec	Error $<$ 30 μsec	Error $<$ 62 μsec	Error $<$ 125 μsec
8K	9.62	40.96	85.69%	85.69%	99.98%
16K	9.62	25.29	71.46%	99.96%	99.98%
32K	9.67	44.47	99.36%	99.99%	99.99%

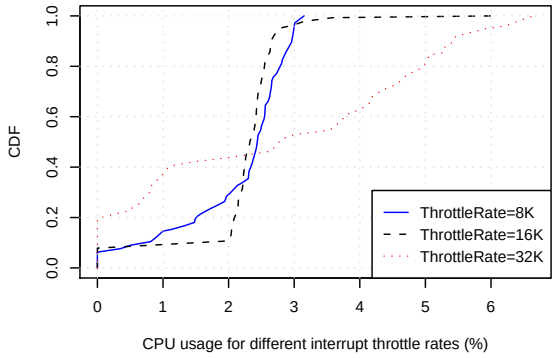

Fig. 13. CPU usage for different interrupt throttling rates - abs timer = 0 μsec

situations the CPU usage did not exceed $\sim$3% for the test duration. An interrupt throttling rate of 32K resulted in the highest CPU usage with a maximum value of 6%.

Figure 13 shows the CPU usages for the same tests with the absolute receive timer set at 0 μsec. Compared to the results presented in Figure 12, setting the absolute receive timer to 0μsec resulted in slightly heavier CPU usage.

Figure 14 shows the packet loss rate for all test configurations. Reducing the absolute receive timers to 0 resulted in significant increase in packet loss for throttling rates of 16K and 32K (by $\sim$ 0.3 and 0.7% respectively). However, by keeping the default absolute receive timer delay of 66μsec, doubling the default throttling rate resulted in only a slight increase in packet loss rate of 0.023%(from 0.125% to 0.147%)

TABLE VI
SUMMARY OF RESULTS - PACKET IAT FOR DIFFERENT THROTTLE RATES - ABS. RECV. TIMER = 0 μSEC)

Packet IAT	Mean μsec	Std.Dev. μsec	Error $<$ 30 μsec	Error $<$ 62 μsec	Error $<$ 125 μsec
8K	9.62	38.91	85.69%	85.69%	99.98%
16K	9.62	25.13	71.44%	99.98%	99.98%
32K	9.64	17.89	99.37%	99.99%	99.99%

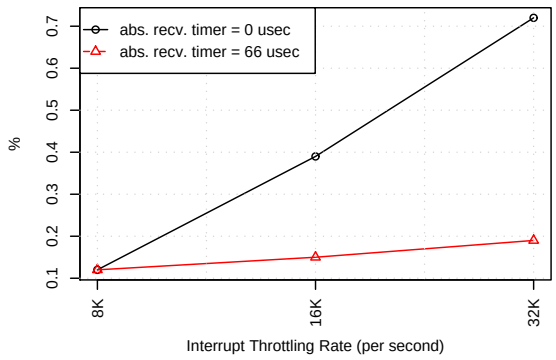

Fig. 14. Packet loss rate for different interrupt throttling and absolute timer rates

Considering the time-stamping error, CPU usage, and packet loss rate, our results suggest configuring the em0 controller with 16K of maximum interrupt rate per second, and to keep absolute receive timer at the

default value of 66 μsec. This configuration would help to reduce the time-stamping error rate for more than 99.95% of total packets to less than 62.5 μsec (compared to 125 μsec with the default configuration) with a slightly increased CPU load (increase by 0.78% on average comparing to the default configuration) and $\sim$0.023% increase in packet loss rate comparing to the default configuration.

V. Time stamping accuracy with an intermediate switch

Similar tests have been done to compare the time-stamping accuracy with direct connection and connection via an intermediate switch.

Network card configuration: Both fxp0 and em0 are active, explicitly set to 100Mbps and full-duplex according to findings in [2]

Smartbits Packet Interarrival Time of 25usec

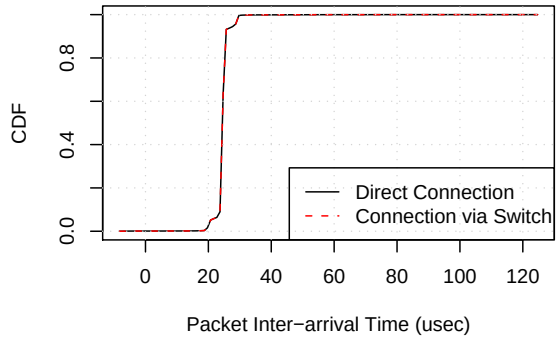

Fig. 15. Packet IAT of 25μsec - fxp0 interface

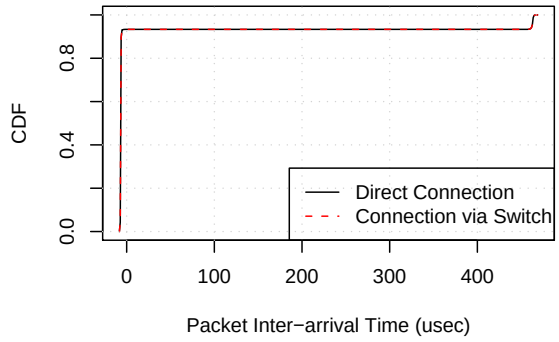

Fig. 16. Packet IAT of 25μsec - em0 interface

Smartbits Packet Interarrival Time of 125usec

As shown in Figure 15 16 17 18, the introduction of an intermediate switch did not affect the packet time-stamping accuracy.

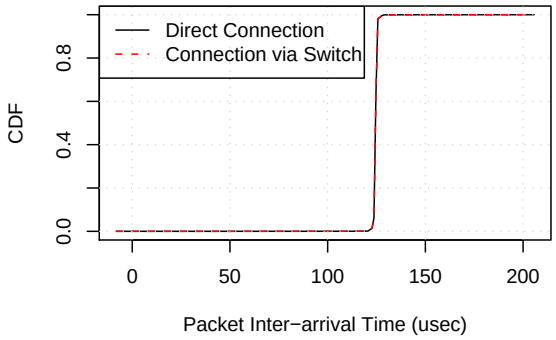

Fig. 17. Packet IAT of 125μsec - fxp0 interface

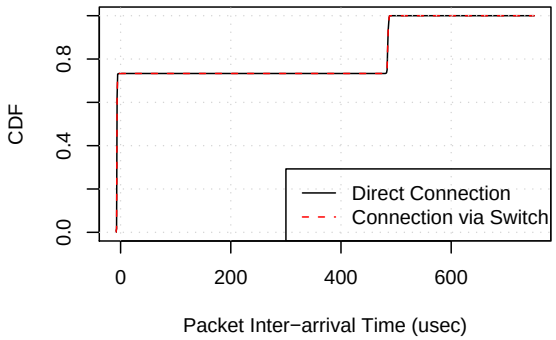

Fig. 18. Packet IAT of 125μsec - em0 interface

VI. Implications on measuring game flows

As shown in the previous section, the interrupt moderation scheme makes a significant impact on per-packet timestamps for high rate traffic. However, for game traffic, it is less likely that traffic flow would have within-flow packet inter-arrival time of less than 1msec [7]. So it is expected that the interrupt bundling scheme will have a relatively small effect on the application's per flow packet inter-arrival time.

To elaborate on this point, a number of tests have been run. With the use of 1 to 4 smartbits cards concurrently, we simulate the scenarios of 1 to 4 concurrent UDP flows with packet length of 60 bytes (without 4 byte CRC) and packet inter-arrival time of 1msec and 25msec. The aggregated back-to-back traffic resulted in a similar packet IAT pattern as seen in previous section. Figure 19 shows per-packet IAT of aggregated traffic for 1, 2, 3 and 4 identical concurrent traffic flows with a packet IAT of 1msec. Approximately 2000 packets per flow were collected by tcpdump at the em0 interface. Interface em0 was explicitly set in full-duplex mode with a link rate of 1000Mbps.

As shown in Figure 19, a high percentage of packets have a small packet IAT of $\sim$3 μsec (more than 30% for 2 concurrent flows, and more than 50% and 60% for the

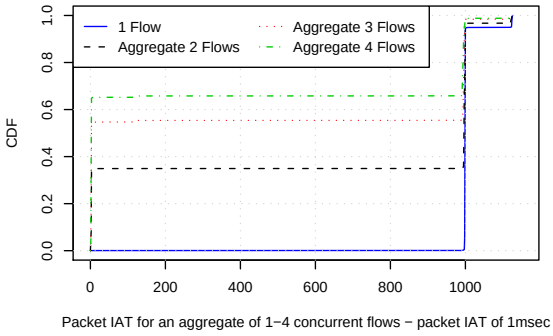

Fig. 19. Packet IAT of 1msec for 1, 2, 3, and 4 concurrent UDP flows- em0 interface (μsec)

3 and 4 concurrent flows respectively).

However, the packet IAT per flow is less affected. Figure 20 shows per-single-flow packet IAT when having up to 4 identical traffic flow(s) with packet IAT of 1msec running concurrently.

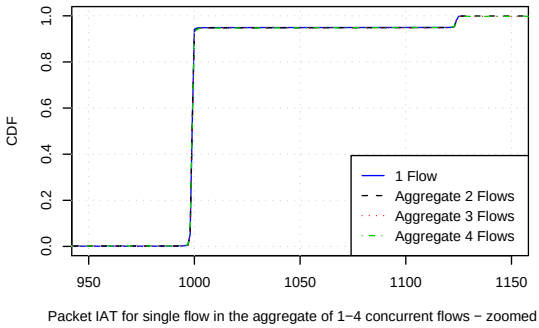

Fig. 20. Packet IAT for single traffic flow with upto 4 UDP flows running concurrently (μsec)

Test results show that within-flow packet IAT has a mean of 1000.3 μsec and std. dev of 167.1 μsec (16.7% of the mean value), 94.7% of all packets have a packet IAT error of less than 10μsec.

Similar results have been seen with packet IATs of 25msec. Figure 21 shows per-packet IAT of aggregated traffic for 1, 2, 3 and 4 identical concurrent traffic flows.

As shown in Figure 21, again a high percentage of packets have small packet IATs of less than 250μsec ($\sim$ 45% for 2 concurrent flows, and $\sim$ 65% and 70% for 3 and 4 concurrent flows respectively).

However, the packet IAT per flow is less affected. Figure 22 shows per-single-flow packet IAT when having up to 4 identical traffic flow(s) with packet IAT of 1msec running concurrently.

Test results show that within-flow packet IAT has a mean of 24984.33 μsec and std. dev of 939.03 μsec (3.76% of the mean value), 82.7% of all packets have a

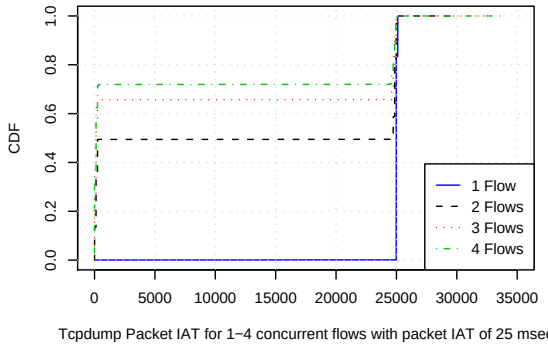

Fig. 21. Packet IAT of 25msec for 1, 2, 3, and 4 concurrent UDP flows- em0 interface (μsec)

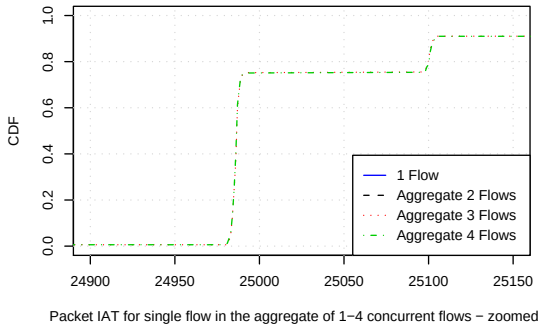

Fig. 22. Packet IAT (μsec) for single traffic flow with upto 4 UDP flows running concurrently

packet IAT error of less than 10μsec and 93.7% of all packets have a packet IAT error of less than 125μsec.

VII. Conclusion

Time-stamping Accuracy of the two Ethernet interfaces of the Flow Meter box has been tested. For capture rates of less than 100Mbps, capturing traffic at the fxp0 interface is recommended, as it has an error rate of 10μsec for more than 99.95% packets. For higher capture rates, traffic must be captured at the em0 interface. With the interrupt moderation scheme employed by the Ethernet controller, per packet time-stamping is significantly affected with default controller configuration (with an error rate in the order of 100μsec and bundled packets having packet inter-arrival times of 2-3 μsec for packet rate of $>$ 8000 packet per second). Time-stamping accuracy can be improved for this case with the trade-offs of CPU usages. However, for game traffic, most of traffic flow has packet inter-arrival time in the order of msec. Test results show that per-flow packet inter-arrival time had been skewed by a small percentage relative to the true packet IAT distribution.

ACKNOWLEDGMENT

This work was supported from 2005 to early 2007 by the Smart Internet Technology Cooperative Research Centre, http://www.smartinternet.com.au.

The author would also like to thank Lawrence Stewart for his help with the testbed setup, Grenville Armitage for his valuable comments and suggestions, Jason But, Sebastian Zander and Nigel Williams for the helpful discussions.

REFERENCES

[1] J. But, N. Williams, S. Zander, L. Stewart, and G. Armitage, "ANGEL - Automated Network Games Enhancement Layer," in *Proceedings of Netgames 2006*, Singapore, October 2006.

[2] A. M. Pavlicic and T. Nguyen, "Evaluating asus p4sgx-mx atx motherboards for the bart project," CAIA Technical Report, Tech. Rep. 031202B, 2003.

[3] M. Pozzobon, "Evaluation of laboratory computers for use as traffic monitors," CAIA Technical Report, Tech. Rep. 020924A, 2002.

[4] *Industry Standard Network Performance Analysis System SmartBits 2000*, Spirent Communications, http://www.spirentcom.com/documents/4077.pdf, as of January 2007.

[5] (as of September 28th 2006) Intel interrupt moderation using intel gigabit ethernet controllers - application note. [Online]. Available: http://www.intel.com/design/network/applnots/ap450.pdf

[6] (as of September 28th 2006) Source code of freebsd* driver for the intel(r) pro/1000 family of adapters. [Online]. Available: http://downloadmirror.intel.com/df-support/10957/ENG/README.txt

[7] G. Armitage, M. Claypool, and P. Branch, *Networking and Online Games - Understanding and Engineering Multiplayer Internet Games*. John Wiley & Sons, April 2006.