ISPCS 2008 – International IEEE Symposium on Precision Clock
Synchronization for Measurement, Control and Communication
Ann Arbor, Michigan, September 22–26, 2008

Semiconductor Manufacturing Equipment Data Acquisition Simulation for Timing Performance Analysis

Ya-Shian Li-Baboud[1], Xiao Zhu[2], Dhananjay Anand[2], Sulaiman Hussaini[2], James Moyne[2]
[1]Semiconductor Electronics Division,
National Institute of Standards and Technology, Gaithersburg, MD 20899-8120
[2]Engineering Research Center for Reconfigurable Manufacturing Systems,
University of Michigan, Ann Arbor, MI 48109-2125

Abstract *– The ability to acquire quality equipment and process data is important for future real-time process control systems to maximize opportunities for semiconductor manufacturing yield enhancement and equipment efficiency. Clock synchronization for accurate time-stamping and maintaining a consistent frequency in trace data collection are essential for accurate merging of data from heterogeneous sources. To characterize the factors impacting data collection synchronization and performance, a configurable fab-wide equipment data acquisition (EDA) simulator has been developed. By understanding the factors impacting clock synchronization and accurate time-stamping, the simulator is used to identify and explore methods to mitigate the latencies and provide guidance on accurate time-stamping for equipment data acquisition systems.*

Keywords *–time synchronization, data acquisition, semiconductor manufacturing, data quality, Equipment Data Acquisition standard*

I. INTRODUCTION

In order for technological advancements in semiconductor manufacturing to be economically viable, yield improvement must be achieved in the face of tighter processing requirements. Improved yield and throughput can be achieved in large part through the use of effective real-time control based on the ability of future Advanced Process Control (APC) systems to access quality data in a timely and accurate manner. The shrinking process tolerances due to decreasing device sizes and increasing chip complexity as well as the advent of larger wafer sizes, such as 450 mm wafer processing, can all benefit from real-time process control through integrated and in-situ metrology.

The data gathered from the sensors in the equipment must be rapidly transmitted and processed by the APC system in order to realize real-time control of processing parameters. This data must be synchronized so that, at minimum, ordering of information is preserved at remote nodes analyzing the information. Realizing the importance of high-speed data acquisition, the Semiconductor Equipment and Materials International (SEMI), the standards organization for the semiconductor manufacturing industry, has developed the Equipment Data Acquisition (EDA) standard to govern the communication of real-time equipment and metrology diagnostics information in the semiconductor factory [4,5,6,7].

To characterize the factors impacting data synchronization and network degradation in the semiconductor manufacturing industry, the National Institute of Standards and Technology and the University of Michigan developed a scalable and configurable fab-wide EDA system simulator. The simulator is designed to enable analysis of standardized factory data communication with respect to data quality aspects such as network latency, impact of data collection frequency, and impact of time synchronization and time-stamping on rapid data fusion from distributed factory data sources. Additionally, the performance of data collection and time-stamping at the equipment level is also assessed.

A. SEMI EDA

SEMI Interface 'A', also known as the EDA standard, provides a suite of specifications to facilitate communication between equipment data sources and various equipment and factory control systems [4,5,6,7]. The interface utilizes eXtensible Markup Language (XML) messaging over HTTP connections. Typically, the EDA server consists of a piece of equipment, while the EDA client is a host system that gathers data for further analysis in order to feed information into control systems. The EDA client generates Data Collection Plans (DCPs), which dictate what types of data, frequency, and other parameters the APC or similar software analysis tool would need. The EDA server would respond with Data Collection Reports (DCRs) based on the specifications in the DCP. The EDA interface provides a significant data source to APC systems requiring data to perform fault diagnosis, process yield optimizations, and virtual metrology. One of the significant challenges of APC is the ability to ensure quality data from the EDA interface. Data quality aspects include the ability to acquire data in a timely manner and at consistent frequencies. Context data, such as time-stamps associated with an EDA parameter data can also have significant impact on accurate analysis of the data for APC. For example, to perform fault diagnosis it is necessary to determine the exact sequence of events, which requires accurate timestamps. Poor data quality can result in costly errors in control decisions made by the APC applications. Therefore, by developing the simulator to characterize aspects of data quality would provide industry with recommended practices for optimizing data collection.

978-1-4244-2275-3/08/$25.00 ©2008 IEEE

B. Timing Requirements

In the EDA standard, there are three types of data collection modes, namely, *event*, *exception*, and *trace*. Event data are taken the moment an event occurs, such as a process end point. Exception data are taken when an error or warning arises in the manufacturing process or the equipment. Event and exception data can benefit from accurate time-stamps for cause-effect correlation in applications such as fault detection classification (FDC) and end-point detection. With trace data, data points are taken continuously for a time interval at specified frequency; this mode also requires accurately synchronized clocks in order to ensure data are taken at consistent frequencies for optimal results.

Most events in the factory occur at frequencies or durations of 10 ms or greater. Therefore a general requirement for time synchronization and time-stamping is about 1 ms accuracy, since it is ideal to have a measurement process that is one order of magnitude better than the desired resolution [1]. However, certain fault events, such as arc conditions in physical vapor deposition (PVD) chambers, have durations as brief as 1 to 100 microseconds [2]. To effectively detect the arc and time-stamp their occurrence, it is necessary to support microsecond-level resolution timing or better for specific modules or sub-systems within the equipment.

C. EDA Simulator

The objective of the simulator is to characterize the factors impacting data quality including network, clock synchronization and time-stamping performance for specific equipment configurations, network traffic patterns, and data acquisition protocols used by industry equipment. The architecture, as shown in Figure 1, allows the simulator to provide insight to the different data quality issues, including clock synchronization and time-stamping issues, at each level, from the factory control to the data Input/Output (I/O) device levels where the sensors and actuators reside. At the control level, there is a simulator controller along with a Network Time Protocol (NTP) factory time server. The simulator controller is intended to serve as a means to specify a simulation configuration through a user interface via a single point of control. The NTP time server provides a synchronized time to all the simulated EDA servers. The simulated EDA servers are part of the equipment interface, and therefore reside at the equipment tool level.

The time synchronization protocol, Precision Time Protocol (PTP - based on IEEE 1588 version 1) have been applied at the equipment subsystem level of the simulator using IEEE 1588 network cards. Since the cards only provided hardware time-stamping of the PTP messages, the data was time-stamped at the application layer. The results had shown that as the interval between transmitted packets increased, the standard deviation of the delay decreases. Therefore to ensure 1 ms or better accuracy in time-stamping required data transmission rates of 1 s[3]. Once the network, clock synchronization and time-stamping factors impacting data quality are determined, the simulator is intended to be used in the future for testing various methods to improve these aspects of data quality including timely arrival and processing of data, precision clock synchronization and data time-stamping for meeting current and future industry requirements.

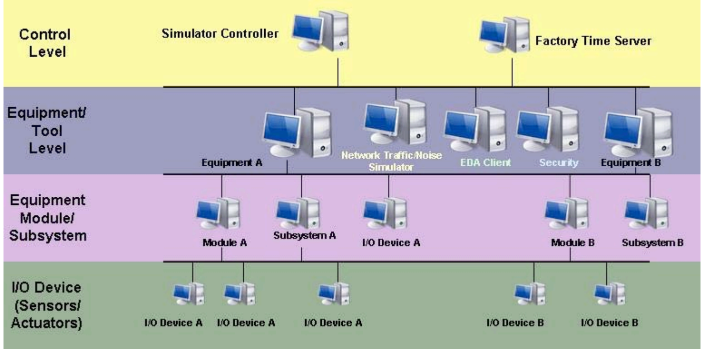


Figure 1. Factory Simulation Architecture

II. DESIGN AND IMPLEMENTATION

A typical semiconductor fabrication facility houses hundreds of process equipment, where each piece of equipment is capable of acting as EDA servers. Therefore, to provide an improved perspective of factory-wide data collection, the intent was to add scalability by enabling simulation of multiple pieces of equipment.

The simulator is implemented using Java on the Windows platform. However, due to limitations in time-stamp and scheduling resolution of Java, C++ with Java Native Interface (JNI) was used to enable better time-stamping resolution of 1 ms. In this initial phase of the simulator development [3], the EDA communication infrastructure is implemented to be able to generate and transmit DCPs and DCRs [7]. The simulator interface enables generation of various types of factory data including event reports, trace reports, and exception reports. The EDA server simulator is able to generate data at specified rates and to time-stamp the data. Noise generators in the form of “dummy” EDA nodes as well as configurable traffic generators are used to assess the impact of network loading.

During the initial testing, it was found that due to limited computing capacity of a normal desktop computer with dual-core 2.13 MHz Central Processor Units (CPUs) and 2 GB of Random Access Memory (RAM), the present architecture of the simulator cannot be scaled up by using threads or processes implementing the full EDA server simulator. With only smart nodes simulated on a single computer, the computer’s CPU rapidly reached full usage with 10 nodes. Thus in order to simulate a more realistic factory environment on a limited number of computers, the architecture was modified, with the objectives to allow the support of approximately 200 EDA servers per computer, and to allow for scalability by supporting the addition of as many computers as necessary for enabling a flexible simulation configuration. These objectives were realized by developing a simulation design that supports two types of servers, namely "smart equipment" and "dummy equipment" as shown in Figure 2.

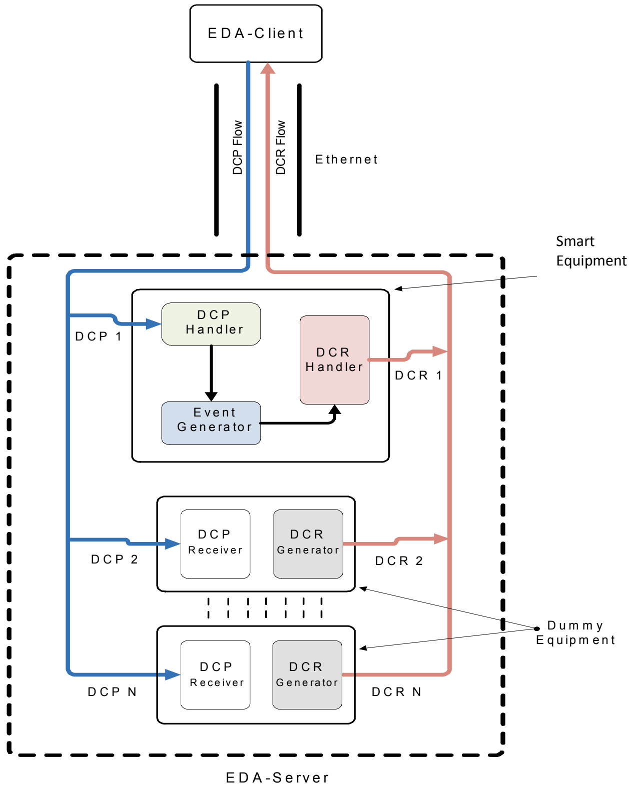


Figure 2. A scalable EDA simulation architecture

The "smart" equipment enables a detailed study of clock synchronization, time-stamping, and CPU loads. The simplified or "dummy" equipment provides a more realistic model of the network loads by simply sending pre-generated DCRs to the client, therefore avoiding the overhead of message processing, such as marshalling and unmarshalling of DCPs and DCRs. Removing the DCP and DCR processing enables the simulator to run multiple equipment servers, within the constraint of the computing power of a single desktop computer. Moreover it was also found that sending the pre-generated DCRs does not require a significant amount of CPU time, so scaling the new module is quite practical.

With respect to the programming architecture, Java threads are used to implement the multiple equipment servers. The factory equipment server architecture is distributable across multiple computers as necessary to support the number of equipment simulators required. Each equipment server in the scalable equipment simulator is a single thread and communicates on a separate port. The simulator allows the user to specify the number of equipment servers as well as the frequency of DCRs to be sent across the network.

Once the simulation system enables multiple pieces of equipment to be simultaneously rendered, a factory time synchronization server can be deployed to determine the impact of various methods of data collection on CPU loads, network traffic, clock synchronization and data time-stamping. Similarly the simulator also examines the impact of network traffic caused by EDA and how various levels of network traffic affect NTP time synchronization accuracy.

As shown in Figure 3, the current simulation system is comprised of an EDA client which receives the generated data reports, an EDA server node which simulates the smart and dummy equipment servers, a noise generator which can be configured to disseminate specified amounts of network traffic for examining performance capabilities, a noise client to configure the amount of traffic to send, an NTP time server, and a network analysis tool, WireShark [8].

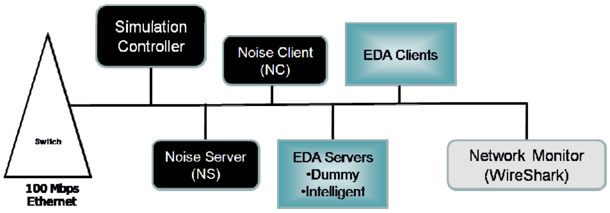


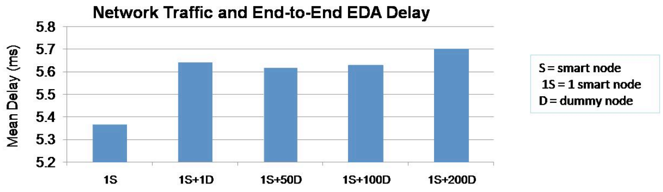
Figure 3. EDA Simulation System Components

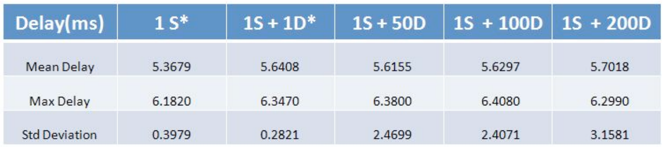
III. RESULTS

Using the EDA simulator, several tests were conducted to determine the impact of the equipment data collection on end-to-end network performance and ultimately, its affect on network time synchronization and time-stamping. The main performance bottlenecks lie at the higher levels of the end nodes including the application software and operating system based on initial tests using several instances of the smart equipment on a single computer. Processing of data reports, such as marshalling and unmarshalling XML files require significant computing resources, which can compromise data quality in terms of timely accessibility, time-stamping, and time synchronization. The programming language, operating system, and computing resource availability, can also limit scheduling of data collection frequency. Therefore to ensure the ability to schedule the levels of data collection frequency used in current factories, the simulation system must allow enough computing resources. The smart equipment consumes 15-20% of the CPU for marshalling the messages, and therefore was not scalable to simulate factory data collection as full CPU usage was seen after rendering 10 equipment server simulations. The ability of the simulator to generate the number of DCR of the full and simplified equipment EDA server has been verified using the network analysis tool, WireShark[8]. The current implementation can support up to 200 simplified equipment servers, and transmit DCRs at a rate of 250 ms per simulated dummy node.

A. EDA on end-to-end delay

Figure 4 demonstrates the impact of the simulated EDA network traffic on end-to-end mean delay. The delay associated with a single smart node ranged from 5.4 ms with a standard deviation of 0.4 ms with zero dummy nodes to 5.7 ms with a standard deviation of 3.2 ms with 200 dummy nodes. The traffic of 200 dummy and 1 full nodes was equivalent to 8-10 percent network usage. This analysis reveals that *the network delay caused by EDA traffic does not have a significant impact on end-to-end delay for EDA node counts representative of a typical semiconductor manufacturing facility*. However, since the simulation contains multiple nodes on a single computer, it does not take into account the additional switch delays incurred when nodes each have individual network cards. Therefore, additional tests were conducted to determine the equivalent number of EDA nodes to flood the network; these tests are described in the next section





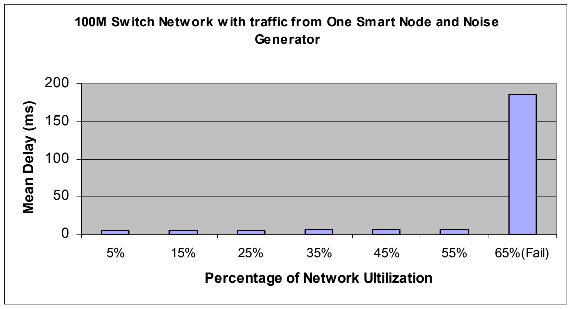
Delay(ms)	1 S*	1S + 1D*	1S + 50D	1S + 100D	1S + 200D
Mean Delay	5.3679	5.6408	5.6155	5.6297	5.7018
Max Delay	6.1820	6.3470	6.3800	6.4080	6.2990
Std Deviation	0.3979	0.2821	2.4699	2.4071	3.1581

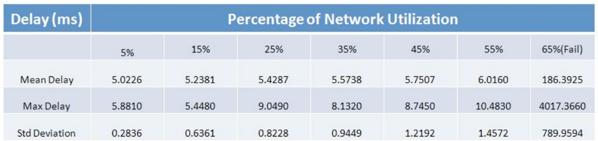
Figure 4. Network traffic of scalable equipment generator

B. Switch v. hub networks

To compare the performance impact of switch and hubs, a 100 Mbps switch and a 10 Mbps hub was used. Ideally, the experiment would utilize a 100 Mbps hub; however, since the initial focus was on performance degradation patterns, it was not mandatory to have a hub with the exact bandwidth as the switch. Comparing the performance of switch-based versus hub-based networks, the noise generator was configured to add increasing loads. The network had a single EDA server generating traffic to determine if the DCRs would arrive in a timely manner at the EDA client. In the experiment, the noise generator provided a 5 to 65 percent network utilization in the switch network. Similarly, the noise generator was able to provide 10 to 99 percent of traffic data for the hub network. Overall, the noise generator was able to generate more traffic for the switch network to attain the necessary network utilization to observe and compare the traffic delay and performance degradation patterns when approaching network reliability limits. The time of arrival of messages were measured using the network monitoring tool, WireShark. Failure is currently defined as greater than 10 ms delay and jitter, where lost data transmissions are no longer negligible for most APC applications.

Switched networks provided more deterministic arrival of data than hub-based networks. In describing network performance, failure occurs when significant delays are incurred. The failure of switched networks degrades suddenly as depicted in the sudden, prolonged delay and jitter shown in Figure 5; the failure occurs at 65 percent network utilization, which is equivalent to 1400 EDA nodes transmitting about 830 bytes at 10 Hz each. Actual failure of switched networks is close to 100 percent net capacity, while this graph plots against gross capacity, which includes switch delay against capacity.

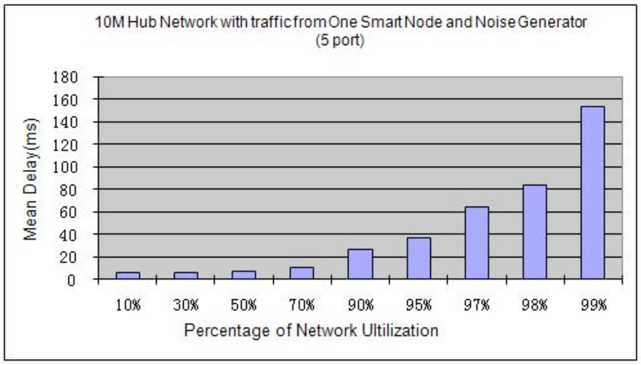


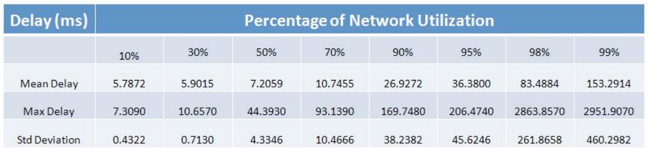


Delay (ms)	Percentage of Network Utilization						
	5%	15%	25%	35%	45%	55%	65%(Fail)
Mean Delay	5.0226	5.2381	5.4287	5.5738	5.7507	6.0160	186.3925
Max Delay	5.8810	5.4480	9.0490	8.1320	8.7450	10.4830	4017.3660
Std Deviation	0.2836	0.6361	0.8228	0.9449	1.2192	1.4572	789.9594

Figure 5. Network utilization for 100 Mbps switch

In the hub network, the failure occurs more gradually than in the switched network. Significant delays begin at 50 percent network utilization, but do not fail until 70 percent.





Delay (ms)	Percentage of Network Utilization							
	10%	30%	50%	70%	90%	95%	98%	99%
Mean Delay	5.7872	5.9015	7.2059	10.7455	26.9272	36.3800	83.4884	153.2914
Max Delay	7.3090	10.6570	44.3930	93.1390	169.7480	206.4740	2863.8570	2951.9070
Std Deviation	0.4322	0.7130	4.3346	10.4666	38.2382	45.6246	261.8658	460.2982

Figure 6. Network utilization for 10 Mbps hub

The degradation shown in Figure 6 depicts a more gradual curve, rather than sudden degradation in the switch network. However, in an actual 200-node EDA network, the failure would occur sooner, as each node would have different network addresses and the potential for message collisions and retries would be much higher.

C. NTP Clock Synchronization Performance

Once the maximum network utilization was established for EDA data collection, the EDA simulation system was used to understand the feasibility of clock synchronization given the EDA network traffic. A basic NTP time server was deployed in the network to synchronize the full EDA server node and the simplified EDA server node. NTP was configured to run at synchronization intervals of 16 s for a period of 30 minutes to one hour before taking the offset and jitter values. The results, shown in Figure 7, indicate that the NTP offset can vary greatly with a potentially long settling time, depending on the how synchronized the clocks are initially as well as network traffic. The convergence times can be attributed to the NTP synchronization algorithms such as phase-lock loop (PLL).

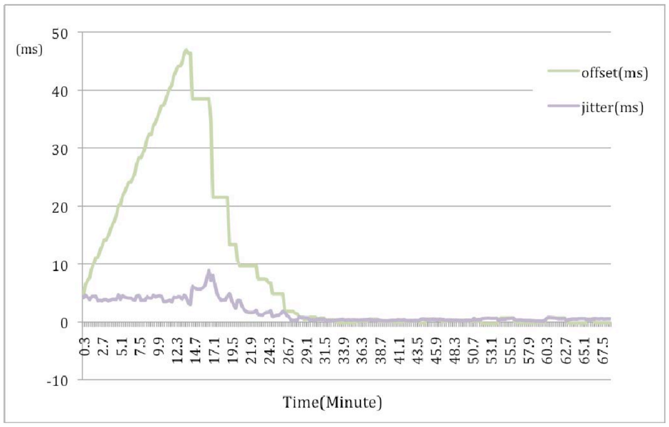


Figure 7. NTP stabilization

Using the NTP query tool, offset and jitter between the EDA servers were measured at 16 second intervals (Figure 8). The offset for a single full and simplified node was in the range of 0.4 to 0.6 ms and jitter ranging from 0.2 to 0.7 ms. Similarly, with up to 200 simplified nodes, the network traffic continued to support levels of synchronization offset below 1 ms, although with slightly greater jitter, some above 1 ms. The offset value denotes the difference in timestamp value between the NTP timeserver timestamp received by the client and the client's clock value.

The results of NTP synchronization analysis in the EDA environment indicate that NTP synchronization is sufficient to address the time synchronization requirements currently in semiconductor manufacturing, however attention should be paid to the settling time issue. Further the impact of application delay and the point in the end-to-end transmission where the data time stamping takes place remain as key issues to address.

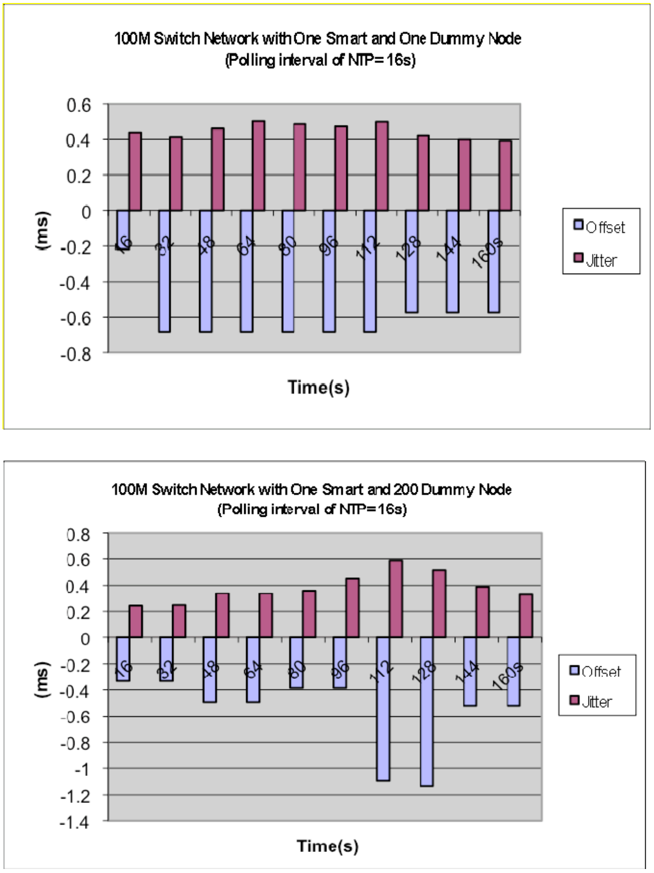


Figure 8. Comparison of NTP offset and jitter in 100 Mbps switch network

IV. CONCLUSION

The current simulation tool provides a scalable capability to determine the impact of delay and network congestion on factory data collection, time synchronization, and time stamping for accurate data consolidation. The simulation architecture allows for an initial assessment of end-to-end network performance based on EDA traffic. The programming language and operating system significantly impact time-stamping at the equipment level, and a language solution was chosen that minimizes this drawback within the restrictions of the Windows operating system. End-to-end delays for EDA messaging was measured as a function of network loading. The network loading was further analyzed as number of EDA nodes that would produce network failure. The accuracy of the clock synchronization between the equipment was analyzed based on various network loads and network type.

Based on the results, it is concluded that higher level software delay dominates over network delay in typical EDA communications. Also, factory EDA traffic does not cause significant network congestion in today's factories with 100 Mbps switched Ethernet. However, as data collection requirements increase, it is necessary to ensure the network remains below saturation in order to allow data to arrive timely manner and similarly, to ensure a sufficient level of accuracy for NTP synchronization. It is also concluded that hubs should be avoided in favor of switched networks in real-time data collection environments as hubs are susceptible to traffic bursts which compromises determinism and lead to poor data quality and clock synchronization accuracy.

It is further concluded that NTP time synchronization is sufficient to the task of EDA synchronization in scenarios typical of today's full-factory semiconductor manufacturing scenarios; however, care should be taken to be aware of potentially long NTP convergence times. Specifically, NTP synchronization is generally sufficient for equipment level applications as the simulation system indicated clock offsets between equipment simulators well under 10 ms and often below 1 ms.

However, to ensure accurate frequency and timestamps, NTP synchronization must be stable before performing data collection. It is recommended to allow NTP to continuously run when possible to avoid long startup stabilization times. In verifying a 1 ms accuracy capability, it will be necessary to use the simulator to continue exploring methods to ensure the offset and jitter remain below 1 ms given the current simulator environment.

Additionally, for specialized applications such as arc detection, more accurate synchronization will be needed. At the subsystem level, IEEE 1588 network cards can be employed for more precise synchronization between modules within the equipment to achieve great time synchronization accuracy. However, commercial products were not available to the experiment to perform data time-stamping at the hardware level. Attaining 1 ms time-stamping accuracy was done at the application layer, and required sufficient interval between transmission of data packets, around 1 s, to ensure the time-stamp delay is below 1 ms. Hardware time-stamping is required to improve the time-stamping capability, especially for high frequency data collection. To meet cost and performance tradeoffs for accommodating different synchronization requirements, a hybrid solution for clock synchronization is recommended.

The current simulation system has some limitations. Foremost among these, as all the EDA server traffic is sent from a single computer, it does not fully reflect the impact on switched networks. This limitation will be addressed in future versions of the simulator. The accuracy of current performance measurement methods and exploration of other measurement methods to characterize time-stamping and time synchronization delays would also be beneficial. Other future simulator extensions will include a study of wireless systems to evaluate the determinism for factory data collection and time-stamping, the study of embedded hardware to analyze clock synchronization and data time-stamping at the equipment sub-system level, the exploration of data time-stamping based upon utilization of a hybrid IEEE 1588 version 2 and NTP approach that optimizes cost as well as clock synchronization and time-stamping performance.

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States. *Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.*

REFERENCES

[1] Kalappa, N., Moyne, J., Parrott, J. and Li, Y. "Practical Aspects Impacting Time Synchronization Data Quality in Semiconductor Manufacturing," *Proceedings of AEC/APC Symposium*, October 2006.

[2] Parker, J., Reath, M., Krauss, A.F., Campbell, W.J. "Monitoring and Preventing Arc-Induced Wafer Damage in 300mm Manufacturing," *2004 International Conference on Integrated Circuit Design and Technology*, p.131-134.

[3] Anandarajah, V., Kalappa, N., Sangole, R., Hussaini, S., Li, Y., Baboud, J., and Moyne, J. "Precise Time Synchronization in Semiconductor Manufacturing," *2007 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, p.78-84.

[4] SEMI E120 Common Equipment Model Specification

[5] SEMI E125 Equipment Self Description Specification

[6] SEMI E132 Equipment Client Authentication and Authorization Specification

[7] SEMI E134 Data Collection Specification

[8] Wireshark website: http://www.wireshark.org