



IEEE 488

APPLICATION
BULLETIN

Preventing Communication Loss and Resource Exhaustion with ICS's Ethernet Interfaces

INTRODUCTION

There are several ways that a user can lose communication with an ICS Ethernet interface. The causes vary from miss-set timeouts, resource exhaustion and even anti-virus network scans by your friendly IT people.

This Application Note describes how a user can properly setup an ICS Ethernet interface to minimize communication loss.

CAUSES OF COMMUNICATION LOSS

There are several major causes of communication loss that this application note seeks to address. The first is an unexpected closure of the socket in the ICS Ethernet Interface. You have not done anything for a while and then, when you try to send a command you get a communication error.

Second is resource exhaustion. Here, the simple act of opening a VISA oriented application, uses up all of the sockets in the ICS Ethernet Interface, forcing you to either wait for COMM_timeout to recover the sockets or power cycle the Ethernet Interface to restore communication.

The third cause can be one of several things, a server changing the IP address, the network IT guy doing a routine virus scan or another application that accesses your device and accidentally sends it commands or setting locks that interferes with or blocks your communication.

ICS ETHERNET INTERFACE SETTINGS

All ICS Ethernet Interfaces have three parameters that can be set to avoid some of these problems. KeepAlive, COMM_timeout and Auto Disconnect. ICS Ethernet Interfaces operate as a server and the controlling PC runs the client application.

KeepAlive operates at the IP level and checks for a valid connection to the client PC if there has not been any communication between them for a 2 hour period. The test is like a ping to verify the connection. If the connection is good, KeepAlive goes to sleep for another 2 hours. If the connection is broken, KeepAlive closes the socket in the ICS Ethernet Interface and recovers all resources associated with that socket so they can be used by another client. KeepAlive can only be set On or Off. The 2 hour time period is not adjustable.

COMM_timeout is a user adjustable time period that closes the socket and frees all resources associated with the socket if there has not been any messages from the client for the user set time period. COMM_timeout operates faster than KeepAlive to give you a quicker recovery time. COMM_timeout is factory set to two minutes but it can be increased to cover expected periods of inactivity such as lunch breaks etc. but, with an increased risk of running out of sockets.

With COMM_timeout, there is always a balance between not wanting it to prematurely close sockets and yet for it to quickly recover orphaned sockets. The best way to do this is to put a background task in the client application that periodically does some link activity if the link has not been used in a while. A one minute interval works fine for the background task. Set COMM_timeout to 2 x the task interval. Let the task do something that will not affect your setup such as opening/closing a link or writing to the ESE Register.

Auto Disconnect looks at the link count and immediately closes the socket when the link count goes to zero. Client software like VISA can scan the network looking for Ethernet devices when it starts up. When an Ethernet device is found, the client software (VISA) tests to find all inst personalities in the Ethernet Interface by rapidly opening and closing links and sockets. Windows operating systems can take 10s of seconds to gracefully close a socket while VISA can open the next socket in a millisecond or two. ICS's Ethernet Interfaces have only 15 sockets, so VISA or

a similar scanning software can quickly exhaust all of the interface's sockets before Windows can close the first socket. Auto Disconnect lets the Ethernet Interface sense the link closure and then immediately close the socket, preventing socket exhaustion. Auto Disconnect can only be set On or Off.

Auto Disconnect is not desirable in all applications. In some cases, the client application might close a link and then open a link to another personality in the device or in the case of a GPIB gateway, link to a different instrument. Automatically closing the socket and losing communication with the Ethernet Interface would not be desirable in this case.

RECOMMENDED CONNECTION

The best connection for an Ethernet Interface is to be on a private network with the controlling PC as shown in Figure 1. This eliminates all external influences and simplifies the setup. Making a private network is easily done by adding an inexpensive NIC card to the PC or by making the PC a stand alone computer that is not connected to the company network.

The recommended settings for a private network are KeepAlive on and COMM_timeout set to a reasonably short period. COMM_timeout is factory set to 2 minutes to quickly recover sockets that are left open during program development. With a working program, COMM_timeout can be extended to cover periods of inactivity such as lunch breaks etc., but, again with the increased risk of socket exhaustion. An alternative to extending COMM_timeout is to add a background thread to the application that performs some periodic activity on the link once a minute if no messages have been sent. Some examples of the link activity routine are shown in the Appendix.

Auto Disconnect should be set on if you are using VISA or Agilent IO Libraries. Else it can be left off.

Figure 2 shows the Ethernet Interface connected to the company network. Sometimes this connection is necessary to accommodate

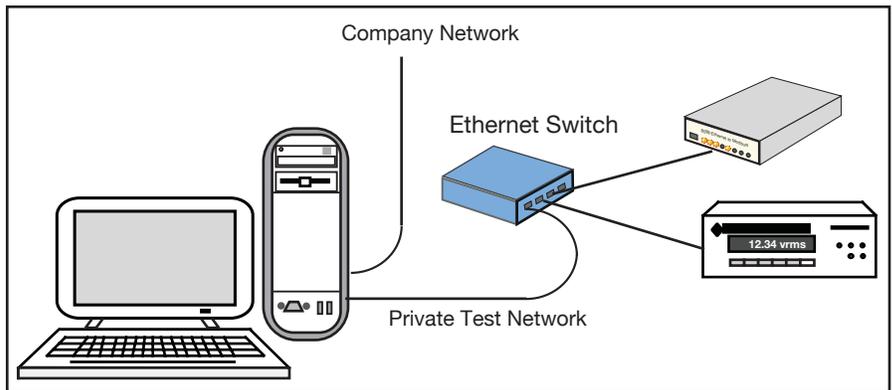


Figure 1 Test instruments connected on a private network

the different physical locations of the controlling PC and the test instruments. This can be a sub-net in the manufacturing area, between buildings or even over the external Internet to a remote location. In any case, this type of connection opens up the Ethernet Instruments to interference from 3rd party applications, possible outsiders and to internally generated problems such as periodic network scans.

With this type of connection, KeepAlive and Auto Disconnect should be set on. COMM_timeout should be kept relatively short and having a background task in the application programs is highly recommended. Appendix 1 list some example routines.

Be aware of other problems that can occur when connected to the company network. Another client can access your Ethernet Interface at the same time you are and either read your response message or change the instrument setup. Use the VXI locking mechanism to protect an atomic operation such as a query to assure that you are getting the correct responses.

If the Ethernet Interface seems to die at the same time each day or at the same time each week, talk with your IT person. Its possible he is running some network test or monitor program like a virus scan that can cause the Ethernet Interface to stop communicating. He can arrange to block the Ethernet Interface from being scanned.

Add error checking to your program. If you get a connection error, have your program reconnect to the Ethernet Interface and retry the failed instruction. Verify command operation with queries or by checking the ESR Register for errors.

SUMMARY

This Application Note has described the three parameters in ICS's Ethernet Interfaces that can be used to avoid communication loss and has provided recommendations on how to set these parameters for private test networks and company network connections.

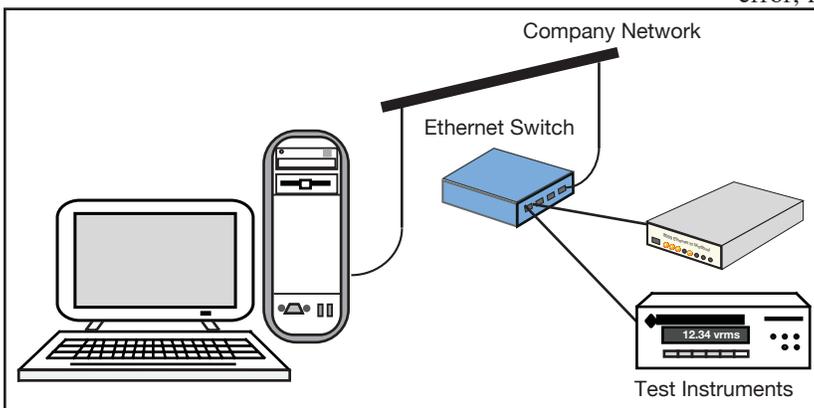


Figure 2 Test instruments connected on the company network

APPENDIX 1 LINK EXERCISE ROUTINES

VISA EXAMPLE

```
Private Sub Timer2_Timer() 'exercise the link by locking nd unlocking the link
    Dim result&

    If MsgSentFlg = 0 Then '0 needs a keep alive exercise
        If udlink& <> 0 Then 'exercise a link lock
            lockType& = VI_EXCLUSIVE_LOCK
            locktimeout& = 30000 '30 seconds
            requestedKey$ = VI_NULL
            accessKey$ = VI_NULL
            status = viLock&(udlink&, lockType&, locktimeout&, requestedKey$, accessKey$)
            If (status < VI_SUCCESS) Then GoTo VisaErrorHandler
            lblKeepAliveMsg.Visible = True 'blinks a label to show link exercised
            lblKeepAliveMsg.Refresh
            status = viUnlock(udlink&)
            If (status < VI_SUCCESS) Then GoTo VisaErrorHandler
            Call Tdelay(0.1)
            lblKeepAliveMsg.Visible = False '
            lblKeepAliveMsg.Refresh
        End If
    End If
End Sub

GoTo Timer2exit:

VisaErrorHandler:
Call viStatusDesc(defrm, status, strVisaErr)
txtError.Text = "*** VISA Error : " + strVisaErr + Str$(vbExclamation)
txtError.Visible = True
Beep
Timer2exit:
MsgSentFlg = 0 'clears flag - link exercised
End Sub
```