Migrating from Serial to Ethernet- The Impact of Latency

White Paper

www.digi.com

DIGI

Overview

When considering the migration of a directly linked device to the world of networked devices, it is important to understand the context of the connection. In other words, it is not enough to simply plug a device into a network, decoupling it from its original host, without understanding the implications of such a change. This paper will discuss the barriers one might encounter when decoupling a closely coupled device across a network, and the specific role that latency plays in this process.

The Network Experience

We have all been in a social situation where we relied on many verbal and non-verbal mechanisms to create a cohesive conversation. While we still may have difficulties inferring a persons' body language and mannerisms, we can always detect general communication mechanics, such as when someone has completed a sentence, or when to end a conversation that has gone too far. Most often then not, interruptions cause a significant pause or cessation of a conversation...

Now, take the conversation scenario to another level where the people involved are in two different locations. Assume the telephone connection has at least one part over a satellite and that each party is using a speakerphone. With the advances of voice quality in telecommunications, the participants can probably speak at the same rate as they did in the face-to-face conversation. The main missing details are the non-verbal clues as to whether a message is being understood and the near instantaneous feedback resulting from a predictable delay in the voice going directly from mouth to ear. As many have experienced, these factors in non face-to-face communication combine to produce additional, unnecessary voice information and can result in breaks in the audio stream – resulting in requests to "retransmit" or to speak more clearly. Many times participants experience lapses in conversation because the speakerphone has picked up other noise in the room. This causes for breaks in conversation and can lead to additional time delay (latency), which can be very frustrating to both parties. Hence, by introducing the additional mechanisms of the speakerphone and satellite telephone connection we have added requirements for new error checking and feedback mechanisms to ensure accurate and efficient communication.

Of course, with today's level of technology knowledge, we recognize the need for adding more error checking and flow control mechanisms to ensure that our valuable conversation data is not lost. Some techniques can be imposed to create more clear conversations, such as improve the detection mechanisms in speaker phones, replace high latency satellite links with fiber optic cable and add video so that one day we can re-establish the instantaneous non-verbal feedback. Unfortunately, many of the devices that we move from their closely coupled serial connections are not as adaptable, so we need to find alternative ways to compensate for this changing environment.

Placing Devices on a Network

Now that we are all familiar with the issues associated with conducting telephone conversations via speakerphone, we can apply some of the same principles to the process of placing the formerly closely coupled serial device on a network.

Figure 1 shows the traditional world of a serial device connected to a host computer using an EIA-232 connection. For our example, the device happens to be a sensor in a power plant, which relays appropriate power drain characteristics. The objective is to both monitor power usage and trigger alarms based on certain criteria. As such, the host application may query the device for status or right configuration. In addition, an unsolicited critical alarm may also come from the device. Since both the device and host assume a local connection, the applications are set up to detect whether or not the device is connected, but generally do not provide flow control. Instead, the protocol between the device and the host assumes a rapid query and response, so as to trigger a series of retries, culminating in an alarm.

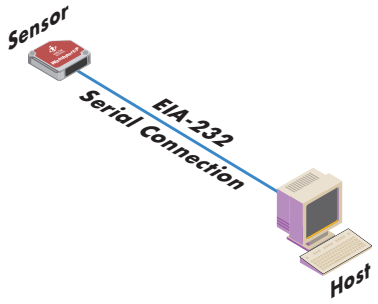

Figure 1

Now, in an effort to be more efficient by aggregating and centralizing computing resources, the host is remotely located with other machines of its kind, and then connected to the world through its Ethernet connection. The device is connected to the serial port on a device server, which then connects to an Ethernet network. See Figure 2.

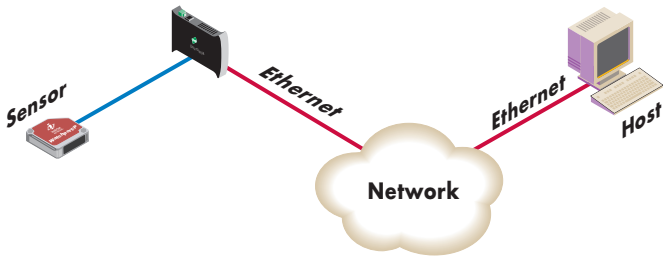

Figure 2

Like before, the same messages are communicated back and forth with the device. In this case, the serial communication is with the serial port on the device server instead of the host computer. However, we have now added three additional components to the system: the device server, the Ethernet network, and the network software and driver on the host. Since the system has now changed, it is important to understand the impact these additions may have.

What is Latency

The word "latency" by definition refers to being late, or in technology terms, how much delay is inserted into a system. Additional delay, if not expected or compensated for, can result in unwanted and unexpected results. Remember our telephone conversation example. The additional delay or latency introduced extra data into the system and required the people involved in the conversation to compensate in order to effectively communicate. When our two people were having a conversation in a room, there was an expected time for one voice to go from one person's mouth to the other person's ear. Adding the additional delay of the telephone link made the speakers voice late in arriving at the listener's ear. It also makes any kind of acknowledgement from the listener back to the speaker impeded.

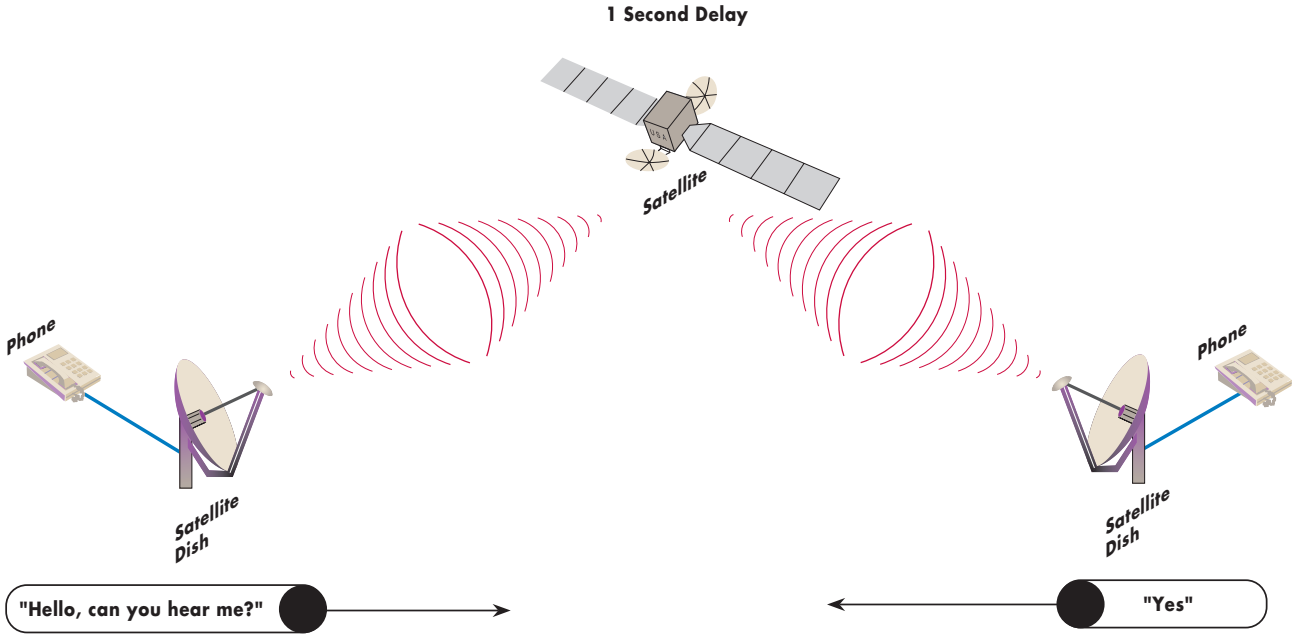

Figure 3
Imagine the speaker repeatedly saying, "Hello, can you hear me?" followed by a pause. Assume it takes about four seconds to utter the phrase and the speaker pauses about 1 second before repeating it. The person will stop saying the phrase when an appropriate acknowledgement is heard. This works fine in a room because if you haven't heard a response in that 1 second pause, unless the person probably wasn't paying attention. Feedback is immediate and predictable. We would expect the phrase to uttered only once because of the near immediate response.

Now, let's do the same test over our satellite telephone call which introduces a delay of one second. The process goes as follows, first the speaker utters the same phrase, and then the listener hears the phrase one second later and responds. Because the response also incurs a one second delay, the speaker does not hear it before he starts speaking the phrase again. We define the latency of this system as two seconds. If the speaker is able to hear the response and stop speaking, then he will perhaps strand part of the phrase (e.g. "Hello, ca…") on the network.

The term strand is used because this is really garbage data. We refer to the speed in which the system reacts and returns to normalcy as slew. The system will work, but we need to be conscious of the speed in which the speaker can stop transmitting (slew) the second time as well as the amount of garbage information that must be thrown away. This is easy for voice communication because most humans have advanced communications skills, which allow them to adapt. It is much harder when two machines are attempting to send strings of digital data (string of 1's and 0's) back and forth.

Finally, to further complicate things, remember that in our example we had speakerphones, so while the person was talking, they could not be listening to the person on the other end. In this case, the telephone call system completely breaks down because the speaker has already begun speaking again, and never hears the response.

Latency on a Network

Now we apply the latency principles to the sensor – host system discussed earlier. In the system where the device is directly connected to the host via a serial port, the time for information to cross the serial interface is known and designed into the system. Hence, the latency of the network system is the additional delay added by crossing the device server, crossing the network and crossing the host's network interface in both directions. See Figure 4.

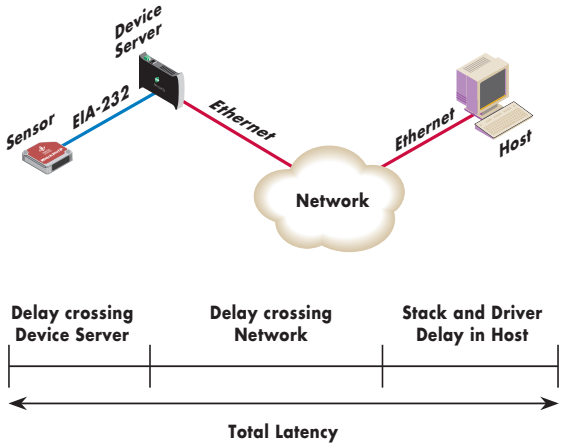

Figure 4

Latency will vary depending upon many characteristics, including the size of the messages to and from the device, the amount of traffic going in and out of the device server, the traffic volume and speed of the network, and the efficiency of the network stack and driver on the host system.

Message Size –
The application on the host can be an influencer, because it looks for information across the serial interface, and is usually closely coupled to the serial port. Since the device server does not have specific knowledge of the application, it needs to be intelligent in how it looks at the incoming data and how it efficiently forwards it to the network. For example, a system designed only to look for 10 character messages will not work well when there are single-character messages.

Device Server Traffic –
The device server needs to forward packaged serial data out of the network as effectively as possible, because other data that uses processing and memory resources within the device server can impact the speed at which the data is forwarded. For example, a device server serving high-speed data on multiple ports can have a higher delay versus one only handling data on one port.

Network Data –
The time it takes to traverse a 100 Mbit/sec Ethernet segment that has no other traffic is very deterministic and very minimal (10 nanoseconds). However, this time is closely related to the amount of other traffic on the network and the number of hops required across a network. For example, if the network is the Internet, the number of hops, routing table look-ups and traffic can push the network portion of the latency into the realm of milliseconds or even seconds. It is very important to characterize this latency through testing.

Network Stack and Driver –
The time it takes to enter an Ethernet interface on a host and the associated stack and drivers to relay the data through the operating system to the application. While it can be related to how busy the host processor is, the delay here tends to be directly related to the tuning of the network stack and driver; specifically, how they poll for data and the efficiency by which this data is forwarded to the application.

Digi Solution

Digi characterizes the latency of a network system by measuring the time it takes to: send one character into a serial port on a device server, across an empty Ethernet network, and into a host Windows host system to an application and back again. This offers a benchmark for comparing different device server products, and host drivers. The system is then tuned to ensure that the latency across a device server varies minimally with message size and other traffic running through it. This is also true on the RealPort host driver system. However, due to the variance in different operating system architectures and IP protocols stacks, results can vary. For example, the Windows 2000/XP systems tend to operate more efficiently than the comparable Linux based systems.

The following table represents the benchmark values for Digi compared to some competitors as performed by E-Testing Labs.

Device Server	Average Latency (ms)
Digi One Family	5.81
Moxa NPort Express DE-311	10.49
Comtrol DeviceMaster DMSH-1	12.18
Comtrol RocketPort Serial Hub IA	29.31
Lantronix UDS100	586.29
Lantronix CoBox-DR1	707.35
Lantronix MSS-VIA	762.73
Lantronix UDS-10	860.80

In order to determine the precise performance profile for a specific deployment, special care must be taken to measure the performance of the network.

PH: 877-912-3444
www.digi.com

Digi International
9350 Excelsior Blvd.
Suite 700
Hopkins, MN 55343

Digi International - Germany
+49-89-540-428-0

Digi International - Japan
+81-3-5428-0261

Digi International - Singapore
+65-6213-5380

Digi International - China
+86-21-5049-2199

91001240
B3/1118