Optimizing Test Systems for Highest Throughput, Lowest Cost and Ease of Integration with LXI Instruments

Application Note

A sampling of
Agilent LXI-compliant products

As instrument control busses have evolved over time, from GPIB in the early 1970's to FireWire and VXI in the 80's and 90's, to PXI today, data transfer speeds have increased from about 1 MBits/s to more than 400 MBits/s the busses have always required special interface cards to be added to PCs to enable their use. During the same time period, Ethernet (often called LAN, an acronym for Local Area Network) speed has increased from 3 MBits/s to 10 Gigabits/s and it has become commonplace on most modern PCs. See **Figure 1**. Now the LAN standard is the basis for a new class of instruments called LXI, or LAN eXtensions for Instrumentation.

LAN was not originally designed for instrument control but for high-speed data transfer of large *packets*. There is an overhead (latency) associated with every data transmission. Because short bursts of data are common with instrument control, latency issues can cause data transfers to take longer over a LAN than over GPIB. So, although LAN control is inexpensive and easy to use, if you have throughput-critical applications, you need to exercise care to make the latency issue less of a factor. This application note explains how to minimize latency when using LAN to control your instrumentation and explains why LXI is an excellent choice for cost-savings and ease of integration.

Programming Considerations

When you send a command to an instrument over any interface, there are two ways to send the command from your program to the hardware:

- *Drivers* – software that provides function calls to control the instrument, available in a variety of programming environments, such as LabVIEW™, Basic, C++®, etc. Drivers for register-based instruments such as PXI create the actual register data needed to set up an instrument and read data from it.
- *Direct I/O using SCPI (Standard Commands for Programmable Instruments)* industry-standard ASCII command language used nearly universally on GPIB instruments, also usable via the VXI-11 specification with LAN and USB. Can be sent to many interfaces from many languages using simple low-level read and write commands.

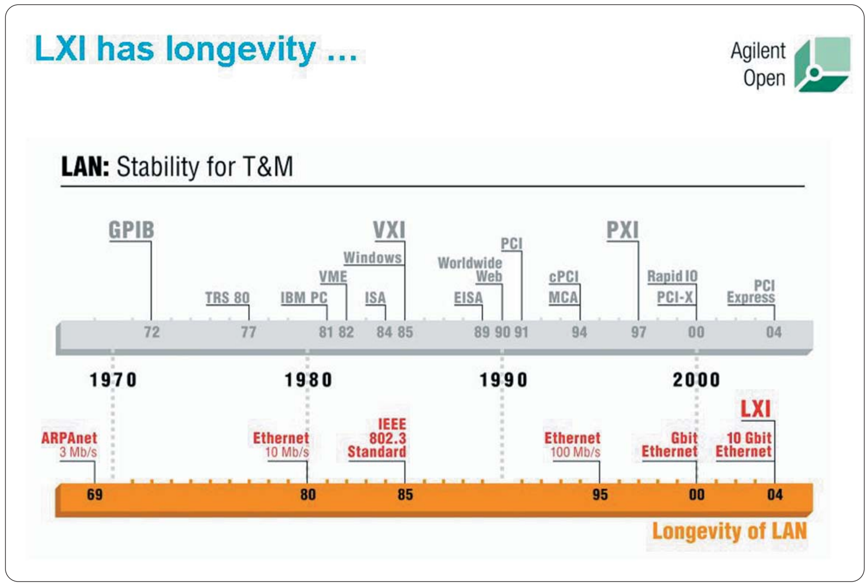


Figure 1. LAN has been with us as other busses have come and gone

Drivers can be written to control SCPI-based instruments too. Such drivers actually produce SCPI as their end result, so they *can* be inherently *slower* than programs that send SCPI directly because they must spend time constructing the SCPI command before sending it. However, they can also be *faster* than manually generated SCPI if they keep track of state information (that is, sending only changed information to the instrument), and if they concatenate commands efficiently. Unfortunately, there is almost no way to tell ahead of time if a driver is coded efficiently. As we will see shortly, bus latency can add dramatically to execution times if commands that can be sent as a single string are instead sent as separate strings.

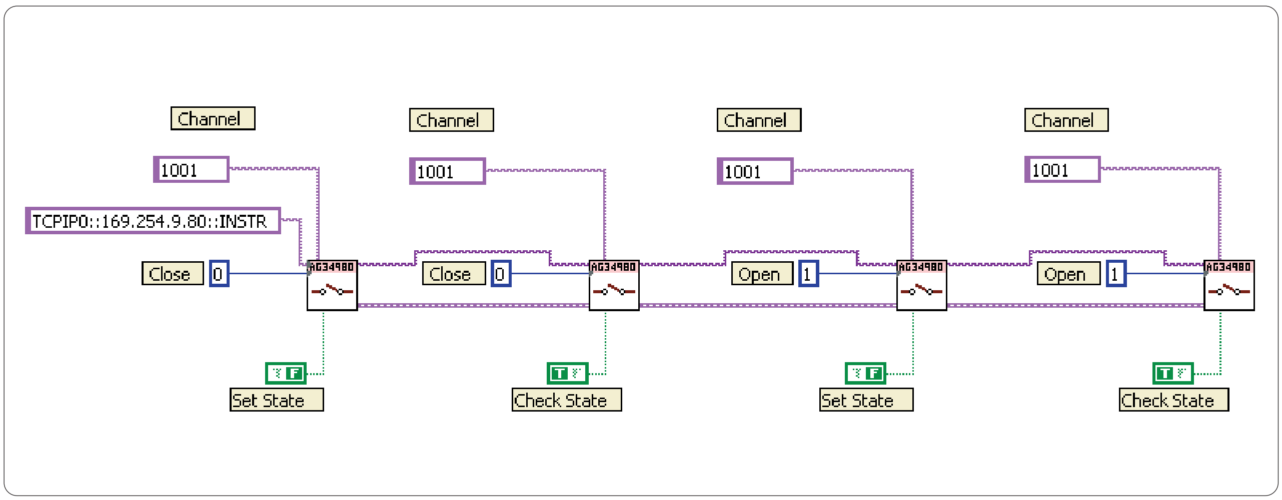
Because all PXI devices and most VXI devices are register-based, not ASCII-based, they *require* you to use a driver provided by the card vendor for your chosen development environment. This driver in turn communicates with an *interface driver* (generally supplied as part of I/O libraries) that handles the mundane tasks of sending and receiving data on the chosen interface (LAN, USB, GPIB, RS-232, FireWire, MXI-4, etc.). If you are using PXI via a remote PC, the interface driver communicates over the NI proprietary MXI interface via a proprietary interface card that you must first install in your PC. GPIB interfaces also typically require a proprietary card, which are available from several vendors, that must be plugged into your PC. LAN interfaces, in contrast, are standard on most PCs and require only inexpensive interconnect cables.

When you use an instrument driver, you have to trust that person who wrote the driver implemented the necessary functionality correctly and efficiently. In the case of register-based devices, the driver writer must implement 100% of the instrument's functionality, since that is the only way to control the instrument. With SCPI-based devices, drivers frequently do not have all of the possible instrument functions implemented, since the instrument itself has 100% of the functionality accessible via SCPI commands. Driver writers sometimes implement a "SCPI pass-through" mode, allowing you to send commands directly that are not implemented in the driver. You can also use VISA or VISA-COM to send SCPI commands directly to the instrument, saving you the extra effort to download, install and learn the instrument-specific driver. However, you then need to learn the relevant SCPI commands for the instrument. This is not a bad thing. It is not uncommon to find defects in vendor-supplied software drivers. If you take the time to understand the SCPI commands that control the instrument, debugging is much easier because you can turn on an I/O monitor such as the one provided in Agilent's T&M Toolkit or NI Spy and observe the SCPI messages that the driver is sending. If you find a bug in a driver that uses SCPI, you can code around it yourself while waiting for an update from the vendor. If you find a bug in a register-based driver, you have no choice but to wait for a fix or to not use that particular function.

Drivers for register-based instruments need only send binary data over a very fast backplane, and no parsing (decoding) is required by the instrument. Since SCPI is an ASCII-based language, it generally takes longer to send the strings down an interface, and it requires parsing inside the instrument. The total time to handle such an ASCII transfer can be on the order of 1 ms with modern instrument processors and high-speed LAN, but that time is realistically independent of command length. In fact, the time to send a single byte or a hundred bytes over LAN is about 1 ms with 100BaseT Ethernet interfaces.

How do I get around the latency issue?
If you use a LabVIEW driver to close a relay, check to see that it closed, open it and check to see that it opened, you will send four commands to the instrument and receive two responses from the instrument. Using a typical PXI reed relay switch card, it takes about 3.7 ms to complete this task. Without optimization, an equivalent LabVIEW program for the L4433A LXI reed relay matrix takes more than 8 ms.

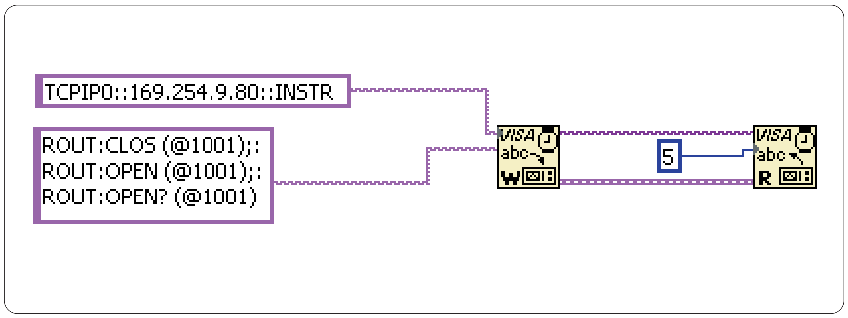
It is possible to construct SCPI commands that will optimize the LAN traffic into only 2 packets, thus saving about 4.6 ms and allowing the L4433A LXI reed relay matrix to toggle a relay in essentially the same time as its PXI competition – 3.9 ms. Let's look at the relay toggle example as implemented in LabVIEW without any optimization:



This LabVIEW program emits the following SCPI code:

ROUT:CLOS (@1001)	1.4 ms	(Set state)
ROUT:CLOS? (@1001)	1.3 ms	(Check state)
Read 1	1.5 ms	
ROUT:OPEN (@1001)	1.6 ms	(Set state)
ROUT:OPEN? (@1001)	1.0 ms	(Check state)
Read 1	1.5 ms	
Total:	8.3 ms	(6 LAN packets)

If you use concatenated SCPI commands carefully, you can greatly simplify this sequence. However, instead of using the LabVIEW driver for the L4433A, you would use VISA calls:



This LabVIEW program emits the following code:

ROUT:CLOS(@1001);:ROUT:OPEN (@1001);:ROUT:OPEN? (@1001)	1.4 ms
Read "1"	2.5 ms
Total:	3.9 ms (2 LAN packets)

From a textual language such as Visual Basic, such commands can be output directly using VISA or VISA-COM statements. For example, using Agilent's T&M Toolkit in the .NET environment, a typical statement is:

```
myAgilentL4433A.WriteLine ("ROUT:CLOS (@1001);:ROUT:OPEN (@1001);:ROUT:OPEN?(@1001)")
s = myAgilentL4433A.Read()
```

Note that there was no need to execute a ROUT:CLOS? (@1001) between the CLOS and OPEN because the instrument does not execute the OPEN until the CLOS is done. If you added it anyway, the Read would return "1,1" and there would be no impact on execution time. This points out the benefit of understanding all the nuances of your instrument in order to extract the best possible throughput. When you take the time to learn the SCPI commands for the instrument, you gain a better understanding of how it works than you might get from using a driver.

If execution speed is not important in your application, drivers are an excellent choice. They provide canned functionality and are relatively quick to implement, especially using Visual Studio.NET® tools. LXI instruments offer you the choice. Register-based cardcages require you to use a driver.

Figure 3. Rear panel of Agilent L4433A showing LAN cable. This inexpensive cable connects directly to many PCs or networks without the need to install an I/O card

Agilent L4433A 128-channel crosspoint reed matrix $2350

LAN cable: ~$5 USD

Installation time: < 1 minute

Other considerations
Speed is not the only factor you must consider when choosing instrumentation. Cost and ease of integration into a test system are two other big factors. **Figures 3** and **4** describe these issues.

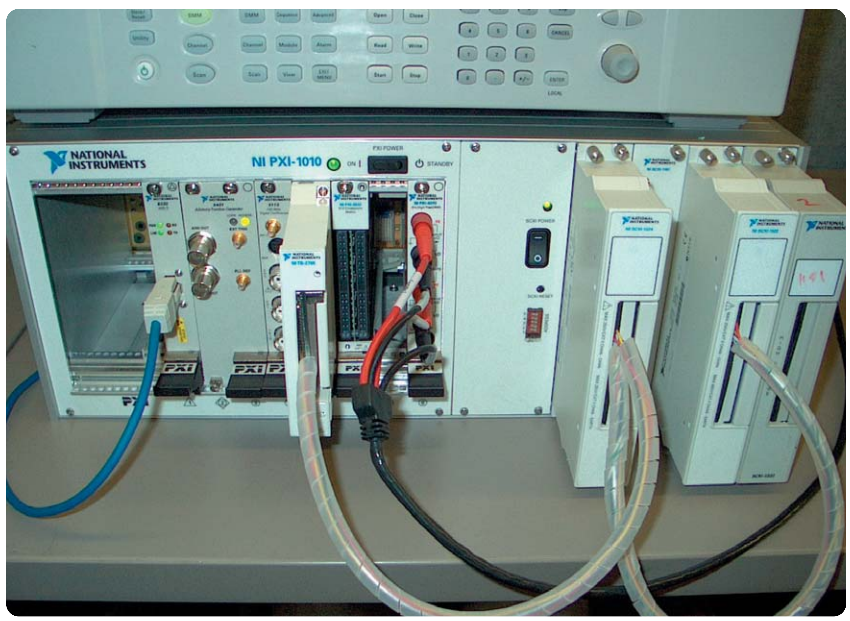


Figure 4. An NI PXI/SCXI cage with an NI MXI-3 interface. If you use a PXI cardcage, typically you need either an embedded PC or an external PC with a controller card installed in it.

NI PXI 8-slot cage: ~$1000 (4 slot) to $4500 (18-slot) USD (picture shows older PXI-1010 model with 4 SCXI slots, new version is PXI-1050 ~$2800) – overhead per slot: ~$275

MXI-4 cable and I/O card for PC and for PXI cage: ~$1500 USD

PXI-2530 128-channel crosspoint reed matrix: ~$1700 USD

Installation time: ~10 minutes (requires PC reboot)

Prices from NI Web site as of February 2006

LXI instruments offer these advantages over PXI:

- No unused slot overhead – get only what you need, when you need it
- Multi-vendor support from top companies in the test and measurement industry (see **www.lxistandard.org**)
- No dedicated, proprietary computer interface needed
- Easier install process with no PC reboot required
- Better shielding from stray magnetic fields
- Each box is powered from independent AC power supplies that are optimized for the instrument, eliminating the possibility of insufficient backplane power
- Up to 3 fewer EIA units of rack space (1 U vs. 4 U) depending upon the number of LXI modules you need. If you need more than 6, consider using an Agilent 34980A switch/measure unit LXI-C compliant cardcage with a built-in 6.5-digit DMM and the ability to hold up to 8 modules, all in 3 U of rack space.

Summary

If program execution time is important to you, think about the code that is being sent to the instrument by your program and try to concatenate as many SCPI strings as possible to save LAN overhead. Drivers do not typically do this as efficiently as a human, so when ultimate throughput is your goal, consider using SCPI, using VISA or VISA-COM to control the instrument yourself rather than relying on a driver.

If cost is important to you, then LXI solutions can be lower cost than PXI because a mainframe and a specialized PC interface are not required. You can also choose to buy just what your budget allows today and expand it tomorrow as your needs and/or budget change.

If rack space is paramount, LXI modules use only 1 EIA unit and half rack width per module, allowing up to 8 such modules in the same space as a PXI cage. This gives you more flexibility in arranging instruments in your rack. Other LXI-compliant instruments, such as Agilent's 34980A, can save rack space and cost when the number of LXI modules exceeds 6.

If location is important, the LAN connectivity and small size of LXI allow you to locate instruments where they are needed without the need for a PC or mainframe.

Related Agilent literature

Brochure
5989-2042EN
Open the door to simpler system creation

Application Note
5989-1412EN
System Developer Guide Using LAN in Test Systems: The Basics

For copies of this literature, contact your Agilent representative or visit **www.agilent.com/find/systemcomponents**

Agilent Technologies' Test and Measurement Support, Services, and Assistance
Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Two concepts underlie Agilent's overall support policy: "Our Promise" and "Your Advantage."

Our Promise
Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you receive your new Agilent equipment, we can help verify that it works properly and help with initial product operation.

Your Advantage
Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional engineering services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.

Agilent Email Updates

www.agilent.com/find/emailupdates
Get the latest information on the products and applications you select.

Agilent Direct

www.agilent.com/find/agilentdirect
Quickly choose and use your test equipment solutions with confidence.

Agilent Open

www.agilent.com/find/open
Agilent Open simplifies the process of connecting and programming test systems to help engineers design, validate and manufacture electronic products. Agilent offers open connectivity for a broad range of system-ready instruments, open industry software, PC-standard I/O and global support, which are combined to more easily integrate test system development.

www.agilent.com

For more information on Agilent Technologies' products, applications or services, please contact your local Agilent office. The complete list is available at:

www.agilent.com/find/contactus

Phone or Fax

United States:
(tel) 800 829 4444
(fax) 800 829 4433

Canada:
(tel) 877 894 4414
(fax) 800 746 4866

China:
(tel) 800 810 0189
(fax) 800 820 2816

Europe:
(tel) 31 20 547 2111

Japan:
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Korea:
(tel) (080) 769 0800
(fax) (080) 769 0900

Latin America:
(tel) (305) 269 7500

Taiwan:
(tel) 0800 047 866
(fax) 0800 286 331

Other Asia Pacific Countries:
(tel) (65) 6375 8100
(fax) (65) 6755 0042
Email: tm_ap@agilent.com
Contacts revised: 09/26/05

Product specifications and descriptions in this document subject to change without notice.

Printed in the USA March 24, 2006
5989-4886EN

Agilent Technologies