A Factory-Wide EDA Data Quality Performance Simulation for APC Capabilities Analysis

Xiao Zhu, Dhananjay Anand, Sulaiman Hussaini,
Ya-Shian Li-Baboud and James Moyne

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

The University of Michigan
Engineering Research Center for Reconfigurable Manufacturing Systems

AEC advanced equipment control APC advanced process control
SYMPOSIUM XX
October 4-8, 2008 • Salt Lake City, Utah

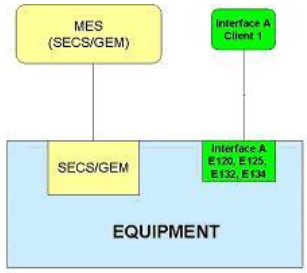
Outline

- EDA, data quality and time synchronization
- EDA end-to-end performance environment
- Characterizing factory-wide EDA performance
  - EDA simulator
- Results
  - End-to-end performance
  - Time synchronization accuracy
- Conclusions and future efforts

Background: Equipment Data Acquisition (EDA)

SEMI Standard for Ethernet-based Data Acquisition

- Second communication port on equipment
- **High-speed, XML-based messaging**
- Structured to allow multi-host solutions
- Facilitates data merging from multiple sources



Current focus of project:

- Characterizing the performance impact
- Emphasizing factory-level applications

Today's Fabs and EDA

Switched fast Ethernet (100 Mbps)

- No collisions, Low utilization

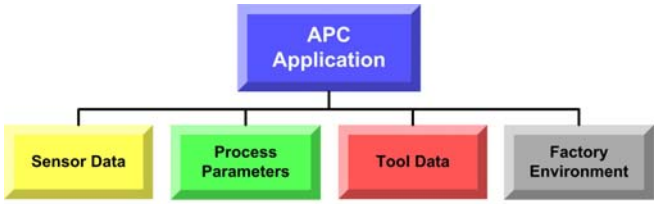
Protocols on top of Ethernet for end-to-end data communication, security, etc.

- EDA over Ethernet
  - UDP or TCP for data transport
  - OPC
  - VPN for security (E132, interface "C")

Fabs need to understand:

- Current network systems' ability to support factory-wide data acquisition
- Impact of EDA network system on data quality aspects
- How to ensure EDA delivers the capabilities for needed for optimal APC

EDA Use Dictates Areas of Concern



Precision Time Stamping to Merge Various Data Streams

Advanced Process Control

- Fault Detection Classification
- e-diagnostics
- Process Optimization
- Virtual Metrology

- **High-speed data collection**
- **Real-time consolidation**
- **Accurate, rapid control**
- **Quality data required**

Potential EDA Pain Points in APC

Availability

✓ Delay and *delay variability*
- Data are received out-of-order, at inconsistent rates
- Inability to support high data collection rates with good data quality

✓ "Out-of-control" situations for Run-to-Run (R2R) controllers

Time-stamps

✓ **Cannot synchronize data across multiple systems (e.g., equipment & metrology)**
- Out-of-order data, inaccurate time-stamping
- Time-stamping at point of sending instead of point of event occurrence

✓ **"False Positives"**
- Fault detection systems bring equipment down unnecessarily

Accessibility
- Inability to migrate from the equipment level to the factory-wide level with APC systems

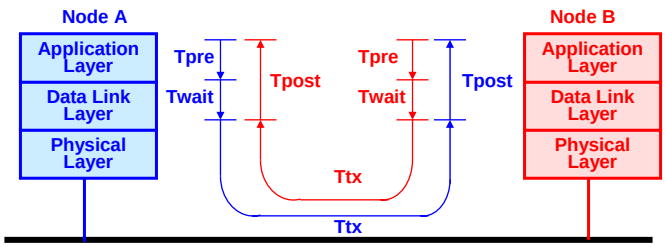
Etc…

How can EDA data quality be improved?

Outline

- EDA, data quality and time synchronization
- The EDA end-to-end performance environment
- Characterizing fab-wide EDA performance
  - EDA simulator
- Results
  - End-to-end performance
  - Time synchronization
- Conclusions and future efforts

Components of End-to-end Delay



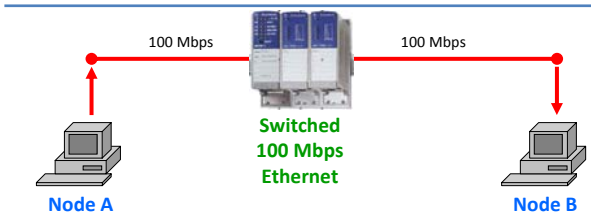
Total end-to-end delay is the sum of

- Pre-processing time: microprocessor
- Waiting time: network protocol - MAC
- Transmission time: data rate & length
- Post-processing time: microprocessor

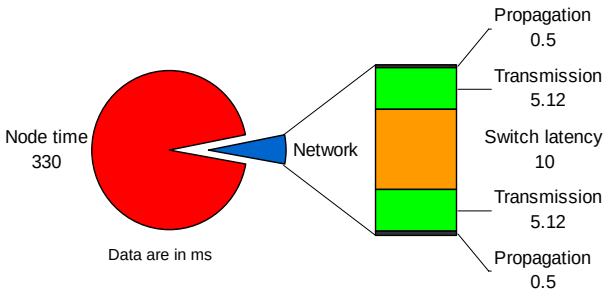
Device Delays

Network Delays

Switched Ethernet System: Where Does the Delay Come From?



64-byte packet



In low-load switch networks, delay is in the node software: operating system, drivers, applications (Tpre, Twait, Tpost)

Impact of Ethernet Overhead on Delay

	UDP	VPN (UDP)	OPC (TCP)	DeviceNet
Delay Average (ms)	0.33	1.21	1.48	0.3-1.2
Delay Variation (3σ) (ms)	0.09	0.49	2.43	0.005-0.2
Minimum Network Contribution (ms)	0.035	0.035	0.035	0.188
% of Delay Due to Network	11%	3%	2%	63%

Application layer protocols contribute greater delays.

Delay and delay variability can reduce data quality by causing inconsistent availability and rendering time-stamps to be inaccurate.

Ideally, performance improvement of application protocols need to be optimized and data must be time-stamped at point of measurement for optimal accuracy.

Need to implement standards for time synchronization (SEMI E148) and follow data quality guidelines on *where to time-stamp* and how to *minimize delay variability* in end-to-end communication.

Outline

- EDA, data quality and time synchronization
- The EDA end-to-end performance environment
- Characterizing fab-wide EDA performance
  - EDA simulator
- Results
  - End-to-end performance
  - Time synchronization
- Conclusions and future efforts

NIST / University of Michigan Project

Benchmark common protocol scenarios (XML, VPN, OPC)

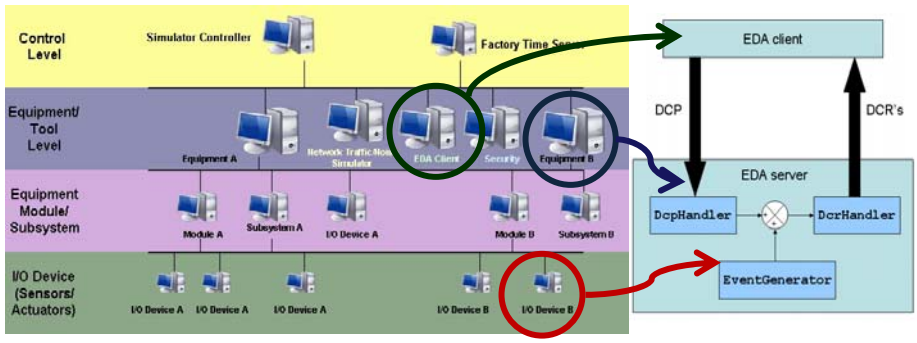
EDA traffic volume analysis and performance benchmarking

- EDA traffic simulation
- Analyze fab-wide scenarios, e.g, for APC

Input to current and future SEMI standards efforts and ITRS

- Improving data quality
  - Time synchronization
  - When / where to timestamp
  - Improve delay variability
- EDA performance roadmap
  - Determine EDA performance metrics
  - Roadmap goals

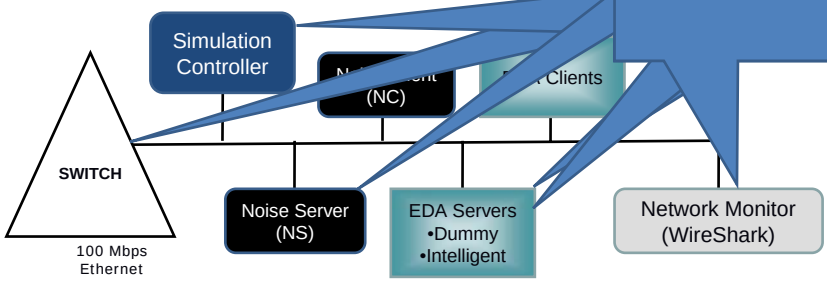
EDA Simulator architecture



Semiconductor factory simulation to study performance and benefits of data quality aspects.

Simulator Components

A real-time Ethernet packet monitoring tool is utilized to evaluate network performance.



Experimentation Approach

- Set up "dummy" and "intelligent" EDA client-server communications
  - Communicating SEMI EDA compliant messages
  - For these experiments each server ➔ 830 Bytes @ 10 HZ
- Simulate noise traffic levels
- Analyze performance of smart EDA server
  - WireShark

Outline

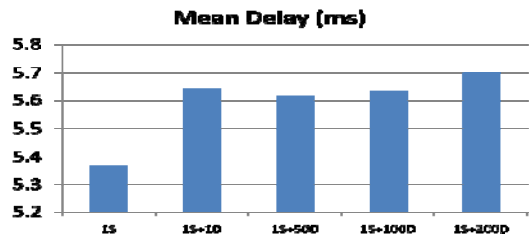
- EDA, data quality and time synchronization
- The EDA end-to-end performance environment
- Characterizing fab-wide EDA performance
  - EDA simulator
- Results
  - End-to-end performance
  - Time synchronization
- Conclusions and future efforts

Impact of Network Traffic on End-to-End EDA Delay

Setup (100 Mbps switched network)
- 1 smart EDA transmitting node
- Up to 200 "other" EDA transmitting nodes on network (typical of a fab environment)
- Measure end to end delay of smart EDA node

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

*S: smart node. 1S means one smart node. *D: dummy node.



Impact of Network Traffic on End-to-End EDA Delay

Observations

- Network delay does not increase significantly increase with number of dummy nodes
- Network congestion is *not* an issue in simulated factory-wide data collection

Issues

Since multiple "dummy" EDA transmitters were simulated at one IP node, the measured delay does not reflect switch delay in an actual system

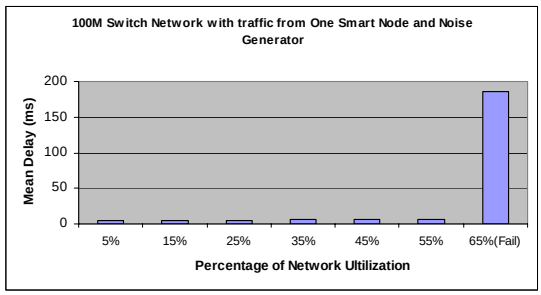
➔ Actual delay in practice will have to include additional switch delay (small, more on this later)

Maximum EDA Node Capacity for Switch Network

Setup (100 Mbps switched network)

- 1 smart EDA transmitting node
- Configurable noise generator used to flood the network
- From these numbers determine the equivalent number of EDA nodes that would flood the network

Delay (ms)	Percentage of Network Utilization						
	5%	15%	25%	35%	45%	55%	65%
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



Switched Network Traffic on End-to-End EDA Delay

Observations

- Switched nodes can handle more capacity than hub-based and are much more deterministic (no collisions); however, performance degrades drastically at the point of network saturation.
- Actual failure is close to 100% of net capacity; this graph plots against gross capacity, which includes switch delay against capacity
- This traffic level translates to network failure with about 1400 EDA nodes transmitting 830 Bytes at 10Hz each

Recommendations

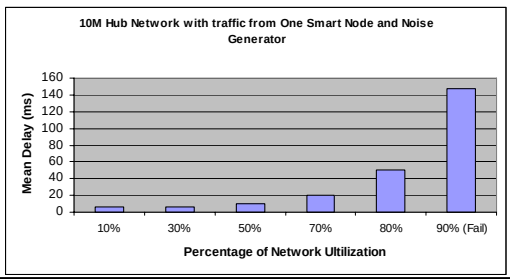
- Understand the limits of the EDA network, ensure data collection is under the network saturation limits
- Results indicate that you can safely operate 200 EDA nodes on a single 100M switched network

Maximum EDA Node Capacity for a Hub Network

Setup (10 Mbps hub-based network)

- 1 smart EDA transmitting node
- Configurable noise generator used to flood the network
- From these numbers determine the equivalent number of EDA nodes that would flood the network

Delay (ms)	Percentage of Network Utilization					
	10%	30%	50%	70%	80%	90%
Mean Delay	6.6195	6.8792	10.1909	20.6377	50.9104	*146.9667
Max Delay	8.0670	15.0600	138.0530	192.0510	183.0780	920.1160
Std Delay	3.3481	3.6316	13.4771	27.2453	37.2479	177.5112



Hub-Based Network Traffic on End-to-End EDA Delay

Observations

- Hub-based nodes are less deterministic, however performance degradation is more gradual.
- In an actual EDA network, failure would occur at a lower percentage of network utilization (e.g., 40%) because the EDA nodes would be on many different IP's (more collisions and retries).

160

Recommendations

Hub-based networks should NOT be used for EDA due to:

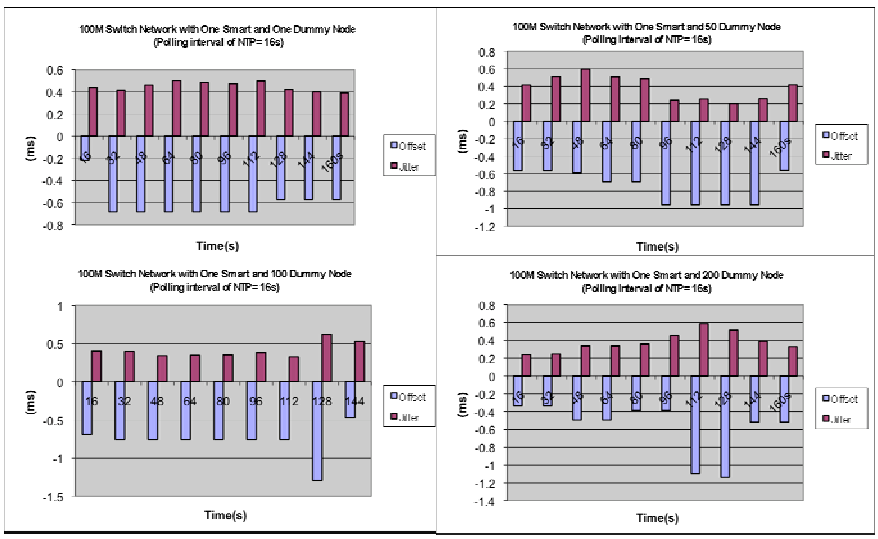
- Low determinism
- High delays

when traffic bursts occur

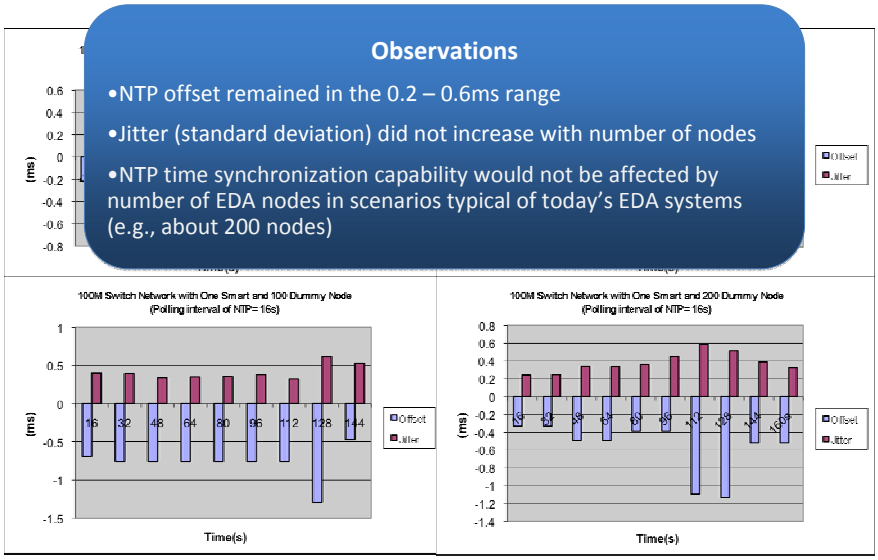
Clock Offset and Jitter with NTP Synchronization

Setup (100M switched network)

- **1 smart EDA transmitting node; vary number of dummy nodes**
- **Analyze NTP synchronization offset and jitter**



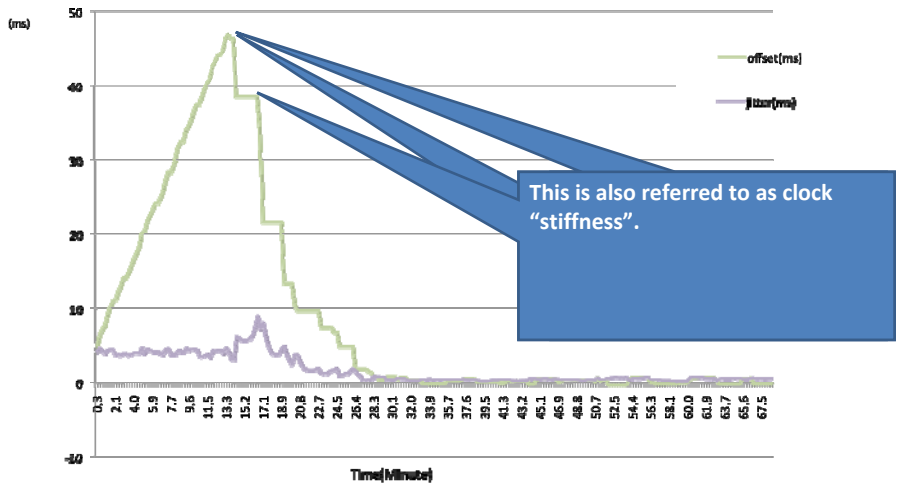
Clock Offset and Jitter with NTP Synchronization – Steady State



Observations

- NTP offset remained in the 0.2 – 0.6ms range
- Jitter (standard deviation) did not increase with number of nodes
- NTP time synchronization capability would not be affected by number of EDA nodes in scenarios typical of today's EDA systems (e.g., about 200 nodes)

Clock Offset and Jitter with NTP Synchronization - Transient Analysis

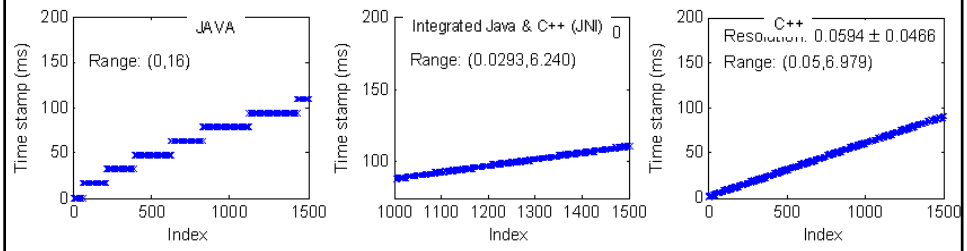


This is also referred to as clock "stiffness".

Time-Stamping Initial Results

Time-stamping

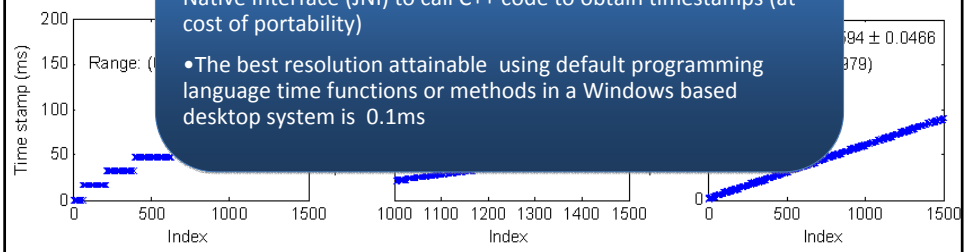
- Developed simple time-stamping application in Java, C++ and hybrid
- Observed time-stamping resolution



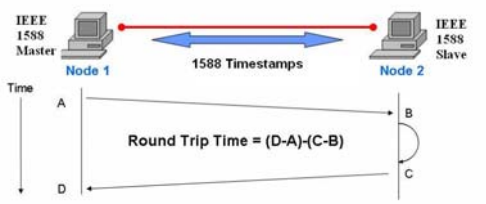
Time-Stamping Initial Results

Observations

- Java time stamping has ~10-15 ms resolution using the Windows operating system
- This can be improved without recoding in C++ by using the Java Native Interface (JNI) to call C++ code to obtain timestamps (at cost of portability)
- The best resolution attainable using default programming language time functions or methods in a Windows based desktop system is 0.1ms



Emerging Time Synchronization Technologies: Tests with 1588 PCI cards



Delay between packets	Max (ms)	Mean (ms)	Min (ms)	Std Dev (ms)
250ms	1.188	0.251	0.201	0.053
1s	0.317	0.243	0.234	0.003

Observations

- IEEE 1588 card does not support data time-stamping, therefore reducing usefulness in the EDA environment – sent recommendations to manufacturer.
- Application processing at node remains the critical issue.

Outline

- EDA, data quality and time synchronization
- The EDA end-to-end performance environment
- Characterizing fab-wide EDA performance
  - EDA simulator
- Results
  - End-to-end performance
  - Time synchronization
- Conclusions and future efforts

Summary of Observations

EDA end-to-end delay is largely in the higher levels of software at the nodes when networks are not saturated.

Network congestion (and delay) is generally not an issue for the EDA data collection scenarios examined.

•100 Mbps switched Ethernet network failure would occur with about 1400 EDA nodes transmitting 830 Bytes at 10Hz each

Use switched rather than hub-based networks for EDA

Data in hub-based networks are susceptible to:

- Low determinism and significant delays
- Especially during network traffic bursts

Summary of Observations

NTP time synchronization capability would not be affected by number of EDA nodes in scenarios typical of today's EDA systems (e.g., about 200 nodes)

- NTP accuracy is in the 0.4 ms range in 100 Mbps switched Ethernet

There is an issue with application level time-stamping accuracy in Windows systems

- Java application level time-stamping accuracy is 10-15 ms
- Can improve with hybrid of Java and C++ (about 1 ms)
- Best observed is down to 0.1 ms (C++) for Windows desktop system

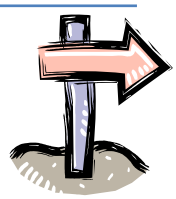
IEEE 1588 data acquisition solution tested was not found to be readily practical for use in EDA systems yet.

Conclusions

- **EDA and PCS can only be realized with *quality data*.**
- **Need to understand sources of network delay and delay variability. Such data quality issues lead to costly errors and ineffective control decisions.**
- **The weak link is often the tool software performance, which impact *reliable data availability* and time-stamping.**
- ***Where you time stamp* is very important.**
- **NTP is sufficient for equipment clock synchronization accuracy of 1 ms; however, NTP requires up to 1 hour to stabilize.**
- **NIST, The University of Michigan, ISMI and SEMI are addressing data quality issues, with the end result being standards, guidelines and prototypes for PCS, EDA and e-diagnostics.**

Future Efforts

Investigate the impact of *where* you timestamp on data quality in EDA systems



Study of EDA over wireless

- Actually quite plausible, even when performance and security are taken into consideration

Technology transfer

- SEMI standards for data quality
- Education; raise awareness of the data quality issue in EDA
- International Technology Roadmap for Semiconductors (ITRS)

Further Information

Contacts:

Ya-Shian Li-Baboud: ya-shian.li-baboud@nist.gov

James Moyne: moyne@umich.edu

References

- N. Kalappa, Y. Li, J. Baboud, V. Anandarajah and J. Moyne, “Precise Time Synchronization in Semiconductor Manufacturing') *2007 International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS) 2007*, Vienna, Austria, (September 2007).
- J. T. Parrott, J. R. Moyne, D. M. Tilbury, "Experimental Determination of Network Quality of Service in Ethernet: UDP, OPC, and VPN," *Proceedings of the American Control Conference,* Minneapolis, MN, June 2006.

Acknowledgments

- University of Michigan Ann Arbor
- NSF Engineering Research Center for Reconfigurable Manufacturing Systems (UM-ERC-RMS)