Migrating from Serial to Ethernet-Performance Impacts

White Paper

www.digi.com

Digi®

Connectware™

Overview

As the world continues its migration from an environment of directly linked devices to networked devices, it is important to consider how this will impact the underlying communication system. 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. Specifically, there are a number of different performance considerations, which must be understood in order to guarantee an error free system. In a previous paper we discussed one aspect of this performance, and focused our analysis in understanding, analyzing and overcoming the latency barriers that are encountered when establishing a network.

This paper will expand the discussion of performance characteristics beyond latency, and look closely at a number of additional factors. In so doing, it will define dependencies and discuss how best to understand a system in order to maximize performance for the given application. The specific aspects of the system will include, slew, jitter, payload size, throughput and latency.

Dependencies of a Network System

In our previous paper, we introduced the concept of a device directly connected to a host computer system via serial cable. Figure 1 demonstrates the traditional world of a serial device connected to a host computer using an EIA-232 connection. For our example, the device is a sensor in a power plant, which relays appropriate power drain characteristics back to a central monitoring host system. The objective is to both monitor power usage, and trigger alarms when certain criteria are not met. As such, the host application may query the device for status, or programmed configuration. On occasion, 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.

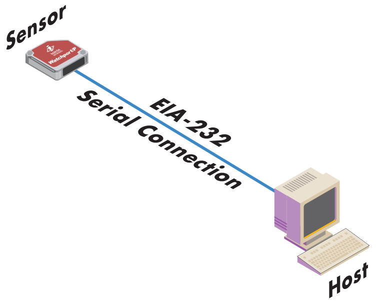

Figure 1

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

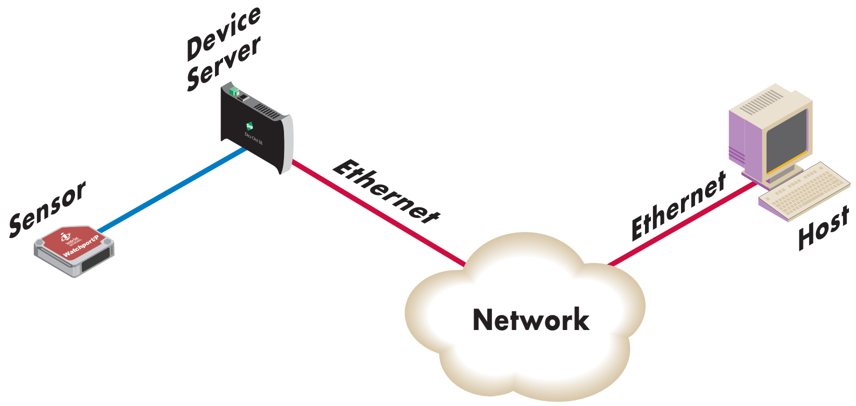

Figure 2

With the introduction of the device server and network environment into the system, we have changed the communication characteristics in a number of ways. The characteristic that has the most immediate impact is the extra time it takes for a message to travel between the device and the host. The extra delay a message takes to transmit is known as "latency." However, while latency may be the most obvious and easily understood of the changes, the network environment, coupled with its latency can impact other elements of the communication. This becomes even more important in closely coupled and/or more complex systems.

In order to better demonstrate and explain the changes in communication between the device and host, we will look at four additional characteristics. For now, we will look at their meanings and then elaborate later. They are:

• Payload size –
A term used to describe message size, or the amount of data enclosed within a transaction. Messages can vary in size ranging from very short, only a few bytes, while others can be very large. You may want to think of this as the difference between a single passenger car and a bus.

• Throughput –
A term used to describe the amount of data sent or transacted during a defined period of time. This is directly related to speed and payload size. Using our car and bus example, we might define throughput as the total number of people that go from point A to point B in an hour. Even though the bus might have a speed slower than the car, its larger payload still may offer better throughput.

• Slew –
A term used to describe the impact on a system due to some event. We expand this to include slew rate, which is the time it takes for the system to react to the event. Slew comes into play because there is not an instantaneous reaction, and often times some data may be thrown away or lost as a result. Using the bus example, think of slew as the impact that a traffic light might have on our car or bus – some traffic always runs through the yellow light. Eventually we return to steady state.

• Jitter –
This is variance in flow between the input and output. Data goes into a system in a certain sequence and certain spacing in time. When the data emerges, the timing and spacing may be different. Think of a line of cars merging onto a freeway system, and assume they will all get off at the same exit. When they are on the on-ramp, they are all spaced at even intervals, traveling at a similar speed. When they exit, they're sequence and spacing may be different.

Now, returning to our directly connected system in Figure 1, since there are no intermediate steps in between the host and device, there is no additional latency. Furthermore, the throughput is purely determined by the speed of the link (baud rate). Payload size is irrelevant because there is only one, simple defined path. Because we have a fixed link, the input must equal the output, so the system has no jitter. Finally, slew is easily quantified to very precise levels. With no other variables, the time and impact of an event is directly related to the speed of the data transmission. For example, if the device tells the host "STOP," the slew is always the amount of data stranded on the link during the time it takes to say "STOP." Adding a network will change all of these elements.

The Freeway System

In order to look at the overall performance impact of inserting a network into a closely coupled system, we created a model of a network using an example which hits closer to home – namely a freeway system and the feeder roads. Instead of the sensor connecting to a host computer, imagine that you and your family are going to travel from your home to your office in a metropolitan setting. The system where the device is directly connected to the host is a kin to you and your family having a dedicated, narrow, two-lane private road, which connects directly from your house to your business office. See Figure 3.

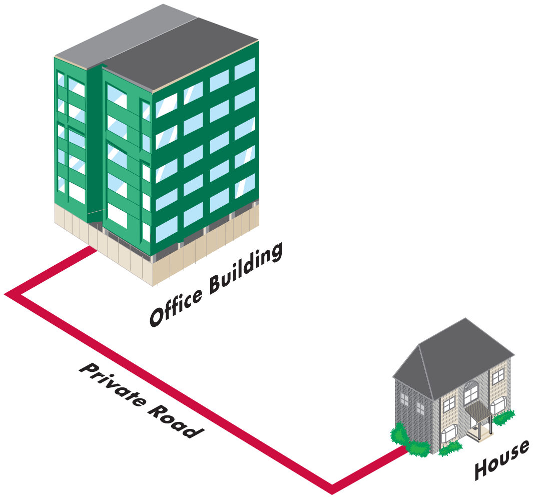

Figure 3.

Since there is no other traffic and it is your road, you can set the speed. Unfortunately, because it is a narrow road, without banked turns and shoulders, you probably aren't going to go too fast. Nonetheless, because it is your lightly traveled road, it doesn't really matter whether you all go in the family mini-van or you each take separate cars, since there is no passing, you will all get there within the same interval that you left. Finally, because it is your road to control and there aren't any other cars other than those going from your house to your office, you get to control all the events that would impact your flow. The result is completely quantifiable. There is no additional latency, throughput is completely dependent on speed, jitter is non-existent and slew, if present, is completely deterministic. However, like the serial cable, we all know that it is not practical to have many long, private roads. Of course, you many have more options depending on the proximity of your office – the shorter the distance, the more practical the private road (i.e. a driveway).

Now, consider a more typical environment where your business office is a fair distance away, requiring you to take a low traveled feeder street onto a shared freeway. Fortunately, your business office is close to the freeway exit. See Figure 4. The system where the device is connected through a device server to a networked host, is similar to the freeway system example. To complete the picture, there is a mass transit bus station close to the freeway entrance and across the street from your business office. Also, traffic signals provide arbitration functions at the entrance and exits.

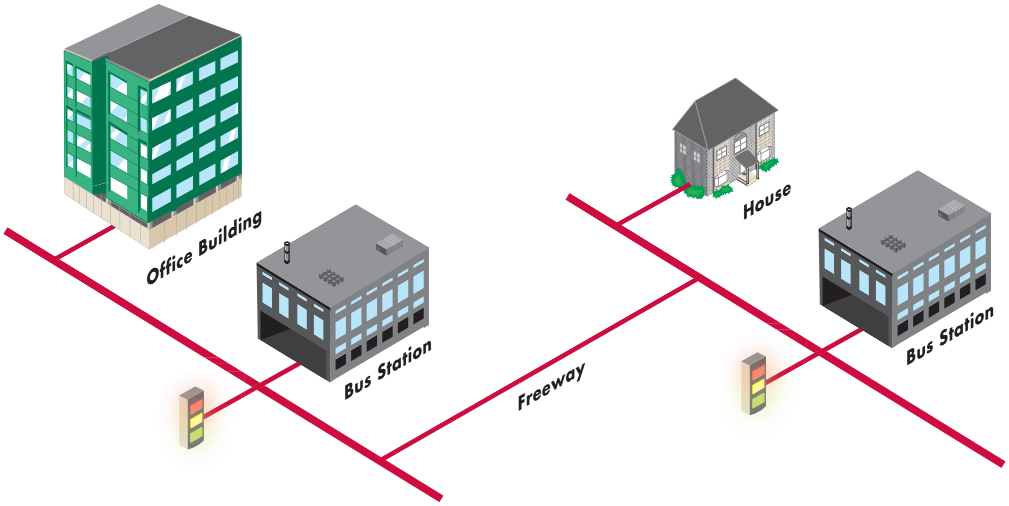

Figure 4.

Now, in order to best understand the impacts and dependencies on the performance, we look a number of different commuting scenarios, comparing them, of course, to the aforementioned private road option.

Scenario 1 – Avoiding Rush Hour.

You and other family members drive in separate cars, taking the freeway in both directions. Because you don't like heavy traffic, you all leave for your business office at 4:00 am and leave for home at 8:00 pm. As a result, this scenario offers very little delay (or latency). You leave the house, pause for a short traffic light, quickly merge onto the freeway, travel at high speed, exit the freeway, pause for another short traffic light and arrive at your business office.

Analysis: Because the freeway and surrounding streets are lightly loaded, the system throughput is mainly dependent upon the speed of the individual cars. In addition, the latency of the system is a combination of the two traffic lights, and the time on the freeway. Again, because the traffic system is lightly loaded, the slew is predictable and quantifiable. The expectation with low traffic volume is that you and all your family members, even if you are in different cars would probably stay pretty close together – as a result, everyone would make it through the traffic lights at the same interval. Finally, due to the lightly loaded system, jitter is mostly likely small, however it is definitely not zero. If each car left home 30 seconds apart, they would most likely encounter a mini traffic jam and spread out again at each traffic light. This would lead to a definite variance in their arrival times at the business office.

Scenario 2 – Rush Hour Congestion.

Similar to Scenario 1, you and other family members drive in separate cars, however this time you are driving during rush hour. Due to the traffic, your individual cars get interspersed with many other cars leaving a couple to sit through two lights before getting on the freeway. Being the aggressive, impatient driver that you are, you ignore the pending yellow traffic light and speed on through. Your family is not so lucky. Once on the freeway, traffic still moves at a good pace, but the volume forces you to make a number of lane changes to keep moving. All is going well until an accident on the other side of the freeway has caused a gaper's delay, forcing all the cars on your side to slow down. After bypassing the delay, all your cars finally exit, and move through the final traffic light to the business office.

Analysis: In this scenario, an increase in traffic volume has a profound impact on latency, slew, throughput and jitter. Not only did latency increase for all the cars, but the variance in the latency did as well – note that some cars got stuck in the additional traffic light, and accident slow down. Had the accident happened in Scenario 1, it most likely would have had very little impact. Since the payload size didn't change, throughput was slowed only by the reduction in speed on each leg. Finally, slew and jitter are also related to the increase in traffic. Like Scenario 1, the cars left home 30 seconds apart. Now, because the additional traffic light and gaper's delay, the arrival rate jitter is significant. Note also that slew and jitter also are directly impacted by a small payload. For example, if everyone would have ridden in the same car, the latency may still have been greater, but slew, jitter and variance in latency would have been very small.

Scenario 3 – Mass Transit.

In this scenario, you and your family members drive to the bus station, board a bus, arrive at the remote bus station, get off the bus and walk to your business office. Of course, the nice element about the bus is that it gets to travel in the HOV lane. However, on this same day, traffic is heavy due to an accident, which causes the high volume groups of vehicles to slow down, This will have little impact on the bus traveling in an HOV lane. Unfortunately, we all know that there are a couple of drawbacks to mass transit. The busses leave on a set schedule, and there is the added time to load and unload the bus – unacceptable conditions to many of us.

Analysis: In this scenario, the benefits and shortfalls of dealing with a larger payload are engaged. Because the bus moves a large volume of people, it reduces the vehicle overhead on the road, and is rewarded with travel in the HOV lane. As such, the component of latency on the freeway will be smaller than in Scenario 2. Unfortunately, this benefit may be partially offset by the waiting, loading and unloading time at the bus station. The bus operators are challenged to provide efficient bus scheduling such that people don't have to wait too long, and the busses maximize their load capabilities. However, full busses still improve overall throughput because the payload size of a bus can handle the amount of people that would fill 50 single occupancy cars. Slew is also easier to quantify because it happens in units of entire busloads instead of individual cars. For this, think about the slew associated with driving through yellow light. If the bus goes through, it takes the whole load. Maybe an additional car will slip through as well, but not many – the slew is large, but deterministic. Compare this to individual single cars. Of the different measures, jitter is the most difficult to predict because it will be directly related to the unloading of the bus at the end destination, and less related to the traffic encountered by the bus.

Devices on a Network

Now that we understand the differences and interdependencies of latency, payload size, throughput, slew, jitter and speed, we can begin to address why it is important and how to best characterize a network system. When we understand the most important aspects of communication between the host and the device, we can engineer the system to ensure that the correct thresholds are met.

In the previous section we presented an analysis of different traffic scenarios for a family commuting from home to work. It is now time to map the metaphor to what we know about networks. When we return to the systems of Figure 1 and Figure 2, the serial ports represent the equivalent of the feeder streets and the network represents the equivalent of the freeway. The device server and host drivers are represented by the bus station, traffic lights and on-ramps. In order to illustrate how to best engineer appropriate network connectivity we need to understand the characteristics of the communication when directly connected, and map the important characteristics to the network view.

Assume for example, that the host in Figure 1 communicates with the sensor device using a large volume of very short messages. If the system doesn't get a response it resets the guard timer, and launches a duplicate message. Since each message is a separate part of individual query-response transaction, it is not important if their arrival rate varies and we occasionally lose a message. In order to insert a network system, it will be important that the individual messages get transferred as quickly as possible. This means that both the device server and host driver must be sure to ship off data as soon as it is available as its number one priority – low latency and throughput are the most important, slew and jitter are irrelevant. In this case, each short message must be placed in its own "car."

Now assume that instead the sensor sends a sequence of readings every hour. Since all the readings are in order, it is important that the order is preserved, and that individual readings are not lost. On the host application, it records these individual readings by time interval. In this case, low latency is not important. Instead, slew must be at the level of the complete bundle of readings, and jitter between the readings must be kept at an absolute minimum. In this case, the best approach is for the device server to wait until all readings are received before forwarding them to the host. Any disruption or loss of data would then eliminate the entire bundle. Jitter within the bundle is kept to an absolute minimum. In this case, the individual readers are placed on a "bus" with the other readings in its bundle.

Digi Solution

The Digi solution for device networking does not assume that all network pipes need to behave the same. Rather we provide a complete characterization of the throughput, slew, jitter, and latency parameters for a range of message sizes for different network conditions. The result is a tunable system, which can match most applications for networking. In doing so, we provide industry leading performance and performance characterizations – whether it is minimum latency, minimum jitter or maximum throughput. Table 1 provides a profile of the performance ratings for different Digi Device Server products.

Product Family	Latency (ms)		Throughput (% of max) RealPort driver ver. 2.6.82.0				Slew at 9600bps H/W Flow		S/W Flow		Jitter (ms) at 100ms		
	Avg	SD	9.6 Kbps	57.6 Kbps	115 Kbps	230 Kbps	Avg	SD	Avg	SD	1 B	10 B	32 B
Digi One IA RealPort	5.51	1.18	98.1%	98.1%	98.1%	98.6%	1.76	0.61	1.77	0.53	1.4	1.9	2.3
Digi One TS	5.74	0.94	98.1%	98.1%	98.1%	98.7%	1.74	0.60	1.82	0.60	1.4	7.3	7.4
Digi One SP	5.88	0.60	98.1%	98.1%	98.1%	98.6%	1.62	0.64	1.58	0.64	1.2	2.0	2.0
PortServer TS 2	5.57	3.13	98.1%	98.1%	98.1%	98.6%	1.78	0.65	1.76	0.60	1.5	2.2	2.3
PortServer TS 4	6.04	0.81	98.1%	98.0%	97.9%	87.1%	1.74	0.63	1.78	0.55	1.4	2.0	2.7

Table 1. Performance Ratings

The first column defines the specific product being characterized. This is followed by columns defining the products performance rating, described as follows:

• Latency – The network system latency is measured as the time it takes a message to move round trip through the device server, network and host system. Even though different applications use different message sizes, a latency rating is calculated for messages of 1 byte because the important variances will show up when looking at slew and jitter. For this rating, serial ports are configured at 9600 baud. Digi is well known to have the lowest latency for a TCP/IP system across all the range of message sizes.

• Throughput – This is measured as the actual data throughput compared to the theoretical maximum. For example, when a serial port is configured for 57.6 Kbps, the maximum data throughput through the network system would be 57.6 Kbps. Due to the constraints we discussed in this paper, the actual throughput is a percentage of this maximum. Most manufacturers quote the baud rates, but do not provide a measure of actual throughput. This is done because throughput usually drops off dramatically at higher baud rates. We measure throughput by sending a continuous stream of serial data through a round trip system and measuring the actual data rate received at the end of the round trip. The rating is a percentage of actual versus the initial data rate.

• Slew – This is measured by assessing the extra character drip which occurs after a both a hardware and software flow control event at 9600 baud. A continuous data stream is inserted into a round trip system. After flow control is inserted, the drip of additional characters is counted. As we discussed earlier, flow control on a standard serial link has an almost immediate and quantifiable effect. It is important to have this same effect measured and trended when a network system is inserted so that correct buffering, data management and retransmissions tools may be used to prevent unnecessary data loss.

• Jitter – This is defined as the percent variance to one standard deviation from the input data interval spacing and indicates the degree of variability of the system. For this rating, we insert characters into a roundtrip network system, spaced at 10 100 ms. Jitter is then measured on the output input by looking at how the output input character spacing varies over time from 10 100 ms.

A more detailed discussion of measurement methodology is included in an Appendix to this paper. Using the above, customers can evaluate their individual system and create an integrated, highly functional network system, based on the important performance aspects of their individual system.

Appendix

Test Methodologies

Latency

Latency tests are performed by measuring the time it takes for serial data packet to travel round-trip through the system under test (SUT). As illustrated in Figure 5, the device under test (DUT), in this case Device Server, is an integral portion of the SUT, but not the only piece.

The Latency test uses two PC systems, a Master System which is used for generating and measuring the data and a Slave System which is used simply as a loop-back. A serial data packet is generated from the Master System and sent out the native serial port (see figure 5). Once the packet has been sent, the round-trip timer is started. The packet is forwarded through the Device Server over the network to the Slave System. A program running on the Slave System “listens” for data on a virtual com port, using Digi’s RealPort software. Once the slave application receives the data it immediately echoes the data back to the virtual com port, thus sending the data back through the network to the Device Server which then forwards the data back to the native serial port on the Master System. Once the data is received by the test generator application, the timer is stopped. This test is repeated 10,000 times from which an average and standard deviation measurement are taken.

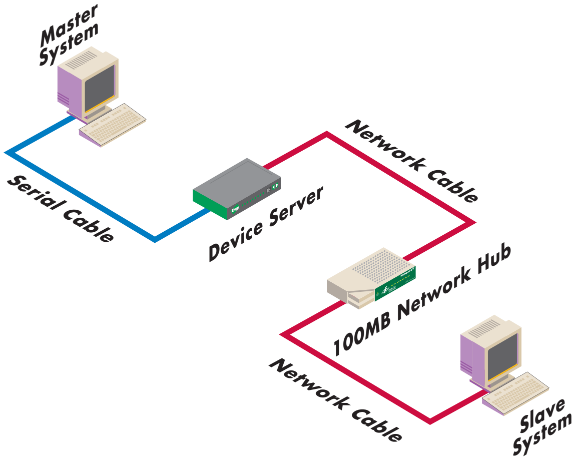

Figure 5.

Throughput

The throughput test measures the ability of the device under test (DUT) to transfer data bi-directionally through the its serial and networking subsystems. Data is generated and measured from a test system which is running Digi's RealPort virtual com port service.

A test application running on the test system sends data at a specified speed over the network to the DUT. The DUT then forwards the data to its serial port. There is a loop-back plug on the DUT's serial port which physically loops the data back to the serial port where the DUT forwards the data back through the network application. Tests are performed at different data rates using different data packet sizes. The test results that are reported in this document are 1024 byte packets sent at the data rates reported. The throughput results reported are an average of 1000 individual tests

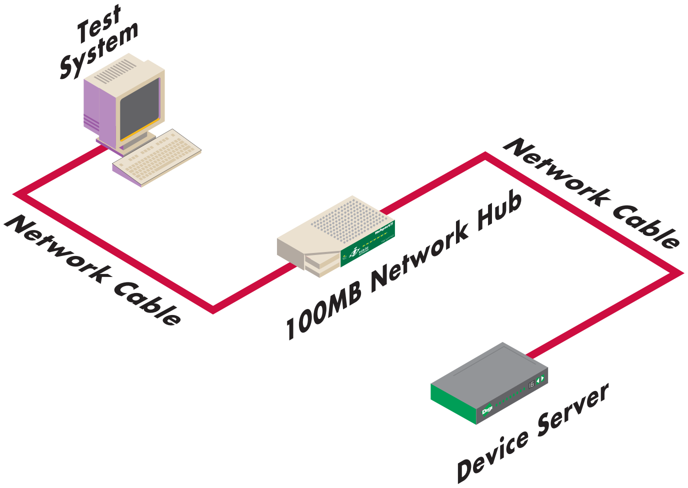

Figure 6.

Slew

Slew is measured by counting the number of characters that "drip" out of the system under test (SUT) once flow control is asserted. The test beds are very similar for software and hardware flow control (see figures 7 and 8). The main difference is that, with software flow control, the receiver PC issues the appropriate flow control character for start (0x13) and stop (0x11) and the hardware flow control test bed uses a break-out box to raise and lower the appropriate hardware flow control signal (in this case, CTS).

Testing is performed by sending a large text file from the generator system (see figures 7 and 8) then asserting flow control by either lowering CTS or sending the 0x11 xoff character for hardware or software flow control respectively. You will be able to see when the signal state changes or when the xoff character was sent on the serial analyzer. You simply count the number of character that appear from the generator system after flow control has been asserted.

The test is performed at 9600 baud and repeated several times. The average and standard deviation are calculated and reported as the character slew rate at 9600 bps.

Software Flow Control

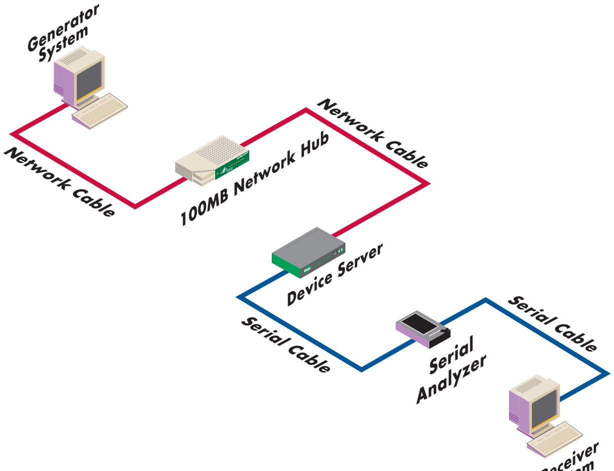

Figure 7.

Hardware Flow Control

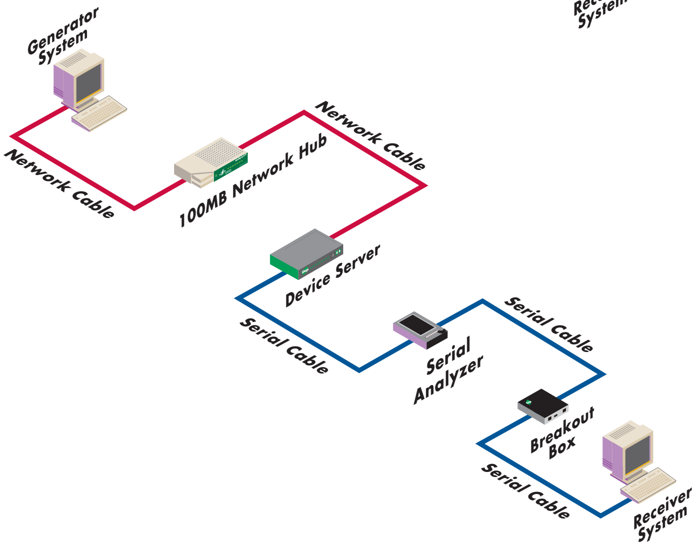

Figure 8.

Jitter

Jitter is measured by performing the latency tests (see figure 5). The main difference is that Jitter is not a measurement of round-trip time, rather it is the measurement of the variance between receive time intervals during the duration of the test. Since we know the interval of the transmit time we are only concerned with the accuracy or amount of "Jitter" in the receive time measurements.

This test is run using various data packet sizes such as 1, 10 and 32 bytes. This illustrates how consistent the system under test (SUT) is when handling different data packet sizes. We take the standard deviation of the receive time over the duration of the test (usually 10,000 samples) and report the results as the variance to 1 standard deviation at interval. If the interval is 100ms and the standard deviation of the receive time interval is 1.5, the reported "Jitter" is 1.5ms at 100ms.

Digi International
11001 Bren Road E.
Minnetonka, MN 55343 USA
PH: 877-912-3444
952-912-3444
FX: 952-912-4952
Email: info@digi.com
www.digi.com

Digi International GmbH
Joseph-von-Fraunhofer Str. 23
D-44227 Dortmund
Germany
PH: +49-231-9747-0
FX: +49-231-9747-111
www.digi.de

Digi International (HK) Limited
Suite 1703-05, 17/F.
K Wah Centre
191 Java Road
North Point, Hong Kong
PH: +852-2833-1008
FX: +852-2572-9989
www.digi.cn

Connectware™