ISPCS 2009 International IEEE Symposium on Precision Clock
Synchronization for Measurement, Control and Communication
Brescia, Italy, October 12-16, 2009

EDA performance and clock synchronization over a wireless network: Analysis, experimentation and application to semiconductor manufacturing

D. M. Anand, D. Sharma, Y. Li-Baboud and J. Moyne

Abstract—**Shrinking process tolerances due to decreasing device sizes and increasing chip complexity in semiconductor manufacturing are motivating efforts to improve methods of Equipment Data Acquisition (EDA). Prior work shows that the lack of precise time-stamping and clock synchronization is a critical hindrance to reliable data acquisition and real-time process control systems. The ultimate goal in the development of EDA standards for performance is to meet industry demands such as modularity, reconfigurability, decentralization, interoperability and low cost. While precision in timing addresses some of these requirements, the need for scalable modularity, flexibility and lower cost is also responsible for a recent interest in performing EDA functions over wireless networks. This paper presents an analysis of data acquisition and clock synchronization performance over a wireless network. Clock synchronization accuracy in a real world EDA environment was determined by using a configurable fab-wide EDA system simulator designed to recreate the specific equipment configurations, network traffic patterns, and data acquisition protocols used by industry standard equipment. The data packets from the EDA simulator were routed through a wireless network testbed described in Section III. The results show that while wireless networks are significantly noisier in terms of time delay variation, a sufficient level of time-synchronization among wireless nodes should be achievable, given additional improvements for meeting semiconductor manufacturing requirements. Hence, time stamping of EDA data can greatly improve data quality, and open up avenues for the design of controllers that are better suited to leverage wireless networks.**

Keywords: clock synchronization, semiconductor manufacturing, data quality, Equipment Data Acquisition standard, NTP, wireless network, time stamping

I. Introduction

Wireless networks are widely implemented as a part of the IT infrastructure in modern manufacturing plants and are able to provide the quality of service required for network communication where there are no hard, real-time constraints. Most wireless protocols are designed to assure data integrity and a nominal data flow rate, their mechanism for medium arbitration and collision recovery result in non-deterministic delays. While multiple standards and technologies exist for wireless networks, we will focus our analysis on IEEE 802.11g (Wi-Fi), highlighting the effect of its use for Equipment Data Acquisition (EDA) systems and for clock synchronization.

In this paper we will first quantify the magnitude and jitter of the delay in transmitting process control messages in a plant floor environment over an IEEE802.11g wireless interface. We will then discuss the benefits of time stamping for control communication, and the performance of the commonly adopted Network Time Protocol (NTP) clock synchronization service over wireless. The extension of these concepts to the IEEE-1588 standard for precision clock synchronization and low level time stamping are then discussed as part of conclusions and future work.

A. The SEMI EDA specification

SEMI interface 'A' provides a suite of specifications [1], [2], [3], [4] for communication between data sources on the plant floor (e.g., equipment) and data consumers. In a factory control environment, data sources or servers are devices that compile and report process control data in formatted Extensible Markup Language (XML) reports called Data Collection Reports (DCRs). These reports are generated in response to a query sent to the device from a client, such as a data storage system or a supervisory plant floor controller. In SEMI EDA terminology this query is called a Data Collection Plan (DCP) since it also carries information about the format, frequency and type of data to be sent out in the DCR. The specification allows multiple simultaneous connections to be maintained between clients and servers, employing the SOAP (Simple Object Access Protocol) messages over HTTP or HTTPS connections.

B. Timing requirements for EDA systems

Systems built around the SEMI EDA architecture are tasked with multiple functions including fault detection, discrete event monitoring and virtual metrology. In order to satisfy these functions three data types are typically used; *event, exception* and *trace* . Event data are generated in response to trigger events determined in the control logic. Exceptions are recorded and reported when faults or warnings arise in the server or the manufacturing process. Both these data types are time sensitive, especially when detected faults have to be correlated with other logged events and then classified to determine cause-effect relationships. Trace data are generated from continually polled processes. High throughput and determinism are required in a network carrying trace data to ensure a consistent sampling rate and good measurement fidelity. In all three cases, accurately recording the order in which reports were generated is vital. The failure to do so could result in disingenuous error reports leading to faulty and sometimes costly conclusions.

Most sampled events in a semiconductor manufacturing process occur at intervals of 10 milliseconds or larger [16]. In order to capture these events, the authors in [10] suggest that clock synchronization between EDA nodes be accurate within

1 millisecond. This requirement is continually shrinking with tighter process control norms and pervasive distributed logic. For example, the authors in [15] identify fault conditions such as electric arcing in the semiconductor fabrication process that occur over a time frame of 1 to 100 microseconds and state that these fast events may also have to be accurately reported over the EDA system in the future [12]. The experimental verification of EDA performance presented in [12] however limits the nominal poll rate for the simulated sensors to 4 Hertz (one sample every 250ms). The EDA simulation in this paper uses the same sampling rate.

C. The IEEE 802.11g protocol

The 802.11 or 'Wi-Fi' specification is designed to extend wireless access to the Ethernet infrastructure [6]. The protocol typically operates with a bandwidth of tens of Megabits per second with a range of about 100 meters. Wi-Fi is designed to operate seamlessly with all Ethernet protocols and readily supports the EDA message frame. The implementation of a Wi-Fi wireless cell could be ad-hoc or self organizing; however, most applications use a centralized access point executing a coordination function between several remote nodes. Given the range of 100 meters, a central access point can cover an entire production floor even in larger facilities. This centralized approach fits the EDA architecture discussed in Section I-A, where multiple nodes connect to a data collection hub or client.

IEEE 802.11g is a high-speed extension within the Wi-Fi specification. While being backwards compatible with the IEEE 802.11b, a slower predecessor, it also supports an Orthogonal Frequency Division Multiplexed (OFDM) multi-carrier physical layer which is capable of a 54 MBits/s modulation. IEEE 802.11g occupies about 25MHz bandwidth in the ISM radio band, between 2402 and 2482 MHz.

II. EDA over IEEE802.11g

Datagrams in the SEMI specification range in size. Large batch reports approach packet sizes of hundreds of Kilobytes. Equipment self description and client authentication exchanges can also inflate the size of the packets to be communicated. There are, however, several cases where the payloads to be transmitted are fairly small. In the case of trace data, for example, the payload typically carries about 50 variables of 1 Byte each [4]; the resulting payload is therefore about 60 Bytes.

The frame format for an 802.11g transmission includes large overheads in the form of a Preamble, MAC Header and CRC suffix; the overhead for each packet could be as high as 64 bytes. In all variants of the 802.11 specification, large data packets are able to better utilize the available bandwidth since buffer and padding overheads in most protocols change very little with increased data payload. Thus small data sizes result in reduced throughput since more packets are used for the same effective data transmission. Further reductions in throughput may occur depending on the choice of higher-level protocols. Additionally, the use of TCP/IP or an equivalent acknowledged transmission protocol reduces the effective throughput by requiring more network bandwidth for acknowledgement messages and application layer retries. The drop in throughput is not purely a function of the large packet overhead. With the centralized access point implementation, the mandatory coordination function requires all the client nodes to compete for the medium using the non-deterministic CSMA/CA algorithm. Even when the medium is available, transmitted packets are subject to some mandatory delays. For example, an inter-frame spacing of 50 μs is applied before every transmission attempt to accommodate time inaccuracies.

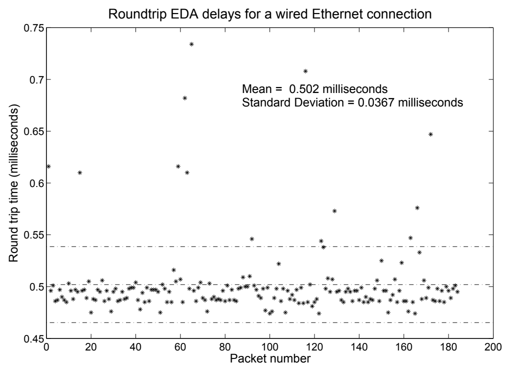

Fig. 1. Round trip delays for 60 Byte packets over a wired Ethernet connection between two nodes. The mean value is 0.502 milliseconds, and the standard deviation is 0.04 milliseconds.

Using trace data with 50 variables per data frame as an example, we can measure the round trip delays for packets with a payload of 60 Bytes. This will allow us to compare the nominal performance of both the wired Ethernet and wireless Wi-Fi interface, in near ideal conditions, with just one client-server pair. Figure 1 shows typical round trip delays for a 60 Byte data-frame collected over a wired Ethernet connection. Figure 3 illustrates this round-trip delay measurement over a wireless connection. The analysis shows an order of magnitude difference in mean round trip delay (compared to wired) and, more significantly, two orders of magnitude difference in the standard deviation (or jitter) of time delays. A clear contrast is also seen in the nature of the jitter. Histograms of the round trip delays (Figures 2 and 4) show that the delay spread is non-deterministic and characterized by large, sporadically occurring outliers. The round-trip delays computed here are for test payloads and not EDA data. EDA conversations exchanging DCPs and DCRs involve multiple packets and several higher level protocol functions. The performance of the wireless system for EDA data is presented in Section III.

III. EDA simulation over wireless

A. Experimental setup

To study the performance of factory scale EDA over the 802.11g interface, we used the EDA simulator introduced in [12] in conjunction with a wireless network testbed. This simulator is a C++ and Java native interface implementation of the EDA communication infrastructure [4]. The simulator recreates network traffic expected from a real world, factory scale EDA implementation, including various data types such as; exception reports, event reports and trace reports. Further, the simulator can be scaled up to simultaneously

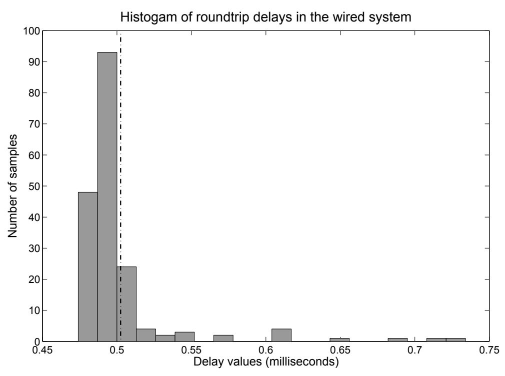

Fig. 2. Histogram of the delay values in Figure 1. The distribution shows most of the delay values clustered about the mean and a small number of outliers clustered around 0.7 milliseconds. The maximum delay is approximately 46 % greater than the mean.

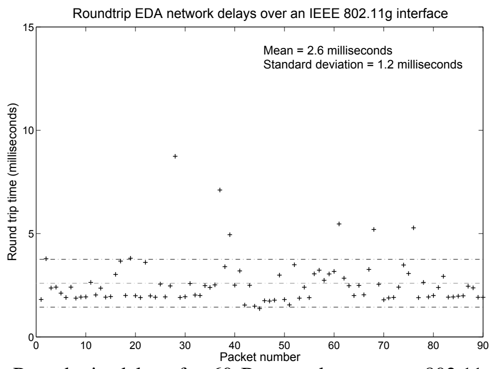

Fig. 3. Round trip delays for 60 Byte packets over a 802.11g interface at a range of 10 meters. The mean value is 2.6 milliseconds and the standard deviation is 1.2 milliseconds.

maintain DCP–DCR exchanges between a client and hundreds of servers. Only one of these servers executes computationally intensive data parsing functions, called the Intelligent node; the rest of the servers called Dummy nodes, are designed instead to respond to DCPs with pre-generated DCRs, thereby avoiding the overhead of message processing. Removing the DCP and DCR processing enables the simulator to run multiple equipment servers (similar to a real factory) on a single desktop computer. Figure 5 shows a functional layout of the the simulator, including the physical wireless hardware

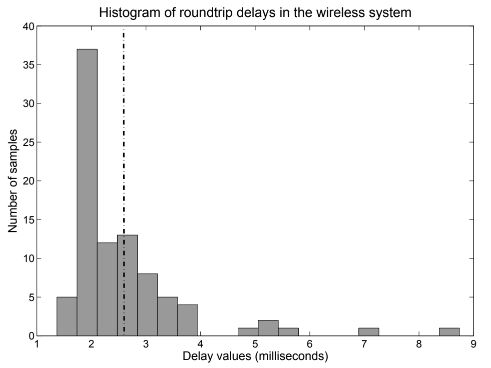

Fig. 4. Histogram of the delay values in Figure 3. The distribution shows the presence of sporadic outliers. The maximum delay in this case is approximately 230 % greater than the mean.

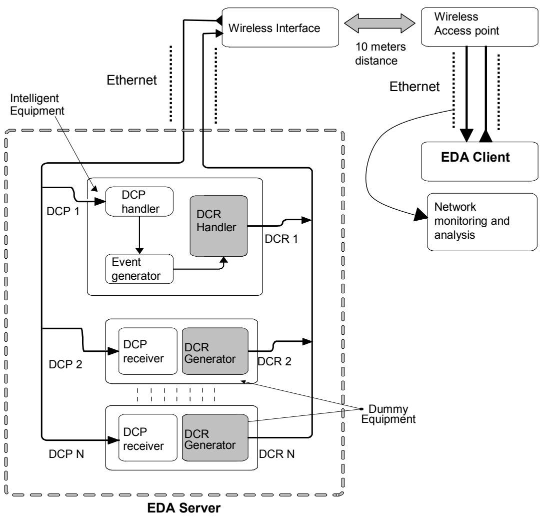

Fig. 5. A schematic showing the functional modules and their interconnections within the EDA simulator. The EDA client module is a combination of the *intelligent* module and several *dummy* modules.

[1] between the server and client side of the simulation.

The wireless component of the communication channel was designed to recreate a best-case operating scenario, with only one paired set of wireless interfaces. All of the available bandwidth was dedicated to the simulation, eliminating the effects of network arbitration with other un-managed wireless nodes. The physical channel was maintained clear of active interference and the nodes were well within the rated distance of 100 meters for 802.11g. The test was conducted in a physical space with similar radio channel properties to the plant floor to recreate passive radio conditions that could be expected.

Network traffic was monitored in real time at the client side of the communication channel. We used analysis software developed in-house in conjunction with the commercial network protocol analyzer *Wireshark*[2], to monitor DCP/DCR exchanges tapped off the wired Ethernet link between the client computer and the Wi-Fi access point. Running an independent protocol parser and delay measurement tool allowed the client to execute EDA functions unhindered.

B. Results

This section uses the following terms to evaluate EDA performance.

- T_{mean}, The mean time for one EDA conversation[3] between client and server.
- T_{max}, The maximum time for a complete conversation.
- T_{min} The minimum time for a complete conversation.
- σ_T The standard deviation of the conversation times.

The results in Table I suggest a marked difference in performance between the wired and wireless case. Experiments

[1]Network components used for these experiments were provided by *Phoenix Contact, USA.*

[2]Wireshark is distributed under the GNU general public license. *www.wireshark.org*

[3]The EDA conversation includes DCP–DCR exchanges and data transmission acknowledgements. 5 TCP packets on average

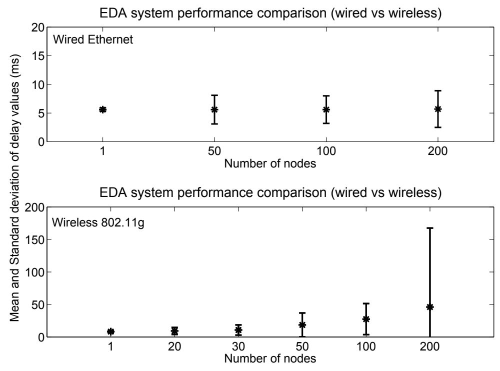

Fig. 6. End-to-end mean delay with increasing number of server nodes.

TABLE I
COMPARISON OF PERFORMANCE PARAMETERS FOR WIRED VS. WIRELESS DATA ACQUISITION AT 4HZ SAMPLING RATE. 1I CORRESPONDS TO 1 INTELLIGENT NODE, SIMILARLY 50D CORRESPONDS TO 50 DUMMY NODES AND SO ON.

Wired parameters	T_{mean}	T_{max}	σ_T
1I	5.4ms	6.2ms	0.4ms
1I+1D	5.6ms	6.3ms	0.3ms
1I+50D	5.6ms	6.4ms	2.5ms
1I+100D	5.6ms	6.4ms	2.4ms
1I+200D	5.7ms	6.3ms	3.2ms

Wireless parameters	T_{mean}	T_{max}	T_{min}	σ_T
1I	9.4ms	12.7ms	7.4ms	1.1ms
1I+1D	8.2ms	12.3ms	7.1ms	1.2ms
1I+20D	9.3ms	39.8ms	7.1ms	5.4ms
1I+30D	10.7ms	43.5ms	7.1ms	7.8ms
1I+50D	18.5ms	82.3ms	7.1ms	18.4ms
1I+100D	27.5ms	85.6ms	7.2ms	23.9ms
1I+200D	46.1ms	402.4ms	7.2ms	121.4ms

conducted on the wireless system show that at 4Hz sampling, the system has both a larger (T_{mean}) (close to double) and an order of magnitude higher (σ_T). The larger mean delay in the wireless case, limits the sampling frequency of the system. From Table I, a system with 200 nodes has delays above 100ms, therefore limiting a system with true 10Hz sampling to fewer than 200 nodes. Figure 6 also shows a large increase in jitter as more nodes are added. The number of nodes, in this case, pertains only to the number of simulated EDA servers in the simulation. If every server were to independently access the wireless medium, then the connection arbitration limits for the Wi-Fi access point and the supporting wireless infrastructure come into play, these effects are beyond the scope of this paper as the assumption here is that the wireless infrastructure is installed to handle a data network of similar bandwidth requirements.

The jitter in delay is the more significant parameter to address since this is a control network, also, jitter directly affects the uncertainty in the clock synchronization algorithm. As the number of nodes increases, the mean delay changes less than the standard deviation of the delay, as illustrated in the second plot in Figure 6. Jitter in the delay affects trace data quality since a definite sampling rate and maintenance of data order across multiple systems cannot be assured. The EDA system must be pessimistically designed to be tolerant of this jitter. For example, in Table I, looking at maximum delay (T_{max}), minimum delay (T_{min}) and mean delay T_{mean}, when there are 100 server nodes; $(T_{max} - T_{min})/T_{mean} = 2.85$. Therefore a variation of 285% in the sampling rate must be accommodated by the system architecture. When T_{max} is twice as large as the sampling interval, the client can potentially receive and log trace samples out of sequence causing an erroneous state. This occurs when there are 200 server nodes present, sampling at 4Hz. Event and exception reporting is also affected by the jitter, since priory of occurrence and cause–effect analysis depends on deterministic data delivery. With undetermined delays, event logs may record events out of order or in an inaccurate state context [16]. Undetermined delays are of critical concern with distributed control systems as well. Synchronized data streams are fundamental to systems that merge data from heterogenous sources. A case is made therefore for time stamping of data packets at all transmitting nodes. This alleviates the need for hard constraints on network determinism.

IV. ARGUMENT FOR TIME STAMPING OF EDA DATA

With accurately time stamped data, the absolute time at which the data was recorded is conserved despite delays in transmission. Since the receiving node can reconstruct the exact time at which the data was stamped, it no longer has to rely on the packet's arrival time to estimate the absolute time of data generation. Network jitter effects are therefore minimized [8]. In the example of trace data, time stamps eliminate the need to make the control system tolerant of large variations in data reporting frequency. The wireless EDA system is therefore an ideal candidate for time stamping. The benefits of time stamping, though, only apply as long as the system operates within the bandwidth limits of the network, i.e, below the saturation region in Figure 7.

Figure 7 shows the change in effective network throughput as the cumulative data rate is increased. The figure shows that effective throughput is linearly related to increasing data rate until 10 Megabits per second. This corresponds to a theoretical maximum of 510 DCR/DCP exchanges per second with a 1Kilobyte payload. Adding overheads and network latency effects on multi-packet exchanges, the theoretical network capacity is approximately 50 server nodes transmitting simultaneously at 10Hz or approximately 126 nodes at 4Hz. Until these limits are met, the network can match the data rate of the nodes, therefore there should be no significant change in the mean delay. A large increase in the jitter is expected, however, as more packets are engaged in non-deterministic network arbitration. Both these suppositions are corroborated by the results in Table I, particularly so in the wireless case. We can conclude that, with time stamps, the system can be operated up to this limit of 126 server nodes without loss in performance.

The accuracy of the time stamps themselves, however, depends fundamentally on the accuracy of synchronization between server and client clocks. Since these clocks are synchronized over the very network where the data is exchanged, the impact of jitter, discussed above, on clock synchronization accuracy needs to be studied.

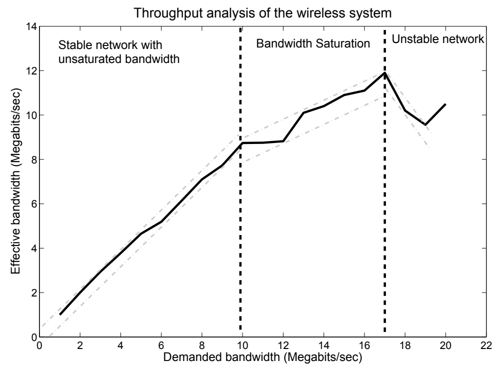

Fig. 7. Figure showing the available effective bandwidth in the wireless connection. The effective bandwidth saturates at about 10 Megabits/sec data rate, the network is unstable at data rates higher than 17 Megabits/sec. Theoretically, 10 Mbps bandwidth corresponds to about 40 EDA servers generating 1Kilobyte DCRs at 10 Hz.

V. Clock Synchronization over Wireless

All network-based clock synchronization strategies require accurate assessment of network delay. These estimates are made by analyzing specific time-calibrated communications between the nodes. We will focus our analysis on one such commonly used network based clock synchronization method, the 'Network Time Protocol' or NTP. The authors in [14] list 'asymmetric propagation delays' and 'fluctuations in network protocol stack' as two of the dominant factors affecting timing accuracy in NTP. Both these effects are heavily pronounced in Wi-Fi networks. Even under ideal physical conditions, there are large outliers in the network delay as illustrated in Figure 3. Under heavy traffic conditions; this jitter is further exaggerated. Looking at the wireless jitter values in Figure 6, we see that, with 200 nodes, σ_T is 2.5 times T_{mean}. Several performance metrics for NTP under adverse network conditions are discussed in [13]. The EDA simulator gives us the ability to study the performance degradation of NTP for wireless EDA networks.

To achieve this, NTP was implemented on two nodes connected over an IEEE 802.11g connection. This connection was also shared by the EDA simulator running 100 server nodes and one client sampling at 4Hz. The results presented in Figures 8 and 9 compare the jitter and offset values computed by the NTP algorithm. Figure 9 shows a 20 times increase in the jitter for the wireless case over wired. The jitter in the NTP algorithm is a weighted moving estimate of the dispersion of the offset values [13]. The same two clocks (computers running NTP) were synchronized in the wired and wireless case to ensure that the jitter characteristics of the internal hardware clock and the software implementation of the local system time were identical in both cases. The increase in jitter therefore is almost entirely an artifact of the jitter in network propagation.

The jitter of 2.8ms in the wireless case is a worst case upper bound on the uncertainty in time accuracy, and is still orders of magnitude better than σ_T. It is however almost three times higher than the clock synchronization accuracy recommended by the authors in [10] and marginally higher than the 2.5ms accuracy required for 100 nodes sampling at 4Hz.

The actual accuracy of synchronization may be much better

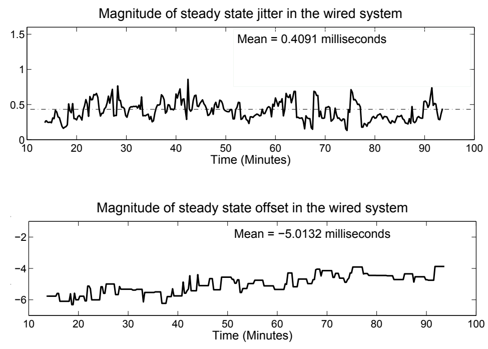

Fig. 8. One node was synchronized to a time source using NTP. The magnitude of time jitter and offset between time queries was recorded over a 90 minute period.

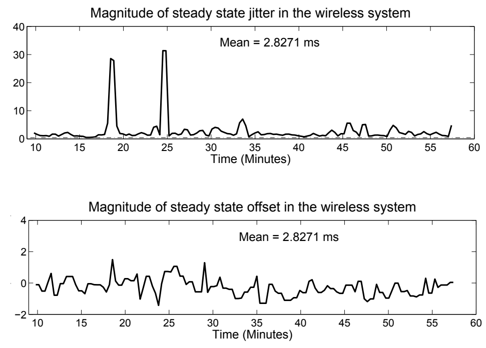

Fig. 9. The synchronization experiment was repeated between the same nodes using an IEEE 802.11g wireless link instead of a wired Ethernet connection.

than 2.8ms since algorithms for smoothing, filtering and classifying noise in the offset estimates are fundamental to the operation of NTP [13]. Time convergence algorithms specifically for real-time systems are also well documented [11]. The NTP data-filtering algorithm, which attempts to improve the offset estimate for a single clock, given a series of round-trip delay measurements, assumes that the network is operating below the knee of the throughput–delay curve (The point on 7, where the bandwidth is saturated). In such a case, the best offset samples should occur at the lowest round-trip delays, since the probability that all NTP packets will encounter busy queues in both directions is very low. NTP therefore employs a minimum-filter to find the best offset estimates. Looking at the round-trip delay measurements in Table I, we see that with increasing number of nodes, the minimum round-trip time T_{min} remains constant at approximately 7.2ms, even as the T_{mean} and σ_T increase many fold. This is an important result, proving that even at high throughput conditions, the minimum-filter in the NTP algorithm will be able select samples to make accurate offset estimates. There are continuing improvements being made to the NTP filter algorithms. For example, the authors in [9] suggest a method to compute the best offset from multiple samples with asymmetric propagation delays. These samples might otherwise have been discarded by the minimum filter. Potentially, this will allow the algorithm to converge to an accurate offset estimate much faster, even while operating under close to saturated or noisy networks.

VI. Conclusions and Future Work

With wireless capabilities, EDA networks would reap significant benefits through lower integration costs and greater flexibility for next-generation semiconductor fabrication facilities. In this paper we have illustrated through experimentation that care must be taken in the migration to wireless as network performance can differ significantly from wired systems. Wireless introduces delays with less determinism, where high levels of jitter occur even under ideal radio conditions. Data shows that while wireless systems provide sufficient bandwidth to support the required traffic in EDA systems with node counts typical of semiconductor facilities, jitter can be significant, which could lead to poor data quality in terms of out-of-order and significantly delayed data. This in-turn can lead to issues with EDA system operation such as false positives in fault diagnostics systems. These data quality issues can be addressed to a large extent by synchronizing nodes utilizing clock synchronization protocols such as NTP and then utilizing precise time stamping to improve data quality and avoid issues such as out-of-order data and varying sampling intervals. However our experimental results indicate that even clock synchronization capabilities are reduced with the move to wireless. Though the currently used NTP algorithm can partially address the increased jitter in offset estimates, the upper bounds on the NTP algorithm operating in the wireless setting presented here are well above the recommended accuracy recommended for 10Hz sampling. This loss of accuracy must be addressed when configuring systems with respect to poll time and minimum level of signal granularity.

These results provide direction to a number of areas for future work. There are techniques and modifications to the NTP algorithm that can improve accuracy of synchronization further, despite the added jitter. There is also ongoing work to improve the native quality of service for time critical communication over IEEE 802.11 wireless interfaces. The IEEE 802.11e standard [7], for example, adds enhancements to the coordination function to allow prioritization of high priority communications on a shared wireless link. The technologies encompassed in the standard include: EDCA (Enhanced Distributed Channel Access) which allows the access point to offer preferential transmit opportunities to nodes with higher priority. This priority assignment is enabled by HCCA (Hybrid Coordinator Function Controlled Channel Access), which allows nodes to communicate a 'traffic class' and desired parameters for bandwidth and jitter. The access point then uses informed scheduling to achieve this quality of service. It is important to note that this strategy hinges on the preferential selection of some nodes over others, an opportunity not always present in a control network. Enhancements to the wireless protocol, like improvements to the NTP algorithm will allow more robust correction estimation and faster convergence to synchronization.

Parallel to the efforts in improving the low-level mechanics of wireless communication and clock synchronization, there is scope for improving system wide time synchronization quality in the semiconductor industry through the development of clock synchronization standards at the application level. Currently a general clock synchronization standard, E148 specifies a basic NTP-based approach and options for clock synchronization of nodes across a semiconductor facility, and identifies a clock object for communication of application level timing information [5]. Research efforts have begun towards enhancing this standard to specify an IEEE 1588 integration standard for (at minimum) low level device synchronization. One idea that has been proposed within this standard effort is that all high speed sensor/actuator/controller devices support 1588 clock synchronization. This standards effort opens up two prominent areas of future research. First, methods of deploying 1588 implementations and time stamping to support various levels of clock synchronization and other precision timing capabilities on sensor systems (with real-time hardware and software) need to be evaluated and best practices developed. Second, once reliable time stamping of sensory and actuation data becomes commonplace, new time-based methodologies for diagnostics and control can be developed to meet the performance requirements of next-generation manufacturing automation solutions.

References

[1] SEMI E120-1.V1104 specification for the common equipment model.

[2] SEMI E125-1.V0305 specification for equipment self description.

[3] SEMI E132-1.V0305 specification for equipment client authentication and authorization.

[4] SEMI E134-1.V1105 specification for data collection management.

[5] SEMI E148-1.V0309 specification for the definition of time synchronization method and format.

[6] IEEE standard for information technology- telecommunications and information exchange between systems- part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications. Specification IEEE Std 802.11-1997, IEEE, 1997.

[7] IEEE standard for information technology- telecommunications and information exchange between systems- Amendment 8: Medium access control (MAC) quality of service enhancements. Specification IEEE Std 802.11e-2005, IEEE, 2005.

[8] John C. Eidson. *Measurement, Control, and Communication Using IEEE 1588.* Springer London, 2006.

[9] O. Gurewitz, I. Cidon, and M. Sidi. Network time synchronization using clock offset optimization. pages 212–221, 2003.

[10] N. Kalappa, J.R. Moyne, J. Parrott, and Y. Li. Practical aspects impacting time synchronization data quality in semiconductor manufacturing. In *Proc. AEC/APC Symposium*, 2006.

[11] H. Kopetz and W. Ochsenreiter. Clock synchronization in distributed real-time systems. In *IEEE Trans. Comput.*, 1987.

[12] Ya-Shian Li-Baboud, X. Zhu, D.M. Anand, S. Hussaini, and J.R. Moyne. Semiconductor manufacturing equipment data acquisition simulation for timing performance analysis. In *Proc. IEEE ISPCS*, 2008.

[13] D.L. Mills. Internet time synchronization: the network time protocol. *Trans. IEEE Communications*, 39(10):1482–1493, Oct 1991.

[14] T. Neagoe, V. Cristea, and L. Banica. NTP versus PTP in computer networks clock synchronization. In *Proc. IEEE ISIE*, volume 1, 2006.

[15] J. Parker, M. Reath, A. F. Krauss, and W. J. Campbell. Monitoring and preventing arc-induced wafer damage in 300mm manufacturing. In *Proc. ICICDT*, 2004.

[16] Anandarajah V., Kalappa N., Sangole R., Hussaini S., Li Y., and Moyne J. R. Precise time synchronization in semiconductor manufacturing. In *Proc. IEEE ISPCS*, 2007.