VMEbus Extensions for Instrumentation

TCP/IP-VXIbus Interface Specification

VXI-11.1

Revision 1.0

July 17, 1995

NOTICE

The information contained in this document is subject to change without notice.

The VXIbus Consortium, Inc. makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The VXIbus Consortium, Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

VMEbus Extensions for Instrumentation TCP/IP-VXIbus Interface Specification VXI-11.1, Revision 1.0 is authored by the VXIbus Consortium, Inc. and its sponsor members:

GenRad, Inc.
Hewlett-Packard Co.
National Instruments, Corp.
Racal Instruments, Inc.
Tektronix, Inc.
Wavetek, Inc.

Table of Contents

List of Figures

VMEbus Extensions for Instrumentation:

TCP/IP-VXIbus Interface Specification

A. INTRODUCTION

The need to connect VXIbus instruments to computer networks has developed in the test and measurement industry. The connections required may be to either local-area networks (LANs) or wide-area networks (WANs). Along with this comes the need to have a standard that specifies the functionality of a TCP/IP-VXIbus Interface Device. This specification, which is part of the VXIbus set of specifications, describes how VXIbus instrumentation can be connected to industry-standard networks. The communications and programming paradigms supported by this specification are similar in nature to the techniques supported by VXIbus Instruments and VXIbus IEEE-488.2 Instruments. The methods described allow ASCII-based communications to take place between a controller and a device over a TCP/IP network. The reader should be knowledgeable about networks, the Internet Protocol Suite, ONC RPC, IEEE 488.2, and VXIbus.

A.1. SCOPE

This specification is part of the VXIbus set of specifications and defines a TCP/IP-VXIbus Interface Device.

The only networks directly considered by this specification are those which support the Internet Protocol Suite. The techniques defined in this specification could be used over other networks, such as networks which support the OSI protocol standards, but this document does not address that mapping. This specification uses Open Network Computing (ONC) remote procedure calls on top of the Internet Protocol Suite.

Other network protocols may also be supported by a TCP/IP-VXIbus Interface Device.

A.2. DOCUMENT STRUCTURE

This document is divided into 2 sections. The first section, an introduction, is intended to familiarize readers with the intent and scope of the document.

The second section, TCP/IP-VXIbus Interface, defines the operation of a TCP/IP-VXIbus Interface Device, including the mapping between the network transactions defined by VXI-11, TCP/IP Instrument Protocol Specification, and the VXIbus word-serial protocol.

A.3. SPECIFICATION OBJECTIVES

This specification has the following objectives in addition to those outlined in the TCP/IP Instrument Protocol Specification, VXI-11:

1. To define the operation of a TCP/IP-VXIbus Interface Device.
2. To define a mapping from network transactions to VXIbus word-serial protocol transactions.

A.4. DEFINITION of TERMS

This specification uses the following terms in addition to those defined in the TCP/IP Instrument Protocol Specification, VXI-11:

VXIbus instrument: an instrument as defined by the IEEE Std. 1155-1992, VMEbus Extensions for Instrumentation: VXIbus.

The following terms are used to identify the contents of paragraphs, as in other VXIbus Specifications. These definitions are the same as those in IEEE 1155-1992.

RULE: Rules **SHALL** be followed to ensure compatibility for cards in the system. A rule is characterized by the use of the words **SHALL** and **SHALL NOT**. These words are not used for any other purpose other than stating rules.

RECOMMENDATION: Recommendations consist of advice to implementors which will affect the usability of the final device. Discussions of particular hardware to enhance throughput would fall under a recommendation. These should be followed to avoid problems and to obtain optimum performance.

PERMISSION: Permissions are included to clarify the areas of the specification that are not specifically prohibited. Permissions reassure the reader that a certain approach is acceptable, and will cause no problems. The word **MAY** is reserved for indicating permissions.

OBSERVATION: Observations spell out implications of rules and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain rules, so that the reader understands why the rule must be followed.

Any text that appears without a heading should be considered as description of the standard.

A.5. REFERENCES

This specification references the following documents in addition to those referenced by the TCP/IP Instrument Protocol Specification, VXI-11:

[1] IEEE Std 1155-1992, IEEE Standard for VMEbus Extensions for Instrumentation: VXIbus.
[2] VMEbus Extensions for Instrumentation: VXIbus, Common ASCII System Commands Specification, VXI-5, Revision 1.0, June 23, 1991.

A.6. RELATED DOCUMENTS

This specification is one document in a set of specifications which describe a method for ASCII-based communication over a network between controllers and devices. This specification describes the mapping

from the protocol to VXIbus word-serial operations, along with the operation of a TCP/IP-VXIbus Interface Device. Other specifications in the group describe the protocol itself as well as the mapping from the protocol to other interface types. Those specifications listed below are currently part of this group:

[1] VMEbus Extensions for Instrumentation: TCP/IP Instrument Protocol Specification, VXI-11, Revision 1.0.
[2] VMEbus Extensions for Instrumentation: TCP/IP-IEEE 488.1 Interface Specification, VXI-11.2, Revision 1.0.
[3] VMEbus Extensions for Instrumentation: TCP/IP-IEEE 488.2 Instrument Interface Specification, VXI-11.3, Revision 1.0.

B. TCP/IP-VXIbus INTERFACE

The TCP/IP-VXIbus Interface Device converts TCP/IP instrument protocol messages to VXIbus communication protocols. It allows controllers to control VXIbus instrument devices (see section 4, "VXIbus Device Implementations" of IEEE 1155-1992) connected to the LAN via the TCP/IP-VXIbus interface. The TCP/IP-VXIbus interface receives instrument control requests from the LAN, converts them to VXIbus transactions, and routes the request to the appropriate VXIbus instrument device. Register based, memory, and non-instrument message based devices are beyond the scope of this specification, although the command support described in section B.5 of this specification does provide limited support for non-instrument devices. A VXIbus system may or may not provide a TCP/IP-VXIbus interface.

OBSERVATION B.1:
A VXIbus instrument responds to the *Read Protocol* command with the *I* bit cleared to zero(0). The *I4* bit may also be cleared to zero(0).

RULE B.1:
A TCP/IP-VXIbus Interface Device **SHALL** be a Message Based Commander.

RULE B.2:
A TCP/IP-VXIbus Interface Device **SHALL** support the receipt of signals.

RULE B.3:
A TCP/IP-VXIbus Interface Device **SHALL** be an Interrupt Handler.

RULE B.4:
When a TCP/IP-VXIbus Interface Device experiences a backplane bus error during the execution of a requested operation, the interface **SHALL** indicate this fact by returning 17, I/O error, in the *error* field of the RPC reply message.

RECOMMENDATION B.1:
A TCP/IP-VXIbus Interface Device should support two or more *network instrument* servers simultaneously.

RULE B.5:
A TCP/IP-VXIbus Interface Device **SHALL** support at least 64 concurrent links per *network instrument* server.

RECOMMENDATION B.2:
The number of *network instrument* servers and links supported by a TCP/IP-VXIbus Interface Device should be based on available resources, not on arbitrary predetermined limits.

OBSERVATION B.2:
A VXIbus instrument is accessible to a *network instrument* server if the *network instrument* server's host is the instrument's Commander. Accessibility as defined here applies to RULE B.4.1 of VXI-11, TCP/IP Instrument Protocol Specification.

Figure B.1 shows a typical TCP/IP-VXIbus system.

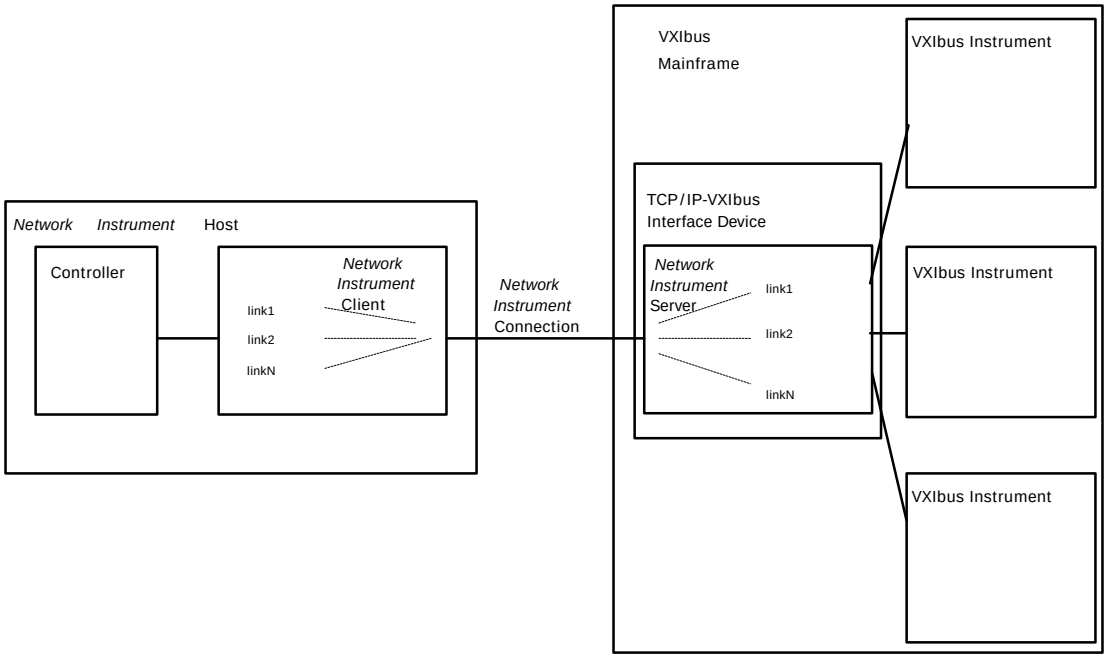

Figure B.1 Typical TCP/IP-VXIbus System

B.1. DEVICE STRING FORMAT

The routing of messages from the LAN to the appropriate VXIbus device takes place via the *create_link* RPC. This RPC is used to create a *network instrument* link. The created link is associated with a particular VXIbus device via the *device* parameter. This parameter is a character string which is parsed by the TCP/IP-VXIbus interface device to determine which VXIbus device the link is associated with.

RULE B.1.1:

A TCP/IP-VXIbus Interface Device **SHALL** support a *device* string of the following format:

```
<intf_name>[,<logical_addr>]
```

where:

- <intf_name> - A name corresponding to the address of a single VXIbus interface. This name **SHALL** uniquely identify the interface on the TCP/IP-VXIbus interface device.
- <logical_addr> - The logical address of a VXIbus device on the VXIbus interface. (optional)

RULE B.1.2:

A *device* which contains just <intf_name> **SHALL** be interpreted by the TCP/IP-VXIbus Interface Device as a link to the interface's command processor (see section B.5, "TCP/IP-VXIbus Interface Device Commands").

RULE B.1.3:

The TCP/IP-VXIbus Interface Device **SHALL** recognize "vxi0" as the first or only interface within the device. Additional interfaces **SHALL** be identified by "vxiN" where N is a non-negative integer assigned sequentially beginning at one.

Examples of valid device parameters are:

vxi0 link is associated with first VXI interface in the host
vxi0,128 link is associated with device at logical address 128

RULE B.1.4:
If an address is received for a device which is not a servant of the TCP/IP-VXIbus Interface Device, then *create_link* **SHALL** return *error* set to 3, device not accessible.

B.2. LAN FUNCTIONS

RULE B.2.1:
A TCP/IP-VXIbus Interface Device **SHALL** support the *network instrument* protocol, and **SHALL** accept and execute all of the RPCs defined by that protocol. Support of the *network instrument* protocol includes support for the entire protocol stack defined for use by *network instrument* devices.

PERMISSION B.2.1:
A TCP/IP-VXIbus Interface Device **MAY** support other LAN protocols.

RULE B.2.2:
A TCP/IP-VXIbus Interface Device **SHALL** support an Ethernet/802.3 Data Link Layer and an 802.3/10BASE-T Physical Layer with an RJ-45 connector for 10BASE-T.

OBSERVATION B.2.1:
The intent of this rule is that the default configuration of the TCP/IP-VXIbus Interface Device provides an RJ-45 connector for 10BASE-T. Other connectors may be supported, and may replace the RJ-45 connector.

The resulting protocol stack is shown in Figure B.2.

Application	*Network Instrument*
Presentation	XDR
Session	ONC/RPC
Transport	TCP
Network	IP
Data Link	Ethernet/802.3
Physical	802.3/10BASE-T

Figure B.2 TCP/IP-VXIbus Interface Device Protocol Stack

PERMISSION B.2.2:
A TCP/IP-VXIbus Interface Device **MAY** support other protocols at the Data Link and Physical layers.

B.3. LINK MANAGEMENT

This section covers the operations which manage links or operations on links. These operations are implemented primarily within the TCP/IP-VXIbus Interface Device itself.

B.3.1. Link Creation

The *create_link* and *destroy_link* RPCs do not directly map to any VXIbus operations.

RULE B.3.1:
The *create_link* and *destroy_link* RPCs **SHALL** be implemented entirely within the TCP/IP-VXIbus Interface Device. The TCP/IP-VXIbus interface device **SHALL** maintain links as defined by the *network instrument* protocol.

B.3.2. Interrupt Channel Creation

The *create_intr_chan* and *destroy_intr_chan* RPCs do not directly map to any VXIbus operations.

RULE B.3.2:
The *create_intr_chan* and *destroy_intr_chan* RPCs **SHALL** be implemented entirely within the TCP/IP-VXIbus Interface Device. The TCP/IP-VXIbus interface device **SHALL** maintain the interrupt channel as defined by the *network instrument* protocol.

B.3.3. Device Locking

The *device_lock* and *device_unlock* RPCs do not directly map to any VXIbus operations.

RULE B.3.3:
The *device_lock* and *device_unlock* RPCs **SHALL** be implemented entirely within the TCP/IP-VXIbus Interface Device.

B.3.4. Abort Operation

The *device_abort* RPC does not directly map to any VXIbus operations.

RULE B.3.4:
The *device_abort* RPC **SHALL** be implemented entirely within the TCP/IP-VXIbus Interface Device to abort any active operations associated with the link.

B.4. INSTRUMENT COMMUNICATIONS

This section covers the operations which communicate directly with VXIbus instruments. The VXIbus specifications provide two mechanisms for transferring ASCII messages, the Byte Transfer Protocol (see IEEE 1155-1992, section 3.3.3.3, "Byte Transfer Protocol") and the Fast Data Channel Message Transfer

Protocol (see VXI-10, Fast Data Channel Specification). When possible, the TCP/IP-VXIbus Interface Device will use the Message Transfer Protocol, otherwise the Byte Transfer Protocol will be used.

The rules in this section only apply to communication with VXIbus instruments and not to the interface's command processor.

RULE B.4.1:
TCP/IP-VXIbus Interface Devices and VXIbus Instruments **SHALL** be logically partitioned in the same manner as defined in IEEE 1155-1992, section 4.2.3, "VXIbus Instrument Protocol".

RULE B.4.2:
The TCP/IP-VXIbus Interface Device **SHALL** report on the success or failure of the word serial protocol transaction associated with the RPC by returning an appropriate error value. The TCP/IP-VXIbus Interface Device **SHALL** perform the appropriate word serial operations to determine the success or failure of the operation, as specified in IEEE 1155-1992, OBSERVATION 3.3.17.

RECOMMENDATION B.4.1:
The TCP/IP-VXIbus Interface Device should support the Fast Data Channel Message Transfer Protocol.

PERMISSION B.4.1:
If a TCP/IP-VXIbus Interface Device detects that a device has entered the Fast Handshake active state, it **MAY** perform Fast Handshake transfers.

B.4.1. Data Transfer from Interface Device to VXIbus Instrument

RULE B.4.3:
When the TCP/IP-VXIbus Interface Device receives a *device_write* RPC, it **SHALL** transfer the bytes in the *data* parameter to the VXIbus instrument associated with the link via either the Byte Transfer Protocol or the Message Transfer Protocol. If the end flag is set in the *flags* parameter, then the last byte **SHALL** be sent to the device with END asserted.

RULE B.4.4:
If the TCP/IP-VXIbus Interface Device supports the Fast Data Channel Message Transfer Protocol and determines via local configuration that the VXIbus instrument associated with the link does as well, the TCP/IP-VXIbus Interface Device **SHALL** transfer the bytes in the *data* parameter to the VXIbus instrument associated with the link via the Message Transfer Protocol.

B.4.2. Data Transfer from VXIbus Instrument to Interface Device

RULE B.4.5:
When the TCP/IP-VXIbus Interface Device receives a *device_read* RPC, it **SHALL** transfer bytes from the VXIbus instrument associated with the link into the *data* response parameter using either the Byte Transfer Protocol or the Message Transfer Protocol until a termination condition is encountered.

RULE B.4.6:
If the TCP/IP-VXIbus Interface Device supports the Fast Data Channel Message Transfer Protocol and determines via local configuration that the VXIbus instrument associated with the link does as well, the TCP/IP-VXIbus Interface Device **SHALL** transfer bytes from the VXIbus instrument associated with the link into the *data* response parameter via the Message Transfer Protocol until a termination condition is encountered.

B.4.3. Device Clear Operation

RULE B.4.7:
When the TCP/IP-VXIbus Interface Device receives a *device_clear* RPC, it **SHALL** send the VXIbus word serial *Clear* command to the device associated with the link.

B.4.4. Trigger Operation

RULE B.4.8:
When the TCP/IP-VXIbus Interface Device receives a *device_trigger* RPC, it **SHALL** send the VXIbus word serial *Trigger* command to the device associated with the link. If the *Trigger* command is not supported by the device, *error* **SHALL** be set to 8, operation not supported.

PERMISSION B.4.2:
If a TCP/IP-VXIbus Interface Device knows that the device does not support the *Trigger* command, it **MAY** choose to not send the *Trigger* command, but simply return with *error* set to 8, operation not supported.

RULE B.4.9:
A TCP/IP-VXIbus Interface Device **SHALL NOT** send a VXIbus word serial *Trigger* command to an instrument until *DIR* bit is set one(1).

B.4.5. Remote/Local Operation

RULE B.4.10:
When the TCP/IP-VXIbus Interface Device receives a *device_remote* RPC, it **SHALL** send the VXIbus word serial *Set Lock* command to the device associated with the link. If the *Set Lock* command is not supported by the device, *error* **SHALL** be set to 8, operation not supported.

RULE B.4.11:
When the TCP/IP-VXIbus Interface Device receives a *device_local* RPC, it **SHALL** send the VXIbus word serial *Clear Lock* command to the device associated with the link. If the *Clear Lock* command is not supported by the device, *error* **SHALL** be set to 8, operation not supported.

PERMISSION B.4.3:
If a TCP/IP-VXIbus Interface Device knows that the device does not support the *Set Lock* or *Clear Lock* command, it **MAY** choose to not send the *Set Lock* or *Clear Lock* command, but simply return with *error* set to 8, operation not supported.

B.4.6. SRQ Operation

The TCP/IP-VXIbus Interface Device sends the *device_intr_srq* RPC based on the RQS state as depicted in Figure B.3 and the enabled state of SRQ interrupts in general.

RULE B.4.12:
The TCP/IP-VXIbus Interface Device **SHALL** maintain the RQS state as depicted in Figure B.3 for each VXIbus instrument associated with an active link.

RULE B.4.13:
For each link, if the interrupt channel is already established and service requests are already enabled by *device_enable_srq* when the RQS state changes from FALSE to TRUE for the associated VXIbus instrument, the TCP/IP-VXIbus Interface Device **SHALL** send *device_intr_srq*.

RULE B.4.14:
For each link, if the interrupt channel is already established and the RQS state is already TRUE when service requests are enabled by *device_enable_srq* when they were previously disabled , the TCP/IP-VXIbus Interface Device **SHALL** send *device_intr_srq*.

RULE B.4.15:
The TCP/IP-VXIbus Interface Device **SHALL NOT** send *device_intr_srq* under any other circumstances.

RECOMMENDATION B.4.2:
A *network instrument* client should create the interrupt channel before enabling service requests and maintain the interrupt channel while service requests are enabled. Otherwise, service requests might be lost.

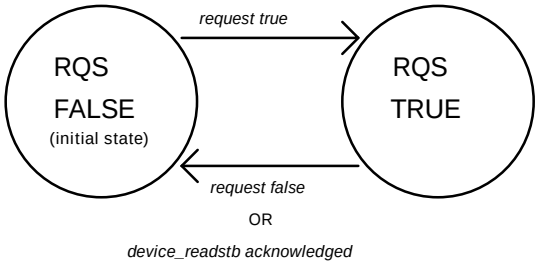

Figure B.3 RQS State Diagram

- *request true* event received from the VXIbus instrument: RQS TRUE state is entered
- *request false* event received from VXIbus instrument: RQS FALSE state is entered
- *device_readstb* is executed for the VXIbus instrument: RQS FALSE state is entered

OBSERVATION B.4.1:
The *request true* VXIbus event does not generate a *device_intr_srq* RPC if the current state is RQS TRUE.

B.4.7. Read Status Byte Operation

RULE B.4.16:
When the TCP/IP-VXIbus Interface Device receives *device_readstb*, if the VXIbus instrument supports the *Read STB* command, then the TCP/IP-VXIbus Interface Device **SHALL** send the VXIbus word serial *Read STB* command and respond to the RPC with the result obtained from the *Read STB* command.

RULE B.4.17:
If the VXIbus instrument does not support the *Read STB* command, then the TCP/IP-VXIbus Interface Device **SHALL** respond to the RPC with bit 6 of the response byte properly indicating the RQS state of the instrument. Bit 6 **SHALL** be set to one (1) if the instrument is requesting service, otherwise it **SHALL** be cleared to zero (0).

RULE B.4.18:
If the VXIbus instrument does not support the *Read STB* command the bits other than bit 6 of the response byte **SHALL** be cleared to zero(0) by the TCP/IP-VXIbus Interface Device.

B.5. TCP/IP-VXIbus INTERFACE DEVICE COMMANDS

This section covers the operations which communicate with an interface. A link can be associated with an interface. Such a link has a VXI-5 command processor (parser) and communication to this VXI-5 command processor does not directly involve any VXIbus operations, though the VXI-5 command processor may perform VXIbus operations as a result of certain commands.

The commands in VXI-5 provide system information and configuration control which is necessary for system components to interoperate.

RULE B.5.1:
A TCP/IP-VXIbus Interface Device **SHALL** provide a VXI-5 command processor for each VXIbus interface within the TCP/IP-VXIbus Interface Device.

RULE: B.5.2:
Each VXI-5 command processor within a TCP/IP-VXIbus Interface Device **SHALL** comply with all the requirements and restrictions specified in VXI-5, "Common ASCII System Commands".

OBSERVATION: B.5.1:
Changes to VXI-5 are anticipated as a working group has been formed to review its technical requirements. Forward compatibilty is not guaranteed to the next revision.

PERMISSION: B.5.1:
Each VXI-5 command processor within a TCP/IP-VXIbus Interface Device **MAY** comply with all the requirements and restrictions specified in sections B.4, "Instrument Communications", and B.5, "Relationship to IEEE 488.2", of VXI-11.3, "TCP/IP-IEEE 488.2 Instrument Interface Specification", Revision 1.0.

OBSERVATION: B.5.2:
This permission implies that each VXI-5 command processor within a TCP/IP-VXIbus Interface Device is allowed to behave like a TCP/IP-IEEE 488.2 Instrument Interface.

OBSERVATION: B.5.3:
Due to the work referred to in observation B.5.1, exactly what it means for a VXI-5 command processor to behave in a manner consistent with IEEE 488.2 is yet to be determined.

B.6. DEVICE_DOCMD COMMANDS

There are no *device_docmd* commands which a TCP/IP-VXIbus Interface Device accepts.

RULE B.6.1:
If a TCP/IP-VXIbus Interface Device receives the *device_docmd* message, it **SHALL** reply with *error* set to 8, operation not supported.