Time-stamping accuracy in virtualized environments

Ida Mutia, Rafika; Sadeque, Nayeema; Andersson, Jens A; Johnsson, Andreas

Published in:
[Host publication title missing]

2011

Time-Stamping Accuracy in Virtualized Environments

Rafika.I.Mutia*, N.Sadeque*, J.A.Andersson*, A.Johnsson**
* Department of Electrical and Information Technology, Lund University, Sweden
** Ericsson Research, Stockholm, Sweden
wx08rm7@student.lth.se, wx08ns1@student.lth.se, Jens_A.Andersson@eit.lth.se, andreas.a.johnsson@ericsson.com

Abstract—**The swift acceptance and the widespread use of virtual environments has substantially increased the stress on networks as each of the operating systems running in parallel sends out packets over the same network. As an increased number of packets traverse a network, the importance to continuously monitor whether the network is providing satisfactory service has increased. The metrics for such analyses includes delay, jitter, packet loss and available path capacity. The basic parameter required to quantify these metrics is the time-stamping values of the packets. The purpose of this research work is to characterize the impact of virtualization on a PC's time-stamping process. Operating system virtualization is employed where the time-stamping is done by the operating system kernel. CPU load and packet generating parameters are varied in order to study about the effect of virtualization on the time-stamping process under different operating conditions.**

Keywords — **Virtualization, Time-stamping Accuracy, Inter-Packet Separation, Packet Size, Packet Rate, CPU Load**

I. INTRODUCTION AND MOTIVATION

As the demands for high-speed networks grow, it is required to better quantify performance parameters such as packet loss, latency, jitter and round-trip time, including improved estimation of the available bandwidth and the bottleneck capacity of a network. These network metrics are needed to effectively manage the network and also to be used for rate adaptation for various applications. One of the parameter needed is an effective time-stamping mechanism which can be used to study the inter-packet separation gap. From the inter-packet separation gap obtained, the above performance metrics can then be determined.

This paper investigates how the time-stamping process is affected by environments that employ multiple operating systems running in parallel on a single PC or embedded controller. This technique is known as operating system virtualization and its adaptation is rapidly minimizing the amount of hardware required for an organization to work, increasing efficiency of multi-core processors including the offer to save cost, reduce carbon footprint, the ability to consolidate systems and also offering increased system security. The purpose of this paper is to investigate the effect of virtualization on the time-stamping process. This aims to comprehend how the stability of time-stamping process is affected in the presence of virtual environments compared to non-virtual environments. For this, the difference in consecutive time-stamped values forms the basis for the analysis in this paper.

The remainder of the paper is organized as follows. Section 2 provides a review of some related work. The software tools used to carry out the investigations are given in the following section, Section 3. The basic setup is explained in Section 4 followed by detailed descriptions of the results obtained and the analyses carried out have been put forth in Section 5. Significant conclusions are proposed in Section 6, ending with potential areas for future work in Section 7

II. RELATED WORKS

Characterizing a network is important and endless research has been carried out to discover new and more reliable approaches. Examples of network analysis include [1] and [2]. The former performs packet delay and delay variation analyses for different live networks derived from the time-stamped values of the arriving packets. The later involves measuring the available bandwidth of a path in the network using packet dispersion techniques by sending equal-sized packet pairs.

Detailed investigations of the accuracy of the time-stamp values has been done in [3] which quantifies time-stamping error and converts this error to data inaccuracy which has been proved to be a significant portion of it. In [4] the available end-to-end bandwidth is estimated by active probing. Here, consecutive packet probes called a packet train are sent. The inter-packet separation strain of the consecutive probe packets caused by cross-traffic is then used to estimate the available bandwidth. This paper uses this concept and terms the inter-packet separation gap as delta (Δ).

Endace Technology Ltd has introduced hardware in [5] as an answer to this problem called the DAG card. It offers resolution in the order of 15 nanoseconds for its time-stamps compared to the microsecond resolution of the traditional NIC cards. However, the DAG aspect mentioned would then be as a result of the literature research and current advances in the area, not related to the implementation aspect of the research. Reference [3], [4] and [5] establishes a need for a high precision time stamping and the fact that improved estimation of network parameter relies on time-stamped values of network packets.

Certain terminologies used in this paper are explained further. The Host Operating System (Host OS) is the operating system of the physical computer on which VirtualBox was installed while the Guest Operating System (Guest OS) is the operating system that is running inside the virtual machine.

III. BACKGROUND AND TOOLS

Capturing network packets for inspection is needed to monitor the network. As packets are sent or received, the time of their arrival or departure are stamped, respectively. The time-stamping value which can be used for calculating the inter-packet separation gap between packets inside one train, is useful for estimation of available bandwidth and bottleneck capacity.[4]

This paper is focused on time-stamping accuracy and aims to investigate the degree of this accuracy in virtualization environments compared to non-virtualized environments. For this, a single parameter is solely derived from the time stamps of consecutive network packets: the delta (Δ) time. The delta (Δ) time is the inter-packet separation gap between the time stamps of the subsequent packets at sender or the receiver side as illustrated Figure 1.

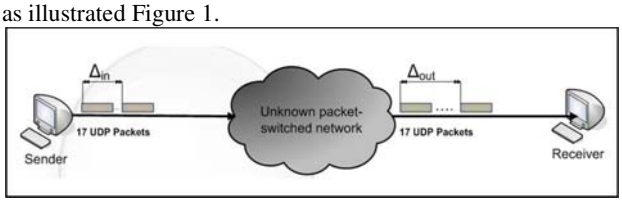

Figure 1. Illustration of Inter-packet Separation

To be able to generate, send and receive packets over the network the Distributed Internet Traffic Generator (D-ITG version 2.7.0-Beta 2) was used [6]. This is an open source packet level traffic generator to produce IPv4/IPv6 traffic. In order to view the time stamps of the captured packets, an open-source packet analyzer, Wireshark was used [7]. The UDP packets were filtered from other synchronization-purposed packets and their time stamps were examined. The preference to obtain the difference in the timestamps of the consecutive packets (Δ) sent and received were also set.

Reference [5] describes how packets are captured by the NIC which in turn also notifies the kernel by issuing an interrupt. When the kernel services this interrupt, the NIC driver discovers that a packet has been received and a time stamp is recorded. The time-stamp value is then saved in *libpcap* library, from where the Wireshark obtain this information. By Wireshark, this information is saved in capture file and can be loaded every time capture data is needed to be displayed.

The PC's clock provides microsecond resolution; hence the time-stamps values for the Δ also have microsecond precision. Figure 2 provides an illustration of the process of time stamping using the system kernel with Wireshark as the network packet analyzer. A similar process takes place for a virtualized environment.

In our research, the Sun VirtualBox was used as the Virtual Machine. When a virtual environment is created, the virtual operating system also consists of Virtualized NIC which shown in Figure 3.

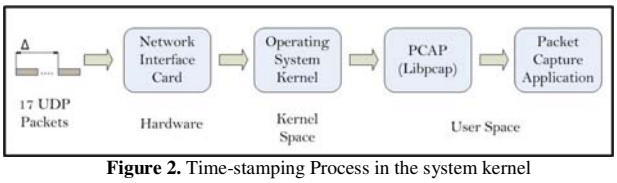

Figure 2. Time-stamping Process in the system kernel

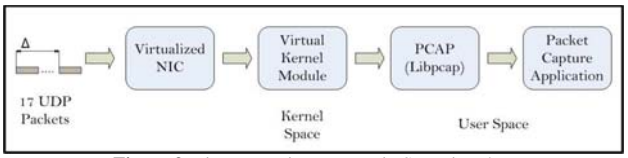

Figure 3. Time-stamping Process in Sun VirtualBox

IV. METHODOLOGY

A. Basic Setup

The basic configuration in this research is set up as in Figure 1. Two computers are connected by Ethernet crossover cable. The sender generates UDP packets by using D-ITG Network Traffic Generator with a particular packet size, packet rate and duration. In both stations, Wireshark will capture the time and determine the inter-packet separation, or Δ, between the packets. The ITGRecv executable displays complete information about the traffic and from which the occurrence of packet loss can be perceived.

1) ***System Parameter:*** The packet parameters varied are packet size and packet rate. The aim is to analyse the impact of these parameters on time-stamping accuracy and performance. The packets are sent in trains of 17 packets, often used when measuring performance parameters such as available bandwidth [4] and in our research, only one train is sent per experiment. The sender and receiver nodes are interconnected by a lab network; that is no other traffic is occupying the network.

As result of packet rate variation and constant train length, the duration of packet transfer will be dependent on the packet rate value. On the other hand, variation of network speed can be derived from variation of packet sizes.

In the following details of the variation in system parameters used corresponding to the input arguments of the ITGSend executable of D-ITG are shown:

- Packet Sizes: 64, 500 and 1460 bytes
- Packet Rate: 1pkt/s, 10pkt/s and 100pkt/s, which results in Δ of 1 second, 0.1 second and 0.01 second, respectively.

2) ***Platform Scenarios:*** The measurements were carried out in different scenarios. The CPU load is one variable of interest for time-stamping accuracy. An application is run and boost up the CPU load up to 100%, in order to see the effect of fully-loaded CPU in time-stamping accuracy.

Moreover, packets are also sent from different platforms; from host OS, from guest OS, from host OS with presence of guest OS and from guest OS with presence of another guest

OS. In the receiver, the receiving platform is always in the host OS.

The following are the different sending scenarios used to carry out the experiments.

- Scenario 1: Packets sent from host OS
- Scenario 2: Packets sent from host OS with fully loaded host CPU
- Scenario 3: Packets sent from guest OS
- Scenario 4: Packets sent from host OS with fully loaded host CPU and presence of guest OS
- Scenario 5: Packets sent from guest OS with fully loaded guest CPU
- Scenario 6: Packets sent from guest OS and presence of another guest OS with fully loaded CPU

V. RESULTS AND ANALYSIS

A. Statistical Analysis

Three runs were carried out for each experiment in order to provide more statistical data. Two statistical property tests were performed on the Δ values obtained: distribution and dependency of the data. The distribution for the three runs observed in the Figure 4 is a normal distribution or Gaussian distribution where the data set is clustering around the mean. However, there is a slight difference between the theoretical values and experimental values where the mean value shown in the figure is less than the theoretical mean value, which is 1 second for 1 packet/second. But the differences are insignificant and still show that the Δ values are normally distributed.

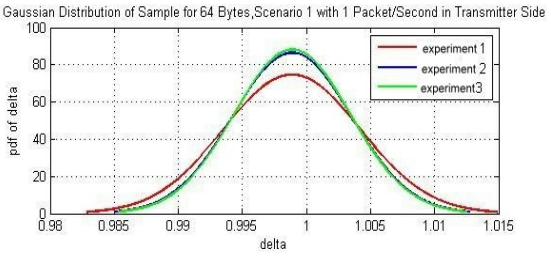

Figure 4. Gaussian Distribution for 64 bytes, Scenario 1 with 1 Packet/Second

The next statistical property evaluated is the dependency of each Δ in one train. The dependency property can be investigated by analyzing the correlation and covariance of the packets. Correlation is defined as the statistical relationship between the random variables or the observed data values. On the other hand, covariance is defined as the measure of how much two variables change together.

Covariance Equation [8]:

$$\gamma = E[(x - E[x]) * (y - E[y])]$$

Correlation Equation [8]:

$$\varphi = E[(x - E[x]) * (y - E[y])]/[\delta(x) * \delta(y)]$$

From the equations, dependencies of packets are represented in covariance plot and scatter plot. Figure 5 shows that the covariance is maximum when Δ = 0 and there is a steep decrease thereafter. From this test, a pre-conclusion is made that each Δ within one train are uncorrelated and independent. This also applies for each Δ between trains.

Next, dependency of delta values from correlation equation is analysed. Correlation reflects the linear relationship between data values. The scatter plot in Figure 6 shows that there is no linear relationship between each Δ. Each Δ within one train is independent to each other and the behaviour is shown both in sender and receiver side. This also applies for each Δ between trains.

Hence from both these tests, it can be concluded that there is no correlation between the Δ values within one train as well as between trains.

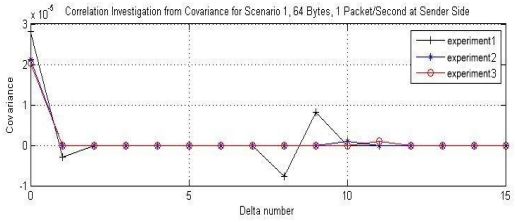

Figure 5. Correlation for 64 bytes, Scenario 1 with 1 Packet/Second

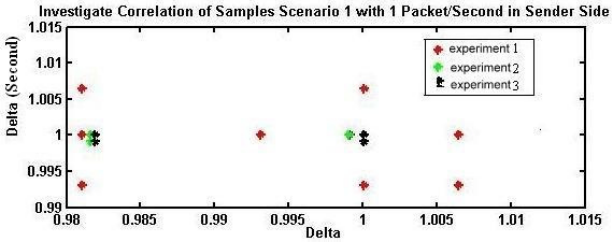

Figure 6. Scatter Plot for 64 bytes, Scenario 1 with 1 Packet/Second

B. Accuracy of Time-Stamping

1) Time-Stamping Accuracy in Variation of Scenarios:

The effect of fully-loaded CPU is investigated in Figure 7. In this figure, Scenario 1 shows steady distribution while there are fluctuations in Scenario 2 which corresponds to decreased time-stamping accuracy. Furthermore, both of the curves start with a low value of Δ in the beginning and attain the ideal value of the inter-packet separation gap from the second or the third packet.

Figure 8 illustrates a comparison of the time-stamping accuracy when sending packets from host OS or guest OS in the presence of minimal CPU load. From the figure, it can be seen that Scenario 1 has more accurate Δ values while there are more deviations in Scenario 3. Even without cross-traffic traffic or fully-loaded CPU, the time-stamping accuracy in the virtualized environment has decreased compared to the non-virtualized environment.

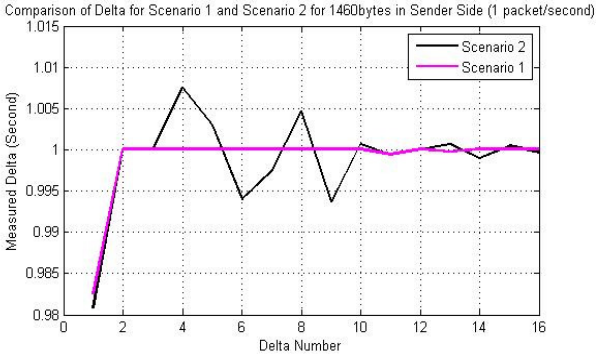

Figure 7. Comparison of Delta for Scenario 1 and Scenario 2 for 1460Bytes with 1packet/s

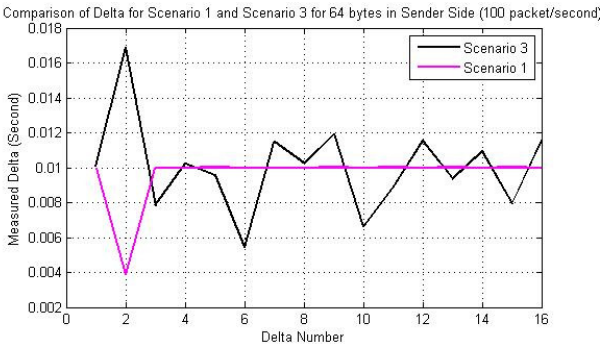

Figure 8. Comparison of Delta for Scenario 1 and Scenario 3 for 64Bytes with 100packets/s

However, when packets were sent at lower packet rate, e.g. 1 packet/second or 10 packets/second, the time-stamping performance of Scenario 3 is better compared to Scenario 2

In Figure 9, the comparison of several scenarios where the guest OS is used to send the packets were investigated. Scenario 3 gives the highest accuracy of the time-stamping, whereas in most cases, Scenario 4 shows less accuracy compared to Scenario 5.

However, it can be concluded that increasing CPU load inside host OS while sending packet from the guest OS results in lower time-stamping accuracy compared to increasing the CPU load of the guest OS and sending packet at the same time.

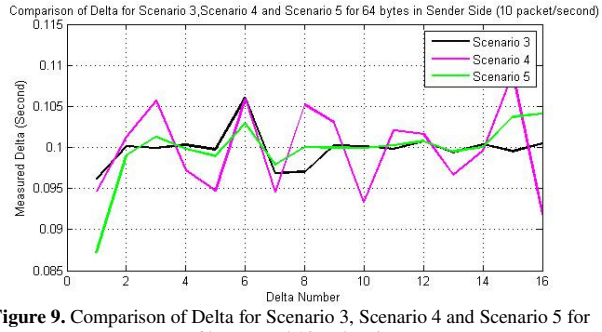

Figure 9. Comparison of Delta for Scenario 3, Scenario 4 and Scenario 5 for 64Bytes and 10packets/s

Accuracy of the time-stamping in the presence of two guest OSes in one computer, as in Scenario 6, is displayed in Figure 10. One of the guest OS runs application to increase the CPU load to 100% while the other is used to generate and send packets. With Scenario 3 used as a reference of the non-virtualized performance, the figure shows that Scenario 6 has lower precision in time-stamping. It shows that running two guest OSes and having fully-loaded CPU in one of them reduce the time-stamping accuracy.

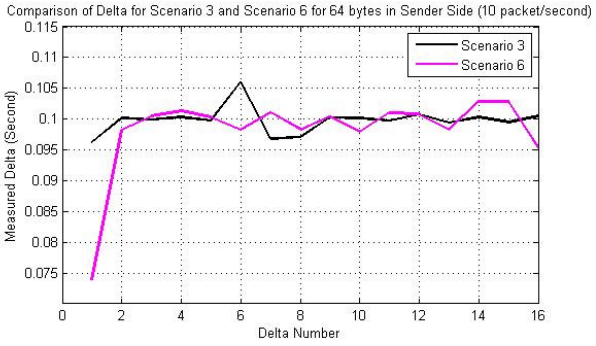

Figure 10. Comparison for Delta in Scenario 3 and 6 with 64 Bytes and 10 Packets/Second

However, Scenario 6 still performs better compared to Scenario 4, but performs worse compared to Scenario 5. It means that the Scenario 4, with fully loaded host CPU and presence of guest OS gives lower time-stamping accuracy.

The reason for this behaviour is the Virtual Machine (VM) that emulates the hardware and creates its own kernel inside the Virtual Machine. Increasing the CPU load of the guest OS means increasing the load in its part of the kernel, while loading host OS up to 100% means increasing the load of the whole kernel in the system. Hence, packets sent from a host OS with 100% CPU load and presence of Guest OS in simultaneous time gives the worst time-stamping performance among all the scenarios.

2) ***Time-stamping Accuracy in Variation of Packet Size:*** The packet size chosen for the experiments are 64 bytes, 500 bytes and 1460 bytes. In order to avoid fragmentation of packets, the maximum payload of 1460 bytes is chosen to keep all the packet sizes within the Maximum Transmission Unit (MTU) for Ethernet. Without any fragmentation of packets, the time-stamping in different packet sizes will be comparable.

Using low packet rate and low CPU load, large packet sizes are more accurate in time-stamping. As packet rate is increased, varying packet sizes and/or fully-loaded CPU do not give significant impact. This trend is noticed from the plot in Figure 11 and Figure 12, which CPU load is low in both cases.

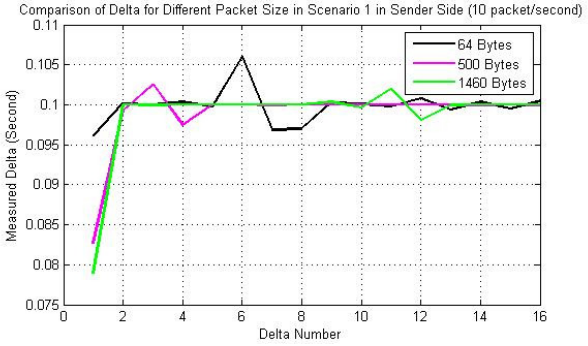

Figure 11. Comparison of Delta for Different Packet Sizes in Scenario 1 with 10packets/s

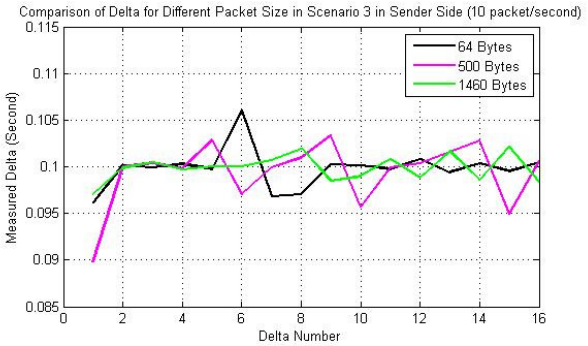

Figure 12. Comparison of Delta Different Packet Sizes in Scenario 3 with 10packets/s

3) Time-stamping Accuracy in Variation of Packet Rate:

Figure 13 shows that low packet rate has better time-stamping accuracy. As the packet rate increased, the time-stamping accuracy decreases. This trend also applies under virtualized environments as shown in Figure 14.

Furthermore, in the virtualized environment, packets can only be sent with rate up to 100 packets/second, while in non-virtualized environment, packet can be sent up to 1000 packets/second. This is because the network appears to have suffered from lost time-stamps [5] when sending packets with higher packet rate.

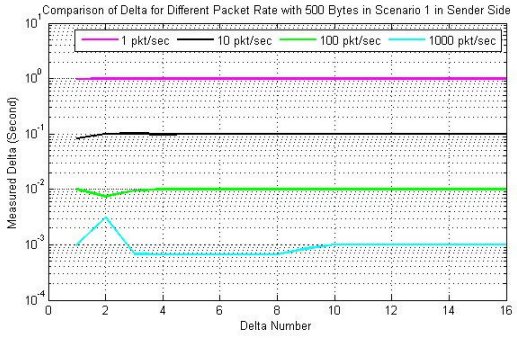

Figure 13. Comparison of Delta Different Packet Rates in Scenario 1 with 500Bytes

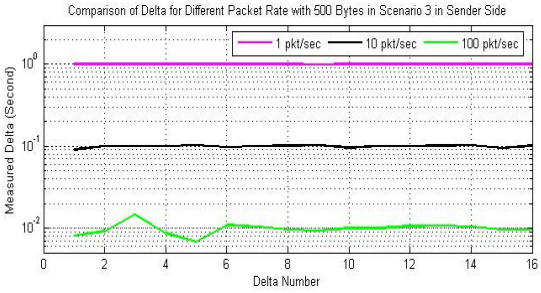

Figure 14. Comparison of Delta for Different Packet Rates in Scenario 3 with 500Bytes

C. Confidence Interval and Prediction Interval

Further analyses are made concerning reliability assurance of the Δ values obtained. The analyses are represented with two types of reliability intervals, that is confidence interval and prediction interval. Confidence intervals are used to indicate the reliability of an estimate. The probability that the values are within this interval is determined by confidence level or confidence coefficient. On the other hand, prediction interval is an estimate of an interval in which future observations will fall, with a certain probability from the previous experiments observed.

The interval range for 95% confidence interval and the 95% prediction interval are computed and fitted against the data collected. Figure 15 shows the confidence interval of 64 bytes packets with rate of 1 packet/second in the sender and Figure 16 shows the range for prediction interval of the same packet parameters.

The range for the prediction interval is wider than the confidence interval since the probability of error for estimating a particular measurement is larger compared to the probability of error for estimating the average value in the future. The reason is to provide a bigger toleration for larger probability of error in prediction interval compared to confidence interval.

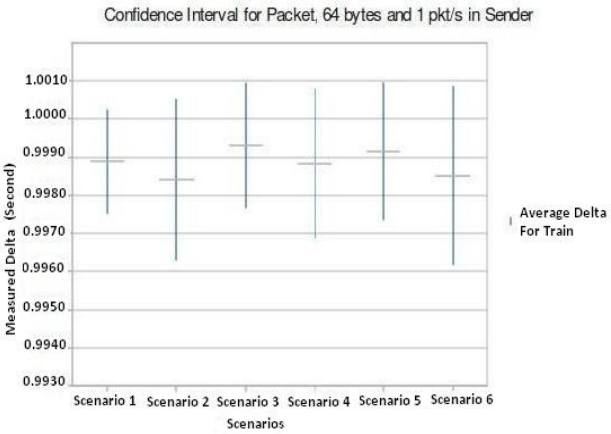

Figure 15. Confidence Interval for 64Bytes and a rate of 1pkt/s in the Sender Side

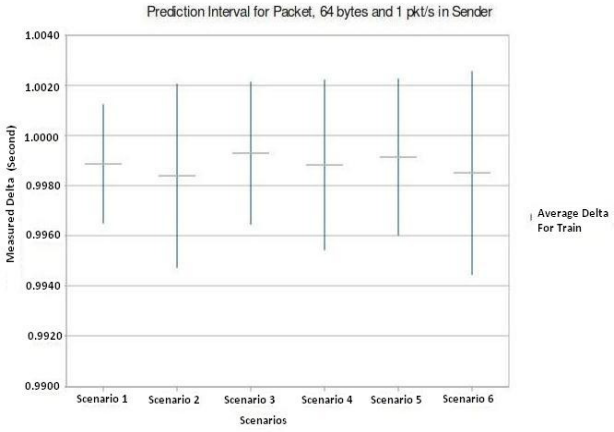

Figure 16. Prediction Interval for 64Bytes and a rate of 1pkt/s in the Sender Side

As the range of both confidence and prediction interval are wider for bigger toleration of error probability under disadvantaged time-stamping environment, wider ranges of confidence interval and prediction interval are needed in the conditions when packet rate is high, packet size is low and fully-loaded CPU. Furthermore, the time-stamping accuracy in virtualized environments also needs wider range of confidence interval and prediction interval compared to the non-virtualized environments since the virtualized environment has lower time-stamping accuracy and also higher Normalized Mean Squared Error (NMSE) as shown in Table 1.

TABLE 1. COMPARISON OF CONFIDENCE INTERVAL AND PREDICTION INTERVAL FOR 500 BYTES PACKETS AND RATE OF 100 PACKETS/SECOND

Scenario	NMSE	Range of Confidence Interval	Range of Prediction Interval
Scenario 1	0.1082	0.000625	0.001082
Scenario 2	0.1814	0.001063	0.001841
Scenario 3	0.4763	0.002786	0.004825
Scenario 4	1.0473	0.006138	0.010631
Scenario 5	0.5897	0.003447	0.005970
Scenario 6	0.8272	0.004845	0.008391

VI. CONCLUSIONS

The statistical properties of the Δ values show the inter-packet separation gap in the packets and between packets are independent and identically distributed which means deviation of one time-stamp will not cause deviation of time-stamp value in another packet. This in turn indicates that the time-stamping value obtained from the kernel are reliable even though there exists a time-stamping interference in one of the packets.

In non-virtualized environment, the measured Δ values are close to their expected values. As the packet parameters and scenario tests varied, their impact on time-stamping accuracy also varied.

There is a small deviation in time-stamping values in presence of virtualized environment. And this performance gets more severe at high packet rate and fully-loaded CPU. When the packet rate is low, large packet size shows very small changes in time-stamping performance and so increasing packet size does not show any significant impact.

Overall, the time-stamping performance in virtualized environments still gives satisfactory result but only under certain conditions, i.e. low packet rate, large packet size and low CPU load.

VII. FUTURE WORK

The packet rate in our research is limited up to 1000 packets/s in the native environment and 100 packets/s in the virtualized environment. The limitations are due to lost time-stamp that occurs because of interrupt-based signalling problem. Interrupt-based signalling is used by NIC to notify the PC of a newly captured packet as stated in [5].

To reduce the load on the host, numbers of interrupts are decreased; hence some NICs do not always generate an interrupt for every packet that is received. For interrupt rate reducing purposes, NICs may generate an interrupt after several packets have been received. Thus, there is only one time-stamp available for multiple packets, especially at high packet rate.

Furthermore, with a high rate of packets sent in high-speed network nowadays, capability of time-stamping with frequency up to 8000 Hz as stated in [9], should have been achieved. Further investigation is needed to analyze several other causes that could impact on lost time stamping.

ACKNOWLEDGMENT

We would like to thank Johan Sandberg from Department of Mathematics at Lund University for the discussions on statistics properties for this research.

REFERENCES

[1] Lee Cosart, *Studying Network Timing With Precision Packet Delay Measurements*, 40th Annual Precise and Time Interval (PTTI) Meeting, 1 December 2008.

[2] C. Dovrolis, P. Ramanathan, D. Moore, *Packet Dispersion Techniques and Capacity Estimation Methodology,* IEEE/ACM Transaction on Networking, December 2004.

[3] A. . Botta, A. Dainotti, A. Pescape, *Multi-protocol and Multi-platform Traffic Generation and Measurement,* INFOCOM 2007 Demo Session, Anchorage, Alaska, USA, May 2007.

[4] S.Ekelin, M.Nilsson, Erik Hartikainen, A. Johnsson, et.al, *Real Time Measurement of End-to-End Available Bandwidth Using Kalman Filtering,* IEEE/IFIP NOMS, Vancouver, 2006.

[5] Stephen Donelly, *Endace DAG, Time-Stamping Whitepaper,* Endace Technology Ltd, 2007.

[6] The D-ITG, Distributed Internet Traffic Generator website. [Online]. Available: http://www.grid.unina.it/software/ITG/ChangeLog.php

[7] The Wireshark website. [Online]. Available: http://www.wireshark.org

[8] The Statistics: Covariance and Correlation website. [Online]. Available: http://investing.calsci.com/statistics3.html.

[9] (December 2009) The RTP Statistics website. [Online]. Available: http://wiki.wireshark.org/RTP_statistics/.