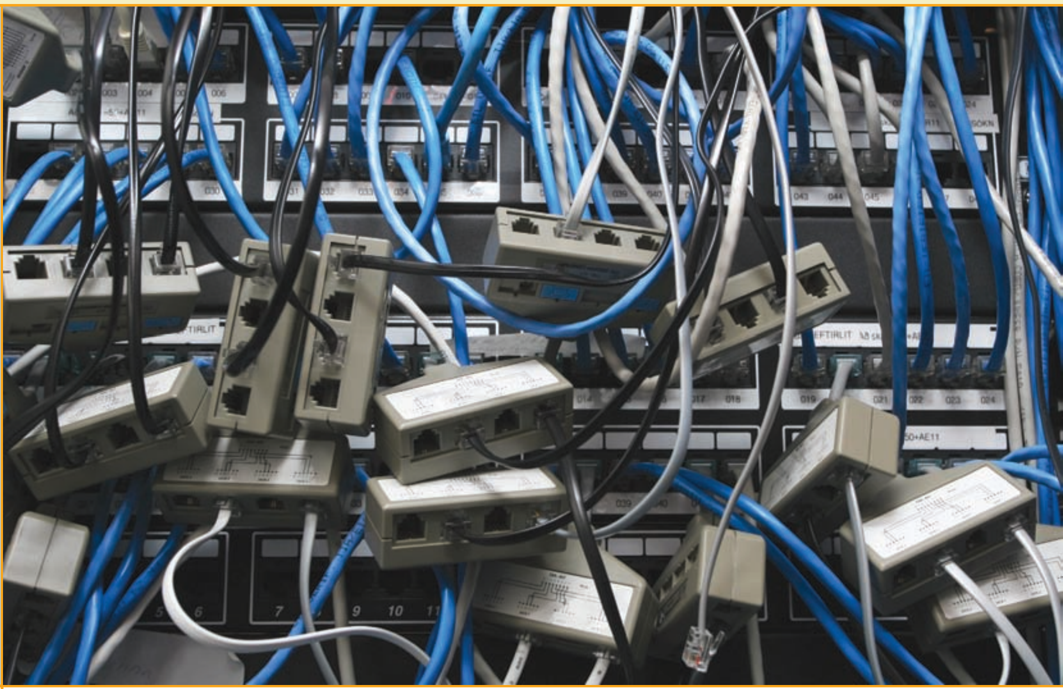
chapter 15

Network Troubleshooting

"The trouble with doing something right the first time is that nobody appreciates how difficult it was."

—Walt West



In this chapter, you will learn how to

- **Describe appropriate troubleshooting tools and their functions**
- **Analyze and discuss the troubleshooting process**
- **Tackle a variety of troubleshooting scenarios**

Have you ever seen a tech walk up to a network and seem to know all the answers, effortlessly typing in a few commands and magically making the system or network work? I've always been intrigued by how they do this. Observing such techs over the years, I've noticed that they tend to follow the same steps for similar problems—looking in the same places, typing the same commands, and so on. When someone performs a task the same way every time, I figure they're probably following a plan. They understand what tools they have to work with, and they know where to start and what to do second and third and fourth until they find the problem. This chapter's lofty goal is to consolidate my observations on how these "übertechs" fix networks. We'll look at the primary troubleshooting tools, formulate a troubleshooting process, and learn where to look for different sorts of problems. At the end of the chapter, we'll apply this knowledge to some common troubleshooting scenarios.

Test Specific

Troubleshooting Tools

While working through the process of finding the cause of a problem, you sometimes need tools. These are the software and hardware tools that provide information about your network and enact repairs. We covered a number of tools already: hardware tools like cable testers and crimpers plus software utilities like PING and TRACERT. The trick is knowing when and how to use these tools to solve your network problems.

No matter what the problem, always consider the safety of your data first. Ask yourself this question before you perform any troubleshooting action: "Can what I'm about to do potentially damage my data?"

Almost every new networking person I teach will at some point ask me: "What tools do I need to buy?" My answer shocks them: "None. Don't buy a thing." It's not so much that you don't need tools but more that your different networking jobs require wildly different tools. Plenty of network techs never crimp a cable. An equal number never open a system. Some techs do nothing all day but pull cable. The tools you need are defined by your job. You'll know by the end of the first day what you'll need.

This answer is especially true with software tools. Almost all the network problems I encounter in established networks don't require me to use any tools other than the classic ones provided by the operating system. I've fixed more network problems with PING than with any other single tool. As you gain skill in this area you'll find yourself pounded by vendors selling you the latest, greatest networking diagnostic tools. You may like these tools. All I can say is that I've never needed a software diagnostics tool that I had to purchase.

Hardware Tools

In multiple chapters in this book you've read about a few hardware tools you use when configuring your network. These **hardware tools** include cable testers, TDRs, OTDRs, certifiers, voltage event recorders, protocol analyzers, cable strippers, multimeters, tone probes/generators, butt sets, and punchdown tools. Some of these tools can also be used in troubleshooting scenarios to help you eliminate or narrow down the possible causes of certain problems. Let's review the tools as listed in the *CompTIA Network+ Exam Objectives*.

Read this section! The CompTIA Network+ exam is filled with repair scenarios and you must know what every tool does and when to use it.

Cable Testers, TDRs, and OTDRs

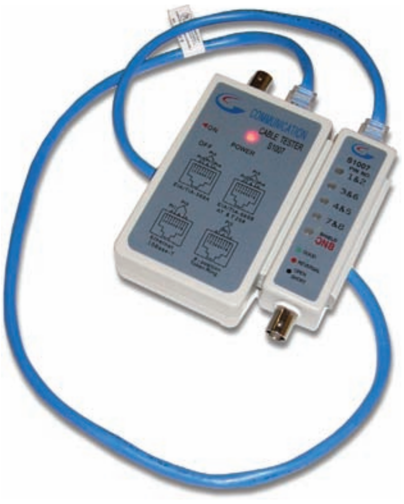
The vast majority of cabling problems take place when the network is first installed or when a change takes place. Once a cable has been made, installed, and tested, the chances of it failing are pretty small compared to all of the other network problems that might take place. Imagine what happens when you can't connect to a resource and ask yourself, "Is there a chance the cable is bad?" Broken cables don't make intermittent problems, and they don't slow down data: they make permanent disconnects.

Network techs define a "broken" cable in numerous ways. First, a broken cable might have an *open circuit*, where one or more of the wires in a cable simply don't connect from one end of the cable to the other. The signal lacks *continuity*. Second, a cable might have a *short*, where one or more of the

The CompTIA Network+ objectives use the term "open impedance mismatch" (echo) at the time of this writing. There should be a comma after "open" so you know you're looking at two different types of broken cable issues.

wires in a cable connect to another wire in the cable. Within a normal cable, none of the wires connect to another wire. Third, a cable might have a *wire map problem*, where one or more of the wires in a cable don't connect to the proper location on the jack or plug. This can be caused by improperly crimping a cable, for example. Fourth, the cable might experience *crosstalk*, where the electrical signal bleeds from one wire pair to another, creating interference. Fifth, a broken cable might pick up *noise*, spurious signals usually caused by faulty hardware or poorly crimped jacks. Finally, a broken cable might have *impedance mismatch*. Impedance is the natural electrical resistance of a cable. When cables of different types—think thickness, composition of the metal, and so on—connect and the flow of electrons is not uniform, this can cause a unique type of electrical noise, called an *echo*.

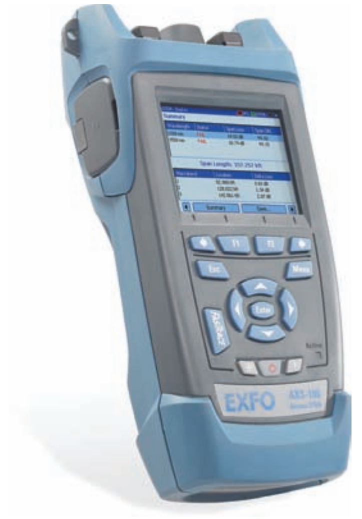
Network technicians use three different devices to deal with broken cables. **Cable testers** can tell you if there's a continuity problem or if a wire map isn't correct (Figure 15.1). Time domain reflectometers (TDRs) and optical time domain reflectometers (OTDRs) can tell you where a break is on the cable (Figure 15.2). A TDR works with copper cables and an OTDR works with fiber optics, but otherwise they share the same function. If a problem shows itself as a disconnect and you've first checked easier issues that would manifest as disconnects, such as loss of permissions, an unplugged cable, or a server shut off, then think about using these tools.



• Figure 15.1 Typical cable tester

Certifiers

Certifiers test a cable to ensure that it can handle its rated amount of capacity. When a cable is not broken but it's not moving data the way it should, you turn to a certifier. Look for problems that cause a cable to underperform. A bad installation might increase crosstalk, excess attenuation, or interference. A certifier can pick up an impedance mismatch as well. Most of these problems show up at installation but it's never a bad idea to run a certifier to eliminate cabling as a problem. Don't use a certifier for disconnects, only slowdowns. Last, all certifiers need some kind of loopback on the other end of the cable run.



• Figure 15.2 An EXFO AXS-100 OTDR (photo courtesy of EXFO)

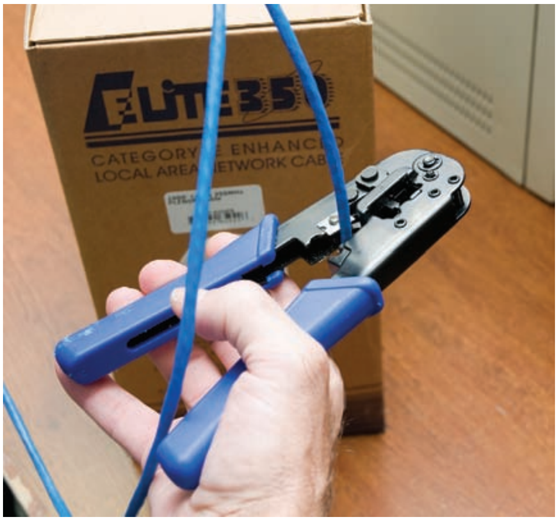
Voltage Event Recorder/Temperature Monitor

Networks need the proper temperature and adequate power, but most network techs tend to view these issues as outside of the normal places to look for problems. That's too bad, because both heat and power problems invariably manifest themselves as intermittent problems. Look for problems that might point to heat or power issues: server rooms that get too hot at certain times of the day, switches that fail whenever an air conditioning system kicks on, and so on. You can use a **voltage event recorder** and a **temperature monitor** to monitor server rooms over time to detect and record issues with electricity or heat, respectively. They're great for those "something happened last night" types of issues.

Protocol Analyzers

Protocol analyzers monitor the different protocols running at different layers on the network. A good protocol analyzer will give you Application, Session, Network, and Data Link layer information on every frame going through your network. Even though the CompTIA Network+ exam places protocol analyzers in a hardware category, they aren't necessarily always hardware. Some of the best and most useful protocol analyzers are software.

Use a protocol analyzer when being able to see the data on the network will help you answer a question. Is something trying to start a session and not getting an answer? Maybe a DNS server isn't responding. Is some computer on the network placing confusing information on the network. Is a rogue DHCP server sending out responses to DHCP requests? In the same vein, a protocol analyzer helps you determine slowdowns on a network by giving you an idea of excess or unexpected traffic (see the "Packet Sniffer" subsection under "Software Tools" later in this chapter).



• Figure 15.3 A cable stripping and crimping tool

Cable Strippers/Snips

A **cable stripper** or **snip** (Figure 15.3) enables you to make UTP cables. Even though the CompTIA Network+ competencies don't mention crimpers, don't forget you'll need them too. You don't need these tools to punch down 66- or 110-blocks. You would use a punchdown tool for that.

Multimeters

Multimeters test voltage (AC and DC), resistance, and continuity. They are the unsung heroes of cabling infrastructures because no other tool can tell you how much voltage is on a line. They are also a great fallback for continuity testing when you don't have a cable tester handy.

Tone Probes and Tone Generators

Tone probes and their partners, **tone generators**, have only one job: to help you locate a particular cable. You'll never use a tone probe without a tone generator.

Butt Sets

Butt sets are the telephone person's best friend. Use a butt set to tap into a 66- or 110-block to see if a particular line is working.

Punchdown Tools

Punchdown tools (Figure 15.4) put UTP wires into 66- and 110-blocks. The only time you would use a punchdown tool in a diagnostic environment is a quick repunch of a connection to make sure all the contacts are properly set.

Try This!

Shopping Spree

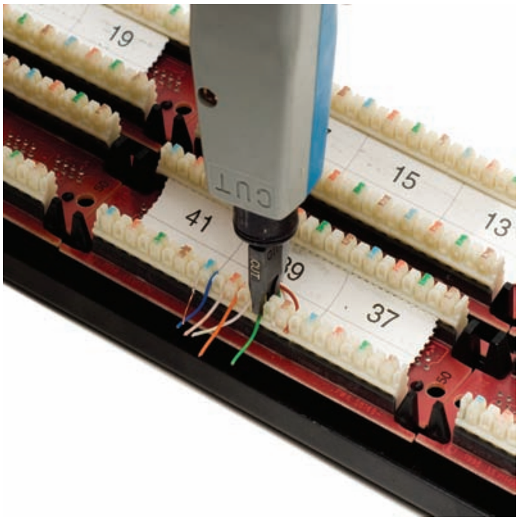
As more and more people have networks installed in their homes, the big box hardware stores stock an increasing number of network-specific tools. Everybody loves shopping, right? So try this! Go to your local hardware store—big box, like Home Depot or Lowes, if there's one near you—and check out their tools. What do they offer? Write down prices and features and compare with what your classmates found.

Tech Tip

Never Buy Cheap Tools

There's an old adage used by carpenters and other craftsmen that goes, "Never buy cheap tools." Cheap tools save you money at the beginning, but they often break more readily than higher-quality tools and, more importantly, make it harder to get the job done. This adage definitely applies to multimeters! There's always a temptation to go for the $10 model that looks pretty much like the $25 model, but chances are the leads will break or the readings will lie on the cheaper model. Buy a decent one and you'll never have to worry about it.

CompTIA and many techs refer to the probe as a toner probe rather than a tone probe or simply a probe. Don't be surprised on the exam. You always need both a probe and a tone generator to use this tool properly.



• **Figure 15.4** A punchdown tool in action

The CompTIA Network+ exam tests your ability to recognize the output from all of the built-in tools. Take some time to memorize example outputs on all of these tools.

Software Tools

Make the CompTIA Network+ exam (and real life) easier by separating your software tools into two groups: those that come built into every operating system and those that are third-party tools. Typical built-in tools are TRACERT/TRACEROUTE, IPCONFIG/IFCONFIG, PING, ARP PING, NSLOOKUP/DIG, HOSTNAME, ROUTE, NBTSTAT, and NETSTAT. Third-party tools fall into the categories of packet sniffers and port scanners.

TRACERT/TRACEROUTE

TRACEROUTE (Windows calls it **TRACERT**) is used to trace all of the routers between two points. We use TRACERT when we are having problems reaching a remote system, to diagnose where the problem lies. If a TRACEROUTE stops at a certain router, you know the problem is either the next router or the connections between them. TRACEROUTE isn't perfect, because many routers block TRACERT packets. Try PATHPING when you're not sure whether TRACEROUTE is working properly.

Here's sample TRACEROUTE output:

```
Tracing route to adsl-208-190-121-38.dsl.hstntx.swbell.net
[208.190.121.38]
over a maximum of 30 hops:

  1     1 ms    <1 ms     1 ms  Router.totalhome
[192.168.4.1]
  2    38 ms    41 ms    70 ms  adsl-208-190-121-
38.dsl.hstntx.swbell.net [208.190.121.38]
```

Try This!

Playing Along

This section contains many command-line tools that you've seen earlier in the book in various places. Now is a great time to refresh your memory about how each one works, so try this! After I review each command, run it yourself. Then type `help` followed by the command to see the available switches for that command. Run some to see what they do. It's more fun to do it than just to read about it; plus it'll help solidify the knowledge you need to master.

IPCONFIG/IFCONFIG

IPCONFIG (Windows) and **IFCONFIG** (everyone else) tells you anything you want to know about a particular computer's IP settings. Make sure you know that typing `ipconfig` alone only gives basic information. Typing `ipconfig /all` gives detailed information (like DNS servers and MAC addresses).

Here's sample IPCONFIG output:

```
Ethernet adapter Main:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:470:bf88:1:fc2d:aeb2:99d2:e2b4
   Temporary IPv6 Address. . . . . . : 2001:470:bf88:1:5e4:c1ef:7b30:ddd6
   Link-local IPv6 Address . . . . . : fe80::fc2d:aeb2:99d2:e2b4%8
   IPv4 Address. . . . . . . . . . . : 192.168.4.27
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::223:4ff:fe8c:b720%8
                                       192.168.4.1

Tunnel adapter Local Area Connection* 6:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

And here's sample IFCONFIG output:

```
eth0      Link encap:Ethernet  HWaddr 00:02:b3:8a:7d:ae
          inet addr:192.168.4.19  Bcast:192.168.4.255  Mask:255.255.255.0
          inet6 addr: 2001:470:bf88:1:202:b3ff:fe8a:7dae/64 Scope:Global
          inet6 addr: fe80::202:b3ff:fe8a:7dae/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2206320 errors:0 dropped:0 overruns:0 frame:0
          TX packets:925034 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:292522698 (292.5 MB)  TX bytes:132985596 (132.9 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:15414 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15414 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1006671 (1.0 MB)  TX bytes:1006671 (1.0 MB)
```

PING and ARP PING

You can use **PING** to see if you can reach another system. Whereas TRACEROUTE tells you about each hop to a connection, PING only tells you if it can reach the other system. PING is the most heavily used networking tool. PING uses ICMP packets, and some devices block ICMP.

If PING doesn't work you can try an **ARP PING**. An ARP PING uses the ARP command instead of ICMP. The only downside to ARP is that ARPs do not cross routers, so you can only use it within a broadcast domain. Windows does not have ARP PING. UNIX and UNIX-like systems use the ARPING utility to perform an ARP PING.

Here's sample PING output:

```
Pinging 192.168.4.19 with 32 bytes of data:
Reply from 192.168.4.19: bytes=32 time<1ms TTL=64
Reply from 192.168.4.19: bytes=32 time<1ms TTL=64
Reply from 192.168.4.19: bytes=32 time<1ms TTL=64
Reply from 192.168.4.19: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.4.19:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Next is sample ARPING output:

```
ARPING 192.168.4.27 from 192.168.4.19 eth0
Unicast reply from 192.168.4.27 [00:1D:60:DD:92:C6]  0.875ms
Unicast reply from 192.168.4.27 [00:1D:60:DD:92:C6]  0.897ms
Unicast reply from 192.168.4.27 [00:1D:60:DD:92:C6]  0.924ms
Unicast reply from 192.168.4.27 [00:1D:60:DD:92:C6]  0.977ms
```

The PING command has the word "pinging" in the output. The ARPING command has the word "ARPING." Don't assume that the CompTIA Network+ exam will include those words in its sample outputs.

NSLOOKUP/DIG

NSLOOKUP (all operating systems) and **DIG** (everyone but Windows) are used to diagnose DNS problems. These are very powerful tools, but the CompTIA Network+ exam won't ask you more than basic questions, such as how to use them to see if a DNS server is working. NSLOOKUP is a poor tool that most everyone considers obsolete. DIG is far more powerful. The DIG example below shows the output of this command:

```
dig mx totalsem.com
```

This command says, "Show me all the MX records for the totalsem.com domain."

Here's the output for that DIG command:

```
; <<>> DiG 9.5.0-P2 <<>> mx totalsem.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6070
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
;; QUESTION SECTION:
;totalsem.com.                   IN      MX
;; ANSWER SECTION:
totalsem.com.           86400    IN      MX      10
mx1c1.megamailservers.com.
```

```
totalsem.com.           86400    IN      MX      100
mx2c1.megamailservers.com.
totalsem.com.           86400    IN      MX      110
mx3c1.megamailservers.com.
```

Running the networking commands several times will help you memorize the functions of the commands as well as the syntax. The CompTIA Network+ exam is big on the switches available for various commands, such as `ipconfig /all`.

HOSTNAME

HOSTNAME is the simplest of all the utilities shown here. When you run it, it returns with the host name of the computer you are on. Here's what it looked like when I ran it on my Windows 7 box:

```
C:\>
C:\>hostname
mike-win7beta
```

MTR

My Traceroute (MTR) is a dynamic (keeps running) equivalent to TRACEROUTE. MTR is not available on Windows.

Here's a sample of MTR output:

```
                             My traceroute  [v0.73]
totaltest (0.0.0.0)
Keys:  Help   Display mode   Restart statistics   Order of
fields   quit
                                       Packets
Pings
 Host                                Loss%   Snt   Last   Avg
Best  Wrst StDev
 1. Router.totalhome                  0.0%     5    0.8   0.8
0.7   0.9   0.1
 2. adsl-208-190-121-38.dsl.hstntx.s  0.0%     4   85.7  90.7
69.5 119.2  20.8
```

ROUTE

The **ROUTE** command gives you the capability to display and edit the local system's routing table. To show the routing table, just type `route print`.

Here's a sample of ROUTE PRINT output:

```
===========================================================================
Interface List
  8 ...00 1d 60 dd 92 c6 ...... Marvell 88E8056 PCI-E Ethernet Controller
  1 ........................... Software Loopback Interface 1
===========================================================================
IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.4.1     192.168.4.27     10
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    306
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    306
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    306
      169.254.0.0      255.255.0.0         On-link      192.168.4.27    286
  169.254.214.185  255.255.255.255         On-link   169.254.214.185    276
  169.254.255.255  255.255.255.255         On-link      192.168.4.27    266
      192.168.4.0    255.255.255.0         On-link      192.168.4.27    266
     192.168.4.27  255.255.255.255         On-link      192.168.4.27    266
    192.168.4.255  255.255.255.255         On-link      192.168.4.27    266
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    306
        224.0.0.0        240.0.0.0         On-link   169.254.214.185    276
        224.0.0.0        240.0.0.0         On-link      192.168.4.27    266
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    306
  255.255.255.255  255.255.255.255         On-link   169.254.214.185    276
  255.255.255.255  255.255.255.255         On-link      192.168.4.27    266
===========================================================================
Persistent Routes:
  None
```

NBTSTAT

NBTSTAT is a Windows-only program that can best be described as a command-line equivalent to Window's My Network Places or Network icon. NBTSTAT must run using a switch. The most useful switch is `-n`, which shows the local NetBIOS names. NBTSTAT is in all versions of Windows through Vista. NBTSTAT is a handy way to see what systems are running on your Windows network. Any systems running SAMBA will also appear here.

Here's an example of running `nbtstat -n` from the command prompt:

```
Main:
Node IpAddress: [192.168.4.27] Scope Id: []
                NetBIOS Local Name Table
       Name                Type         Status
    ---------------------------------------------
    MIKESPC        <00>  UNIQUE      Registered
    TOTALHOME      <00>  GROUP       Registered
    MIKESPC        <20>  UNIQUE      Registered
    TOTALHOME      <1E>  GROUP       Registered
```

NETSTAT

NETSTAT is a very handy tool that displays information on the current state of all of your running IP processes. It shows what sessions are active and can also provide statistics based on ports or protocols (TCP, UDP, and so on). Typing `netstat` by itself only shows current sessions. Typing `netstat -r` shows the routing table (virtually identical to ROUTE PRINT). If you want to know about your current sessions, NETSTAT is the tool to use.

Here's sample NETSTAT output:

```
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:27015        MikesPC:51090          ESTABLISHED
  TCP    127.0.0.1:51090        MikesPC:27015          ESTABLISHED
  TCP    127.0.0.1:52500        MikesPC:52501          ESTABLISHED
  TCP    192.168.4.27:54731     72-165-61-141:27039    CLOSE_WAIT
  TCP    192.168.4.27:55080     63-246-140-18:http     CLOSE_WAIT
  TCP    192.168.4.27:56126     acd4129913:https       ESTABLISHED
  TCP    192.168.4.27:62727     TOTALTEST:ssh          ESTABLISHED
  TCP    192.168.4.27:63325     65.54.165.136:https    TIME_WAIT
  TCP    192.168.4.27:63968     209.8.115.129:http     ESTABLISHED
```

Packet Sniffer

Packet sniffer, protocol analyzer, or packet analyzer: all of these names are used to define a tool that intercepts and logs network packets. You have many choices when it comes to packet sniffers. Some sniffers come as

Try This!

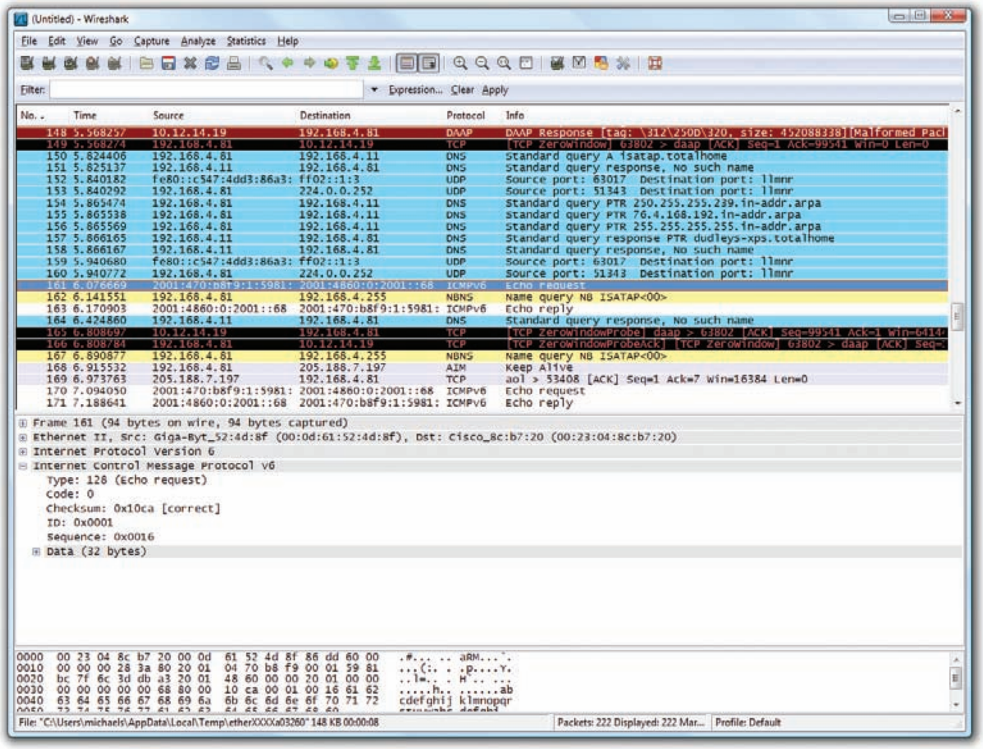
Download and Run Wireshark

You can get the latest version of Wireshark at www.wireshark.org, so fire up your favorite Web browser and try this program! Download a copy of the application and run it on a networked computer. Don't forget to look at the extensive documentation on the Web site to get a sense of all the coolness that is Wireshark.

programs you run on a computer, while others manifest as dedicated hardware devices. Arguably, the most popular (and also mentioned in the CompTIA Network+ objectives) is **Wireshark** (Figure 15.5). You've already seen Wireshark in the book, but here's a shot to jog your memory.

Port Scanners

A **port scanner** is a program that probes ports on another system, logging the state of the scanned ports. These tools are used to look for unintentionally opened ports that might make a system vulnerable to attack. As you might imagine, they also are used by hackers to break into systems. The most famous of all port scanners is probably the powerful and free NMAP. NMAP is designed to work on UNIX systems, so Windows folks tend to look for alternatives like Angry IP Scanner by Anton Keks (Figure 15.6).



(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

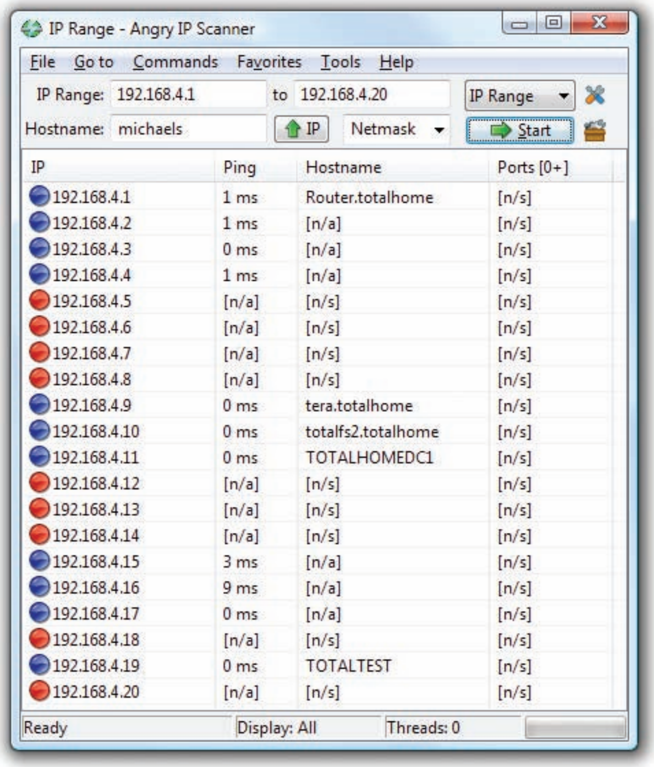
```
No. .  Time      Source                    Destination               Protocol  Info
  148 5.568257  10.12.14.19               192.168.4.81              DAAP      DAAP Response [tag: \312\250D\320, size: 452088338][Malformed Pacl
  149 5.568274  192.168.4.81              10.12.14.19               TCP       [TCP ZeroWindow] 63802 > daap [ACK] Seq=1 Ack=99541 Win=0 Len=0
  150 5.824406  192.168.4.81              192.168.4.11              DNS       Standard query A isatap.totalhome
  151 5.825137  192.168.4.11              192.168.4.81              DNS       Standard query response, No such name
  152 5.840182  fe80::c547:4dd3:86a3:    ff02::1:3                 UDP       Source port: 63017  Destination port: llmnr
  153 5.840292  192.168.4.81              224.0.0.252               UDP       Source port: 51343  Destination port: llmnr
  154 5.865474  192.168.4.81              192.168.4.11              DNS       Standard query PTR 250.255.255.239.in-addr.arpa
  155 5.865538  192.168.4.81              192.168.4.11              DNS       Standard query PTR 76.4.168.192.in-addr.arpa
  156 5.865569  192.168.4.81              192.168.4.11              DNS       Standard query PTR 255.255.255.255.in-addr.arpa
  157 5.866165  192.168.4.11              192.168.4.81              DNS       Standard query response PTR dudleys-xps.totalhome
  158 5.866167  192.168.4.11              192.168.4.81              DNS       Standard query response, No such name
  159 5.940680  fe80::c547:4dd3:86a3:    ff02::1:3                 UDP       Source port: 63017  Destination port: llmnr
  160 5.940772  192.168.4.81              224.0.0.252               UDP       Source port: 51343  Destination port: llmnr
  161 6.076669  2001:470:b8f9:1:5981:    2001:4860:0:2001::68      ICMPv6    Echo request
  162 6.141551  192.168.4.81              192.168.4.255             NBNS      Name query NB ISATAP<00>
  163 6.170903  2001:4860:0:2001::68      2001:470:b8f9:1:5981:     ICMPv6    Echo reply
  164 6.424860  192.168.4.11              192.168.4.81              DNS       Standard query response, No such name
  165 6.808697  10.12.14.19               192.168.4.81              TCP       [TCP ZeroWindowProbe] daap > 63802 [ACK] Seq=99541 Ack=1 Win=6414
  166 6.808784  192.168.4.81              10.12.14.19               TCP       [TCP ZeroWindowProbeAck] [TCP ZeroWindow] 63802 > daap [ACK] Seq=
  167 6.890877  192.168.4.81              192.168.4.255             NBNS      Name query NB ISATAP<00>
  168 6.915532  192.168.4.81              205.188.7.197             AIM       Keep Alive
  169 6.973763  205.188.7.197             192.168.4.81              TCP       aol > 53408 [ACK] Seq=1 Ack=7 Win=16384 Len=0
  170 7.094050  2001:470:b8f9:1:5981:    2001:4860:0:2001::68      ICMPv6    Echo request
  171 7.188641  2001:4860:0:2001::68      2001:470:b8f9:1:5981:     ICMPv6    Echo reply

⊞ Frame 161 (94 bytes on wire, 94 bytes captured)
⊞ Ethernet II, Src: Giga-Byt_52:4d:8f (00:0d:61:52:4d:8f), Dst: Cisco_8c:b7:20 (00:23:04:8c:b7:20)
⊞ Internet Protocol Version 6
⊟ Internet Control Message Protocol v6
    Type: 128 (Echo request)
    Code: 0
    Checksum: 0x10ca [correct]
    ID: 0x0001
    Sequence: 0x0016
  ⊞ Data (32 bytes)

0000  00 23 04 8c b7 20 00 0d  61 52 4d 8f 86 dd 60 00   .#... .. aRM...`.
0010  00 00 00 28 3a 80 20 01  04 70 b8 f9 00 01 59 81   ...(:. . .p....Y.
0020  bc 7f 6c 3d db a3 20 01  48 60 00 00 20 01 00 00   ..l=.. . H`.. ...
0030  00 00 00 00 00 68 80 00  10 ca 00 01 00 16 61 62   .....h.. ......ab
0040  63 64 65 66 67 68 69 6a  6b 6c 6d 6e 6f 70 71 72   cdefghij klmnopqr

File: "C:\Users\michaels\AppData\Local\Temp\etherXXXXa03260" 148 KB 00:00:08   Packets: 222 Displayed: 222 Mar...   Profile: Default
```

• Figure 15.5 Wireshark in action



IP Range: 192.168.4.1 to 192.168.4.20

Hostname: michaels

IP	Ping	Hostname	Ports [0+]
🔵 192.168.4.1	1 ms	Router.totalhome	[n/s]
🔵 192.168.4.2	1 ms	[n/a]	[n/s]
🔵 192.168.4.3	0 ms	[n/a]	[n/s]
🔵 192.168.4.4	1 ms	[n/a]	[n/s]
🔴 192.168.4.5	[n/a]	[n/s]	[n/s]
🔴 192.168.4.6	[n/a]	[n/s]	[n/s]
🔴 192.168.4.7	[n/a]	[n/s]	[n/s]
🔴 192.168.4.8	[n/a]	[n/s]	[n/s]
🔵 192.168.4.9	0 ms	tera.totalhome	[n/s]
🔵 192.168.4.10	0 ms	totalfs2.totalhome	[n/s]
🔵 192.168.4.11	0 ms	TOTALHOMEDC1	[n/s]
🔴 192.168.4.12	[n/a]	[n/s]	[n/s]
🔴 192.168.4.13	[n/a]	[n/s]	[n/s]
🔴 192.168.4.14	[n/a]	[n/s]	[n/s]
🔵 192.168.4.15	3 ms	[n/a]	[n/s]
🔵 192.168.4.16	9 ms	[n/a]	[n/s]
🔵 192.168.4.17	0 ms	[n/a]	[n/s]
🔴 192.168.4.18	[n/a]	[n/s]	[n/s]
🔵 192.168.4.19	0 ms	TOTALTEST	[n/s]
🔴 192.168.4.20	[n/a]	[n/s]	[n/s]

Ready | Display: All | Threads: 0

• Figure 15.6 Angry IP Scanner

The Troubleshooting Process

Troubleshooting is a dynamic, fluid process that requires you to make snap judgments and act on them to try and make the network go. Any attempt to cover every possible scenario would be futile at best, and probably also not in your best interests, because any reference that tried to list every troubleshooting problem would be obsolete the moment it was created. If an exhaustive listing of all network problems is impossible, then how do you decide what to do and in what order?

Before you touch a single console or cable, you should remember two basic rules: To paraphrase the Hippocratic Oath, "First, do no harm." If at all possible, don't make a network problem bigger than it was originally. This is a rule I've broken thousands of times, and you will too. But if we change the good doctor's phrase a bit, it's possible to formulate a rule you can live with: "First, do not trash the data!" My gosh, if I had a dollar for every megabyte of irreplaceable data I've destroyed, I'd be rich! I've learned my lesson, and you should learn from my mistakes. The second rule is: Always make good backups! Computers can be replaced; data that is not backed up is gone forever.

No matter how complex and fancy, any troubleshooting process can be broken down into simple steps. Having a sequence of steps to follow makes the entire troubleshooting process simpler and easier, because you have a clear set of goals to achieve in a specific sequence. The most important steps are the first three—they help you narrow down the cause of the problem to a specific item. The reason this matters so much is that figuring out what's wrong will also probably tell you how to fix the problem, and how to prevent it from happening in the future.

The basics of any troubleshooting process should include the following steps:

1. Gather information—identify symptoms and problems.
2. Identify the affected areas of the network.
3. Establish if anything has changed.
4. Establish the most probable cause.
5. Determine if escalation is necessary.
6. Create an action plan and solution identifying potential effects.
7. Implement and test the solution.
8. Identify the results and effects of the solution.
9. Document the solution and the entire process.

Gather Information—Identify Symptoms and Problems

If you are working directly on the affected system and not relying on somebody on the other end of a telephone to guide you, you will establish the symptoms through your observation of what is (or isn't) happening. If you're troubleshooting over the telephone (always a joy, in my experience), you will need to ask questions based on what the user is telling you. These questions can be *close-ended*, which is to say there can only be a yes or no type answer, such as, "Can you see a light on the front of the monitor?" You can also ask *open-ended* questions, such as, "Tell me what you see on the screen." The type of question you use at any given moment will depend on what information you need, and on the knowledge level of the user. If, for example, the user seems to be technically oriented, you will probably be able to ask more close-ended questions, because they will know what you are talking about. If, on the other hand, the user seems to be confused about what's happening, open-ended questions will allow him to explain in his own words what is going on.

Identify the Affected Areas of the Network

One of the first steps in trying to determine the cause of a problem is to understand the extent of the problem—is it specific to one user or is it network-wide? Sometimes this entails trying the task yourself, both from the user's machine and from your own or another machine.

For example, if a user is experiencing problems logging into the network, you might need to go to that user's machine and try to use their user name to log in. This will tell you whether the problem is a user error of some kind, as well as enable you to see the symptoms of the problem yourself. Next, you probably want to try logging in with your own user name from that machine, or have the user try to log in from another machine. In some cases, you can ask other users in the area if they are experiencing the same problem, to see if the problem is affecting more than one user. Depending on the size of your network, you should find out whether the problem is occurring in only one part of your company or across the entire network.

Eliminating variables is one of the first tools in your arsenal of diagnostic techniques.

What does all of this tell you? Essentially, it tells you how big the problem is. If nobody in an entire remote office can log in, you may be able to assume that the problem is the network link or router connecting that office to the server. If nobody in any office can log in, you may be able to assume that the server is down or not accepting logins. If only that one user in that one location can't log in, it may be a problem with that user, that machine, or that user's account.

Establish if Anything Has Changed

The goal of this step is to identify if anything has changed that might have caused the problem. You may not have to ask many questions before the people using the problem system can tell you what has changed, but in some cases establishing if anything has changed can sometimes take quite a bit of time and involve further work behind the scenes. Here are some examples of questions to ask:

- "Tell me exactly what you were doing when the problem occurred."
- "Has anything been changed on the system recently?"
- "Has the system been moved recently?"

Notice the way I've tactfully avoided the word *you*, as in "Have *you* changed anything on the system recently?" This is a deliberate tactic to avoid any implied blame on the part of the user. Being nice never hurts, and it makes the whole troubleshooting process more friendly.

Avoid aggressive or accusatory questions.

You should be asking some isolating questions *internally* of yourself, such as, "Was that machine involved in the software push last night?" or "Didn't a tech visit that machine this morning?" Note that you will only be able to answer these questions if *your* documentation is up to scratch. Sometimes, isolating a problem may require you to check system and hardware logs (such as those stored by some routers and other network devices), so make sure you know how to do this.

Identify the Most Probable Cause

This step comes down to experience—or good use of the support tools at your disposal, such as your knowledge base. You need to select the most *probable* cause from all the *possible* causes, so the solution you choose fixes the problem the first time. This may not always happen, but whenever possible, you want to avoid spending a whole day stabbing in the dark while the problem snores softly to itself in some cozy, neglected corner of your network.

Determine if Escalation Is Necessary

Escalation has two meanings: either to inform other parties about a problem for guidance or to pass the job off to another authority who has control over the device/issue that's most probably causing the problem. Let's say you have a server with a bad NIC. This server is used heavily by the accounting department and taking it down may cause problems you don't even know about. You need to inform the boss of accounting to consult with them. Alternatively, there are problems over which you have no control or authority. A badly acting server across the country (hopefully) has another person in charge to whom you need to hand over the job.

Create an Action Plan and Solution Identifying Potential Effects

By this point you should have some ideas as to what the problem might be. It's time to "look before you leap." An action plan defines how you are going to fix this problem. Most problems are simple, but if the problem is complex you need to write down the steps. As you do this, think about what else might happen as you go about the repair. If you take out a switch, will the users all stop working? Maybe adding the users to a new switch first is a good idea. If you replace a router, can you restore all the old router's settings to the new one or will you have to rebuild from scratch?

Implement and Test a Solution

Once you think you have isolated the cause of the problem, you should decide what you think is the best way to fix it, and then try your solution, whether that's giving advice over the phone to a user, installing a replacement part, or adding a software patch. All the way through this step, try only one likely solution at a time. There's no point in installing several patches at once, because then you can't tell which one fixed the problem. Similarly, there's no point in replacing several items of hardware (such as a hard disk and its controller cable) at the same time, because then you can't tell which part (or parts) was faulty.

As you try each possibility, always *document* what you do and what results you get. This isn't just for a future problem, either—during a lengthy troubleshooting process, it's easy to forget exactly what you tried two hours before, or which thing you tried produced a particular result. Although being methodical may take longer, it will save time the next time—and it may enable you to pinpoint what needs to be done to stop the problem from recurring at all, thereby reducing future call volume to your support team—and as any support person will tell you, that's definitely worth the effort!

Then it's time to test. This is the part everybody hates. Once you think you've fixed a problem, you should try to make it happen again. If you can't, great! But sometimes you will be able to re-create the problem, and then you know you haven't finished the job at hand. Many techs want to slide away quietly as soon as everything seems to be fine, but trust me on this, it won't impress your customer when their problem flares up again 30 seconds after you've left the building—not to mention that you get the joy of another two-hour car trip the next day to fix the same problem, for an even more unhappy client! In the scenario where you are providing support to someone else rather than working directly on the problem, you should make *him* try to re-create the problem. This will confirm whether he understands what

Always test a solution before you walk away from the job!

you have been telling him, and will educate him at the same time, lessening the chance that he'll call you back later and ask, "Can we just go through that one more time?"

Identify the Results and Effects of the Solution

Okay, now that *you* have changed something on the system in the process of solving one problem, you must think about the wider repercussions of what you have done. If you've replaced a faulty NIC in a server, for instance, will the fact that the MAC address has changed (remember, it's built into the NIC) affect anything else, such as the logon security controls, or your network management and inventory software? If you've installed a patch on a client PC, will this change the default protocol or any other default settings that may affect other functionality? If you've changed a user's security settings, will this affect their ability to access other network resources? This is part of testing your solution to make sure it works properly, but it also makes you think about the impact of your work on the system as a whole.

Memorize these problem analysis steps:

1. Information gathering – identify symptoms and problems.
2. Identify the affected areas of the network.
3. Establish if anything has changed.
4. Establish the most probable cause.
5. Determine if escalation is necessary.
6. Create an action plan and solution identifying potential effects.
7. Implement and test the solution.
8. Identify the results and effects of the solution.
9. Document the solution and the entire process.

Document the Solution and the Entire Process

It is *vital* that you document the problem, symptoms, and solutions of all support calls, for two reasons. First, you're creating a support database that will be a knowledge base for future reference, enabling everyone on the support team to identify new problems as they arise, and know how to deal with them quickly, without having to duplicate someone else's research efforts. Second, documentation enables you to track problem trends and anticipate future workloads, or even to identify a particular brand or model of an item, such as a printer or a NIC, that seems to be less reliable or that creates more work for you than others. Don't skip this step—it *really* is essential!

■ Troubleshooting Scenarios

I want to end this chapter with some good troubleshooting scenarios. Take some time and think about these situations and how you would handle them. What questions would you ask? What tests would you do first? The CompTIA Network+ exam absolutely *loves* to ask scenario questions. The knowledge from the previous chapters combined with the methods you've learned in this chapter should enable you to fix any network!

"I Can't Log In!"

One of the most complex troubleshooting issues is that one set of symptoms, in this case a user's inability to log in, can have many causes. Suppose Woody has called complaining that he cannot log into the company's intranet. Tina Tech first tries accessing the intranet site from her system, and finds she has no problem. Tina might also want to have other users try to log in, or confirm that other users are not having the same problem. Next, Tina should have Woody try to log in from another machine. This will help Tina determine whether the problem lies with Woody's user account's capability to log in, with Woody's system, or with some connectivity issue.

If Woody is unable to log in from another machine, Tina should probably check to be sure Woody is using the correct login ID, password, and procedure when he logs in. On the other hand, if Woody can log in from another user's system, Tina should probably focus on determining whether Woody's system is working properly and connecting to the network. One step she could try here is to PING Woody's system. If Tina can PING Woody's machine successfully, she knows that the machine is up, the TCP/IP protocol is configured correctly, and the system is connected to the network. Tina might then check the configuration of the network client on Woody's system. If Tina is not able to PING the system, however, she might need to test the cables and NIC using cable testers or loopback devices, and verify that TCP/IP was correctly configured using IPCONFIG.

"I Can't Get to This Web Site!"

Reaching external Web sites requires that a variety of components be configured correctly. Some of these components are within your company's internal control; many of them are not. When Fatima calls and tells Tina Tech that she cannot reach www.comptia.org, Tina's first step is to try to reach that site herself. In this case, Tina was also unable to get a response from the comptia.org site. One of her next steps is to PING the site, first by name, and then by IP address. In this case, she gets no response by name, but she does get a normal response when she PINGs the site by IP address. This immediately indicates to her that the problem is name resolution, in this case, DNS.

On the other hand, had Tina been unable to PING successfully using either the IP address or host name, she should consider two possibilities. First, if her company uses a firewall or proxy server to reach the Internet, she should PING that machine. This machine usually has the same IP address as the default gateway TCP/IP setting. If Tina can successfully PING her default gateway, she can be almost certain that the problem is not something she or her company has any control over. To verify this, Tina should attempt to reach some other external sites, both by using PING and a Web browser. If she can reach other sites successfully, the problem is most likely with the comptia.org site or the gateway.

✓ **Cross Check**

DNS Settings

You learned about DNS in detail way back in Chapter 10, "Network Naming," so dust off those memories and see if you can answer these questions. What might cause a DNS server to go down? What's a DNS root server? What are the top-level domain servers? Does DNS use a flat name space or a hierarchical name space? What's the difference?

"Our Web Server Is Sluggish!"

Slow response from a server can be related to a variety of things. Usually, however, the problem can be traced to a connection to the server, or to the server itself. When Wanda calls in from working at home and tells Tina Tech that she is getting a slow response from the company's Web site, Tina Tech leaps into action. Tina tries to reach the offending server and is immediately connected; this indicates a connectivity problem for that user. She asks Wanda to execute a `tracert` command from her system to the slow server. This reveals to Tina that the slowdown stems from one of the intermediate steps through which Wanda's system connects to the server. Because of this, the problem is out of Tina's hands, unless she can offer a direct dial-up option for Wanda.

If Tina finds she cannot reach the offending server quickly when she tries from her system, then the problem may lie with the server itself. Tina checks the Change Log for the Web server, to see if anyone has changed anything recently. She discovers that a new antivirus component was recently added, so she checks the vendor's Web site to make sure there are no known problems or patches for that piece of software. She also uses Performance Monitor to compare the server's current responses to the baseline that she previously recorded. This shows her that the bottleneck is related to excessive paging, indicating that the server may need more physical memory, or RAM.

"I Can't See Anything on the Network!"

When a user is completely cut off from the network, the problem is usually limited to that user's system or network connection. When Tina gets a call from Johnny saying his Windows machine is on, but that he can't log in and can't see any other machines on the company's TCP/IP network, Tina goes to Johnny's office to run some tests. The first test Tina runs is to PING an external machine. She doesn't expect it to work, but tests just to be certain. Next, she tries to PING Johnny's machine using either `ping localhost` or `ping 127.0.0.1` (remember the loopback address?). When this PING doesn't work, Tina guesses that the problem is in the TCP/IP configuration. To view the machine's TCP/IP configuration, Tina uses IPCONFIG, and notices the IP address is blank. After checking her network documentation to verify what IP address Johnny's machine should have, she adds the IP address and he is able to connect to the network.

If Tina's `ping 127.0.0.1` had worked, she would have had to assume the TCP/IP and networking configuration of Johnny's machine was correct. She should then check the hardware, using a network card utility to verify that the NIC itself is working correctly, and a cable tester to verify that the cable from Johnny's system is operating properly. In this case, the cable tester shows that the cable is bad, so she replaces the cable between Johnny's system and the patch panel, and he is able to connect.

"It's Time to Escalate!"

No single person is truly in control of an entire Internet-connected network. Large organizations split network support duties into very skill-specific areas: routers, cable infrastructure, user administration, and so on. Even in a tiny network with a single network support person, problems will arise that go beyond the skill level of the tech or that involve equipment they don't own (usually it's their ISP's gear). In these situations it becomes the tech's job to identify a problem and, instead of trying to fix it on his or her own, escalate the issue.

In network troubleshooting, problem escalation should occur when you face a problem that falls outside the scope of your skills and you need help. In large organization, escalation problems have very clear procedures, such as who to call and what to document. In small organizations, escalation often is nothing more than a technician realizing that he or she needs help. The CompTIA Network+ competencies define some classic networking situations that CompTIA feels should be escalated. Here's how to recognize broadcast storms, switching loops, route problems, routing loops, and proxy ARP.

Broadcast Storms

A **broadcast storm** is the result of one or more devices sending a non-stop flurry of broadcast frames on the network. The first sign of a broadcast storm is when every computer on the broadcast domain suddenly can't connect to the rest of the network. There are usually no clues other than network applications freezing or presenting "can't connect to…" types of error messages. Every activity light on every node is solidly on. Computers on other broadcast domains work perfectly well.

The trick is to isolate; that's where escalation comes in. You need to break down the network quickly by unplugging devices until you can find the one causing trouble. It's usually difficult to get a packet analyzer to work, but at least try. If you can scoop up one packet you'll know what node is causing the trouble. The second the bad node is disconnected, the network returns to normal. But if you have a lot of machines to deal with and a bunch of users who can't get on the network yelling at you, you need help. Call a supervisor to get support to solve the crisis as quickly as possible.

Switching Loops

A **switching loop** is when you connect multiple switches together to cause a loop to appear. Switching loops are rare because most switches use spanning tree protocol, but they do happen. The symptoms are identical to a broadcast storm: every computer on the broadcast domain can no longer access the network.

The good part about switching loops is that they rarely take place on a well-running network. Someone had to create that loop and that means someone, somewhere is messing with patch cables. Escalate the problem and get the team to help you find the person making changes to the switches.

Route Problems

Improperly configured routers aren't going to send packets to the proper destination. The symptoms are clear: every system that uses the misconfigured router as a default gateway is either not able to get packets out or get packets in, or sometimes both. Web pages don't come up, FTP servers suddenly disappear, and e-mail clients can't access their servers. In these cases you need to verify first that everything under your responsibility works. If that is true, then escalate the problem and find the person responsible for the router.

Routing Loops

A **routing loop** occurs when interconnected routers loop traffic, causing the routers to respond slowly or not respond at all. Dynamic routing protocols sometimes cause routing loops when a router goes down, but most routing loops are caused by static routes. Your big clue is a huge amount of traffic—far more than your usual traffic—on the links between the routers. Router loops never cause individual computers to stop responding (unless they happen to be on the same broadcast domain as the looping packets). Like with any route problem, be able to recognize the symptoms and escalate.

Proxy ARP

Proxy ARP is the process of making remotely-connected computers truly act as though they are on the same LAN as local computers. Proxy ARPs are done in a number of different ways, with a Virtual Private Network (VPN) as the classic example. If a laptop in an airport connects to a network through a VPN, that computer takes on the network ID of your local network. In order for all this to work, the VPN concentrator needs to allow some very LAN type traffic go through it that would normally never get through a router. ARP is a great example. If your VPN client wants to talk to another computer on the LAN, it has to ARP to get the IP address. Your VPN device is designed to act as a proxy for all that type of data.

Almost all proxy ARP problems take place on the VPN concentrator. With misconfigured proxy ARP settings, the VPN concentrator can send what looks like a Denial of Service (DoS) attack on the LAN. (A DoS attack is usually directed at a server exposed on the Internet, like a Web server. See Chapter 17, “Protecting Your Network,” for more details on these and other malicious attacks.) If your clients start receiving a large number of packets from the VPN concentrator, assume it is a proxy ARP problem and escalate by getting the person in charge of the VPN to fix it.

Troubleshooting Is Fun!

The art of network troubleshooting can be a fun, frolicsome, and frequently frustrating feature of your network career. By applying a good troubleshooting methodology and constantly increasing your knowledge of networks, you too can develop into a great troubleshooting artist. This takes time, naturally, but stick with it. Begin the training. Use the Force. Learn new stuff, document problems and fixes, talk to other network techs about similar problems. Every bit of knowledge and experience you gain will make things that much easier for you when crunch time comes and a network disaster occurs—and as any experienced network tech can tell you, it will, even in the most robust network.

Chapter 15 Review

Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about network troubleshooting.

Describe appropriate troubleshooting tools and their functions

- Before starting work on any problem, always ask yourself if what you are about to do can potentially harm your data.
- The vast majority of cabling problems take place when the network is first installed or when changes, if any, take place. Cables rarely go bad after they have been made, installed, and tested.
- Broken cables don't create intermittent problems—they make permanent disconnects. A TDR can tell you where a break is on a cable.
- Certifiers test a cable to ensure that it can handle its rated amount of capacity. If a cable isn't broken yet isn't moving data the way it should, test it with a certifier. Use a certifier for slowdowns, not disconnects.
- Heat and power problems manifest as intermittent network problems. Use a voltage event recorder to measure power, and use a temperature monitor to ensure proper temperature.
- A good protocol analyzer will give you the Application, Session, Network, and Data Link layer information on every frame going through your network. Protocol analyzers can be hardware or software.
- A multimeter tests voltage and can tell you how much voltage is on a line.
- Tone generators and tone probes work as a pair to help you locate a particular cable.
- A butt set is used to tap into a 66- or 110-block to see if a particular line is working.
- A punchdown tool places UTP wires into 66- and 110-blocks. It is useful in a diagnostic environment to repunch a connection to make sure all the contacts are properly set.
- Software tools can be organized in two categories: those that come built into your operating system and those that are provided by a third party.
- TRACEROUTE (called TRACERT in Windows) is used to trace all the routers between two points. Use it to diagnose problems reaching a remote system. Because many routers block TRACERT packets, PATHPING is a viable alternative.
- IPCONFIG (on Windows) and IFCONFIG (on everything else) gives you information about a computer's IP settings. The /all switch gives additional detailed information, including DNS server addresses and MAC addresses.
- PING uses ICMP packets to show you if you can simply reach a remote computer. Because some devices block the ICMP packets, ARPING can be used instead. However, ARPING is available only on UNIX systems—and it can't cross routers.
- NSLOOKUP is used to diagnose DNS problems, but is considered obsolete. DIG is a more powerful alternative, but is not available on Windows.
- The HOSTNAME command simply returns the host name of the local computer.
- MTR, which is not available on Windows, is similar to TRACEROUTE except that it keeps running until shut down.
- The ROUTE command enables you to display and edit the local system's routing table.
- NBTSTAT (Windows only) can show all the local NetBIOS names and is a command-line equivalent to My Network Places. NBSTAT must be run with a switch.
- NETSTAT displays information on the current state of all the running IP processes on your computer. Use NETSTAT when you want to know about your current sessions.
- A packet sniffer intercepts and logs network packets. Wireshark is a popular packet sniffer.
- A port scanner probes ports on another system, logging the state of scanned ports. It can be used to find an unintentionally open port so that it can be secured. Hackers like to use port scanners to find vulnerabilities in other systems.

Analyze and discuss the troubleshooting process

- There is no reference guide to troubleshooting every possible network problem because such a guide would be obsolete the moment it was created.
- A basic troubleshooting model may include the following nine steps: (1) Gather information—identify symptoms and problems, (2) Identify the affected areas of the network, (3) Establish if anything has changed, (4) Establish the most probable cause, (5) Determine if escalation is necessary, (6) Create an action plan and solution identifying potential effects, (7) Implement and test the solution, (8) Identify the results and effects of the solution, and (9) Document the solution and the entire process.
- When establishing the symptoms, it may be necessary to ask the user reporting the trouble both closed- and open-ended questions.
- Isolating the cause of the problem includes identifying the scope of the problem, such as determining if it affects a single system or the entire network.
- When trying to determine what recent changes may have caused the problem, it is important to recognize things that are not causes. Re-creating the problem yourself removes user error as a possible cause, and experiencing the problem on another computer removes the possibility of changed settings on the first computer as the cause.
- Once you have determined possible causes, you should identify what you feel is the most probable cause. The ability to identify the most probable cause improves with experience.
- When implementing a solution, be sure to try only one thing at a time. If you perform multiple activities or make multiple changes, you won't know which action actually solved the problem—and you won't know which action made things worse.
- Once a solution has been implemented, test it by trying to re-create the problem. If you can re-create the error, you haven't fixed the problem.
- If you have fixed a problem, you need to recognize what potential problems you may have caused. For example, replacing a NIC in a server may get the server back online, but the new NIC has a different MAC address, which may introduce a whole new set of problems.
- Problems, symptoms, and solutions should be documented so the solutions can be used later in a knowledge base. Additionally, the documentation will help you track problem trends.

Tackle a variety of troubleshooting scenarios

- The CompTIA Network+ exam loves to ask scenario questions, so be familiar with as many scenarios as you can!
- Some common scenarios every tech should be familiar with include users not being able to log in, Web sites not loading, servers or networks appearing slow and sluggish, and My Network Places (or Network) not working as expected.
- Various networking problems that fall outside the scope of a tech's skill set should be escalated. These include broadcast storms, switching loops, route problems, routing loops, and proxy ARP.

Key Terms

ARP PING *(408)*
broadcast storm *(419)*
butt set *(405)*
cable stripper *(405)*
cable testers *(404)*
certifier *(404)*
DIG *(408)*
hardware tools *(403)*
HOSTNAME *(409)*
IFCONFIG *(407)*
IPCONFIG *(407)*
multimeter *(405)*
My Traceroute (MTR) *(409)*
NBTSTAT *(410)*
NETSTAT *(410)*
NSLOOKUP *(408)*
packet sniffer *(410)*
PING *(408)*
port scanner *(411)*
protocol analyzer *(404)*
proxy ARP *(420)*
punchdown tool *(405)*

ROUTE *(409)*
routing loop *(420)*
snip *(405)*
switching loop *(419)*
temperature monitor *(404)*
tone generator *(405)*
tone probe *(405)*
TRACEROUTE *(406)*
TRACERT *(406)*
voltage event recorder *(404)*
Wireshark *(411)*

Key Term Quiz

Use the Key Terms list to complete the sentences that follow. Not all the terms will be used.

1. Use _______________ to locate a problem between two routers.
2. Use a(n) _______________ to put wires into 66- and 110-blocks.
3. A(n) _______________ tests cables to ensure they can handle their rated capacity.
4. If ICMP packets are being blocked, you can use _______________ to test connectivity to another system.
5. _______________ is a popular packet sniffer/protocol analyzer/packet analyzer.
6. To view IP settings on a UNIX computer, use the _______________ command.
7. Use a(n) _______________ to test AC/DC voltage, resistance, and continuity.
8. _______________ is similar to NSLOOKUP, but much more powerful.
9. _______________ uses ICMP packets to test connectivity between two systems.
10. A(n) _______________ is used by telephone technicians to tap into a 66- or 110-block to determine if a particular line is working.

Multiple-Choice Quiz

1. Jordan says she can't access files on the server any more. No other user has reported this problem and she can PING the server from another computer successfully. Typing `ping 127.0.0.1` from Jordan's computer is also successful. Using PING to try to reach the server or any other computer from Jordan's computer fails. A check of IP settings on Jordan's computer shows that her static IP address and other information is good. What is the most likely cause of the problem?
   - **A.** The router that Jordan's computer connects to is down.
   - **B.** Jordan's network card is bad.
   - **C.** The DHCP server is down.
   - **D.** Jordan's Ethernet cable has become unplugged from her computer.
2. You are trying to locate which patch cable in the main switch traces back to a particular computer. Which tool should you use?
   - **A.** Tone probe
   - **B.** Cable tester
   - **C.** Punchdown tool
   - **D.** Butt set
3. The Windows TRACERT tool fails sometimes because many routers block TRACERT packets. What tool can be used as an alternative?
   - **A.** PING
   - **B.** PATHPING
   - **C.** PATHROUTE
   - **D.** TRACER
4. What is the first step in the troubleshooting model?
   - **A.** Implementing the solution
   - **B.** Testing the solution
   - **C.** Isolating the cause
   - **D.** Establishing the symptoms

5. Kay's computer has lost all network access. Which tool should be used to test for a break on the cable?

   A. Certifier

   B. TDR

   C. Voltage event recorder

   D. Crimper

6. Which command shows you detailed IP information, including DNS server addresses and MAC addresses?

   A. `ipconfig`

   B. `ipconfig -a`

   C. `ipconfig /all`

   D. `ipconfig /dns`

7. Which tool uses ICMP packets to test connectivity between two systems?

   A. ARP

   B. ARPPING

   C. PATHPING

   D. PING

8. What tools can you (and hackers) use to discover vulnerabilities on your network? (Select three.)

   A. Port scanner

   B. NMAP

   C. Angry IP Scanner

   D. HOSTNAME

9. Asking a user "Can you start your e-mail program?" is what type of question?

   A. Closed-ended

   B. Open-ended

   C. Leading

   D. Unprofessional

10. If you want to see which other computers on your network are currently connected to you, what tool should you use?

    A. PING

    B. NBTSTAT

    C. NETSTAT

    D. TRACERT

11. One of your users calls you with a complaint that they can't reach the site www.yahoo.com. You try and access the site and discover you can't connect either but you can PING the site with its IP address. What is the most probable culprit?

    A. The workgroup switch is down.

    B. Yahoo! is down.

    C. The gateway is down.

    D. The DNS server is down.

12. A brand new employee is complaining on his second day of work that he can't log into his computer. What is the most probable cause?

    A. The server is down.

    B. His network card is bad.

    C. He forgot or is mistyping his password.

    D. A port on the switch is bad.

13. When should you use a cable tester to troubleshoot a network cable?

    A. When you have a host experiencing a very slow connection

    B. When you have an intermittent connection problem

    C. When you have a dead connection and you suspect a broken cable

    D. Never

14. Which tools should you use to diagnose problems with DNS?

    A. NMAP or Wireshark

    B. NSLOOKUP or DIG

    C. PING or PATHPING

    D. TRACERT or PATHPING

15. Which Windows command displays the local system's routing table?

    A. `route print`

    B. `print route`

    C. `tracert /print`

    D. `tracert /p`

Essay Quiz

1. You and a co-worker are working late trying to fix a problem on the server. Your friend suggests applying three hot fixes and swapping out the network card for another. However, he wants to do all these things at the same time to finish the job quicker. Explain to him why that's not a good idea.
2. Because of your outstanding troubleshooting skills, you have been selected by your supervisor to train a new intern. Explain to her the steps of a basic troubleshooting model.
3. You've read in this chapter: "First, do no harm." First, explain in your own words what that means to you. Then, think of a situation in which you were either the technician or the "victim" in a troubleshooting case where harm was done. What happened?

Lab Projects

• Lab Project 15.1

You've learned about many free software tools in this chapter—some available only for Windows, some only for UNIX/Linux/Mac, and some available for all. Make a chart with five columns: tool name, description, useful switches/options, supported operating system(s), built-in or third party. Fill in the chart with the tools from this chapter and use it as a study guide.

• Lab Project 15.2

Using the chart you created in the previous lab activity, run each of the tools to gain some familiarity with the interface, switches, and output. Are there any tools in your chart you already use on a regular (or semiregular) basis? Which tools are the easiest for you to understand? Which tools do you not completely understand? If there are any that are still unclear, ask your instructor or research the Internet for clarification on the tool's usage. If you've researched the Internet, compare your finding with classmates or verify with your instructor to make sure your research resulted in correct information!