

GPIB and Ethernet: Selecting the Better Instrument Control Bus

by Alex McCarthy, National Instruments
EE August 2005

GPIB or Ethernet? Some early testing results and recommendations can help you focus on the realities of each technology.

Ethernet, with its higher bandwidth and ubiquitous usage, seems a natural alternative to the market-leading GPIB instrument control bus for automating instruments. However, it's not always an accurate conclusion.

There are many factors to consider when determining which bus provides better performance. Most important are the bus bandwidth and latency, the device-side instrument control hardware and software implementation, and the host-side software.

With the recent introduction of several instruments with Ethernet and GPIB ports, National Instruments (NI) decided to compare the actual bus performance of Ethernet and GPIB in typical test use cases. The company is testing different types of instruments with multiple buses from multiple vendors. Even at this early stage, the results indicate that **the optimal bus is use-case-specific because neither bus was better at all tasks.**



Shorter Time Wins

Once you have selected the instrument and defined the tests, you need to decide on an instrument control bus. At this point, the primary performance goal usually is to minimize testing time. This is true for almost any part of the product development cycle from research through design and validation to production.

While the incremental test-time reduction value is higher in production, shorter test times are important throughout the process. One of the many factors in reducing test time is instrument control bus performance.

Before examining the instrument control buses, consider the actual tests. Most tests are made up of six smaller tasks: *set state*, *query*, *data transfer or block move*, *service request*, *trigger*, and *abort*. You can use these six tasks in many combinations to complete the required tests.

The order, number, and mix are dramatically different, depending upon the use case. A good estimate of the overall testing time is the sum of the product task duration multiplied by the number of times you use the task.

Because this article describes the differences in instrument control bus performance, the test setup keeps the host PC and instrument hardware and software constant for all tests. The only difference is the bus. This provides the impact of bus performance on transfer time, and tasks-per-second is the key metric.

Bandwidth vs. Latency

The benefits of higher bandwidth Ethernet over GPIB are not universal. In most test tasks, the higher bandwidth does not overcome the relatively longer latency.

Focusing on just the bus, there are two important components to performance—bandwidth and latency. GPIB is a byte-wide, parallel bus with test and measurement optimizations. Ethernet is a general-purpose serial bus with packet-based protocols.

The bus bandwidth is the data transfer rate, usually measured in bits per second for serial buses such as Ethernet and bytes per second for parallel buses such as GPIB. The bus latency is the time delay in transferring data, usually measured in seconds.

The theoretical maximum Ethernet bandwidth ranges from 10 Mb/s up to 1 Gb/s. Fast Ethernet, operating at 100 Mb/s and compatible with 10 MB/s, is the most common implementation for both PCs and instruments available today.

The maximum bandwidth for GPIB is 1.8 MB/s or 14.4 Mb/s for standard IEEE 488.1 and 7.7 MB/s or 61.6 Mb/s for high-speed IEEE 488.1. This gives 100 Mb/s Ethernet a 7x bandwidth advantage over GPIB.

The results indicate that the optimal bus is use-case-specific because neither bus was better at all tasks.

GPIB, however, has a shorter latency than Ethernet. Typical GPIB latency is about 100 μ s while Ethernet is in the 1,000- μ s range. This gives GPIB a 10x latency advantage over Ethernet. From these bus parameters, GPIB should be better suited for short data transfers and Ethernet better suited for larger data transfers.

GPIB also should be better at test-specific tasks such as instrument service requests and triggers. But even these two important factors are impacted by the actual implementation of the complete instrument control path that includes both PC and instrument hardware and software. The host software is a key part of the test system. The software impacts not only the instrument control performance, but also the capability to use multiple instrument control buses. Virtual instrument software architecture and instrument drivers based on it abstract the instrument control buses and are a key technology for multiple bus testing and usage.

As a result, only the virtual instrument software architecture resource name, such as GPIB::5::INSTR, typically changes when changing the bus. This makes profiling, developing, maintaining, and upgrading instruments faster and easier.

The PC hardware is fairly standard and plays a limited role unless it is so slow that it becomes a bottleneck. NI selected the PC configuration used in these tests so that it would not be a performance bottleneck. So, while the dual 3.0-GHz, Pentium 4 CPU with 4 GB of RAM and 2GbE ports is overkill for most test systems, it will not slow down bus performance tests.

On the bus instrument side, software and hardware also play a key part in performance. Software factors that affect instrument control bus performance include the embedded OS, firmware or kernel software, input and output buffer management, command parser, and control logic. Hardware factors include the bus interface chips, buffer size and type, processor, caching, and memory size and speed.

Given all these factors, it is not surprising that there is a wide range of results from bus to bus and from instrument to instrument. It is important to perform task-profiling tests to understand the real bus performance.

The test setup aims to focus on real instrument bus performance in near-ideal conditions. For Ethernet, this means only a crossover cable—no hub, switch, or general LAN traffic and only one instrument. For GPIB, this means only one instrument.

Figure 1 is a diagram of the test setup. For all tests, the host and device software were the same and the instrument initialized to the same state before each test run. Only the bus cable and the virtual instrument software architecture resource name GPIB::9::INSTR <-> TCP/IP::192.168.1.100::inst0::INSTR changed between test runs.

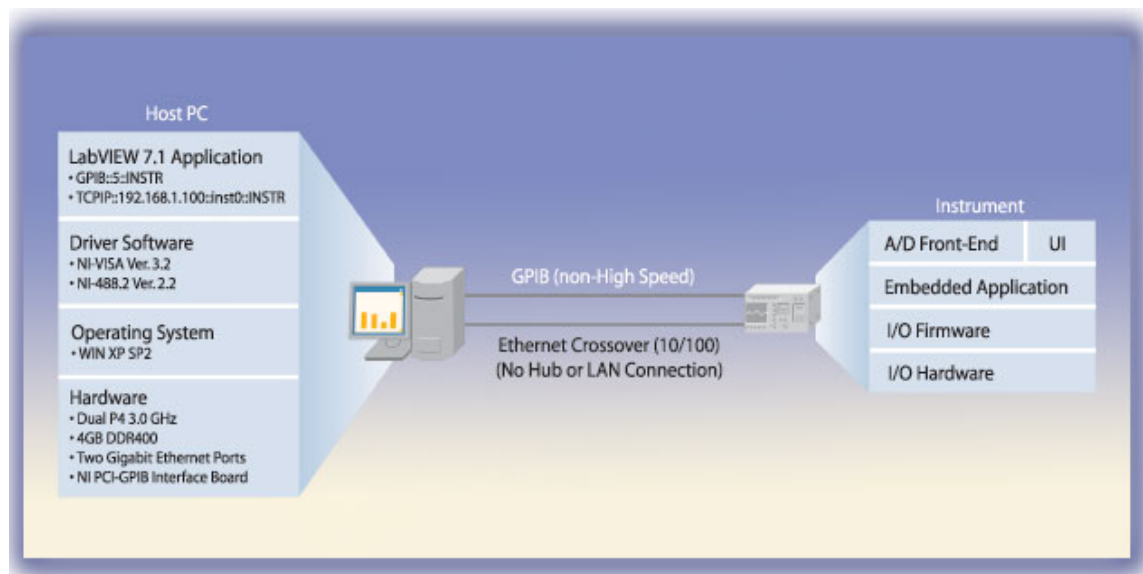


Figure 1. Instrumentation Control Bus Performance Test Setup
[click here to see larger image](#)

For this comparison, NI ran tests for the *set state*, *query*, and *transfer* tasks. The *set state* task used the “*CLS” command because it is widely implemented and does not depend on the instrument analog or digital front end. The *query* task used the “*IDN?” query because it also is widely implemented and does not have delays from the front-end interactions. The *transfers* read the 4-B, 256-B, 4-kB, and 32-kB blocks from the instrument output buffer. Each task ran from 50 to 1,000 iterations with the mean duration reported for each bus. The instrument tested is a mainstream instrument from a top-tier vendor with GPIB and 10/100 Ethernet ports. The results are shown in **Figures 2** and **3** as the number of tasks per second for GPIB and Ethernet.

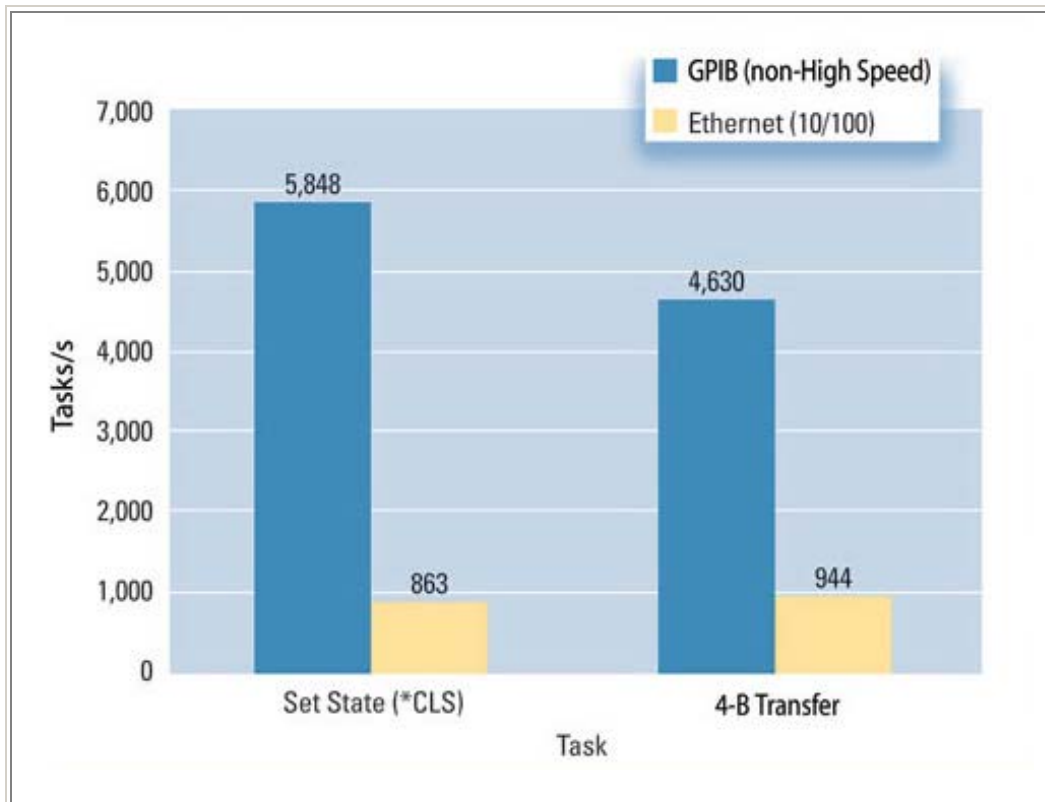


Figure 2. GPIB 578% and 390% Faster Than Ethernet for Set State and Very Small Transfers, Respectively

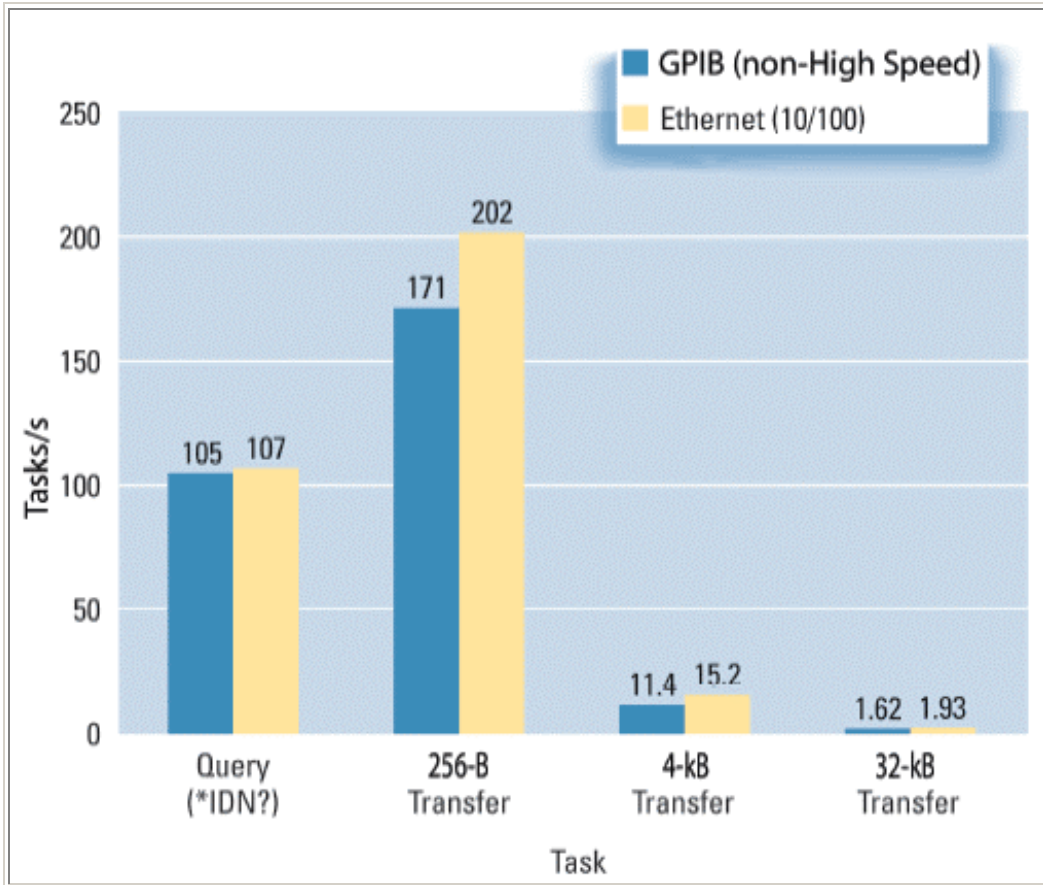


Figure 3. Ethernet 2% to 34% Faster Than GPIB for Queries and Transfers

GPIB is almost 6x to 4x faster than Ethernet for set state tasks and very small transfers. Even though GPIB has the shorter latency and is expected to be faster for small transfers, the magnitude of the difference is surprising.

For query tasks, Ethernet and GPIB essentially are the same. Keep in mind that most query responses are closer to the 4-B transfer size because headers usually are disabled. Finally, Ethernet only is 2% to 33% faster in 256-B, 4-kB, and 32-kB transfers. Although Ethernet at 100 Mb/s has almost 7x the bandwidth of normal GPIB at 14.4 Mb/s, it is only 2% to 34% faster in reality after all the factors are included.

So if GPIB is faster for small-byte tasks, the same for query tasks, and slower for larger transfer tasks, how should you select the better bus? Using these results, consider two hypothetical use cases: a validation tester and a production tester.

Validation test systems usually take higher resolution measurements with more variables and permutations, generating more raw data. This detailed and precise data helps define the test limits and masks for production but usually requires longer test runs.

Production test systems usually take lower resolution measurements or at least transfer 1-B instead of 2-B data, with fewer variables and permutations but run more individual tests. The production test system typically uses many more set-state tasks and returns smaller pass/fail or reduced data results, effecting shorter test times.

Both cases aim to reduce test times. The production requirement for shorter test times is obvious: Longer test times can increase both product costs and capital expenses. The validation value usually is lower than production but still very important as evidenced by Do Not Disturb—Test in Progress signs posted on testers.

Clearly, shorter test times are important. Validation often becomes the critical path for product releases when there are multiple hardware and firmware revisions that must be revalidated to ensure proper performance against the target specifications.

Figure 4 combines the performance results in two hypothetical test systems, one for validation and the other for production. All of the tests required to test one device include an estimate of the number of times the test uses each task. This estimate, divided by the measured tasks per second, produces the transfer time in seconds for each task.

Ethernet is 19% Faster Than GPIB

Hypothetical Validation Tester			
GPIB (non-High Speed)			
Tasks	Number	Tasks/s	Est. Time (s)
Set State	100	5,848	0.0
Query	1,000	105	9.6
Small Transfers	10	11.4	0.9
Large Transfers	1,000	1.62	618.9
			Total: 629.3
Ethernet (10/100)			
Tasks	Number	Tasks/s	Est. Time (s)
Set State	100	863	0.1
Query	1,000	107	9.4
Small Transfers	10	15.2	0.7
Large Transfers	1,000	1.93	517.2
			Total: 527.3
Higher Resolution More Variables and Permutations Longer Test Runs More Raw Data for Analysis			

Figure 4a. GPIB and Ethernet Within a ±25% Performance Range

GPIB is 25% Faster Than Ethernet

Hypothetical Production Tester			
GPIB (non-High Speed)			
Tasks	Number	Tasks/s	Est. Time (s)
Set State	10,000	5,848	1.7
Query	1,000	105	9.6
Small Transfers	100	11.4	8.8
Large Transfers	10	1.62	6.2
Total:			26.3
Ethernet (10/100)			
Tasks	Number	Tasks/s	Est. Time (s)
Set State	10,000	863	11.6
Query	1,000	107	9.4
Small Transfers	100	15.2	6.6
Large Transfers	10	1.93	5.2
Total:			32.7
Lower Resolution Less Variables and Fewer Permutations As Short as Possible More Pass/Fail and Reduced Data			

Figure 4b. GPIB and Ethernet Within a ±25% Performance Range

Summing the time for all the tasks gives an estimate of the total testing transfer time. Comparing total transfer times points to the better instrument control bus.

Figure 4 demonstrates two key observations. First, **GPIB is faster in production, and Ethernet is faster in validation.** This contradicts the bandwidth-only predictor that points to using Ethernet in production even though it actually may be slower. Second, **there is only a 20% to 25% difference either way, not the typical 7x bandwidth multiplier.**

This information should encourage you to use drivers based on virtual instrument software architecture, run some simple task profiling tests on instrument buses, plug in your own task profiles, and reap the reduced test time benefits in all research and development phases.

About the Author

Alex McCarthy is the senior product marketing manager for instrument control and VXI at National Instruments. He has more than 15 years of sales and marketing management, is a board member on the VXIbus Consortium, and has represented National Instruments as a member of the PXI System Alliance board. Mr. McCarthy received a B.S. in chemical engineering from Texas A&M University. National Instruments, 11500 Mopac Expressway, Austin, TX 78759, 512-683-0100, e-mail: alex.mccarthy@ni.com

FOR MORE INFORMATION

about GPIB and Ethernet implementations at NI
www.rsleads.com/508ee-206

Published by EE-Evaluation Engineering
 All contents © 2005 Nelson Publishing Inc.