The Implications of Ethernet as a Control Network[1]

Paul G. Otanez[a], Jonathan T. Parrott[a], James R. Moyne[b], and Dawn M. Tilbury[a]
The University of Michigan
(a) Department of Mechanical Engineering
(b) Electrical Engineering and Computer Science
Ann Arbor, MI 48109-2125
{potanez, jparrott, moyne, tilbury}@umich.edu

Abstract

Centralized point-to-point architectures are increasingly being replaced by distributed common-bus systems to address issues such as the interchangeability and reliability required of modern control systems. Traditional centralized architectures also do not support industry's need for smart sensors or distributed control. For manufacturing systems a distributed architecture allows for modularization of functionality and standard interfaces for interchangeability and interoperability. However, implementing a control system over a common-bus network induces time delays that degrade performance. To date, guidelines to optimize the performance of networked control systems have not been clearly defined. The adoption of a distributed architecture has been motivated by lower cost and improvement of reconfigurability and reliability. Several network types, such as Ethernet, ControlNet, and DeviceNet have been successfully implemented in different control systems. Because of its low cost, availability, and higher communication rates Ethernet is potentially the most practical network solution. The non-deterministic nature of the carrier sense multiple access with collision detection (CSMA/CD) mechanism used by Ethernet to resolve contention is the major drawback to its use as a control network, because response time magnitudes cannot be guaranteed. In this paper, we describe the implications of using Ethernet as a control network and present a set of guidelines describing what conditions are necessary for the successful use of Ethernet as a control network.

1 Background

Traditional control systems composed of interconnected controllers, sensors, and actuators have been successfully implemented using a point-to-point architecture. With this type of architecture, system components are directly wired to various controllers. Modern manufacturing systems, must be reconfigurable in order to be cost-effective and responsive to market changes [5]. A point-to-point architecture is somewhat stagnant from a reconfigurability point of view, as it is difficult to interchange components due to lack of a common communication protocol, and difficult to add, delete or re-configure components since levels of interoperability are generally not defined. As an alternative to point-to-point, the common-bus network architecture offers more efficient reconfigurability and reduces installation and maintenance cost due to the decreased wiring. However, implementing a control system over a common-bus network induces inevitable time delays that degrade performance and can even cause instability [11].

2 Control Networks

Communication networks can be classified as either control networks or data networks. The difference is that control networks have strict real-time requirements for the information sent through the medium [8]. Control networks must shuttle small but frequent messages among a large number of nodes to meet time-critical requirements of real-time control systems. In contrast, data networks have a high data rate and send large but infrequent data packets [16]. In a control network, it is more important that the majority of the information be delivered in a timely manner than that all the information is delivered in the most efficient manner.

A networked control system (NCS) is defined as a control system in which control information (sensor data, actuator commands) is sent over a communication network. It is well-known that time delays in a feedback loop can significantly degrade the performance of a control system or even destabilize it. Thus, the most important factor to consider when evaluating a control network is the end-to-end time delay between sensors, controllers, and actuators. The correct operation of a control system depends on the timeliness of the data coming over the network, and thus, a control network should be able to *guarantee* message delivery within a *bounded* transmission time.

Control networks are specifically adapted to the demands of NCS. The most common types of control networks are

---

[1] The authors are pleased to acknowledge the financial support of the Engineering Research Center for Reconfigurable Manufacturing Systems under NSF grant EEC-9529125.

token passing and CAN-based networks. In a token passing network, such as ControlNet, a deterministic real-time channel is defined as an unidirectional virtual connection between two endpoints that provides a priori deterministic or absolute guarantees for the timely delivery of packets. Nodes are arranged logically in a ring and are only allowed to transmit when they receive the token, eliminating the possibility of message collisions. Under moderate message loads, the performance of ControlNet is stable and during increased loads the token-based algorithm assures fair access with deterministic waiting time delays [2]. In a CAN-based network such as DeviceNet (CAN bus) each message has a priority that determines network access in case of simultaneous transmission. Data is transmitted in frames and message collisions are not destructive since the message with the higher priority is delivered successfully [12]. The lower-priority messages is then re-sent when the network is idle. In both types of control networks, message delivery is guaranteed as long as the network is not saturated. In addition, if the pattern of periodic network traffic is known, a bound on the transmission time can be calculated as in [12].

For networks with low traffic and high data rates, the time delays are typically small and can often be neglected when the controller is designed, although analysis should always be done to validate the controller performance with the network in the closed loop [18, 19]. The smaller the time delay, the lesser the impact on the control performance. Research has also been done on new protocols and enhancements to already existing protocols to improve the real-time characteristics of computer networks [1, 2, 17]. In this paper, we investigate the implications of using standard Ethernet (both 10MBps and 100MBps), without any modifications or enhancements, as a control network.

3 Ethernet for Control

During the last few years there has been significant competition in the fieldbus setting. Due to the fact that there were a lot of options to choose from, it was unclear what protocol would come out the winner. Because of its nondeterministic nature, Ethernet has never been considered the ideal field bus control protocol; however, due to its availability, and high communication rates, Ethernet is becoming a prime network control candidate. In this investigation, Ethernet with a transmission rate of 10 megabits per second and Fast Ethernet with a transmission rate of 100 megabits per second are considered.

3.1 Advantages

Ethernet offers many advantages to a system designer. The major advantage of Ethernet is availability; almost all personal computers sold today come equipped with Ethernet connectivity. This fact also translates into inexpensive training as most plants/offices already have technicians trained to setup and work with Ethernet. The result of more Ethernet devices being used will result in standardization. The second advantage of Ethernet is its flexibility. Ethernet is able to transmit messages from a minimum size of 72 bytes to a maximum of 1500 bytes over a length of 2500 meters (see Table 1). Because of its low medium access overhead, Ethernet uses a simple algorithm for network operation and almost has no delay at low network loads [12]. Therefore no bandwidth is lost to gain access to the network. Ethernet is commonly used for data networks because it is very good for transmitting large files because of its speed. Another important advantage is access to the Internet which makes it possible to implement remote data logging and diagnostics.

3.2 Disadvantages

Ethernet was not created to guarantee the delivery of time-critical information and thus presents disadvantages when used as a control network. The major limitation is the nondeterministic nature of communication. Ethernet uses the carrier sense multiple access with collision detection (CSMA/CD) to resolve contention in case of simultaneous data transmission. If two nodes transmit messages simultaneously, the messages collide and are destroyed (data is lost). The two transmitting nodes listen to the network to detect a message collision. If a collision is detected, the transmitting nodes wait a random length of time to retry transmission. The random length of time is determined by the standard binary exponential backoff algorithm. If sixteen collisions are detected, the node stops transmitting and reports an error. This is an important characteristic of Ethernet because in modelling, both time delays associated with waiting and re-trying and the possibility of message loss must be considered.

When a control system is implemented over a network, network time delays can degrade system performance. For networks with low traffic and high data rates, time delays are relatively small and under some conditions can be neglected when the controller is designed (although analysis should always be done to validate the controller performance with the network in the closed-loop) [18, 19]. In addition, since Ethernet does not support message prioritization, there is no guarantee that a particular message will not collide and eventually be dropped. Message collisions significantly compromise network performance by increasing time delays or causing message loss. Section 3.5.3 discusses what factors affect collisions and increase the possibility of dropped messages. In order to relate collisions to time delays, the blocking time of Ethernet is considered. The blocking time is defined as the time a message must wait once a node is ready to send it. It includes waiting time while other nodes are sending messages and the time needed to resend the message if a collision occurs. The expected blocking time can be described by

$$E\{T_{block}\} = \sum_{k=1}^{16} E\{T_k\} + T_{resid} \tag{1}$$

	Ether.	Fast
Data rate (Mb/s)	10	100
Max. Length (m)	2500	250
Max. number of hubs	4	2

Table 1: Characteristics of Ethernet (10 MBps), Fast Ethernet (100 MBps).

where T_{resid} denotes the residual time until the network is idle, and $E\{T_k\}$ is the expected time of the kth collision. $E\{T_k\}$ depends on the number of backlogged and unbacklogged nodes as well as the message arrival rate at each node. T_{block} is not deterministic and maybe unbounded due to the discarding of messages [12].

3.3 Hardware

3.3.1 Performance and Materials: Ethernet has generally been used on the information technology level of manufacturing operations for years, but there has always been doubt on how it would perform at the factory level. Due to the harsh environments on the factory floor, Ethernet brings up the following questions in terms of durability, noise immunity, power, and reliability [13].

Performance of an Ethernet network can be negatively affected by the factory floor environment. The effective data rate of cables can decrease in an industrial environment due to plant-floor generated noises and and emissions. The noise increases the likelihood of packet errors. The higher the data rate the more sensitive the network will be to noise. For Fast Ethernet, this problem is amplified since the amplitude of the signal is 10 times smaller than that of 10 Megabit Ethernet (see Table 1). It is important to understand, that the plant floor's equipment can generate both electrical and mechanical noise.

Unlike other fieldbuses, Ethernet does not provide power on the wire resulting in implementation problems. To reduce material costs communication, instrumentation, and power conductors are often combined in a single raceway which is highly undesirable as it can exacerbate the data loss problem [13].

3.3.2 Hubs, Switches, Routers, and NICs: An important factor in considering whether a protocol is a suitable control network solution is the magnitude of the message delays. Queuing delay is a form of packet-switching delay as the number of packets in a device increases exponentially as utilization increases. Utilization is defined as the percent of total available data-carrying capability of the network. In the case of Ethernet, a fundamental delay occurs when routers, switches, or bridges forward information. The delay from these devices is caused by the speed of the internal circuitry, and the switching architecture of the internetworking device. Starting from the bottom of the OSI model, hubs introduce the minimum amount of delays because they operate in layer 1. A hub works by repeating an incoming signal to all of its ports, including the port that originally sent the data. Because of their simplicity, hubs are able to handle a lot of information, but are not able to route information to avoid collisions. A switch operates in the data link layer of the OSI model, layer 2. When a frame is received, the switch must observe the destination address, check whether the frame has an error, look up the port of the destination IP address, and direct to the message to the appropriate port. Therefore, switches are able to direct information to only the receiving IP address and therefore reduce or eliminate collisions.

Routers can effectively separate traffic from different subnets. As discussed in Section 3.5, the network collision rate and subsequent network time delays are minimized if the utilization is reduced. For this reason, it is important to minimize the traffic in a control subnet. When a router is used, a subnet sees only network traffic that is directed to it. However, one of the advantages of Ethernet is that many types of of information, such as monitoring information, diagnostics, data files, can be sent through the network medium. When using a router, this non-control information must be directed explicitly to all relevant subnets.

Smart network interface cards (NICs) minimize unnecessary CPU load on the computers in a subnet. In a typical hubbed subnet, every NIC will receive all messages repeated at the hub. The messages are then sent to the computer CPU to determine whether destination IP address matches the particular computer. As the network utilization increases, the CPU on each computer must devote significant processing time to reading the destination of each message. NICs with parallel processing are able to read the destination of each message and only forward messages to the CPU that were directed to that particular IP. This property is very useful to ensure that a computer in a subnet with a high utilization will not get slowed down by traffic not directed to it. Currently, very few vendors offer NICs with parallel processing and in general they are slightly more expensive, however, they isolate the performance of computers in a subnet from the utilization of the network.

3.4 Traffic Smoothing and Reduction

Ethernet's contention-based protocol makes it impossible to predict network delays exactly. By eliminating packet collisions, switched Ethernet can provide deterministic delays, but its higher price has restricted its deployment in industry [7]. Several software changes have been suggested so that Ethernet network delays could be bounded [4]. Venkatramani et al. proposed the implementation of a virtual token-passing scheme that protects the network against token loss due to message collisions [17]. Kweon et al. developed another method to bound delays that is simpler and requires minimal changes in the OS kernel [7]. In their method delays are statistically bounded by limiting the packet arrival rate at the medium access control sublayer (MAC). At each node, a traffic smoother is installed between the TCP/IP layer and the Ethernet MAC sublayer. The traffic smoother regulates the node's outgoing message rate to achieve a certain traffic-generation rate. The traffic-generation rate is allowed to adapt itself to the underlying network load to more effectively eliminate collisions [6].

The most effective way to minimize the effect of time delays on the performance on a NCS is to reduce network traffic. [20] proposes the use of estimators to predict the states of other systems on the network. The motivation behind this method was that increased computational costs from the estimators was outweighed by the performance benefits gained reducing network traffic which would decrease the effect of delays on system performance. The transmission of state updates by a system occurred when the difference between the actual and estimated and actual values of the state exceeded a variable threshold. As an alternative, Otanez et al. proposed the implementation of adjustable deadbands as a solution to reduce network traffic. With a deadband defined on a node, the node does not broadcast a new message if the node signal is within the deadband. There is a trade-off between increasing the size of the deadbands to reduce traffic and the performance of the system. To quantify this trade-off, a performance metric was formulated that takes into account the cost of network traffic and the cost of system performance. By selecting deadbands at the minimum of the performance index, traffic can be minimized and performance maintained. The implementation of deadbands is very promising since nodes do not require knowledge of their environment and thus is better suited for reconfigurability [15].

3.5 Experimental Results

Throughput, CPU usage, collision rate, and position monitoring with network traffic are investigated in this section utilizing a control network testbed. In this testbed, two to four computers are used to represent nodes in a subnet. Traffic is generated using a software packet generator. The packet generator allows the user to control the number of packets generated on the network, the packet size (payload), and the destination IP address. The generator sends messages using the user datagram protocol which is a connectionless and unreliable service, since it does not issue acknowledgements to the sender upon receipt of data, nor does it inform the sender of lost messages. A hardware protocol analyzer was used to record network utilization and collision rate. Software is also used to monitor how much information is sent from a node, i.e. upload rate, and how much information is received, i.e. download rate. The throughput is measured while nodes are connected via a direct connection, cross-over cable, and a hub. Software reports the CPU usage while three computers are connected through a hub. The throughput and collision rate are measured for both Ethernet and Fast Ethernet. Three different types of network interface cards (NICs) were also tested. For a detailed description of the software and hardware used see Section 6.

	PC1	PC2
Upload (kB/s)	1175	146
Download (kB/s)	119	1130
Total (kB/s)	1294	1276

Table 2: Throughput for cross-over cable with largest allowable packet size. PC1 sends messages to PC2. Note: The theoretical limit for a 10 MBps Ethernet is 1250 kB/s.

	PC1	PC2
Upload (kB/s)	202	161
Download (kB/s)	455	415
Total (kB/s)	658	576

Table 3: Throughput for two computers connected by a hub running at 10 MBps with the largest allowable packet size. PC1 sends messages to PC2.

	PC1	PC2
Upload (kB/s)	11585	750
Download (kB/s)	0.9	5782
Total (kB/s)	11586	6532

Table 4: Throughput for cross-over cable with largest allowable packet size. PC1 sends messages to PC2. Note: The theoretical limit for a 100 MBps Ethernet is 12500 kB/s.

	PC1	PC2
Upload (kB/s)	7950	1000
Download (kB/s)	800	7700
Total (kB/s)	8750	8700

Table 5: Throughput for two computers connected by a hub running at 100 MBps with the largest allowable packet size. PC1 sends messages to PC2.

	NIC1	NIC2
Upload (kBps)	1528.9	2339.9
Download (kBps)	1889.9	2610
Total (kBps)	3418.8	4949.9

Table 6: Throughput for two computers connected by a hub running at 100 MBps with the smallest allowable packet size.

	Utilization	CPU Usage
NIC1	90%	90%
NIC2	90%	2%

Table 7: CPU Usage with and without parallel processing NICs (pairs).

3.5.1 Throughput Results: It is important to determine how much information a single computer can send through the network when there is no other traffic. Throughput is defined as the quantity of error-free data successfully transferred between nodes per unit time and was measured in two different ways. In the first set up, two computers were connected by a cross-over cable and the packet rate increased with full payload until a maximum throughput was reached. The payload is the number of bytes of meaningful data transmitted in a packet. The packet size is the total number of byes in a packet, including the payload, the overhead (22 Bytes), and any necessary stuffing to satisfy the minimum packet size of 46 Bytes. For the next test, the computers were connected through an eight port hub and packet rate was increased in the same manner. These tests were run at both 10 and 100 MBps.

Tables 2 and 4 show that a single computer can saturate both 10 and 100 megabit Ethernet when large size packets are transmitted. If a hub is used to connect the computers, Tables 3 and 5 show that the throughput decreases due to the lack of speed of the hub's internal circuitry and due to collisions. A collision occurs when two NICs listen for traffic, hear none, and then transmit simultaneously. These tables also show that the percentage data loss from collisions is much lower for 100 MBps than it is for 10 MBps (0.5% vs. 12.5%). However, it should be mentioned that the throughput can also be dependent on the NIC when smaller packets are transmitted, as testing showed that for computers with the same setup, the throughput was increased by 45% if a different NIC was used (see Table 6).

3.5.2 CPU Usage: A hub sends all the messages it receives to all the nodes connected to it. Therefore, it is important to determine if nodes will be slowed down by having to decode messages that are not directed to them. In this setup, three computers were connected via hub, and information was sent from the first to the second, while the CPU usage of the third was recorded. Testing results, see Table **??**, show that with 90% network utilization, the third CPU had to devote all of its resources to decode messages, while the CPU usage was reduced to 2% when NICs with parallel processing were used. NICs with parallel processing isolate the performance of computers in a subnet from the utilization of the network.

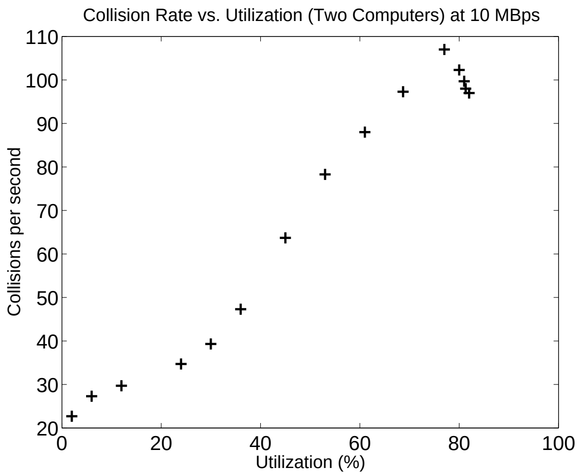

Figure 1: Collision rate as a function of utilization (two computers).

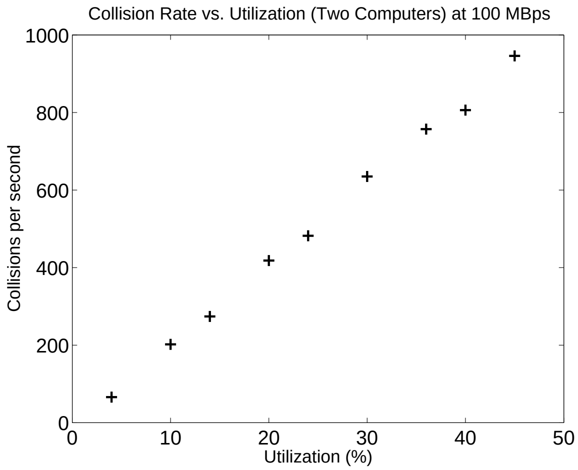

Figure 2: Collision rate as a function of utilization (three computers).

3.5.3 Collisions as a Function of Network Traffic: Long transmission time delays deteriorate NCS performance and could drive the system to instability. In the setup presented, time delays are directly proportional to the network collision rate since there is no retransmission of lost messages. In order to determine how network traffic impacts collisions, two setups are used to measure the network collision rate. In the first, two computers are set up to send packets to each other connected through a hub. In the second setup, the two computers are trying to send packets to a third computer and all are connected through a hub. Figures 1 and 2 show that the mean value of the collision rate is a linear function of utilization. To account for the uncertainty due to the stochastic nature of collisions, the measurements were taken five times, so that the mean value and standard deviation could be computed. In all cases the standard deviation was within plus or minus 5% of the mean value. In [3, 14] discontinuities were observed in the collision rate/utilization relationship at 25% and 37% for 10 and 100 MBps respectively. These break points were not observed during testing with small payloads, but were observed when the maximum payload was used.

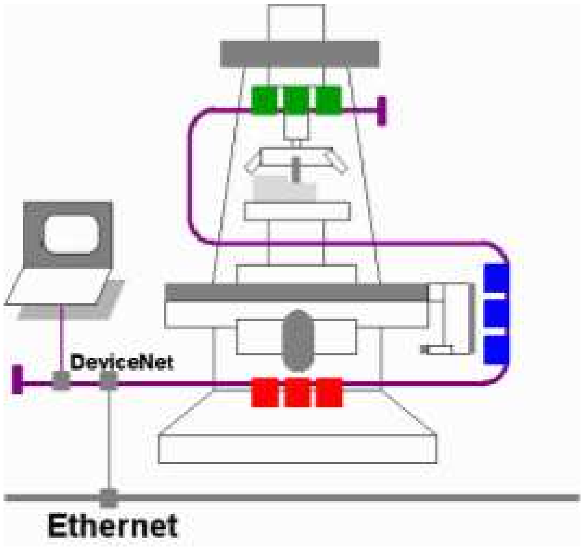

Figure 3: A distributed control system, the Robotool, with DeviceNet and Ethernet devices.

3.5.4 Position Monitoring with Network Traffic:

To demonstrate the effects of network traffic on the performance of a distributed control system, a three-axis milling machine was converted from a point-to-point architecture to a distributed common-bus [9]. As seen in Figure 3, the positions of the machine axes are controlled using DeviceNet I/O modules and a PLC with a scan time of 5 milliseconds. The position signals from the axes are monitored using the National Instruments Fieldpoints distributed I/O modules. The Fieldpoint modules are connected by a 100 Megabit hub and traffic generating software is used to simulate a congested network.

To quantify the effect of utilization on the performance of the system, the utilization was increased from 1% (where all the traffic is generated by the Fieldpoint modules) to 40% by adding noise traffic. It should be noted that only the payload of the noise traffic was changed. The integral of the absolute error (IAE) of the position is also commuted to provide a measure of the response. The response was measured for a packet with no payload and full payload. In general, because of its small size, the empty payload is very much like information sent from a control system. The full payload case does not resemble information sent through a control network, but it was implemented to investigate an extreme case.

Figure 4 shows that even at high utilization (40%), the monitoring performance is acceptable, and Figure 5 shows that is proportional to the error in response. For small utilizations, since the payload is empty, collisions are minimized and the response does not deteriorate dramatically. It should be noted that due to the nondeterministic nature of Ethernet, there will always be uncertainty in the response.

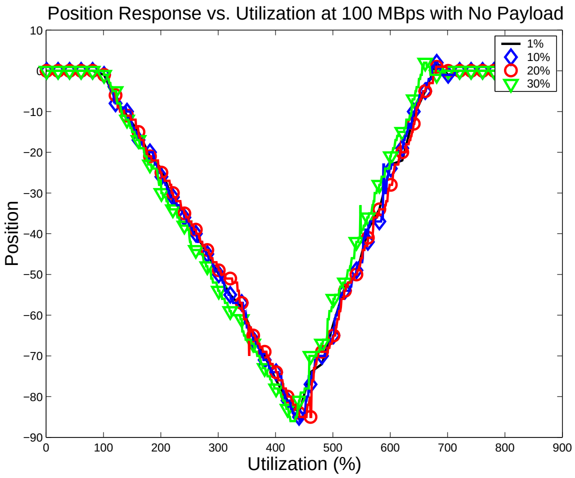

Figure 4: Position response versus utilization at 100 MBps with no payload.

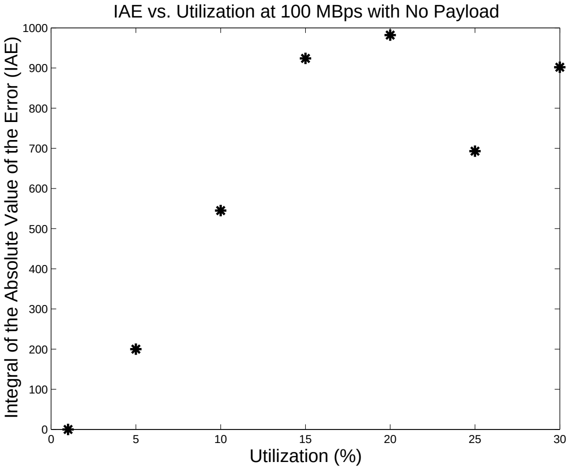

Figure 5: IAE versus utilization at 100 MBps with no payload.

Figures 6 and 7 show that larger payloads results in the network having a more significant influence on the response. From Figure 6 it is clear that at 30% utilization, the traffic prevents accurate monitoring of the position signal. As shown by Figure 7, the relationship between utilization and IAE appears to be linear.

3.6 Recommendations

The following guidelines improve the ability of Ethernet to effectively deliver control information.

- Network utilization should not exceed 25% and 37% for 10 and 100 MBps Ethernet respectively [3].
- For small payloads there is a linear relationship between utilization and collision rate.
- The design of a NCS should consider that most Ethernet wires and connectors are sensitive to factory floor

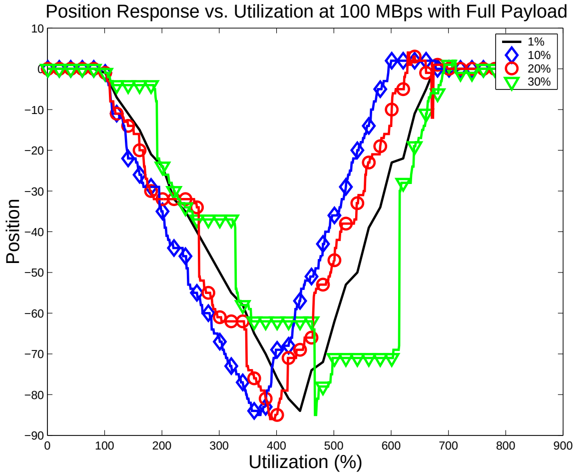

Figure 6: Position response versus utilization at 100 MBps with full payload.

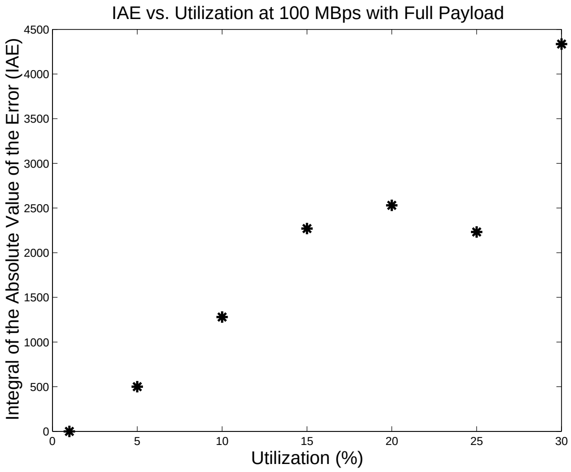

Figure 7: IAE versus utilization at 100 MBps with full payload.

electrical and mechanical noise.

- The optimal sampling time for a NCS is between P_B and P_C defined by

$$P_B = \frac{T_{bw}}{20} - 2T_d \qquad (2)$$

$$P_C = \frac{T_{ttt}}{0.69} \qquad (3)$$

  where T_{bw} is the control system bandwidth, T_d is the time delay, and T_{ttt} is the total transmission time of all cyclic messages [10].

- Routers should be used to isolate traffic between subnets.

- NICs with parallel processing technology should be used to ensure nodes are not slowed down by messages not sent to them.

- 100 MBps Ethernet should be used if possible.

4 Summary and Future Work

There are implications when using Ethernet as a control network, due to its inability to bound network transmission delays. These delays are minimized by using the network bandwidth effectively and by minimizing collisions. The throughput of a single computer was measured using both a direct (cross-over cable) connection and through a hub at two different data rates, 10 and 100 MBps. NICs were also tested to determine their effect on throughput and the resultant CPU usage at a node. The relationship between network utilization and collision rate was also investigated in order to quantify how utilization results in data loss. As an example of the effects of excessive network traffic, the effects of data loss on position monitoring were presented. Finally, guidelines that make Ethernet deliver control information more effectively were formulated. Future work includes, extending testing to include different network topologies and also formulating a more precise relationship between utilization, collision rate, and NCS time delays.

5 Acknowledgements

The authors would like to acknowledge the contributions of John Korsakas of the University of Michigan, Brad Baker of National Instruments and Kathy McLennan of Fluke Networks.

6 Hardware Description

- Computers
  - 2 P4 with 128 MB of RAM
  - 1 P1 with 64 MB of RAM
  - 1 P3 with 128 MB of RAM
- Network Interface Cards
  - 3COM Integrated card
  - 3COM
  - Linksys
- Network Software
  - IPLoad Version 1.11 (traffic generation)
  - DUMeter Version 2.21 (traffic monitoring)
- Traffic Monitoring Hardware
  - OneTouch by FlukeNetworks
- Hubs
  - Asante 8-port hub, 10/100 MBps
  - Soho Basic, 205 Hub, 10 MBps

References

[1] G. Agrawal, B. Chen, W. Zhao, and S. Davari. Guaranteeing synchronous message deadlines with the timed token protocol. *Proceedings of the 12th International Conference on Distributed Computing Systems*, 1992.

[2] B. Englert, L. Rudolph, and A. Shvartsman. Developing and refining an adaptive token-passing strategy. *21st International Conference on Distributed Computing Systems, 2001*, April 2001.

[3] W. Fong and J. Rodgerson. Industrial Ethernet. *Preprint*, 2001.

[4] S. Gang, Y. Cai, X. He, and W. Zhang. A new real-time Ethernet MAC protocol for time-critical applications. *Journal of Central South University of Technology (English Edition)*, March, 2002.

[5] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. Reconfigurable manufacturing systems. *A Keynote paper. CIRP Annals*, November, 1999.

[6] S.-K. Kweon and K. G. Shin. Ethernet-based real time control networks for manufacturing automation systems. 2000.

[7] S.-K. Kweon, K. G. Shin, and Q. Zheng. Statistical real-time communication over ethernet for manufacturing automation systems. *Proceedings of the Fifth, IEEE*, 1999.

[8] F.-L. Lian. *Analysis, Design, Modeling, and Control of Networked Control Systems*. PhD thesis, University of Michigan, May 2001.

[9] F.-L. Lian, J. Moyne, and D. Tilbury. Implementation of networked machine tools in reconfigurable manufacturing systems. *Proceedings of the 2000 Japan-USA Symposium on Flexible Automation, Ann Arbor, MI*, July, 2000.

[10] F.-L. Lian, J. Moyne, and D. Tilbury. Network design consideration for distributed control systems. *IEEE Transactions on Control Systems Technology*, March, 2002.

[11] F.-L. Lian, J. R. Moyne, and D. M. Tilbury. Analysis and modelling of networked control systems: MIMO case with multiple time delays. *Proceedings of the 2001 American Control Conference*, 2001.

[12] F.-L. Lian, J.R. Moyne, and D.M. Tilbury. Performance evaluation of control networks: Ethernet, ControlNet, and DeviceNet. *IEEE Control Systems*, February 2001.

[13] B. Lounsbary and J. Westerman. Building an industrial-strength ethernet. *Integrated Manufacturing Solutions*, October, 2001.

[14] P. Oppenheimer. *Top-down Network Design*. Cisco Press, 1999.

[15] P. Otanez, J. Moyne, and D. Tilbury. Using deadbands to reduce communication in networked control systems. *Proceedings of the 2002 American Control Conference*, May, 2002.

[16] A. Tanenbaum. *Computer Networks*. Prentice Hall, 1981.

[17] C. Venkatramani and T. Chiueh. Supporting real-time traffic on Ethernet. *Procedings of Real-Time Systems Symposium*, 1994.

[18] G. C. Walsh and H. Ye. Scheduling of networked control systems. *IEEE Control Systems Magazine*, 2001.

[19] B. Wittenmark, B. Bastian, and J. Nilsson. Analysis of time delays in synchronous and asynchronous control loops. *Proceedings of the 37th IEEE Conference on Decision and Control*, December 1998.

[20] J. K. Yook, D. M. Tilbury, H. S. Wong, and N. R. Soparkar. Trading computations for bandwidth: State estimators for reduced communication in distributed control systems. *Proceedings of the Japan-USA Symposium on Flexible Automation*, July, 2000.

Paul G. Otanez received his B.S. degree in Mechanical and Aerospace Engineering, *cum laude*, from Cornell University in 2000 and the M.S. degree in Mechanical Engineering from the University of Michigan, Ann Arbor in 2002. Currently he is a Ph.D. student in the Department of Mechanical Engineering at the University of Michigan. His areas of research include distributed control with network communication and the control and autonomy of high performance systems.

Jonathan T. Parrott is fourth-year undergraduate working towards B.S. and M.S. degrees in Mechanical Engineering at the University of Michigan, Ann Arbor. He is currently employed as a research assistant at the NSF Engineering Research Center for Reconfigurable Manufacturing Systems and the Modbus Conformance lab, both at the University of Michigan.

James R. Moyne received his B.S.E.E. and B.S.E. degrees in math and his M.S.E.E. and Ph.D. degrees from the University of Michigan. Since 1992 he has been an Associate Research Scientist in the Department of Electrical Engineering and Computer Science at the University of Michigan. He is also President and co-founder of MiTeX Solutions, Inc. His areas of research include advanced process control, database technology, and sensor bus technology. He has written a number of refereed publications in each of these areas. He also holds a patent and is the author of several semiconductor manufacturing standards.

Dawn M. Tilbury received the B.S. degree in Electrical Engineering, *summa cum laude*, from the University of Minnesota in 1989, and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley, in 1992 and 1994, respectively. In 1995, she joined the faculty of the Mechanical Engineering Department at the University of Michigan, Ann Arbor, where she is currently an Associate Professor. For her work in web-based software tutorials (the Control Tutorials for Matlab), she received an Undergraduate Computational Engineering and Science Award from the US Department of Energy in 1995 and the EDUCOM Medal (jointly with Professor William Messner of Carnegie Mellon University) in 1997. An expanded version, Control Tutorials for Matlab and Simulink, was published by Addison-Wesley in 1999. She received an NSF CAREER award in 1999, and is the 2001 recipient of the Donald P. Eckman Award of the American Automatic Control Council. She is a member of ASME, IEEE, and SWE, and co-chairs the ASME-DSCD technical panel on Computers, Communication, and Control. Her research interests include distributed control of mechanical systems with network communication, logic control of manufacturing systems, performance management and control of computing systems, and trajectory planning for nonlinear systems.