Test Tool for Industrial Ethernet Network Performance

James Gilsinn & Freemon Johnson
Intelligent Systems Division (ISD)
Manufacturing Engineering Laboratory (MEL)
National Institute of Standards and Technology (NIST)

Keywords: industrial, Ethernet, network, performance, test tool, data analysis

INTRODUCTION

Ethernet is being used by a wider variety of industrial devices and applications. Industrial applications and systems require deterministic operations that traditional Ethernet and Transport Control Protocol / Internet Protocol (TCP/IP) suites were not originally designed to support. A standardized way to describe and test industrial devices is needed in order to aid users to characterize the performance of their software and hardware applications.

The Manufacturing Engineering Laboratory (MEL) of the National Institute of Standards and Technology (NIST) has been working to develop a set of standardized network performance metrics, tests, and tools since 2002. NIST has cooperated with standards organizations and other groups during that time.

NIST is presently working on developing an open-source test tool, called Industrial Ethernet Network Performance (IENetP), to aid vendors in characterizing the performance of their devices. The IENetP test tool will be capable of conducting a full series of performance tests and reporting the results to the user. The current version of the software is capable of analyzing network traffic and producing statistics and graphs showing the network performance of a device.

The first section of this paper describes the background of the project, including the problem, why NIST is involved in the project, and a history of NIST's previous work in this area. The second section describes the overall performance testing methodology, including the basic performance metrics, testing methodology used when conducting industrial Ethernet performance tests, and a description of the test system used to conduct the performance tests. The third section describes the data analysis methodology that the IENetP test tool follows. The fourth section describes the IENetP test tool makeup, design process used in its development, and more information about the specific requirements of the test tool. The final section of the paper describes the future plans for the IENetP test tool. NIST has recently released the first public version of the test tool and plans future versions that will expand on different features and capabilities of the project.

BACKGROUND

THE PROBLEM

While Ethernet and the TCP/IP suite are inherently non-deterministic protocols, it is possible to use them for real-time industrial networks. The development of high-speed Ethernet interfaces, switched network infrastructures, and specialized TCP/IP network stacks have allowed a multitude of industrial Ethernet protocols to operate in the millisecond range. The large variety of different protocols and vendors have caused end-users to ask many questions, including:

- Which industrial network performs better for my application?
- Which vendor's products will satisfy my given requirements?
- How will a particular device perform compared to another?
- How does one performance metric compare to another?
- How well will a particular product work in my control system?

Defining performance characteristics of industrial Ethernet applications and devices is analogous to comparing the performance of automobiles. How would one rate the performance of an automobile? Choosing the type of vehicle is one of the first steps when choosing an automobile, since the performance metrics are quite different for each type. Does one's application call for a sports car, an economical commuting car, a large pickup truck, or a minivan? Once the type of vehicle is chosen, it is necessary to compare vendors and choose which one has the best performance characteristics. In the context of a sports car, horsepower, 0 to 60 mph time, and cornering ability all describe different aspects of the performance of a sports car. The weighting that one places on each of those metrics depends on their application. The same idea can be applied in the industrial control system workspace.

Having a standardized way to measure the performance metrics also aids end-users. Using an automobile example again, one standardized metric and method designed by the U.S. Department of Transportation is the fuel economy test. It is based on a standardized series of tests and produces two commonly known metrics, city and highway fuel economy. These metrics can be used to compare vehicles from multiple vendors to determine how the vehicles meet the requirements for their particular application. One of NIST's long-term goals for this project is to develop standardized methods to measure the industrial Ethernet performance metrics.

WHY NIST?

The mission statement of NIST is "to promote U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology in ways that enhance economic security and improve our quality of life." MEL is specifically assigned to "promote innovation and the competitiveness of U.S. manufacturing through measurement science, measurement services, and critical technical contributions to standards. … Developed collaboratively with our external partners in industry, academia, and other government agencies, MEL measurement and standards solutions allow our customers to overcome barriers to product and process innovation, to share manufacturing information seamlessly and accurately, and to take full advantage of the latest technologies essential to their competitiveness and future success."

These broad mission statements give NIST and MEL the authority to investigate standards that help U.S. industries with respect to their long-term objectives. NIST and MEL promote standardization for the U.S. industry, which can easily benefit the entire industrial community.

NIST is uniquely positioned to provide a standardized approach for industrial Ethernet network performance metrics, tests, and tools. NIST can take a wide view at the issue of the industrial Ethernet performance and focus on what will provide the most meaningful metrics for both the vendors and the end-users. NIST has no direct affiliation with any particular group or technology which is why NIST is in a unique position to lead this effort.

HISTORY OF INDUSTRIAL ETHERNET PERFORMANCE AT NIST

NIST began looking at industrial Ethernet network performance in 2002. Industrial Ethernet was seen as an emerging technology and there was no common way to describe the performance of different devices. Due to the prevalence of Common Industrial Protocol (CIP)[1] based networks in the U.S. auto manufacturing industry and existing relationships with those organizations, NIST choose to investigate the Ethernet / Industrial Protocol (EtherNet/IP) for its initial efforts. This led NIST to join the EtherNet/IP Implementors Workshop series, part of ODVA, Inc., to learn more about the network and promote the idea of network performance metrics and tests. [1][2] The workshop series provides an open forum for vendors to discuss topics related to implementing EtherNet/IP and promote interoperability between the vendors.

Large portions of the workshop's efforts have been aimed at developing a set of interoperability recommendations and testing those recommendations at "PlugFests." [3][4][5] The PlugFests are scheduled twice per year and allow vendors to bring their products and engineers to one location to see how well their devices interoperate with other vendor's products in a collaborative environment. NIST has participated in every PlugFest, conducting network performance testing and helping to manage the events. The performance tests have expanded from just capturing network traffic during interoperability testing to conducting one-on-one performance testing on each device present at the PlugFest with pass/fail criteria.

In late 2005, NIST and the United States Council for Automotive Research (USCAR) signed a memorandum of agreement and formed the United States Alliance for Technology and Engineering for Automotive Manufacturing (U.S. A-TEAM). The main focus of this effort is to improve the manufacturing processes used by the members of USCAR in order to reduce their costs and the costs of their first- and second-tier suppliers. Industrial Ethernet performance fell under the Plant Floor Controllers Task Force as part of this agreement. The highest priority task from this group was to form a commercial laboratory where their suppliers could obtain certified performance test results for their devices. USCAR's objective was to improve the overall reliability of their manufacturing systems by knowing the performance characteristics of the devices they purchased for a particular application. In 2006, in response to this task, NIST and ODVA, Inc. formed a collaborative research agreement to develop a software test tool that ODVA could use as the basis for a commercial laboratory capable of

[1] Certain commercial products and vendors are mentioned in this paper for informational purposes only. Their mention does not indicate a recommendation or endorsement by the National Institute of Standards & Technology or the U.S. Department of Commerce.

conducting fee-for-service performance testing on EtherNet/IP devices. NIST transfered the test tool to ODVA at the end of 2007 and ODVA started their performance testing service in 2008.

In parallel to the NIST/ODVA agreement, the EtherNet/IP Implementors Workshop group decided to add performance testing to its interoperability recommendations document as pass/fail criteria. NIST was tasked with leading the effort to develop the performance recommendations and the pass/fail criteria in the spring of 2007. Modifications were made to the proposal until a final set of recommendations was released in the fall of 2008. The performance tests and criteria for EtherNet/IP devices have been added to the PlugFest test plans and are scheduled to become required elements of the PlugFest testing in 2009.

The ODVA performance testing service provides a way for vendors to certify the performance metrics for their devices; however, it does not provide a way for vendors to obtain performance characteristics during their development lifecycle. NIST is continuing its research into industrial Ethernet performance by developing the IENetP test tool. The test tool is freely available to anyone, allowing vendors to conduct performance testing on their devices at any stage of development and under various conditions.

PERFORMANCE TESTING METHODOLOGY

PERFORMANCE METRICS

There are two basic types of communication methods used in most industrial Ethernet devices currently available:

- Publish/Subscribe, also called Peer-to-Peer, and
- Command/Response, also called Master/Slave.

For publish/subscribe or peer-to-peer communications, two or more devices communicate with each other in some way that the devices themselves negotiate. This may be at an understood rate or at some pre-determined condition. For example, device A wants to get a digital input value from device B at a rate of twenty times a second. Device A sends a message to device B requesting the particular value and specifies the particular rate. Device B can accept this request or deny it based on its configuration. If device B accepts the request, then it starts sending messages to device A, and possibly other devices, twenty times a second. Other than the initial request, device A does not dictate when device B will send its messages. The true rate at which the messages are sent depends solely on device B's internal hardware and software architecture.

For command/response or master/slave communications, two devices communicate with each other based on how the commander or master device dictates. Responder or slave devices can be relatively inexpensive and unintelligent, since their sole purpose is to process commands and respond back. Following the prior example, if device A wants to get a digital input value from device B at a rate of twenty times a second, device A sends a message to device B twenty times a second for that particular value. Device B responds back with the value as quickly as it can. The true rate at which the messages are sent depends on both device A's and device B's internal hardware and software architectures and the network connecting the two devices.

Based on these two types of communication methods, two main performance metrics appear:

- Cyclic Frequency Variability/Jitter, and
- Latency.

When communicating at an understood rate, the ability for the devices to maintain the desired message rate is extremely important. Control loops based on this type of communication count on the message streams to be maintained at the desired rate. Control systems theory states that the communications used in a control loop should operate at least twice as fast as the overall loop; however this is not always the case in practice. For tightly coupled control loops that operate at or near the same rates, variability or jitter in the packet interval may affect the system's performance in unintended ways.

When responding to a particular command or pre-determined condition, the ability for a device to process the command or condition quickly is most important. An unexpected delay or latency in the response message being sent from the device may seriously affect the system's performance behavior.

Real-time EtherNet/IP typically uses a form of publish/subscribe communications with two parallel streams of traffic, each flowing in the opposite direction. For EtherNet/IP, the desired packet rate is called the Requested Packet Interval (RPI). When a device is requested to produce network traffic at a particular RPI, it is required to send back an Accepted Packet Interval (API) to the requester. This API value represents the agreed upon rate that each device expects to receive network packets for that particular traffic stream. Most devices use the same API rate for ingoing and outgoing real-time network streams, even though it is not a requirement of the EtherNet/IP specification.

The performance test system uses network capture files to verify that the device under test (DUT) maintains its desired API. The measured packet interval (MPI) is the rate at which the test system receives packets from the DUT.

TESTING METHODOLOGY

The basic methodology for the IENetP test system is fairly simple, regardless of the metric being measured [6]. The process, and test tool engine, does not have to be changed to suit a particular metric, background traffic, or analysis method being used in the test. The following is a procedural listing of the basic methodology used by the performance test system:

1. Begin recording network traffic
2. Establish a connection with the device under test (DUT)
3. Begin transmitting background network traffic, based on the particular test conditions
4. Wait for a given amount of time
5. Stop transmitting background network traffic
6. Close the connection with the DUT
7. Stop recording network traffic
8. Analyze the network traffic capture and report the results

The current version of the IENetP test tool is not capable of communicating directly with the DUT, capturing traffic, or issuing background traffic. The current test tool is primarily a data analysis tool, and is only used for step 8 in this methodology; the user is responsible for communicating with the DUT and capturing the data with the current test tool. Future versions of the software will incorporate a greater portion of this methodology. Until the IENetP test tool is capable of communicating with the DUT directly, NIST plans on producing a recommended testing procedure that requires specific background traffic types and amounts. The user is responsible for transmitting the background traffic on the network with the current version of the tool.

TEST SYSTEM

The performance test system has been designed to be extremely flexible, thus allowing the user to determine the performance metrics for their desired application. The test system can be as simple as attaching a cross-over Ethernet cable between the tester and the DUT, as shown in Figure 1, or it could be as complex as a large set of infrastructure devices between the tester and the DUT, as shown in Figure 4. When testing the performance for one particular device, it is important to isolate the device from a network to remove any latency introduced by other infrastructure devices. That is why it is recommended the test system be directly attached to the DUT to keep the latency to the absolute minimum, as shown in Figure 1. When using a wireless DUT, it may be necessary to use a wireless access point or other network hardware to connect to the DUT unless the tester has a wireless interface, as shown in Figure 2. When trying to analyze the performance of a system, the test system may be connected to the system as shown in Figure 3 and Figure 4. These figures show that the test system may be split into two time-synchronized devices, although there is no requirement to split the functions for test systems with enough network ports. Network taps are shown in these figures since they are designed to introduce no collisions or latency that a conventional network hub might introduce.

The raw data presented in Figure 5 and Figure 6 was captured during one of the EtherNet/IP Interoperability PlugFests. The raw data was reanalyzed using the IENetP Test Tool, resulting in the figures shown in this paper. The tester in this case was a specialized network analyzer directly attached to the DUT, as shown in Figure 1, capable of recording data with multiple orders of magnitude better timing accuracy and precision than the DUT.

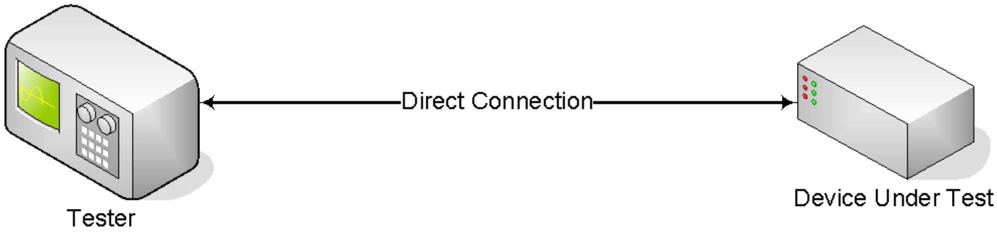


Figure 1 – Test System Directly Connected to DUT

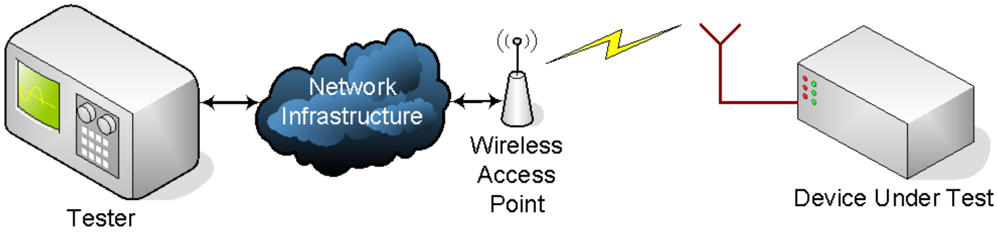


Figure 2 – Test System Connected to Wireless Access Point

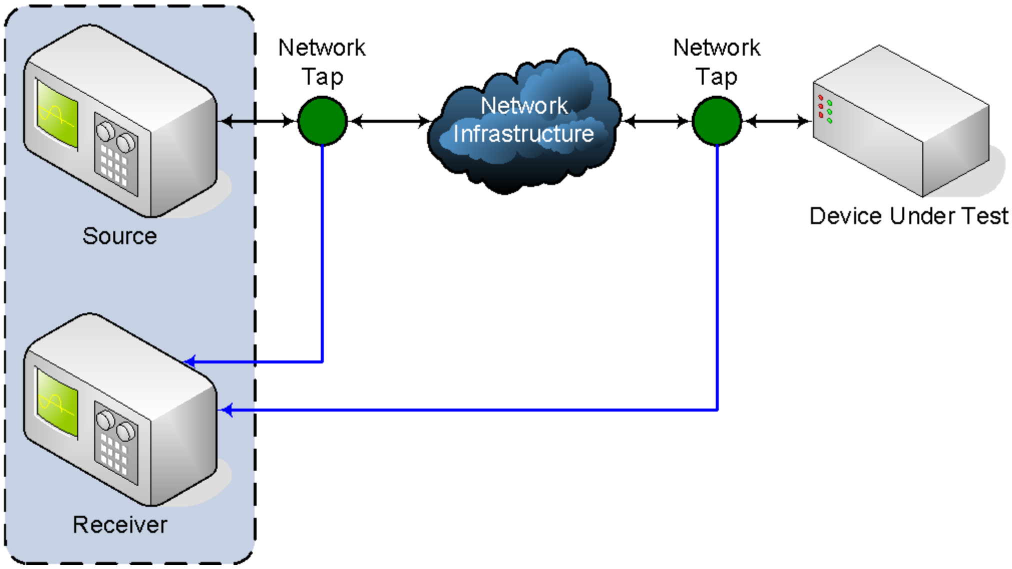


Figure 3 – Test System Connected to DUT through Wired Infrastructure

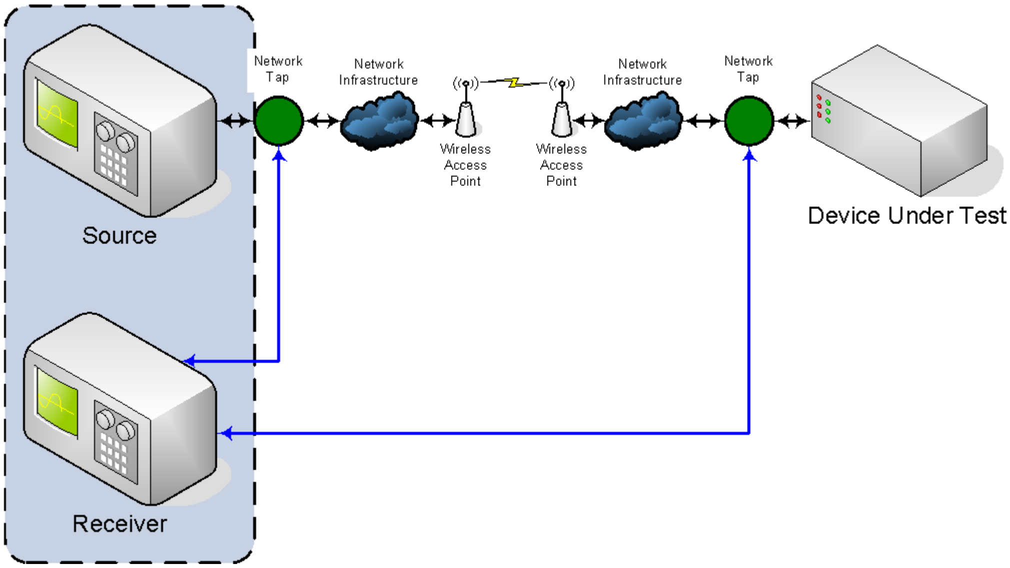


Figure 4 – Test System Connected to DUT through Wireless Infrastructure

DATA ANALYSIS

The IENetP test tool currently supports only data analysis. The data analysis method used in the most recent version of the test tool is a distribution analysis of the cyclic frequency variability/jitter of the MPI, calculating the following values: minimum, maximum, mean, standard deviation, skewness, and kurtosis. The objective is to measure how well the DUT adheres to its configured RPI/API value while operating in a variety of network conditions.

The IENetP test tool is capable of extracting multiple network traffic streams from the same network capture file. The network capture files collected during a performance test usually contain traffic unrelated to the desired network traffic stream, including the injected background network traffic and other extraneous protocols. Each packet in the capture file is analyzed for its source address, destination address, protocol type, and, in the case of EtherNet/IP, the connection identifier. These four items allow the IENetP test tool to parse and collate various network traffic streams.

After the network packets are identified, the timing data is extracted from the capture file and used in the distribution analysis. The last step in the analysis process is to generate graphs of the MPI for the user. The data graphs have provided the most value to the vendors in characterizing the performance of their devices. An example MPI graph is shown in Figure 5. An ideal device would produce an MPI identical to the desired RPI/API. There are many reasons why there are not ideal devices, including limited processing resources, network overhead, and inefficient software designs to name a few.

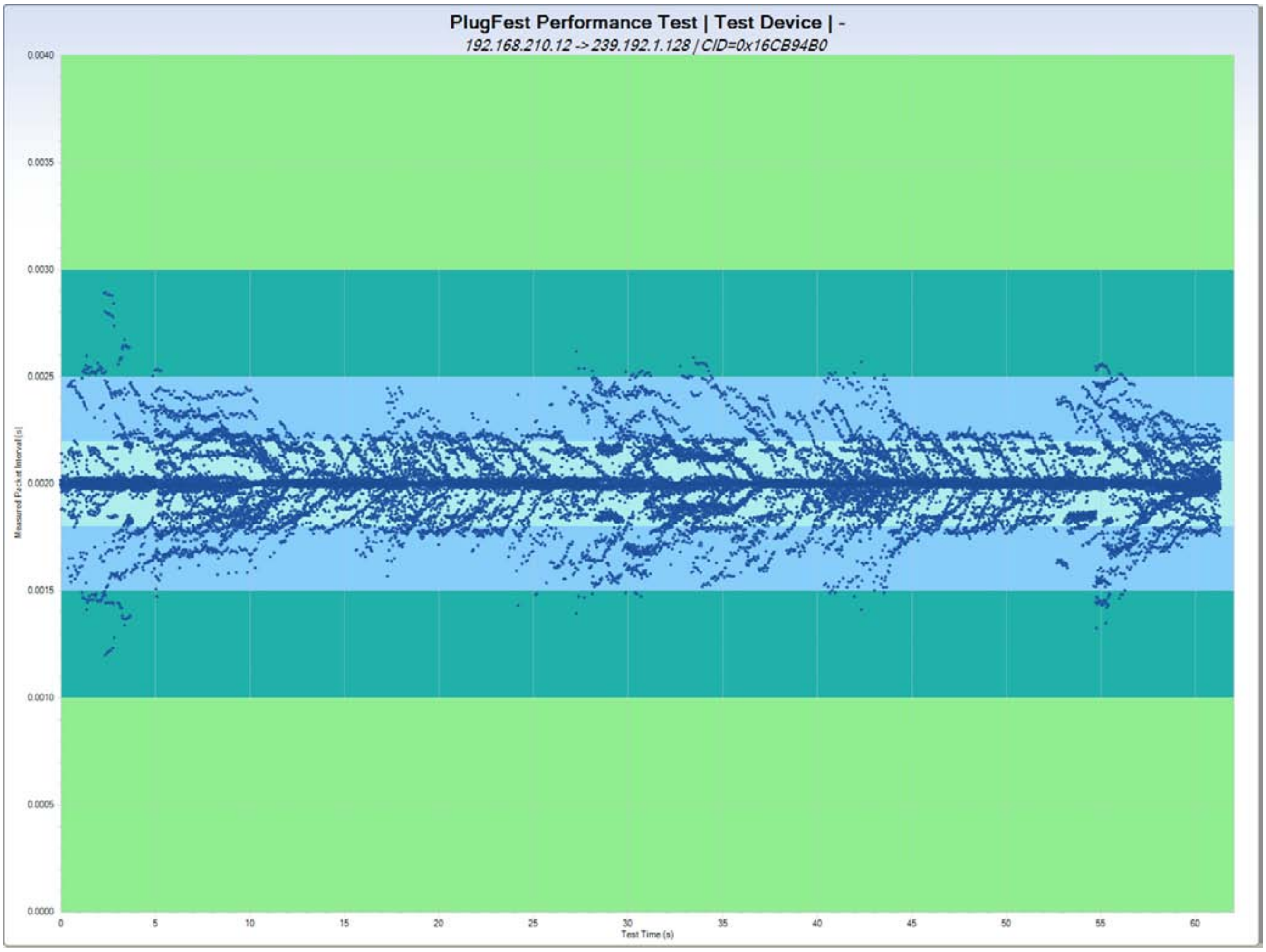


Figure 5 – Example Performance Test Results, Time Domain Graph

Figure 5 and Figure 6 show the graphical results produced by the IENetP test tool from a single test. Figure 5 shows a time-based version of the dataset where the Y-axis of the graph shows the MPI for each packet in the network traffic stream, and the X-axis shows the time at which the particular packet was timed and captured during the test. The origin of the X-axis is the first packet recorded in the network capture file. With respect to that origin, all packets captured thereafter are processed to show the deviation from the desired RPI/API. Colored horizontal bars are used to represent different percentages of the MPI (±10 %, ±25 %, and ±50 % in this particular example). The percentages and colors chosen for the bars are user-selectable based on their particular performance requirements. Figure 6 shows a histogram of the dataset. In this example:

- Desired RPI/API = 2.000 ms
- Mean MPI = 2.000 ms
- Minimum MPI = 1.198 ms
- Maximum MPI = 2.894 ms
- MPI Standard Deviation = 0.142 ms
- MPI Skewness = 0.146389
- MPI Kurtosis = 2.942464

Figure 6 and the data shown above indicate that the dataset has a distribution that is closely centered at the mean with a very sharp peak and longer tail on the right side of the graph.

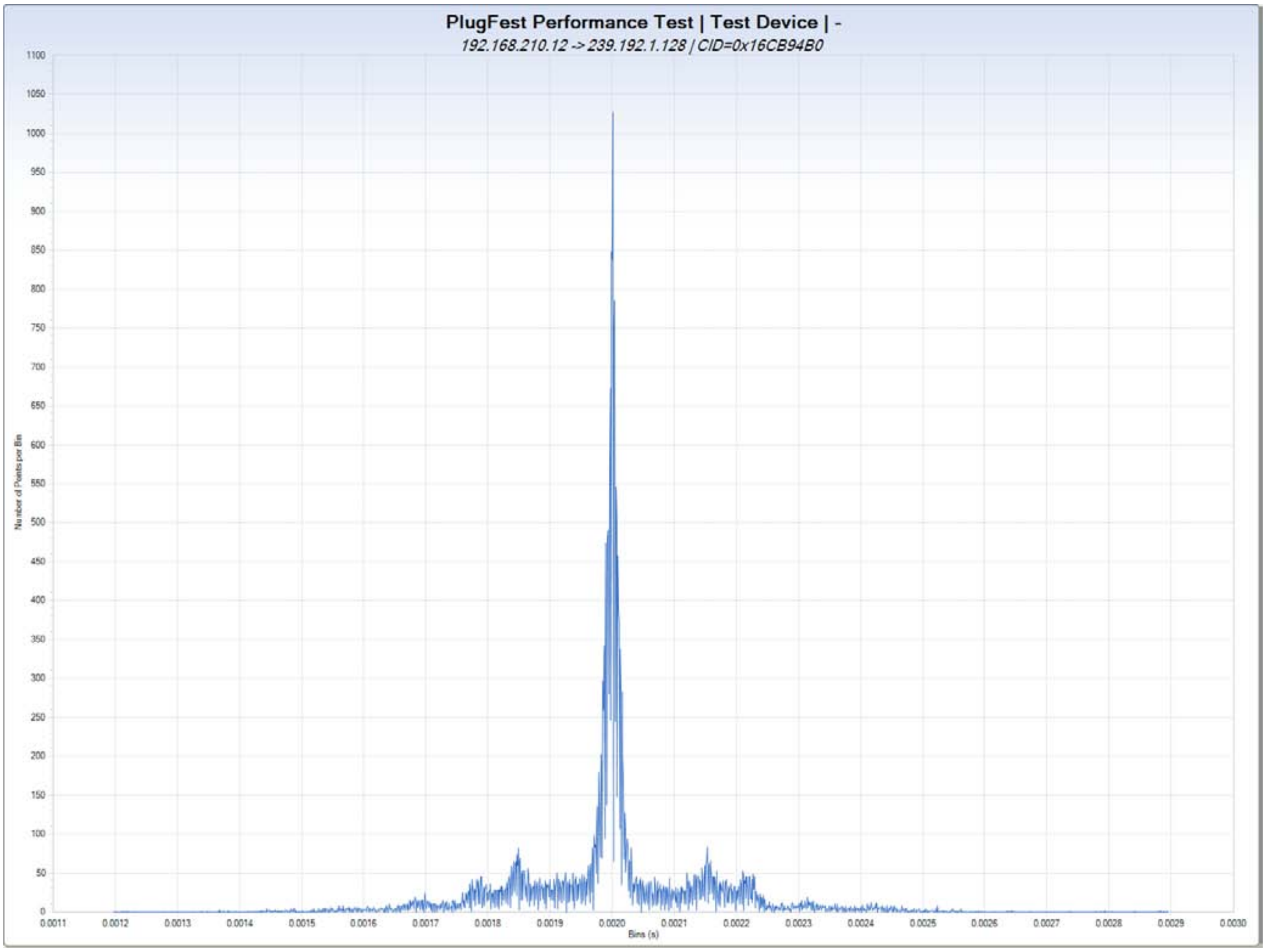
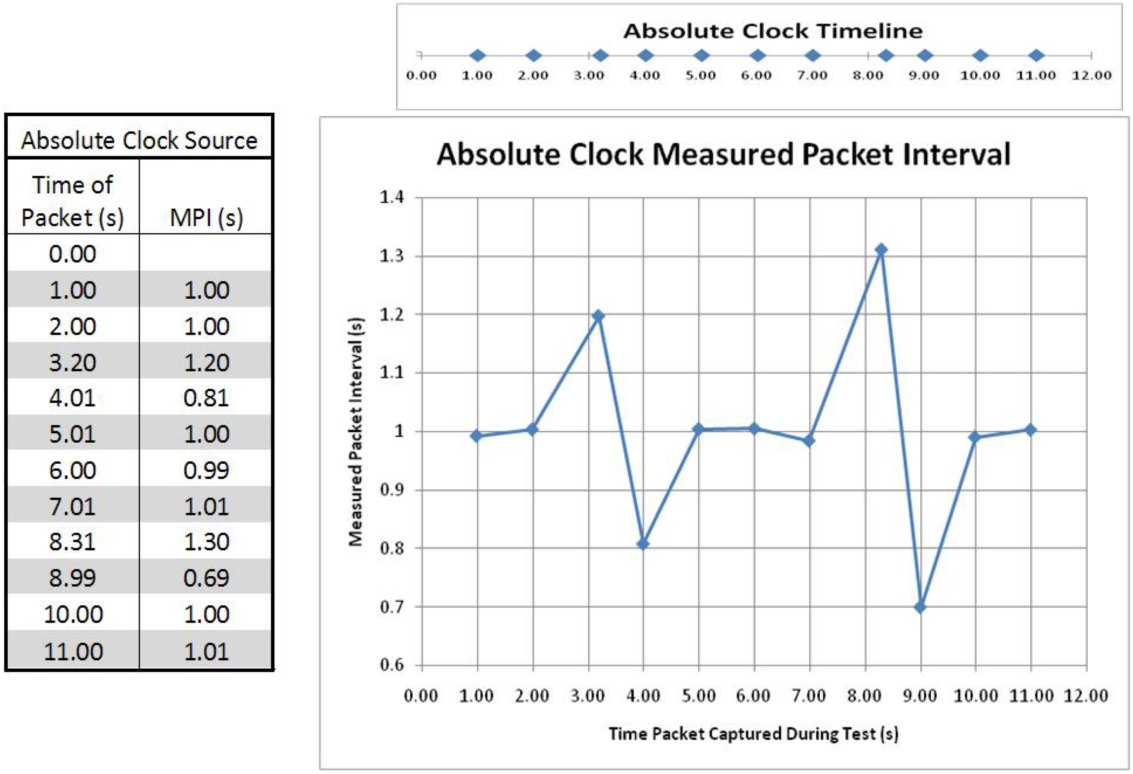


Figure 6 – Example Performance Test Results, Histogram Graph

One phenomenon to observe in Figure 5 is the envelope pattern to the MPI. This envelope pattern may indicate a performance issue internally that the vendor may need to investigate, including issues involving the performance of network stack software, microprocessor overhead, etc. These graphs help to point-out various issues and events for a device during operation that a vendor or manufacture may not have been aware of by other means.

As a result of PlugFests testing, issues surrounding a DUT's internal clock source have manifested. One example is the performance difference between devices using an absolute clock versus ones using a relative clock. In the context of industrial devices, an absolute clock source is one that uses an asynchronous method for determining the timing of particular actions. Many such devices use hardware-assisted, interrupt-driven clocks. The timing of each network packet sent by the device is relative to the clock source, and has no relation to other network packets sent by the device. A relative clock source is one that uses a synchronous method for determining the timing of particular actions. The timing of each network packet is typically related to the previous network packet, not a known clock source.

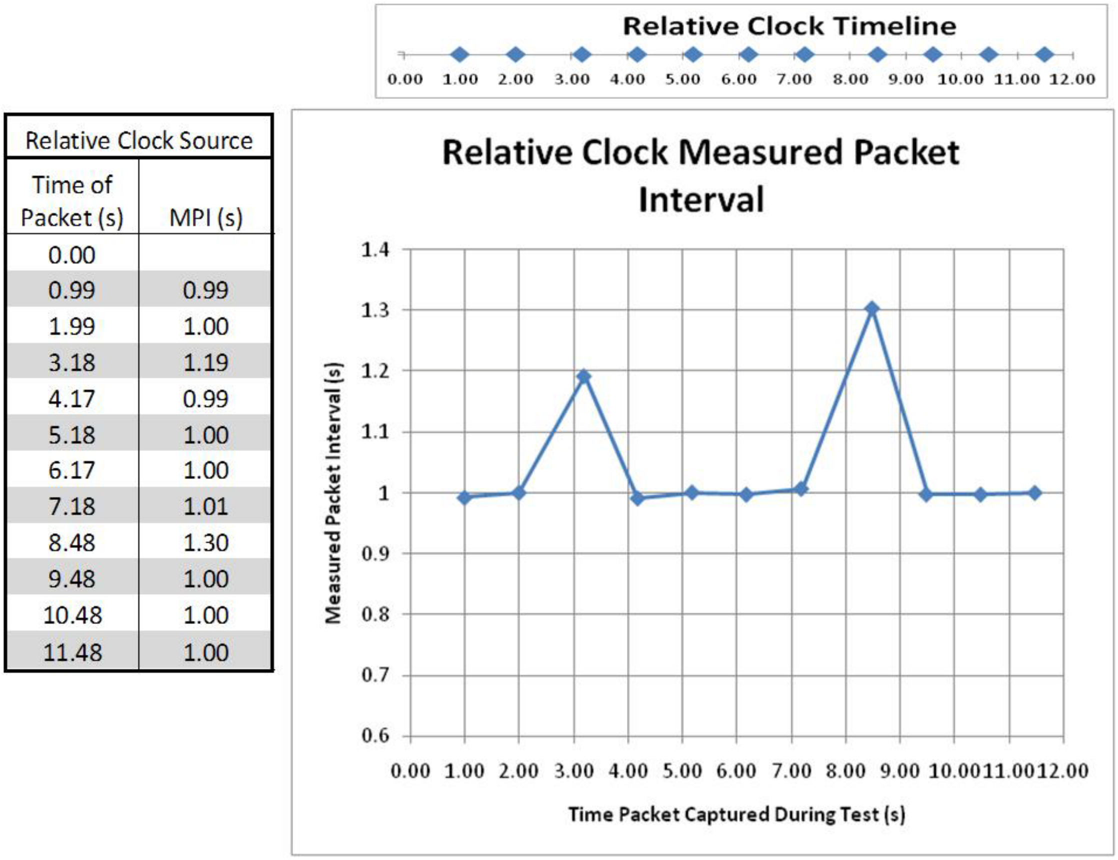
A simulated example of a device using an absolute clock source is shown in Figure 7. In this example, two packets have been delayed, the first by approximately 200 ms and the second by approximately 300 ms. For each of the delayed packets, the next packet in the sequence returns to the original time sequence, appearing to the user to be received early. The amount of time that the first packet comes late appears to be approximately equal to the amount that the next packet comes early, effectively creating a two-sided envelope for the resulting MPI values.



Absolute Clock Source	
Time of Packet (s)	MPI (s)
0.00	
1.00	1.00
2.00	1.00
3.20	1.20
4.01	0.81
5.01	1.00
6.00	0.99
7.01	1.01
8.31	1.30
8.99	0.69
10.00	1.00
11.00	1.01

Figure 7 – Absolute Clock Source Example

An example of a device using a relative clock source is shown in Figure 8. In this example, the same two packets have been delayed as in the absolute clock example. For each of the delayed packets, the next packet in the sequence arrives at approximately the desired RPI/API. The graphs only show delayed packets with no corresponding early packet, creating a single-sided envelope for the apparent MPI values. When related to the desired timeline for packets, each of the packets gets progressively worse with each delay.



Relative Clock Source	
Time of Packet (s)	MPI (s)
0.00	
0.99	0.99
1.99	1.00
3.18	1.19
4.17	0.99
5.18	1.00
6.17	1.00
7.18	1.01
8.48	1.30
9.48	1.00
10.48	1.00
11.48	1.00

Figure 8 – Relative Clock Source Example

TEST TOOL

The objective of the IENetP test tool is to provide meaningful data to the vendor without having to know the statistical analysis in depth. The user interface will be designed to allow the vendor to specify many options to process and display data in various ways while the tool takes care of the underlying statistical analysis yielding those results. The IENetP test tool will allow the user the flexibility to generate customized reports that are relevant and meaningful for their device.

A multi-phase software development lifecycle is being used when developing the IENetP test tool. This will entail all aspects from the analysis through testing and deployment. During those phases, support for various protocols, test designs, data analysis, and other features will be added to the framework.

The IENetP test tool is targeted to allow vendors to produce better devices with respect to performance characteristics. The application is an object-oriented application programmed in Visual C# and built on the .NET version 2.0 framework from Microsoft. To speed the development process and provide the widest array of features, a commercial graphical package is incorporated into the test tool. The application programming interface will also allow users to integrate their own graphical package as well should they choose not purchase and incorporate the package that is presently utilized. The test tool can be used on a computer with Windows 2000 or higher operating system and no extra hardware necessary. The only extra tool required to run the current version of the test tool is Wireshark, a commonly available open-source network capture and packet analysis software package.

The source-code for the IENetP test tool is considered public domain, while the compiled version of the software is considered open-source. Without delving deeply into licensing laws and policies, public domain software "may be freely reproduced, distributed, transmitted, used, modified, built upon, or otherwise exploited by anyone for any purpose, commercial or non-commercial, and in any way, including by methods that have not yet been invented or conceived." [7] Open-source software is very similar to public domain software, although there are some subtle differences. The biggest differences requires that derived works must also be covered under the same license as the original work and that records need to be kept on who makes what changes to the software. [8] The source-code and a compiled installer are available from SourceForge, an open-source project hosting site that provides many different features for developers and users on various computing platforms. [9]

Even though IENetP is designed to use Wireshark for its data analysis, it does not inhibit the user from using alternative methods for capturing network packet files as long as the capture files are in a Wireshark readable format. The choice of hardware used to capture network packets will have an effect on the overall performance reported by the IENetP test tool based on the precision and accuracy of its clock source.

FUTURE PLANS

The basic methodology and capabilities for the IENetP test tool have not changed, in principle, since they were first introduced in 2005 while working on the ODVA, Inc. testing laboratory. No additional performance metrics, mathematical analysis methods, or networks other than Ethernet/IP have been investigated. NIST is planning to release additional versions of the IENetP test tool to add these types of functionality.

NIST released the first version of the IENetP test tool in March 2009. The software is functional, but is still missing many capabilities. Version 2.x of the software will focus on adding additional mathematical analysis methods and performance metrics. While NIST is planning on improving the mathematical analysis methods, the test tool will hide the complexity of the calculations by presenting the user with data that will be easily understood and compared. The next major performance metric to investigate is latency, which will allow the test tool to analyze a larger number of industrial networks and communication protocols. Version 3.x of the software will focus on industrial Ethernet protocols other than EtherNet/IP. Some examples of other protocols are: Modbus/TCP, Profinet, Foundation Fieldbus HSE, ISA-100.11a, IEEE 802.11/WiFi, and Zigbee.

Later versions of the IENetP test tool will progress from an analysis tool to an active testing tool. This will require the test tool to be capable of communicating directly with the DUT and capturing network traffic without the need for additional intervention from the user or any extra hardware or software assistance.

REFERENCES

[1] Performance Test Terminology for EtherNet/IP Devices, Version 1.1, March 14, 2005, PUB00080R1.1, EtherNet/IP Implementors Workshop.

[2] Performance Test Methodology for EtherNet/IP Devices, Version 1.0, March 14, 2005, PUB00081R1, EtherNet/IP Implementors Workshop.

[3] Recommended Functionality for EtherNet/IP Devices, Version 2.0, November 10, 2008, PUB00070R2.0, EtherNet/IP Implementors Workshop.

[4] EtherNet/IP Interoperability Test Procedures, Version 2.0, November 10, 2008, PUB00095R2.0, EtherNet/IP Implementors Workshop.

[5] Knake, K., Gilsinn, J., "EtherNet/IP Interoperability Recommendations," *ODVA 2009 Conference & 13th Annual Meeting*, February 24-26, 2009.

[6] Gilsinn, J., "Real-Time I/O Performance Metrics and Tests for Industrial Ethernet," *ISA Automation West*, April 28, 2004.

[7] Creative Commons – Copyright-Only Dedication (based on United States law) or Public Domain Certification, http://creativecommons.org/licenses/publicdomain/, Website correct as of April 18, 2009.

[8] The Open Source Definition (Annotated) – The Open Source Initiative, http://www.opensource.org/docs/definition.php, Website correct as of April 18, 2009.

[9] Sourceforge IENetP, http://sourceforge.net/projects/ienetp/.