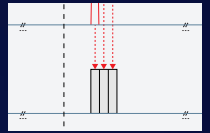


Data Synchronization with Microsecond Latency



How a UEI Synchronized System Works

Overview

It is straightforward to achieve sub-millisecond system latencies with data synchronization in the microsecond range over Ethernet using patented technology developed by UEI.

changed on that cycle and will be updated with the current data on the next cycle. The probability of a lost packet in a typical network is between one in 1E6 and one in 1E7. Most control systems can easily tolerate this as a packet would only be lost once in every 2–20 days (depending on the actual

packet loss statistics). However, in this application, we propose running at an aDMAP refresh rate of 400 Hz, and using data at 100 Hz would require 4 consecutive UDP packets to be lost in order to not have fresh data at each 10 mS update. This is something that would happen somewhere on the order of every 1014 years.

The following section provides a bit more detail on the process described above as well as some background on how VME systems handle this requirement.

Background: VME Synchronization

Control systems based on VME often use synchronization techniques by distributing interrupts. One VME system generates pulses on a DOut pin which are routed into the interrupt input of each system. Process control is implemented either directly in the interrupt

handler, or an interrupt routine sets up semaphores for processing threads to be scheduled. Reads and writes to I/O cards are performed in the interrupt routine or spun thread and tightly synchronized with the pulse train on the DOut pin. A/D and D/A converters on each board are also synchronized with this timing. Threads read input values, write output values and then trigger conversions, or the conversion clock is also tightly aligned to the pulses on DOut pin. The result is a control system operated on a tight real-time schedule, and with this system it is possible to align scans to 100 us with respect to each other or even better.

PowerDNA Synchronization over Ethernet

Each PowerDNA IOM (Cube or RACK) is independent and runs from its own system clock accurate to 10 ppm over the full temperature range (there is 10 ppm initial and 10 ppm over lifetime error as well). However, eventually systems drift apart.

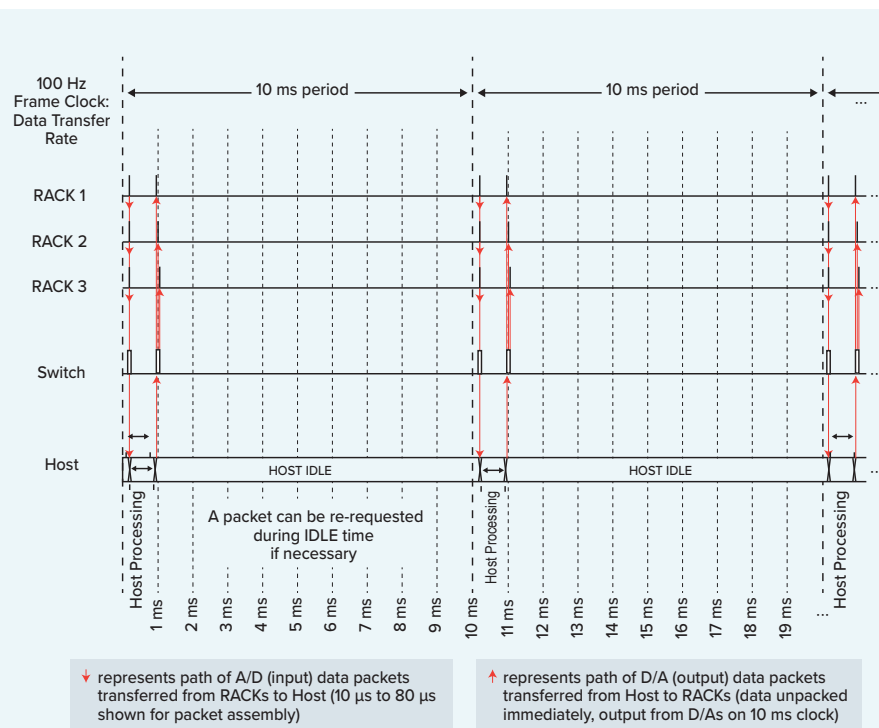


Figure 1 - Data Transfer on Multiple Chassis

UEI offers a data transfer protocol called aDMAP, which is similar in concept to reflective memory. In each aDMAP cycle, all inputs from the I/O Cube/RACK are written to memory in the host PC, and all outputs are updated with the appropriate data that has been written into the host memory. Note that this all happens in the background, and the UEI software takes care of all the memory setup and maintenance. Figure 1 is a graphical representation.

A clock signal sent to each Cube/RACKtangle initiates the periodic update. These updates are synchronized using the Cube/RACK sync port so that all updates in different Cubes/RACKtangles are initiated at the same time. The system takes advantage of the low latency of sending UDP packets, as TCP/IP has too much overhead to be useful in real-time systems.

The aDMAP software ensures that all data is refreshed on each cycle, and these cycles can be run at speeds up to 4 kHz. In all networks, there is a non-zero probability of a lost UDP packet. If a packet is lost, the data in memory is not

To synchronize system clocks on each IOM, UEI developed a 1PPS synchronization mechanism. At the heart of this mechanism is an adaptive digital phase lock loop (ADPLL) implemented on the FPGA of the IOM's CPU board. The ADPLL receives a 1PPS signal from one of the following sources:

- a. 1PPS from SyncIn connector from an external source
- b. 1PPS signal from IRIG-650 card which is synchronized either with GPS or IRIG time source
- c. The ADPLL receives a 1PPS signal and locks to this signal with the resolution of 15 ns to run an internal timekeeper flywheel in sync with the external source (10 ms on IRIG-650 layer). Since the main system clock is very stable, the ADPLL keeps correct timing over a significant time and temperature range even if the external 1PPS signal is lost. The ADPLL also provides built-in protection mechanisms to avoid accepting erroneous clock pulses and performs flawless adjustments to the 1PPS frequency.

The second part of the synchronization mechanism is implemented with an event module (EM). The event module provides the ability to generate an exact number of clocks between the 1PPS reference received from the ADPLL and supports frequency ranges from 1 Hz to 1 MHz. This mechanism places clocks at equidistant time intervals from each other and performs corrections spread across multiple clock cycles to minimize errors. The EM can generate a main clock and up to four derivative clocks produced on dividers. Note that derivative clocks are tightly aligned with the main EM clock.

How Clocks are Routed and Used

1PPS and generated clocks are routed on the SYNC bus, which are four pins on the backplane of the rack. These lines are routed to each of the I/O boards and CPU layers as well as to the SyncOut connector on the front panel and can be used as the conversion clock, timestamp reference or trigger for any I/O board. Each I/O board has the ability to subdivide these clocks into lower frequency clocks and lower resolution timestamps. A 1PPS clock generated on one rack can be routed out to all other racks in the synchronized system via the SyncOut connector and a separate UEI connector panel, the STP-SYNC-1G, (up to 1000').

The degree of synchronizing a 1PPS clock (and all other derived clocks) across all racks depends on the cable length (equal length sync cables alleviate this issue). With all cables equal, we observe clocks drift relative to each other within +/-100 ns in the long term.

Triggering all racks at the same time can be accomplished several ways:

- Provide an external trigger pulse to the SyncIn connector to start all I/O boards at the same time. This trigger can be distributed to other racks via STP-SYNC-1G (up to 1000').
- Arm a broadcast trigger and subscribe the host on the 1PPS event. Upon receiving this event, call an API

function to broadcast the start time to all armed racks in the synchronized system. Since the 1PPS clock is synchronized, as well as the conversion clock, all racks start acquiring data at the same time.

- Broadcast a software command to start or stop an I/O board. This is the least accurate way because the actual start time depends on the time of propagating broadcast packet via switches and network cables.

aDMap/aVMap Clocking

aDMap/aVMap mode is a special version of data mapping that doesn't require a host to send a request packet to racks to receive a slice of fresh data. Instead, the host process initializes all racks into data mapped (DMap) or variable-data-size mapped (VMap) mode, (i.e., selects what channels on which I/O cards are participating in this exchange and configures them). Then the host configures which clocks are to be used as conversion clocks and data delivery clocks. For example, a user can choose to start conversions on input and output cards at a 1000 Hz update rate and send data to the host at 100 Hz without waiting for a request from it upon the same clock.

Since all clocks across the synchronized system of racks are synchronized, the data will be delivered to the host almost simultaneously. At the host, the user application waits on incoming packets in one or multiple threads to process it. Since the packets are sent in real time, the host application is locked into real-time as well.

aDMap/aVMap Timing

An inherent delay exists between sending a packet and receiving it on the host. This delay comes from a few critical events associated with data transfers.

- Time to process an event and form the packet (around 80 us)
- Time to transmit the packet across Ethernet (depends on how many switches a packet must travel through; the delay on the local network could be from 18 us for a single switch)
- Amount of traffic on the network. Switches store and forward packets for each port. The delay depends on the number of aDMap/aVMap packets from each rack, number of racks in the synchronized set and number of NIC cards on the host; this delay contributes up to 18 us for each rack
- Time it takes for the host to process data. This can be widely different and depends on the operating system and data processing algorithms

For a sample system running hardened Linux and consisting of three synchronized racks with two DMaps on each rack using all 12 I/O cards, the processing time for software to write input data to output data is expected to be around 800 us.

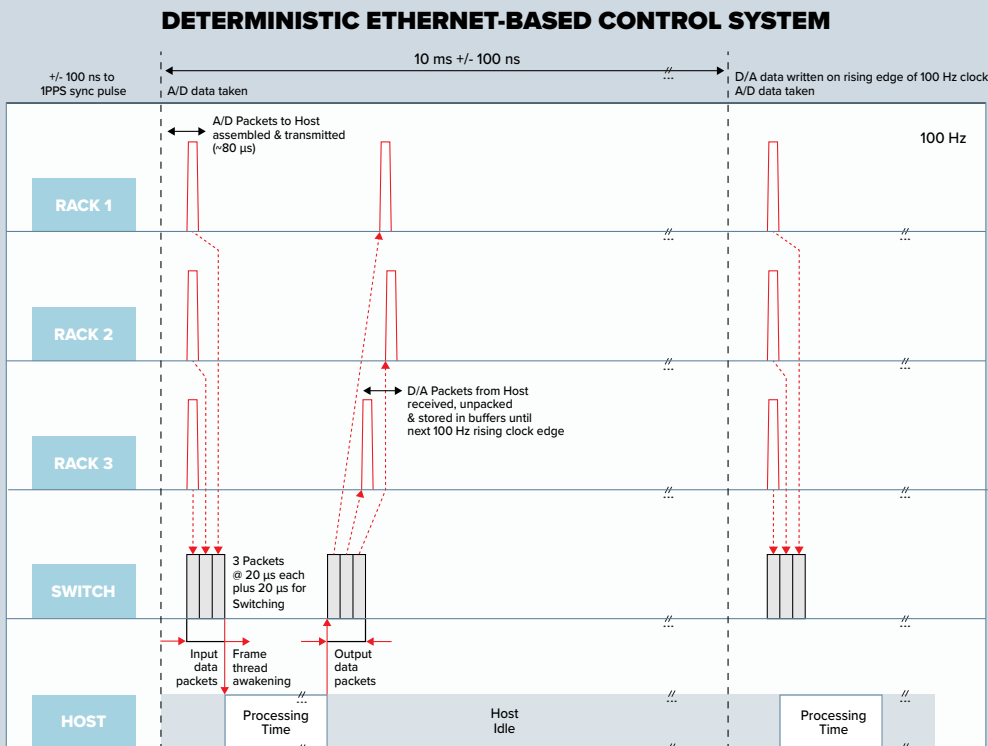


Figure 2: Timing Diagram

Output Update Timing

Outputs are updated the same way DMaps/VMaps are created for aDMap/aVMap input data but in the opposite direction. When the host application decides to update D/A outputs it calls a map refresh function. When the packet is sent to the rack, the update happens upon synchronized clock distributed over SYNC bus. Firmware takes 80-100 µs to write the data into the shadow memory of each card and then the conversion clock propagates this data to the actual outputs.

Packet Numbering and Lost Packet Recovery

Though not required in this application, as mentioned previously, the probability of a packet loss on a local network is 10^{-6} to 10^{-7} , (i.e., one packet in 10 million). aDMap/aVMap packets are numbered from 1 to 64k, and a packet with the counter of zero resets this sequence. The user application receives the packet number in the header of each data map received. In the case of a missing packet, each DMap stores ten previous packets in the rack memory which gives the host a sufficient window of opportunity to call the API to retrieve any missed packets.

aDMap/aVMap Proof of Concept

UEI implemented a multi-chassis system based on the system described in previous examples and captured actual timing data. Using the aDMap protocol to transfer data at a

100 Hz rate, we analyzed actual delay times relative to the 100 Hz clock edge.

Test Setup

The test setup of our implementation was configured with the following hardware:

- 3 UEI racks, each configured with an analog input board, an analog output board, and/or a digital I/O board
- Cisco SG300-20 1G switch
- 1 Acer PC, running Windows 7 and with UEI's PowerDAQ PCI digital I/O board (PDL-DIO-64TS) installed
- 1 Teledyne Lecroy oscilloscope

Test Code Description

Test code ran on the host PC and began with the configuration of the IOMs, which set up aDMap mode, configured I/O boards on the IOMs, configured 100 Hz clocking on the IOMs and configured the PowerDAQ PCI digital I/O board (PDL-DIO-64TS) on the host PC.

After the initial setup, the following loop-around code sequence (or a subset of it) was used for the various tests of the system:

- The IOMs acquire data and upon a 100 Hz clock edge, the CPU packs data samples from the analog input boards into Ethernet packets
- Packets from the IOMs are automatically transmitted to the host PC through a 1G switch
- Packets are received by the host PC, and after the last packet is received, the host PC issues a command to the installed PDL-DIO-64TS to toggle a digital output channel
- The host PC then assembles output packets to each of the IOMs
- Packets from the host PC with analog output data transfer to the IOMs through the 1G switch
- IOMs unpack packets
- Packets from the host PC with DIO toggle data transfer to the IOMs through the 1G switch
- IOMs unpack packets (DIO on IOM toggles to indicate full system loop is complete)

Figure 2 is a good representation of the system.

Measurement Method

The IOM clock and digital outputs on the IOM I/O board and host PC were connected to the oscilloscope. The above test code was run (in part or in full), and transitions were captured in persistence mode on the oscilloscope.

The 100 Hz acquisition clock was accessed from the backplane of the rack chassis. Digital outputs directly on the host PC and the rack IOM were used as test points for measuring the delays. A digital I/O pin on the PowerDAQ board installed on the host PC was used to measure when all packets were received by the host PC. The digital I/O pin on the IOM was used to represent round-trip delay data of when an output packet was received by the IOM.

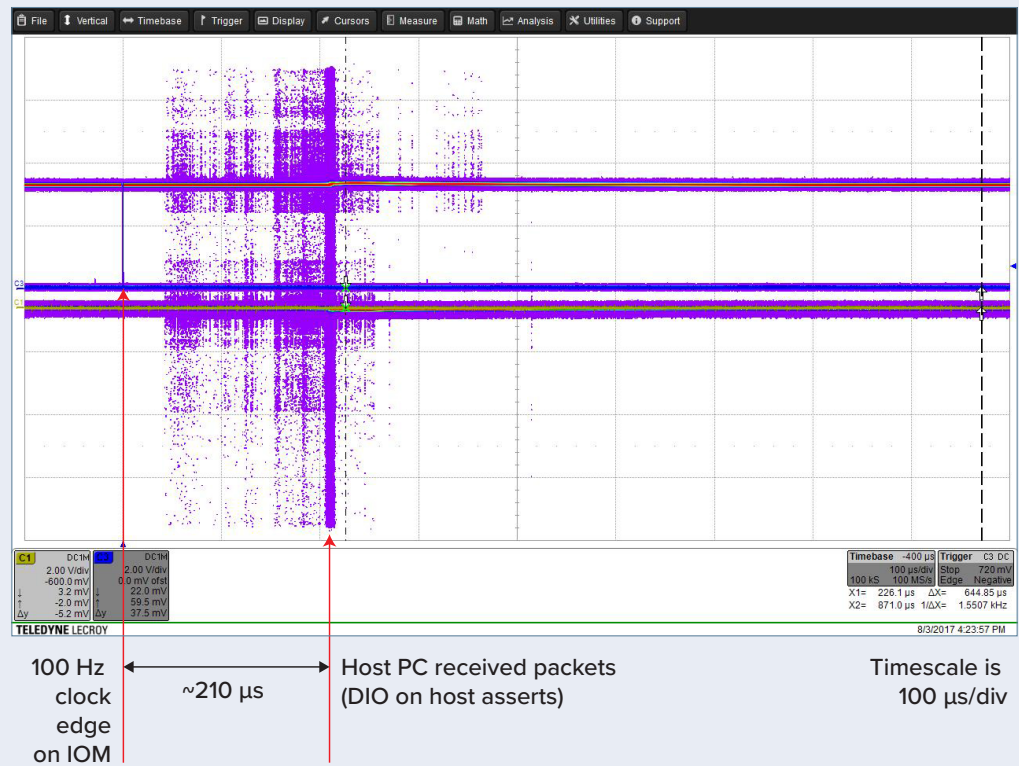


Figure 3: Propagation time from 100 Hz clock edge to packet received on host PC (no switch)

aDMap/aVMap Test Results

The test setup as described above resulted in the following:

- The delay from the rising edge of the clock to the packet being received by the host PC measured approximately 210 μ s on average (with a direct connection)
- The 1G switch added approximately 20 μ s delay per IOM
- The round trip delay measured approximately 790 μ s on average

Scope printouts are provided in the following subsections for the listed test cases:

- Delay from IOM clock edge to host receiving packet: direct connection vs switched connection
- Delay associated with adding IOMs
- Delay associated with round trip of system

Results showing delay from IOM clock edge to host receiving packet

The following scope images show the delay from the 100 Hz clock to the host PC receiving the packet. The scope is capturing multiple, successive measurements in persistence mode. The first scope image shows results of only 1 IOM connected directly to the host PC; the second shows 1 IOM connected to the host PC through a Cisco 1G switch.

1 IOM connected directly: The median delay to assemble and receive a packet with a direct connection is approximately 210 μ s, with the fastest propagation at approximately 50 μ s and the slowest showing 360 μ s (Figure 3).

1 IOM connected through a 1G switch: The median delay to assemble and receive a packet using a 1G switch is approximately 226 μ s.

Note that the “Host PC received packets” measurements represent the PC toggling a DIO pin on a Windows 7 machine. Jitter showing late responses (after median and beyond) on all graphs will show delays associated with Windows 7 OS. Windows 7 is not a real-time operating system, and incurred Windows-related delays, as the >226 μ s to 850 μ s outliers in Figure 4, will show in jitter numbers.

Results showing delay associated with switches

This section shows scope images of delays from the 100 Hz clock to the host PC receiving the packet when additional IOMs are added to the Cisco 1G switch. Each IOM adds approximately 20 μ s delay.

- Figure 4 shows measurements with 1 IOM (~226 μ s)

- Figure 5 (next page) shows results of 2 IOM connected (~253 μ s)
- Figure 6 (next page) shows results of 3 IOM connected (~261 μ s)

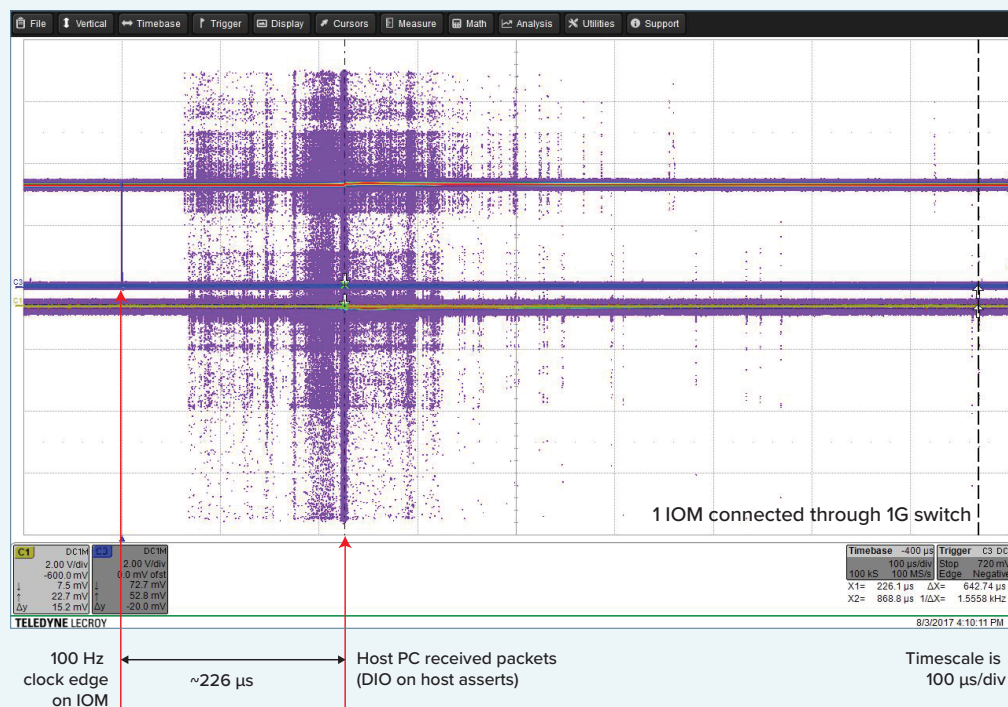


Figure 4: Propagation time from 100 Hz clock edge to packet received on host PC (through 1G switch)

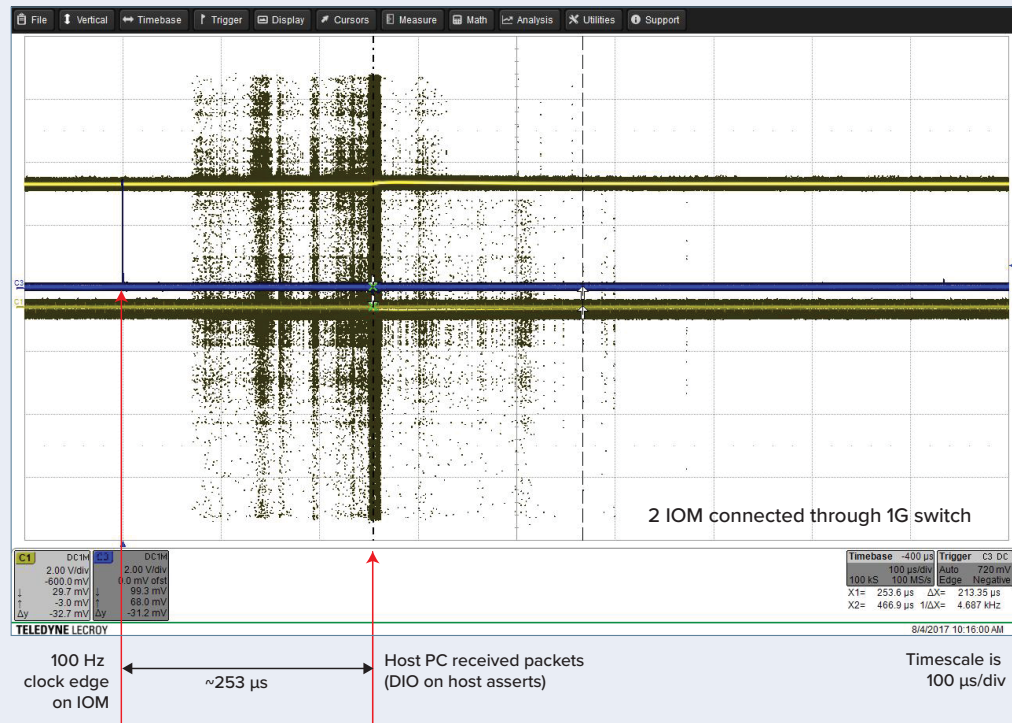


Figure 5: Propagation time from 100 Hz clock edge to packet received on host PC (2 IOM through 1G switch)

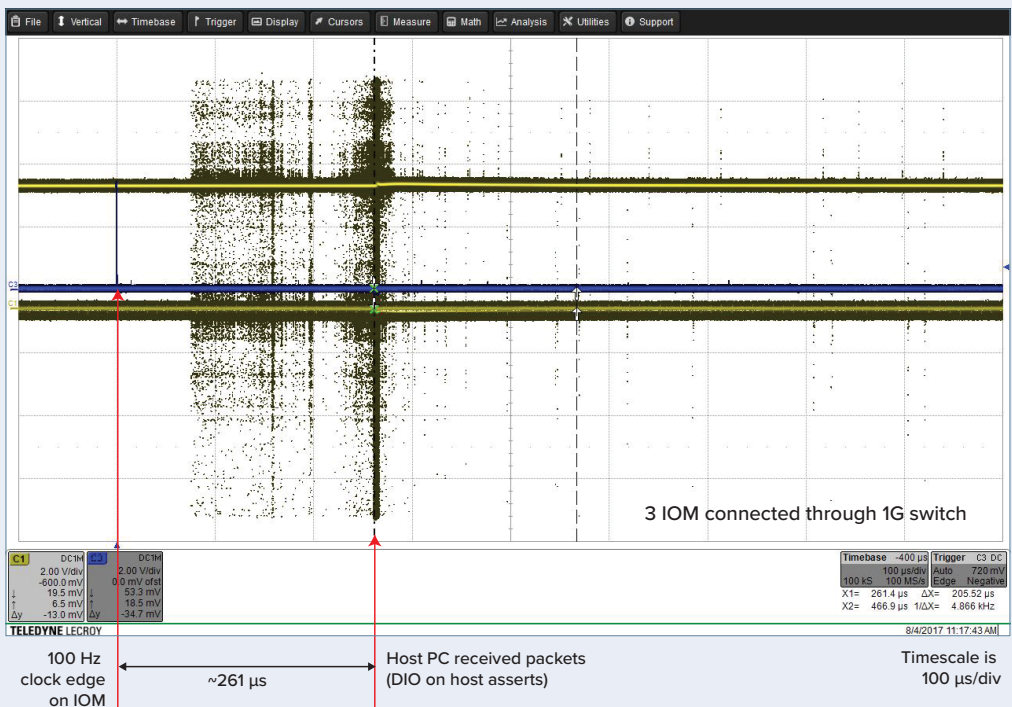


Figure 6: Propagation time from 100 Hz clock edge to packet received on host PC (3 IOM through 1G switch)

Results showing roundtrip delay of 3 IOM system

The following section shows an average roundtrip delay at 790 μ s, clock edge to system is executing the full test path (Figure 7):

- The IOMs are acquiring data and upon 100 Hz clock edge, the CPU packs input data samples from the analog input boards into Ethernet packets
- Packets from the IOMs are automatically transmitted to the host PC through a 1G switch
- Packets are received by the host PC, and after the last packet is received, the host PC issues a command to its locally installed PDL-DIO-64TS to toggle a digital output channel
- The host PC then assembles output packets to each of the IOMs
- Packets from the host PC with analog output data transfer to the IOMs through the 1G switch
- IOMs unpack packets
- Packets from the host PC with DIO toggle data transfer to the IOMs through the 1G switch
- IOMs unpack packets (DIO on IOM toggles to indicate full system loop is complete)

Note that this test case is using a host PC running Windows 7 machine, as is all the test cases in this document. Windows 7 is not a real-time operating system and will show delays associated with Windows 7 OS.

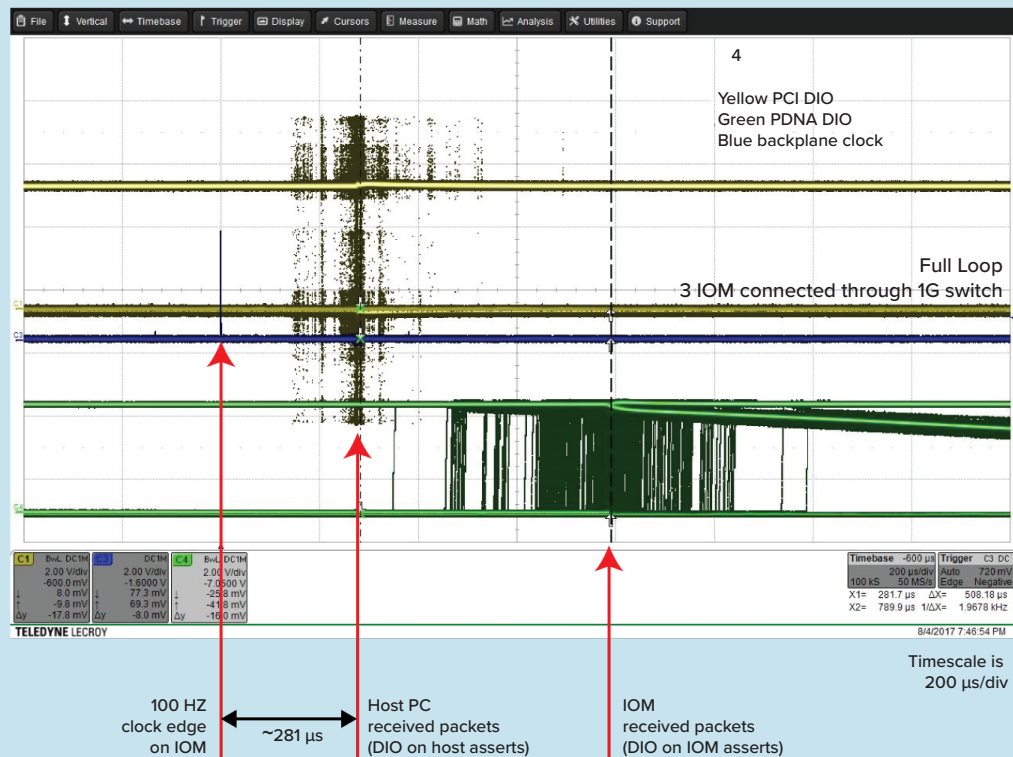


Figure 7: Full loop around time from 100 Hz clock edge to DIO packet received by IOM from host PC (3 IOM through 1G switch)



249 Vanderbilt Avenue, Norwood, MA 02062 USA • Tel: 508-921-4600 Fax: 508-668-2350 • uei.sales@ametek.com
 AMETEK GmbH, Rudolf-Diesel-Straße 16, 40670 Meerbusch, DE • Tel: +49 40 63698136 • uei.salesema@ametek.com

