How Can I Trace The Path Of A Network Packet?

Overview

A lot of people really don't understand how networks work or how their data gets from their computer to another computer and back again. This article will discuss how data packets move through a network and why you should understand how your network is setup and how it works.

What is a network?

A network literally defined is a "system of computers, terminals, and databases connected by communications lines." A network is a group of computers than exchange data between themselves using some sort of method for communication. Networks used to normaly be connect via IPX using cable lines attached directly to each computer with a terminator at the end. Now Ethernet networks are the most dominant as well as the

addition of the WiFi connection methods for wireless access. To get data from one point of the network to another there needs to exist some sort of rules for the transmission of data over the physical layer.

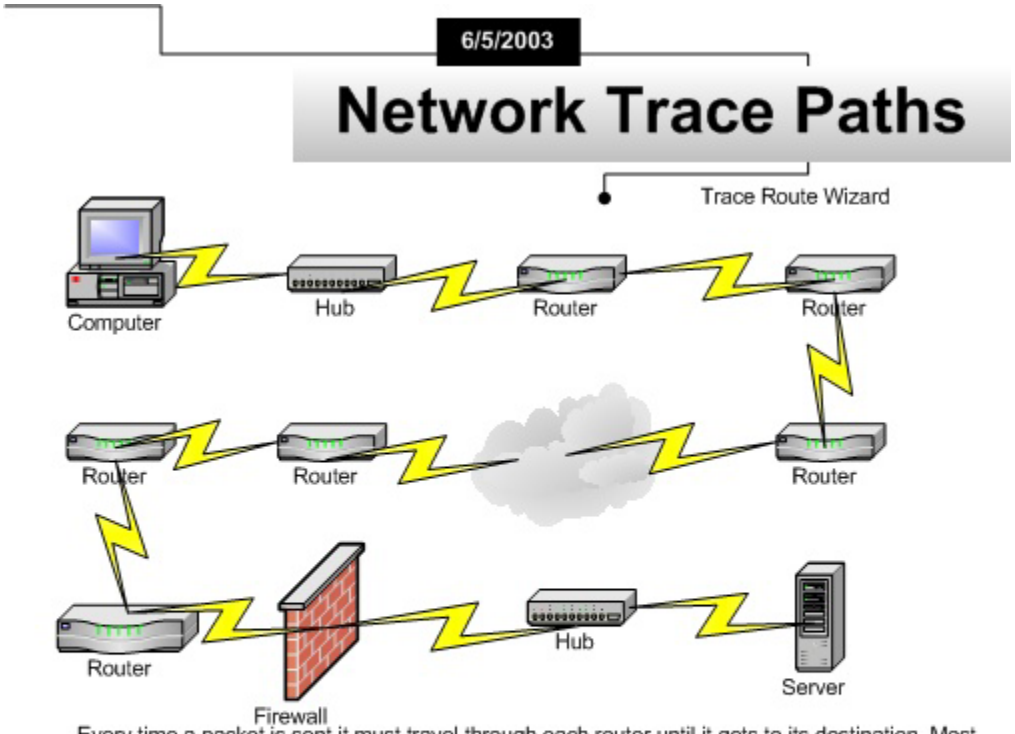
How does a network work?

Hardware handles the difficult task of transmitting the bits and information of the communication mediums but their still is the need for the data to leave the local network of computers and connect with all the other networks. This is currently done using the IP protocol for data communication. Each computer is setup to have a unique ID number that is used for identification of a particular machine. The IP packets are formatted with the ID number of the target, the sender, and also the port number that it is destined to go to. Ports are different points at which a client can connect which allows a single machine to accept multiple requests from multiple clients at the same time. The IP packets are first sent across the local network to see if that network ID, or IP address, is located on the local network. If not, then it is up to a piece of hardware called a router to take over the sending of the packet.

What is a router?

A router is a piece of hardware with a complicated piece of software called firmware that handles all the hard work of moving packets of data where they need to go. The router looks at packets and decides if they need to leave the local network. If it thinks they belong to an IP address somewhere else they attempt to deliver it by sending it to the next router in the path to the recipient until the last router in the path can directly deliver the message. Every time a packet has to go through a router, the time it takes to reach its target is increased. If a router is overloaded by a lot of traffic, they can actually drop, or discard, packets. It is then up to the higher level communications protocol to resend the packet again. This makes it more obvious that the more points your packet has to go through to get to the final destination, the longer it will take due to slow routers, overloaded routers, or the mere processing time of the routers.

Network Path Diagram



Every time a packet is sent it must travel through each router until it gets to its destination. Most ISP's user other ISP's for their internet connections. This shows you that not only do your packets bounce through your ISP's routers, they also bounce through a lot of others to reach the host. This can cause network lag since the time for the packet to reach each host can take quit a while.

Trace Route Wizard lets you see each HOP, or router, that your packets travel through. This is great for diagnostics and other such purposes.

Seekford Solutions, Inc.

http://www.SeekfordSolutions.com

How can I trace the path?

There is really only one good way to trace the path of a packet and that is to trick each router on the path the packet takes into thinking it needs to tell the sender there was an error in communication. The best and most common way is to attempt to ping the target host's IP address using and incremental Time to Live identifier in the IP header. The TTL or Time to Live identification is used to tell routers how many times the packet can be routed before it should just be thrown away and discarded. Every router is considered a hop, and every hop the TTL value is decremented by one. Once it reaches 0 the router that has it discards the packet and then sends an ICMP control message back to the caller telling it that the packet timed out and did not reach the destination. Some routers are not courteous enough to respond with the ICMP control message; these routers are called "black hole router" due to the fact packets go in but may not come out. To find out each router a packet goes through and effectively determine the number of hops it must take requires a simplistic algorithm of sending a special packet with a TTL of 1 and incrementing it to N which is the value TTL is when you reach the target. This algorithmic process is called a Trace Route. A program that comes with Windows called TraceRt.exe is very familiar to a lot of network administrators because it does exactly what was just described. Developers can add this functionality to their programs using an ActiveX control called TraceRoute Wizard.

What is TraceRoute Wizard?

TraceRoute Wizard is an ActiveX control that enables developers to conduct Trace Routes using their network to determine the number of hops a packet must take. This is great for determining which routers are performing poorly, the number of hops a packet is taking out your network and through an ISP's network and more. It is also good for developers because they can show why their program may not be communicating as quickly as they would like due to network congestion. You could give the IT department a list of all the routers your communicating through and show them that they need to upgrade their network or change ISP's. The control provides the Return Transmit Time for each packet and the IP addresses and host names for each hop the packet takes to its destination.

Can I see Sample code:

[Visual Basic]

```
txtTraceData.Text = "Tracing route to " & txtTraceAddress.Text & vbCrLf & "over a maximum of " & CStr(TraceRouteWizard1.MaxHops) & "
hops:" & vbCrLf & vbCrLf
 txtTraceData.Text = txtTraceData.Text & "Hop" & vbTab & "#1" & vbTab & "#2" & vbTab & "#3" & vbTab & "Host IP/Name" & vbCrLf
 txtTraceData.Refresh
 TraceRouteWizard1.ResolveIPsToHostNames = CBool(chkResolveNumericIP)
 TraceRouteWizard1.HopTimeOut = CInt(txtTimeout)
 TraceRouteWizard1.PacketSize = CInt(txtPacketSize)
 UsingStepTrace = True
 Dim i As Integer
 Dim x As Integer
 Dim sAddress As String
 If (TraceRouteWizard1.StepTraceTo(txtTraceAddress.Text)) Then
   Do
     i = i + 1
     txtTraceData.Text = txtTraceData.Text & CStr(i) & vbTab
     sAddress = ""
     txtTraceData.Refresh
     DoEvents
     If (TraceRouteWizard1.HopSuccessful(i)) Then
       For x = 1 To TraceRouteWizard1.HopAttempts
        txtTraceData.Text = txtTraceData.Text & IIf(TraceRouteWizard1.HopAttemptSuccessful(i, x), TraceRouteWizard1.HopRTT(i, x), "---") & "
ms" & vbTab
        If Len(TraceRouteWizard1.HopAddress(i, x)) > 0 Then
         sAddress = TraceRouteWizard1.HopAddressName(i, x) & " [" & TraceRouteWizard1.HopAddress(i, x) & "]"
        End If
       Next
       txtTraceData.Text = txtTraceData.Text & sAddress & vbCrLf
     Else
       txtTraceData.Text = txtTraceData.Text & TraceRouteWizard1.HopLastErrorDescription(i) & vbCrLf
     End If
   Loop Until Not (TraceRouteWizard1.StepTraceNext)
  Else
```

```
   txtTraceData.Text = txtTraceData.Text & "1" & vbTab & TraceRouteWizard1.LastErrorDescription & vbCrLf
 End If
 txtTraceData.Text = txtTraceData.Text & vbCrLf & "Trace complete." & vbCrLf
 UsingStepTrace = False
```

[VBScript]

```
function IIF(a,b,c)
        if a then
                IIF = b
        else
                IIF = c
        end if
end function

        dim sIP
        dim MyTraceRouteWizard
        dim bResolveHostNames
        dim sResults



        'create the object
        set MyTraceRouteWizard  = CreateObject("TRACEROUTEWIZARD.TraceRouteWizardCtrl2")

        sIP = InputBox("Please enter the host to trace: i.e. www.SeekfordSolutions.com or 192.168.0.2","Host to Trace Path To",
"www.SeekfordSolutions.com")
        bResolveHostNames = InputBox("Do you want to resolve IP addresses into Hostnames? This may take a lot longer. Enter Y to resolve,
otherwise enter anything else.","Resolve Host Names?","Y")

        if ( bResolveHostNames = "Y") then
          MyTraceRouteWizard.ResolveIPsToHostNames = TRUE
         else
```

```
        MyTraceRouteWizard.ResolveIPsToHostNames = FALSE
      end if
      msgbox "We are going to Trace now, Please be patient. This could take up to a minute or so depending on network conditions and whether
you chose to Resolve Host Names"
      MyTraceRouteWizard.UnlockTraceRouteWizard("")

      Dim sData,bIsOk,iIndex,iPacketsSent ,iPacketsReceived ,iPacketsLost ,dAveragems ,iMinms ,iMaxms
      Dim i
       Dim x
      dim sAddress
       If (MyTraceRouteWizard.StepTraceTo(sIP)) Then
        Do
         i = i + 1
         sResults = sresults & CStr(i) & vbTab
         sAddress = ""
         If (MyTraceRouteWizard.HopSuccessful(i)) Then
           For x = 1 To MyTraceRouteWizard.HopAttempts
            sResults = sresults & IIf(MyTraceRouteWizard.HopAttemptSuccessful(i, x), MyTraceRouteWizard.HopRTT(i, x), "---") & " ms" &
vbTab
            If Len(MyTraceRouteWizard.HopAddress(i, x)) > 0 Then
             sAddress = MyTraceRouteWizard.HopAddressName(i, x) & " [" & MyTraceRouteWizard.HopAddress(i, x) & "]"
            End If
           Next
      sResults = sresults & sAddress & vbCrLf
          Else
              sResults = sresults & MyTraceRouteWizard.HopLastErrorDescription(i) & vbCrLf
         End If
        Loop Until Not (MyTraceRouteWizard.StepTraceNext)
       Else
         sResults = sresults & "1" & vbTab & MyTraceRouteWizard.LastErrorDescription & vbCrLf
       End If
    sResults = sresults & vbCrLf & "Trace complete. Please visit www.SeekfordSolutions.com to Purchase!" & vbCrLf
      msgbox sResults
```

[Visual FoxPro]

```
local vbCRLF
local vbTab
THISFORM.UsingStepTrace = .t.
vbCRLF = chr(10)
vbTab = chr(9)

 THISFORM.txtTraceData.value = "Tracing route to " + TRIM(THISFORM.txtTraceAddress.value) + vbCrLf + "over a maximum of " +
Str(THISFORM.TraceRouteWizard.MaxHops) + " hops:" + vbCrLf + vbCrLf
 THISFORM.txtTraceData.value = THISFORM.txtTraceData.value + "Hop" + vbTab + "#1" + vbTab + "#2" + vbTab + "#3" + vbTab + "Host
IP/Name" + vbCrLf
 THISFORM.txtTraceData.Refresh
 THISFORM.TraceRouteWizard.ResolveIPsToHostNames = (THISFORM.chkResolve.value =1)
 THISFORM.TraceRouteWizard.HopTimeOut =  VAL(STR(THISFORM.txtTimeout.value))
 THISFORM.TraceRouteWizard.PacketSize = VAL(STR(THISFORM.txtPacketSize.value))
 THISFORM.TraceRouteWizard.MaxHops = VAL(STR(THISFORM.txtMaxHops.value))
 THISFORM.TraceRouteWizard.HopAttempts = VAL(STR(THISFORM.txtHopAttempts.value))
 THISFORM.UsingStepTrace = .t.
 local i
 local x
 local sAddress
 local bGoAgain
 bGoAgain = .t.
 sAddress = ""
 i = 0
 x = 0
 If (THISFORM.TraceRouteWizard.StepTraceTo(TRIM(THISFORM.txtTraceAddress.value))) Then
   Do while (bGoAgain)
     i = i + 1
     THISFORM.txtTraceData.value = THISFORM.txtTraceData.value + STR(i) + vbTab
     sAddress = ""
     THISFORM.txtTraceData.Refresh
     DoEvents
```

```
    If (THISFORM.TraceRouteWizard.HopSuccessful(i)) Then
      For x = 1 To THISFORM.TraceRouteWizard.HopAttempts
       if(THISFORM.TraceRouteWizard.HopAttemptSuccessful(i, x)) then
                THISFORM.txtTraceData.value = THISFORM.txtTraceData.value +  STR(THISFORM.TraceRouteWizard.HopRTT(i, x)) + " ms" +
vbTab
              else
                    THISFORM.txtTraceData.value = THISFORM.txtTraceData.value +  "---" + " ms" + vbTab
            endif
       If Len(THISFORM.TraceRouteWizard.HopAddress(i, x)) > 0 Then
        sAddress = THISFORM.TraceRouteWizard.HopAddressName(i, x) + " [" + THISFORM.TraceRouteWizard.HopAddress(i, x) + "]"
       EndIf
      Next
      THISFORM.txtTraceData.value = THISFORM.txtTraceData.value + sAddress + vbCrLf
     Else
      THISFORM.txtTraceData.value = THISFORM.txtTraceData.value + THISFORM.TraceRouteWizard.HopLastErrorDescription(i) + vbCrLf
    EndIf
    bGoAgain = THISFORM.TraceRouteWizard.StepTraceNext
  enddo
 Else
   THISFORM.txtTraceData.value = THISFORM.txtTraceData.value + "1" + vbTab + THISFORM.TraceRouteWizard.LastErrorDescription +
vbCrLf
 EndIf
 THISFORM.txtTraceData.value = THISFORM.txtTraceData.value + vbCrLf + "Trace complete." + vbCrLf
 THISFORM.UsingStepTrace = .f.
```

[Visual C++]

```
        UpdateData(TRUE);
        m_bStepTracing= TRUE;
        m_TraceRouteWizard.UnlockTraceRouteWizard("");
        m_sReplyData.Format("Tracing route to %s\r\nover a maximum of %d hops:\r\n\r\n",m_sAddress,m_iMaxHops);
        UpdateData(FALSE);
        m_TraceRouteWizard.SetResolveIPsToHostNames(m_bResolve);
        m_TraceRouteWizard.SetHopTimeOut(m_iTimeout);
```

```
	m_TraceRouteWizard.SetPacketSize(m_iPacketSize);
	m_TraceRouteWizard.SetMaxHops(m_iMaxHops);
	m_TraceRouteWizard.SetHopAttempts(m_iHopAttempts);
	int iHopNumber =0;
	int x = 0;
	CString sAddress;
	if(m_TraceRouteWizard.StepTraceTo(m_sAddress))
	{
		do
		{
			iHopNumber++;
			m_sReplyData.Format("%s%d\t",CString(m_sReplyData),iHopNumber);
			RedrawWindow();
			Sleep(0);
			if (m_TraceRouteWizard.GetHopSuccessful(iHopNumber))
			{
				for(x = 1; x <= m_TraceRouteWizard.GetHopAttempts(); x++)
				{
					if(m_TraceRouteWizard.GetHopAttemptSuccessful(iHopNumber,x))
					{
						m_sReplyData.Format("%s%d\t",CString(m_sReplyData),
m_TraceRouteWizard.GetHopRTT(iHopNumber,x));
						sAddress.Format("%s
[%s]",m_TraceRouteWizard.GetHopAddressName(iHopNumber,x),m_TraceRouteWizard.GetHopAddress(iHopNumber,x));
					}else
					{
						m_sReplyData += CString("---\t");
					}
				}
				m_sReplyData.Format("%s%s\r\n",CString(m_sReplyData),sAddress);
			}else
			{
```

```
        m_sReplyData.Format("%s%s\r\n",CString(m_sReplyData),m_TraceRouteWizard.GetHopLastErrorDescription(iHopNumber));
                }
                UpdateData(FALSE);
        }
        while(m_TraceRouteWizard.StepTraceNext());
}else
{
        m_sReplyData.Format("%s1\t%s\r\n",CString(m_sReplyData),m_TraceRouteWizard.GetLastErrorDescription());
}

m_sReplyData += CString("\r\nTrace complete.\r\n");
UpdateData(FALSE);
RedrawWindow();
m_bStepTracing= FALSE;
```

[Active Server Pages]

```
if request.form("p_Address") <> "" then
        set MyTraceRouteWizard  = Server.CreateObject("TRACEROUTEWIZARD.TraceRouteWizardCtrl2")
        Response.write("Trace Route Address:""" + request.form("p_Address") + """" + vbcrlf)
        Response.write("Resolve Host names:""" + request.form("p_ResolveHostnames") + """" + vbcrlf)
        if ( request.form("p_ResolveHostNames") = "TRUE") then
          MyTraceRouteWizard.ResolveIPsToHostNames = TRUE
         else
          MyTraceRouteWizard.ResolveIPsToHostNames = FALSE
        end if
        Response.write("<HR>")
        '////////////////////////////////////////////////////////////
        '//
        '//        This component will only work in ASP once it has been purchased.
        '//
        '////////////////////////////////////////////////////////////
        MyTraceRouteWizard.UnlockTraceRouteWizard("")
```

```
        Response.flush
        Dim sData,bIsOk,iIndex,iPacketsSent ,iPacketsReceived ,iPacketsLost ,dAveragems ,iMinms ,iMaxms
        Dim i
         Dim x
         Dim sAddress
         If (MyTraceRouteWizard.StepTraceTo(request.form("p_Address"))) Then
           Do
            i = i + 1
            Response.write( CStr(i) & vbTab)
            sAddress = ""
            If (MyTraceRouteWizard.HopSuccessful(i)) Then
              For x = 1 To MyTraceRouteWizard.HopAttempts
               Response.write(IIf(MyTraceRouteWizard.HopAttemptSuccessful(i, x), MyTraceRouteWizard.HopRTT(i, x), "---") & " ms" & vbTab)
               If Len(MyTraceRouteWizard.HopAddress(i, x)) > 0 Then
                sAddress = MyTraceRouteWizard.HopAddressName(i, x) & " [" & MyTraceRouteWizard.HopAddress(i, x) & "]"
               End If
              Next
              Response.write( sAddress & vbCrLf)
             Else
              Response.write( MyTraceRouteWizard.HopLastErrorDescription(i) & vbCrLf)
            End If
           Loop Until Not (MyTraceRouteWizard.StepTraceNext)
          Else
            Response.Write( "1" & vbTab & MyTraceRouteWizard.LastErrorDescription & vbCrLf)
         End If
         Reponse.Write( vbCrLf & "Trace complete." & vbCrLf)
    else
        Response.write("To trace a route to a host, please enter one below and click submit.")
end if
```

Where can I find TraceRoute Wizard?

TraceRoute Wizard is publicly available from Seekford Solutions, Inc. on their website @ http://www.SeekfordSolutions.com/Products/

Who is Seekford Solutions, Inc.?

Seekford Solutions, Inc. is a software development corporation specializing in the design and development of state of the art ActiveX controls and custom projects. Their core product line is focused on Internet technologies primarily in the facilitation of the use of the common Internet protocols. The design philosophy is based on ease of use and quick implementation time. They also handle custom projects for clients who are need of specialty software or who need a framework base to use. The company was founded in early 2001. Their website is http://www.SeekfordSolutions.com.