Real-Time I/O Performance Metrics and Tests for Industrial Ethernet

James D. Gilsinn
Electronics Engineer
Manufacturing Engineering Laboratory
National Institute of Standards & Technology (NIST)
100 Bureau Drive, Stop 8230
Gaithersburg, MD 20899-8230
james.gilsinn@nist.gov

KEYWORDS

industrial Ethernet, performance, metrics, testing, EtherNet/IP

ABSTRACT

Real-time input/output (I/O) communication over Ethernet is still fairly new in the industrial environment. There are many questions about the reliability and performance that have to be answered before industrial Ethernet can be utilized in most manufacturing plants. Similar questions are being asked in the information technology (IT) environment now that real-time communications are being used more heavily in such applications as real-time broadcasts and voice over the Internet Protocol (VoIP). Performance metrics and tests for IT infrastructure equipment have been used for many years, but now with real-time communications being used more, users are asking whether their desktop, laptop, or palm computer can handle the performance requirements of these network intensive applications. These same questions can be asked of industrial equipment as well. Can a Programmable Logic Controller (PLC), Distributed Control System (DCS), or Supervisory Control and Data Acquisition (SCADA) system handle the performance requirements for real-time I/O communications over industrial Ethernet? The National Institute of Standards & Technology (NIST) and the Open DeviceNet Vendor Association (ODVA) are developing a set of metrics and tests for real-time I/O performance of industrial devices. These will allow the user to have a common language and method to test different devices. Since performance metrics and tests have been used in the IT environment for many years, many of the same tools and experience can be utilized when developing metrics and tests for industrial Ethernet equipment. Last March, at the ODVA EtherNet/IP Interoperability Demonstration, a basic set of performance tests was conducted and data was collected about the performance of the devices that participated. A second interoperability demonstration was conducted in January of 2004 where a more substantial set of performance tests were conducted and data was collected. The performance metrics and tests used at these two demonstrations, as well as some of the data collected, will be discussed in this paper.

1 INTRODUCTION

One of the new buzzwords within the controls community is “industrial Ethernet”. For the last few years, industrial equipment vendors have been developing products with Ethernet interfaces. This is in stark contrast to Information Technology (IT) equipment, which has had Ethernet built-in for decades.

Industrial equipment vendors and users were previously concerned with Ethernet’s inherent non-deterministic performance characteristics. Ethernet, and many of the upper layer protocols based on it, provides unreliable, connectionless, but best-effort delivery of packets on the wire. In general, there is no guarantee that the data sent over the wire by the source will reach its destination. This is unacceptable in the IT world, so protocols and methods have been developed to overcome these limitations.

In order to determine just how reliable and fast these protocols and methods are, a workgroup of the Internet Engineering Task Force (IETF) developed multiple standard metrics and methodologies for testing IT equipment’s performance. These standard metrics and tests are now used when describing the performance of Ethernet devices. Now that industrial equipment has started using Ethernet and its various protocols, it is useful to look at these standards in order to develop industrial Ethernet performance metrics and tests.

The National Institute of Standards and Technology (NIST) is working with the Open DeviceNet Vendors Association (ODVA) to develop standard performance metrics and tests for EtherNet/IP (Ethernet / Industrial Protocol) devices. These standard metrics and tests will measure the performance of the real-time data Input/Output (I/O) communications of the EtherNet/IP devices. The metrics and tests are being designed to help a user select between multiple devices of the same type based on their application’s requirements.

The first section of this paper will discuss the metrics used to describe the performance of the real-time I/O communications of the device. The next section provides a basic description of EtherNet/IP and the relevant parts of the standard that relate to the performance metrics and tests. Following this section, multiple methodologies to measure the performance of a device are discussed. Some of these are mature, while others are still in their infancy. The next section describes how these performance metrics and tests were demonstrated at the EtherNet/IP Interoperability Plug-Fest. Finally, the paper presents a short summary and description of the future directions of the work.

2 “STANDARD” PERFORMANCE METRICS

The IETF maintains many of the “standards” for networking technology. These are maintained as documents called Request For Comment (RFC) documents. They are posted for a time giving people around the world a chance to make comments on the technology as draft documents, eventually, being published as full RFCs. RFCs can be written on any topic related to the Internet, but some of them are developed along the “standards” track that requires a more stringent review process. Many of the common Internet technologies like the Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) are RFCs maintained by the IETF.

There are multiple RFCs for testing networking equipment performance. Two in particular are RFC 1242 – "Benchmarking Terminology for Network Interconnection Devices" – and RFC 2544 – "Benchmarking Methodology for Network Interconnect Devices". [1,2] Most, if not all, networking infrastructure devices (i.e. switches, routers, hubs, and bridges) are tested against these two RFCs. NIST and ODVA are using these RFCs to develop metrics and tests for industrial equipment.

These RFCs cannot be directly applied to industrial networked equipment, since they were primarily developed for pass-through devices. Pass-through devices take data in one port and pass them out again through another port on the same device. Hubs and switches are examples of pass-through devices. Most industrial equipment within the scope of these metrics and tests are called end-point devices. End point devices (i.e. computers, sensors, and actuators) produce or consume network data packets. The primarily focus of this project has be to create metrics and tests that can use the common terms and methodologies as defined by the RFCs, but are still relevant to industrial equipment.

NIST and ODVA have borrowed four definitions in particular from RFC 1242 that provide a common set of terminology to describe both the tests and results. These terms are:

- **Throughput –** The maximum continuous traffic rate that a device can send/receive without dropping a single packet (frames per second at a given frame size).
- **Latency –** The time interval between a message being sent to a device and a corresponding event occurring.
- **Jitter/Variability –** The amount of change in the measured times for a series of events. This measure includes the difference between the minimum and maximum values as well as the standard deviation of the measured times.
- **Overload Behavior –** A description of the behavior of a device in an overload state. (This is qualitative.)
  - Overload states exist when the device's internal resources either receive too much information to process or bad information and the device goes into a state other than its normal run mode.
  - Data recorded for overload states should:
    - Describe the device behavior when its resources are exhausted.
    - Describe the system management response in an overload state.
    - Describe the device recovery from an overload state.

NIST and ODVA have also added two modifications to the Latency term.

- **Response Latency –** The closed-loop latency of a device to process a command and respond to it (e.g. a request for the device's serial number).
- **Action Latency –** The closed-loop latency of a device to process a command and return a desired physical output (e.g. analog/digital output signal), or the closed loop latency of a device to process a physical action (e.g. analog/digital input signal) and return a response.

The response latency and action latency terms have been added since the devices under test (DUTs) will be end point devices and cannot be simply tested on their ability to forward packets. More about the reasoning behind the development of these terms will be discussed in the section about the testing methodology.

3 BASIC DESCRIPTION OF ETHERNET/IP

ODVA released the EtherNet/IP standard in June of 2001. This standard took the Common Industrial Protocol (CIP) developed for DeviceNet and ControlNet and layered that over Ethernet, Transmission Control Protocol over Internet Protocol (TCP/IP), and User Datagram Protocol over Internet Protocol (UDP/IP). It allows simple input/output (I/O) devices like sensors/actuators or complex control devices like robots, Programmable Logic Controllers (PLCs), welders, and process controllers to exchange time-critical application information. [3]

EtherNet/IP uses a peer-to-peer and producer/consumer architecture for its data exchange versus a master/slave or command/response architecture. This allows for greater flexibility in the network and system designs, which fits better into the Ethernet networking model. In addition, EtherNet/IP splits its communications into configuration and management traffic (explicit messaging) and real-time, data I/O traffic (implicit messaging). Configuration and management traffic uses TCP/IP, and real-time I/O traffic uses UDP/IP. Since real-time I/O traffic uses UDP/IP, it must maintain information about the packet sequencing and connection at the CIP application layer. These are maintained as the EtherNet/IP Sequence Number and EtherNet/IP Connection ID. The performance metrics and tests described in this paper are primarily concerned with the real-time I/O traffic.

The producer/consumer model for EtherNet/IP allows multiple modes of communication to be chosen for real-time data exchange. The most common mode for producing data is called cyclic production. During cyclic production, the producer will send data at a particular rate called the Requested Packet Interval (RPI). The RPI and corresponding Accepted Packet Interval dictates the speed of the data produced over the network regardless of the rate at which the actual data values change.

EtherNet/IP also uses an object-oriented model. Some objects, such as the Identity object, TCP/IP object, and the Ethernet link object, are required by all EtherNet/IP devices. These map basic information about the device into the object model. Other objects are device specific, and while basic definitions of them may exist in the specification, the exact information recorded in the object is specific to the device and application.

4 PERFORMANCE TESTING METHODOLOGY

This section will describe some basic methodologies to test the performance of EtherNet/IP equipment; however, it will not describe the actual implementation of those test methodologies. While these tests are currently being developed for EtherNet/IP, they are generic enough as to be applicable to all industrial Ethernet.

4.1 CYCLIC/RPI TESTING

Since the RPI is the basis for most of the real-time I/O communications over EtherNet/IP, it is a natural place to start when looking at device performance. The ability for a device to maintain that RPI value under different conditions may be very important to the control system. No device will perform perfectly, so these tests will show how closely a particular device is able to maintain the RPI value. They will not determine a pass or fail value. That will have to be determined by the user and whether the device's performance characteristics meet the application's needs.

The basic premise of the cyclic/RPI tests will be to determine the maximum throughput of the device at a particular RPI value. Both producer and consumer devices will be tested. Devices will not be penalized for tests performed outside their published capabilities. The tests will result in a 2D matrix of maximum throughput versus RPI value. Pseudo-code for the test is shown below.

```
1. Setup RPI and Connection Size for the test
2. Establish connection with the device under test (DUT)
3. Communicate with the DUT for some time
4. Check for packet loss
5. Calculate statistics (min, max, average, standard deviation, histogram)
6. If packet loss, increase RPI value (decrease frequency) and continue
7. Else, report RPI and statistics
```

Implementing this pseudo-code is relatively easy for testing producer devices. The test equipment is required to listen to the packets produced by the DUT, allowing it to remain a passive participant in the test. Any device can be used to perform steps 1 to 3 in the above pseudo-code. By eliminating the need to have the test equipment do steps 1 to 3, the error reported due to the test equipment can be minimized.

EtherNet/IP uses a sequence number in its protocol, so checking for packet loss (or even out of sequence packets) in step 4 can be determined by following the sequence numbers that are produced by the DUT. The statistics calculated in step 5 relate to the throughput at the selected RPI and variability/jitter in the throughput value. If the test shows no packet loss then the test is successful and the values are reported as accurate. Otherwise, the test needs to be re-run at a larger RPI value.

While the test implementation for producer devices is relatively simple, the implementation for consumer devices is more complicated. Consumer devices are expecting to receive packets at the RPI rate, and may not indicate packet loss. This is because EtherNet/IP devices are not required to have two-way communications except for a heartbeat. Another difficulty is that the test equipment must be responsible for steps 1 to 3 or the device aiding the test may introduce error. The error in the consumer device would be subject to any error introduced by the producer device. This error can be determined by using the producer test described above, but cannot be overlooked. The only way to eliminate these types of errors would be to use the test equipment, which should have at least an order of magnitude more accurate timing than the DUT, to produce the data at regular intervals.

4.2 LATENCY TESTING

As mentioned earlier, NIST and ODVA have developed two modifications to the latency term for testing industrial devices. These tests relate to actual inputs and outputs to and from the device, and are the most relevant real-world tests. The two terms discussed before will be expanded upon and some possible test methodologies will be proposed.

4.2.1 RESPONSE LATENCY

Response latency tests the ability of a device to respond to a request for information. The test will be used to determine the efficiency of the communications stack. Since the command and response will only be reading information from the device's memory, there should be a minimum of processing overhead associated with the test results.

For EtherNet/IP, the simplest response latency test would be to have the device return its identity object information. Since the identity object is a specific EtherNet/IP and CIP concept, the command would have to be processed through the entire communications stack to the application layer. Other standard objects like the TCP/IP and Ethernet link object could also be selected, but the potential exists for devices to trick the test by responding at a lower level. Another test of the response latency would be to read data from an assembly object. But, the test procedure would have to be customized for each vendor and product, since assembly objects are not standard between devices. A time analysis of the response latency test is shown in Figure 1 and equations (1) – (3).

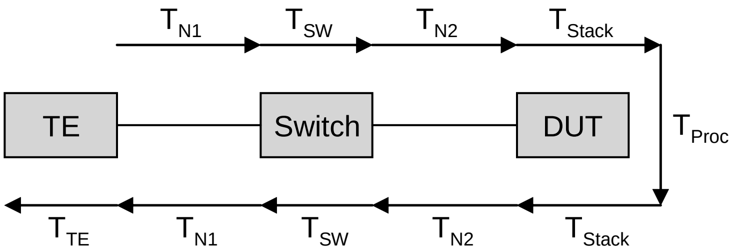

FIGURE 1 – TIME ANALYSIS FOR RESPONSE LATENCY TEST

$$T_{Network} = T_{N1} + T_{SW} + T_{N2} \tag{1}$$

$$T_{DUT\_RL} = 2 \times T_{Stack} + T_{Proc} \tag{2}$$

$$T_{RL} = 2 \times T_{Network} + T_{DUT\_RL} + T_{TE} \tag{3}$$

Where $T_{Network}$ = Latency time due to network overhead (s)
T_{N1} = Latency time due to the first network physical interfaces (s)
T_{SW} = Latency time due to the network switch or other infrastructure equipment (s)
T_{N2} = Latency time due to the second network physical interfaces (s)
T_{DUT_RL} = Latency time due to the Device Under Test for the Response Latency Test (s)
T_{Stack} = Latency time due to the DUT's network protocol stack (s)
T_{Proc} = Latency time due to the DUT's processor overhead (s)
T_{RL} = Latency time for the Response Latency Test (s)
T_{TE} = Latency time due to the test equipment (s)

4.2.2 ACTION LATENCY

Action latency tests the ability for a device to either cause or measure a physical action and determine the time between the action and the associated network packet. If the device is being commanded to act, it is the time between the device receiving the network packet and the action happening. If the device is producing data, it is the time between the physical action and the device sending the network packet. These tests will be highly device specific, and require application level programming on the part of the tester. These tests will also be affected by multiple error sources, since the test equipment may consist of more than one device.

In order to eliminate the need for multiple devices to execute the test, it may be possible to construct a loop-back test. This loop-back test would connect an output on the device to an input, and then command the device to send an output and wait for the input to be measured by the device. While not all devices will have both inputs and outputs, many of the devices subject to this category of testing will have both. The loop-back test would also limit the test equipment to one device, since the test equipment would only have to measure the time delay between network packets.

The loop-back test would be subject to many different types of errors and latencies. The test will be much more valuable to users than to developers, since it will not show the affects of the individual errors of latencies. The major sources of error and latency will probably be from the physical energy conversion creating an output signal and reading the input. These numbers are usually well known by the vendor and can be accounted for in the performance analysis. Another source of error and latency would be due to the processing overhead and network protocol stack. A time analysis of the action latency loop-back test procedure is shown in Figure 2 and equations (4) and (5).

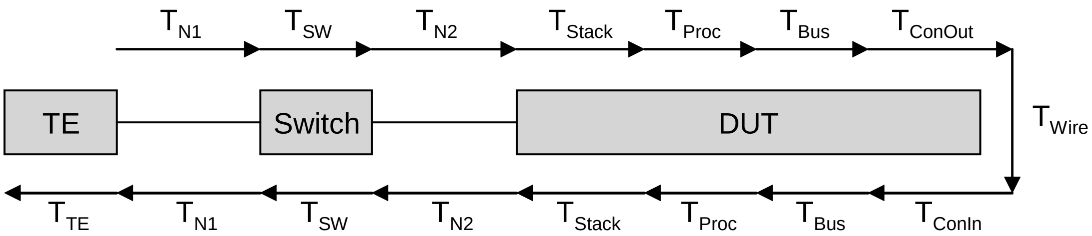

FIGURE 2 - TIME ANALYSIS FOR ACTION LOOP-BACK TEST

$$T_{DUT\_ALLB} = 2\times T_{Stack} + 2\times T_{\mathrm{Pr}oc} + 2\times T_{Bus} + T_{ConOut} + T_{Wire} + T_{ConIn} \tag{4}$$

$$T_{ALLB} = 2\times T_{Network} + T_{DUT\_ALLB} + T_{TE} \tag{5}$$

Where T_{DUT_ALLB} = Latency time for the Device Under Test for the Action latency Loop-Back Test (s)
T_{Bus} = Latency time for the internal device bus, may be zero if device does not use a bus (s)
T_{ConOut} = Latency time to perform the output energy conversion (s)
T_{Wire} = Latency time for the signal to travel along the wire (s)

T_{ConIn} = Latency time to perform the input energy conversion (s)
T_{ALLB} = Latency time for the Action Latency Loop-Back Test (s)

Other tests will need to be developed for measuring the latency in single input or output devices. For digital input or output devices, it may be possible to use some sort of frequency analysis to test this latency. For analog input or output devices, a ramp signal and an independent analog measurement or source device may be used to test the latency. These tests will require additional thought and development before they can be documented further. A time and data analysis of the possible test procedure for a digital or analog output device is shown in Figure 3 and equations (6) and (7).

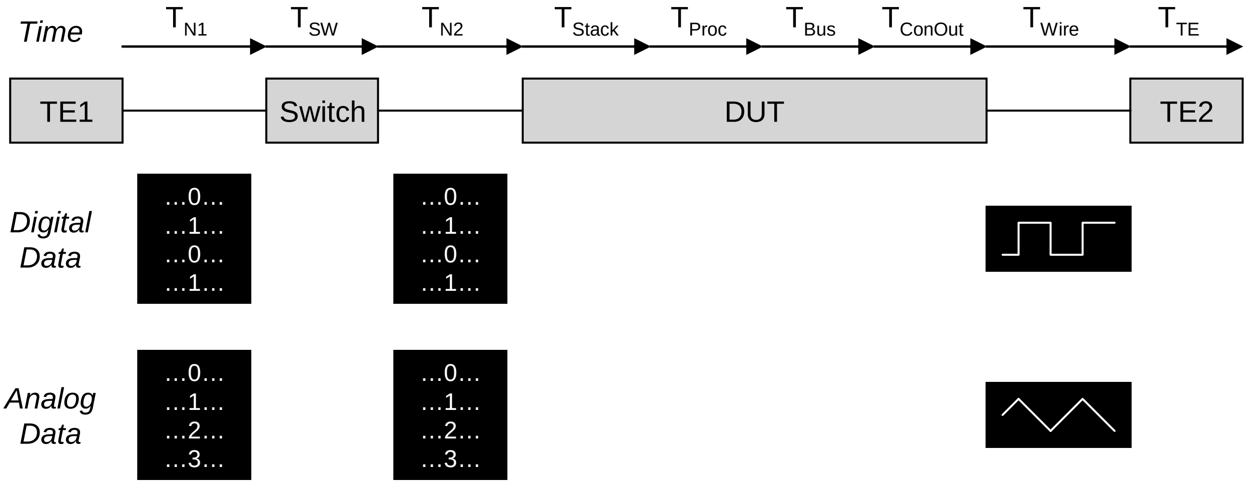

FIGURE 3 - TIME AND DATA ANALYSIS FOR ACTION LATENCY SINGLE OUTPUT FREQUENCY TEST

$$T_{DUT\_ALOF} = T_{Stack} + T_{Proc} + T_{Bus} + T_{ConOut} \tag{6}$$

$$T_{ALOF} = T_{Network} + T_{DUT\_ALOF} + T_{Wire} + T_{TE} \tag{7}$$

Where T_{DUT_ALOF} = Latency time for the Device Under Test for the Action latency Single Output Frequency Test (s)
T_{ALOF} = Latency time for the Action Latency Single Output Frequency Test (s)

5 ETHERNET/IP INTEROPERABILITY PLUG-FESTS

ODVA sponsored two EtherNet/IP Interoperability Plug-Fests where any vendor developing an EtherNet/IP product can come and determine how interoperable their device is with other vendors' products. The first event was held in March of 2003 and the second was held in January of 2004. During each of these events, performance tests were run to determine how well the different devices performed under varying conditions.

5.1 DESCRIPTION OF INTEROPERABILITY TESTING

The interoperability testing was split into two separate phases. During the first phase, every device was tested against every other device individually to determine what features worked and what features did not. The second phase incorporated all the devices into one large system, which had the devices attempt to communicate with one another in groups.

The individual device-to-device testing was intended to determine how well each device communicated with all the other devices. A series of communications tests was run for each device pair, resulting in a set of matrices that was analyzed to determine how well all the devices interoperated on an individual basis. The system-testing phase determined if there were additional interoperability issues communicating among multiple devices at the same time. Another series of communications tests was run for the groups of devices, resulting in a dataset for each group. The groups and tests were chosen based on the results of the device-to-device testing, so that devices were not additionally penalized for features that were already found not to work.

5.2 PERFORMANCE TESTING DURING PLUG-FESTS

While the interoperability testing was taking place, data was collected on the performance of the different devices. Since the performance tests were not the main focus of the event, only the cyclic/RPI producer testing, described earlier in this paper, was performed. This allowed the performance test equipment to remain a passive component in the system, while not disturbing the interoperability testing.

The performance test equipment was attached to the Plug-Fest infrastructure equipment and allowed to listen to all the traffic in the network. The performance test equipment recorded traffic from the network and sorted that traffic based on the Source IP address, Destination IP address, and EtherNet/IP Connection ID. This allowed the individual streams of real-time I/O traffic to be analyzed separately.

5.3 PLUG-FEST PERFORMANCE DATA ANALYSIS

For the first attempt at analyzing the Plug-Fest performance data, some basic statistical values were calculated for each of the individual real-time I/O traffic streams. The statistical values calculated were the average, minimum value, maximum value, and standard deviation. These statistical values relate somewhat to the metrics discussed earlier, but they will have to be refined before a more direct relationship can be shown. The average relates to the throughput and clock skew, while the minimum, maximum, and standard deviation relate to the jitter or variability in the throughput.

While none of the devices performed exactly the same, it was possible to see different trends in the data and determine possible causes for their occurrence. These trends usually related to different spike patterns in the data. The following sections describe some of the more easily identifiable trends.

5.3.1 HARDWARE VERSUS SOFTWARE CLOCK

One of the first trends that were seen in the data is whether a device uses a hardware or software clock. The spike pattern trends in the data sets usually related to some overhead in the processor that caused a packet to be delivered late. For devices using a hardware clock for their RPI timing, the next packet showed a subsequent early packet after each late packet, resulting in a downward spike at almost the same distance. The downward spike was caused by the device returning to the original RPI sequence time. Devices that use a software clock for their RPI timing did not show this downward spike. Their timing was always calculated based on the time the last packet was sent. Figure 4 and Figure 5 show examples of data from a device with a hardware clock and a software clock.

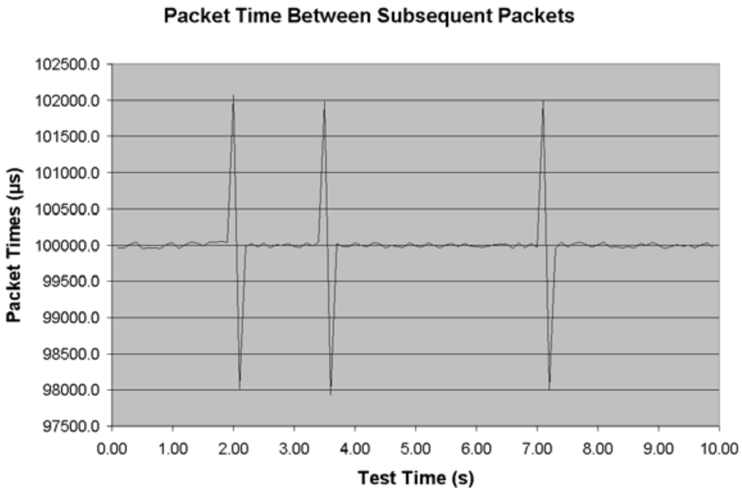

FIGURE 4 - EXAMPLE DATA FROM A DEVICE WITH A HARDWARE CLOCK

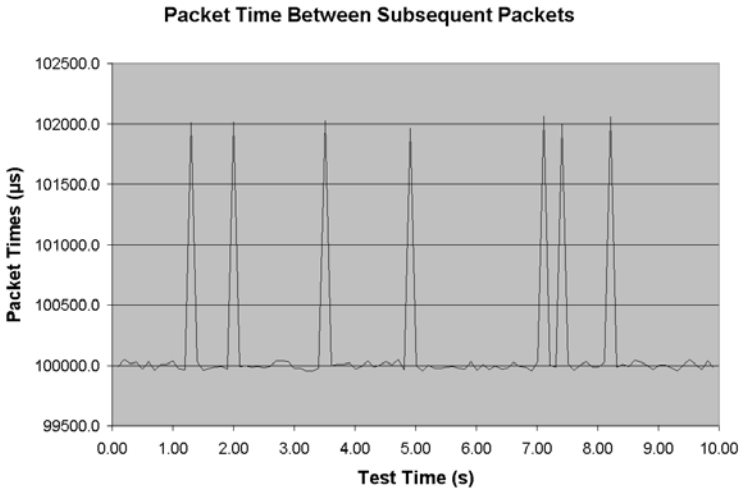

FIGURE 5 - EXAMPLE DATA FROM A DEVICE WITH A SOFTWARE CLOCK

5.3.2 CLOCK RESOLUTION

It was also possible to determine the clock resolution used by some devices. These devices tended to be software clock devices, since hardware clocks run at much higher frequencies than could easily be measured by the test equipment. The data for software clock resolution showed large blocks of spikes with easily identifiable steps in the data. Figure 6 shows an example of a device that uses a software clock and has a clock resolution easily identifiable from the spike pattern.

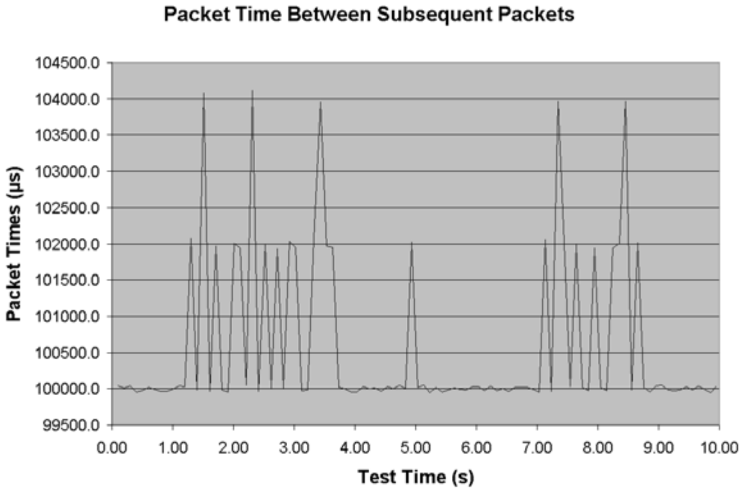

FIGURE 6 - EXAMPLE DATA SHOWING THE SOFTWARE CLOCK RESOLUTION

6 SUMMARY

With some work, it is possible to modify the IT-based metrics and tests for Ethernet device performance and use them to describe the performance of industrial Ethernet devices. This is beneficial, since a great amount of work has gone into developing the IT performance metrics and tests. Many great minds have worked for years to develop these tests, and it would be foolish not to leverage that work for industrial Ethernet.

However, the metrics and tests for industrial Ethernet devices are going to be substantially different, when implemented, due to the limitations of industrial devices. Also, the methodologies developed for industrial Ethernet performance will require different implementations from device to device. Given the benefits to users, who at this point have very limited ways to compare similar products from multiple vendors, it is necessary to try and standardize as much of these metrics and tests as possible.

This paper has presented some possible metrics and tests for industrial Ethernet equipment. While these tests are being specifically developed for EtherNet/IP devices, it is NIST's goal to introduce these metrics and tests to other industrial Ethernet standards groups once they have been proven for EtherNet/IP. By working through all of the development issues with one network first, the migration to other industrial Ethernet networks should be reduced.

7 REFERENCES

1. Bradner, S., *Benchmarking Terminology for Network Interconnection Devices,* RFC 1242, July 1991.
2. Bradner, S., McQuaid, J., *Benchmarking Methodology for Network Interconnection Devices*, RFC 2544, March 1999.
3. *EtherNet/IP Specification*, ControlNet International & Open DeviceNet Vendor Association, Release 1.0, June 5, 2001.
4. Lian, F., Moyne, J., Tilbury, D., "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet", *IEEE Control Systems Magazine*, February 2001, pp. 66-83.