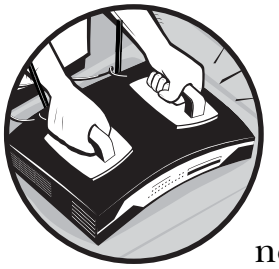
# 5

# TROUBLESHOOTING ROUTERS



Now that you have a basic understanding of how to configure a router and how various network types work, let's consider the most common router problems. These can be boiled down into two groups:

- Router crashes
- Network failure or slowness

By following a few straightforward steps, you'll have a reasonable chance of fixing many of these problems. In the worst case, you can at least be well armed with diagnostic information when you call your ISP.

## Router Crashes

For the most part, Cisco routers keep working. On rare occasions, an untouched router will intermittently reboot itself, or just shut itself off for no apparent reason. In these situations, a SmartNet contract is

invaluable—Cisco devices are mostly black boxes, and if something goes wrong with one, the warranty might be your only recourse. Still, you can try a few things to resolve the problem without a service contract.

A crashing router will often print errors to the serial console. Attach the serial console to the router and leave it there. Any crash messages will remain in the serial terminal's message buffer. Copy the messages exactly, and search on them in Google or http://www.cisco.com; you may well identify the problem. Cisco's technical support will certainly want copies of these messages. Cisco offers articles on how to diagnose router crashes for various models. Many articles are accessible without a SmartNet contract. Always search http://www.cisco.com for help.

If you cannot discover the problem through the crash messages, check the hardware itself; perhaps something's come loose inside the router. Open the case and remove any dust that has worked its way in, preferably with compressed air. Reseat every removable component, such as memory or flash cards, as well as add-on modules such as Cisco WIC cards and port adapters. When you are certain that everything is tightly attached, replace the router in the rack and turn it on. If you're lucky, it will work.

If your router still crashes, you have no choice but to contact Cisco. (Well, all right, you could choose to live with the problem. A few moments of lost connectivity isn't the worst thing that could happen to a company, but it could hurt your job and definitely interfere with your Internet radio listening.) If you don't have a SmartNet contract, Cisco's technical support will happily transfer you to the sales department or direct you to a reseller.

### AUTO-FTPING CISCO CRASH DUMPS

You can also proactively configure your router to automatically write crash dumps to an FTP or TFTP server. This only works if you do not have a NAT device between your router and your (T)FTP server. If you have an FTP server without NAT, however, I highly recommend you do this when installing your router. While you'll still need Cisco tech support to interpret the crash dump, they'll be able to offer a faster resolution to unusual issues. See Cisco's website or search Google for details.

## Network Failure

Network failure is by far the most common router problem, even though, technically, the router itself hasn't failed—the circuit has. Backhoes digging in the wrong spot cause any number of circuit outages, and you'll never fix those yourself. Before you blame the telco or ISP, however, you can do several things to rule out problems on your end and maybe even quickly fix the problem without outside help.

Most network administrators are fairly skilled at troubleshooting Ethernet problems. Is there a link light? Did the duplex setting change? Is the switch port bad, or did someone trip on the cable and break the little plastic thinga-majig on the end? The time-honored ritual of replacing cables and switching

ports solves most Ethernet problems. Internet circuit failures, on the other hand, are far more difficult to troubleshoot, as you only have a small window into the circuit, and most of it is handled by the ISP or telco.

As with all troubleshooting, the first question to ask yourself is, "Did *I* change anything?" If you made a change and the network went away, try undoing that change to see if the network returns. If that doesn't work, log in to the router and look at the interface of the failed network connection.

## Initial Circuit Tests

A phone call from a user screaming, "The Internet is down!" deserves a closer look. But remember, for many users, the Internet is Internet Explorer, or even just Yahoo! (if you're my publisher's mother). I recall one user who insisted the Internet was broken whenever an email took more than five minutes to arrive. Because the email system was badly overloaded, this happened at least three times a week.

If you think that you have an Internet failure, check a variety of network services first. A failure in a nameserver, firewall, proxy server, switch, or other device can appear to be a complete Internet failure to users who depend upon it. The two most helpful diagnostic tests you can perform to determine the extent of an actual Internet circuit outage are ping and traceroute.

### ping

When you ping, you simply send a request to another Internet node asking, "Are you there?" If the remote node receives the request and is not configured to ignore it, it should send a response. A successful ping means that you have basic network connectivity to that node.

When network problems are wide enough that you suspect an Internet circuit failure, your first attempt at resolution should be to log in to your router and try to ping across your Internet circuit. You may have a host name for this circuit, but use the IP address, because actual Internet problems also mean DNS problems. Chances are good that the far side of your circuit is the default route on your router. In the following example, the remote side of our serial line has an IP address of 192.168.88.65. Log in to your router and give the `ping` command and the target IP address.[1]

```
router# ping 192.168.88.65
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.88.65, timeout is 2 seconds:
❶ !!!!!
Success rate is ❷100 percent (5/5), round-trip ❸min/avg/max = 4/4/4 ms
router#
```

[1] It's also possible to try to ping from your desktop computer, but in this case, you're not just testing the router's circuit; you're testing your desktop's connection to the network, any intervening firewalls, and whatever other network equipment lies between your computer and the far side of the circuit. To isolate problems, test only one small piece at a time.

The router sends five packets to the destination IP address and gives each two seconds to return. This is generally more than long enough to ping anything that isn't over a satellite link. Each exclamation point ❶ indicates a packet that returns, while a period shows a packet that has vanished. Finally, we see the success percentages ❷ and some statistics on packet speed ❸, in milliseconds. The circuit shown here is certainly up.

On the other hand, a result like the following indicates your local circuit is down.

```
Sending 5, 100-byte ICMP Echos to 192.168.88.65, timeout is 2 seconds:
❶ .....
Success rate is ❷0 percent (0/5)
```

A period is a dropped packet ❶. Five periods indicate total circuit failure ❷. Start troubleshooting your circuit!

However, going back to the first ping example, just because your circuit is up doesn't mean that you're on the Internet. Perhaps your ISP had a failure, or some major backbone had a run-in with that rogue backhoe. Your next step is to attempt to reach a popular Internet site such as Yahoo! from your router. If your circuit is passing traffic, but you cannot ping such popular and highly available sites such as Yahoo! or CNN.com, you may have a failure between your network and the main Internet. That's when you use traceroute.

#### traceroute

The `traceroute` command sends packets to a remote Internet node and returns the IP address of every node it passes through on the way. Remember that to reach a remote site, your traffic probably travels through your router, your ISP's router, across several backbone routers, through the destination site's ISP's router, and through the destination site's router before reaching the actual target server. If any of these are broken, your request fails.

To use traceroute, just log in to the router and enter the command `traceroute` and the IP address or hostname you're attempting to reach.

```
router# traceroute www.blackhelicopters.org
Translating "www.blackhelicopters.org"...domain server [OK]

Type escape sequence to abort.
Tracing the route to bewilderbeast.blackhelicopters.org (198.22.63.43)

  ❶1 192.150.247.53 [❷AS 26096] ❸4 msec 4 msec 0 msec
  ❹2 192.150.247.38 [AS 26096] 4 msec 8 msec 8 msec
  ❺3 bewilderbeast.blackhelicopters.org (198.22.63.43) [AS 26096] 12 msec
8 msec 8 msec
router#
```

Upon leaving our current router, traffic to the server goes through the router with the IP address of 192.150.247.53 ❶. If this machine had a reverse DNS entry, a hostname would be displayed instead. Of three packets, two return in four milliseconds ❸, and one returns in zero milliseconds. (This

doesn't mean zero time; it just means a smaller time than your router can measure.) This router belongs to the Autonomous System 26096 ❷ (see Chapter 7). The second router is crossed very quickly ❹. On our third hop ❺, we reach a machine with the reverse DNS of "bewilderbeast .blackhelicopters.org," which is where the traceroute ends. Presumably, the IP address of www.blackhelicopters.org is on this machine.

- When a packet is dropped, you will see an asterisk instead of a timestamp.
- If a traceroute ends in a !H, that means that the last node in the traceroute doesn't know how to reach the destination. This is almost certainly a routing issue.
- Traceroutes are said to "die" when they turn into line after line of three asterisks. This may indicate that a firewall is blocking traceroute packets from reaching the target server, or it could be a network problem. The last node that appears knows where to send the packets, but it is not receiving a response from the next hop.

A traceroute can help you clarify the scope of a problem: if all of your traceroutes reach your ISP's router and die, you can assume that your ISP has a problem and that it's time to give them a call.

If the traceroute dies somewhere out in the middle of the Internet, it's a fair bet that there's nothing you can do, but that the damage is limited in scope. Or there may be some Internet "black hole" that causes many different requests to fail. For example, for much of 1996, MCI's Willow Springs router cluster seemed to be the Secret Internet Packet Burial Ground; network requests routed through that system consistently suffered horrible packet loss and latency. It's highly likely that the owner of such a Network of Death is well aware of the problem and is desperately attempting to resolve the issue, but if you have no business relationship with the network owner, there's nothing you or your ISP can do about it. In either event, at least you know.

If you cannot even ping across your circuit, it's time to get your hands dirty and troubleshoot your circuit. But before you can troubleshoot anything, you must understand it.

### Circuit Design

Different phone companies have used many different types of connection equipment over the decades. One of my clients recently replaced a phone switch dating from the 1950s, featuring fuses the width of my thumb and several inches long. (I believe that Dr. Frankenstein used such equipment, and I have suggested offering the equipment to him for spares if he'll only haul it away.) There is no way this sort of equipment can be used for an Internet connection.

Others have systems of a more recent vintage, but a completely unknown origin. If you have such a baroque setup, save yourself a *lot* of trouble and get a modern circuit. We're going to discuss the physical design of a modern T1 circuit, not Frankenstein collectibles.

### The Smartjack

Phone companies usually terminate the circuit at your location with a device called a *smartjack*, which is a small box with green and red lights on it. The smartjack has simple brains that the telco can use for troubleshooting (that's where the "smart" part comes from). The phone company or ISP is usually responsible for the circuit up to the smartjack. Problems between the smartjack and the router are your responsibility.

The point where the smartjack is installed is called the *demarc*. If your smartjack is far from your router, it's not uncommon to have a professionally installed *extended demarc*, where the circuit is run from the smartjack to a point more convenient for the equipment. You'll frequently see this in office buildings, where the phone company delivers all the circuits to one central location but offices are on many floors above.

Eventually, the circuit is delivered to a Channel Service Unit/Data Service Unit (CSU/DSU), which transforms the signals arriving over the T1 line into something the router understands. Most modern Cisco WAN interfaces have an integrated CSU/DSU, so you can just plug the RJ-45 cable from the smartjack directly into the router without having additional clunky boxes sitting around the data center.

#### OTHER CIRCUIT TYPES

Circuits like DS3 or OC3 give their components different names, but the ideas are the same. A DS3 is delivered over coax instead of RJ-45, but it still goes back to a telco-managed box on your wall.

## Examining the Circuit

The first line of the `sho int` output for an interface describes the basic state of the interface, as discussed in Chapter 3. If the line is up, the router is seeing a reasonable signal from the T1. If the line is down, the router is not seeing a signal over the T1. While you still need to check for other errors on the interface, not seeing a signal is a pretty solid sign that something is seriously wrong.

If the circuit is up but the protocol is down, the router cannot understand the signals coming over the T1. If this is a brand-new circuit, it is probably misconfigured, but in mature circuits, this may mean line noise or damage. (It can also mean that someone else somewhere along the circuit touched things, but no ISP or telco would even *dream* of doing that, not even if some underpaid and overworked tech thought you wouldn't notice if he just tweaked one setting while nobody was looking.)

Although you'll want to check the rest of the information available on this interface, there are things you can try first.

### Resetting the Interface

First, if some network device along the circuit is confused, resetting the interface may kick that device back to its senses.

```
router# conf t
router(config)# int s1/0
router(config-int)# shut
```

Count to 10 slowly, and then reopen the interface.

```
router(config-int)# no shut
router(config-int)# ^Z
router#
```

If your circuit is back up, congratulations! If not, the next step is to reboot the router.

### Rebooting the Router

While pulling the power cord out of the back, counting to 10, and plugging it back in will do the trick, as long as you're logged in you might as well restart the router in a slightly more graceful manner.

```
router# reload
```

The router will prompt you for confirmation and then restart itself. If your router has a separate CSU/DSU, power cycle it simultaneously.

### Nothing Worked!

If neither of these works, you must phone your ISP or telco. You can make this call in one of two ways: either armed with lots of information so you can resolve the problem as quickly as possible or in a hysterical panic. When you use the hysterical panic technique, you don't have to bother calming yourself down. An ISP tech usually accepts screaming abuse from customers as part of the service as he tries to extract useful information from your cries. If you prefer the hysterical panic technique, make that call now. If not, arm yourself with all the information your router offers before calling. No matter how good your ISP's customer service department is, failure of your network is more important to *you* than it is to the person answering the phone. After all, that support tech still has *his* email!

When you make that call, however, you will sound much more impressive and serious if you can say, "Our circuit protocol has failed, and we are receiving hundreds of CRC errors a second," than if you say, "Uh, our circuit isn't working." Your router provides you with a wide range of debugging information if you know how to read it.

### Interface Debugging Information

Each router interface provides a full description of the work it is performing, the errors it sees, and what actions it is taking. While we won't cover every scrap of information your router offers, you should understand some of the basics when you have to troubleshoot. Here's the output from a `sho int` on a standard Cisco T1 interface. Your output might look slightly different, depending on your interface and circuit type.

> **INPUT AND OUTPUT**
>
> Remember, a router's "input" is traffic that is entering the router via the interface, while the "output" is leaving the router via that interface. In the case of a T1, input is coming in from the outside world, while output is leaving your network and heading out over the T1. On a typical Ethernet interface, router input is leaving the local network, and output is arriving at the local network.

```
#sho int s1/0
...
❶ Last input 00:00:07, output 00:00:07, output hang never
❷ Last clearing of "show interface" counters 5w6d
...
❸ 5 minute input rate 6000 bits/sec, 6 packets/sec
  5 minute output rate 6000 bits/sec, 2 packets/sec
❹ 6413455 packets input, 2153942875 bytes, 0 no buffer
❺ Received 0 broadcasts, 0 runts, 641 giants, 0 throttles
❻ 36851 input errors, 327 CRC, 36520 frame, 0 overrun, 0 ignored, 4 abort
  2871476 packets output, 608779651 bytes, 0 underruns
❼ 0 output errors, 0 collisions, 31 interface resets
  0 output buffer failures, 0 output buffers swapped out
❽ 27 carrier transitions
❾ DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

The "last input" and "last output" values ❶ show how long it has been since packets entered or left this interface. This particular interface has been idle for seven seconds, which is not surprising with a problematic T1 or a slow network.

Cisco tracks most errors with an incrementing counter. The last clearing of "show interface" counters space ❷ shows the last time this incremental counter was reset to zero. In this example, the counters have been incrementing for five weeks and six days, far too long to be useful for troubleshooting a problem happening right now—those 15 million errors the router has recorded could have happened a month ago or in the last minute, and there's

no way to tell. It's very easy to reset these counters, though, and you don't even have to go into configure mode:

```
router# clear counters
Clear "show interface" counters on all interfaces [confirm]
router#
```

This resets all the counters on all the interfaces to zero. You can now do a `sho int` on your failed interface several times in succession and easily see if any error counters are increasing while the problem is happening. (Yes, you can do this while the error counters are all high, but it's much easier to see the difference between 0 and 300 than to see the difference between 15831594 and 15831894, especially if many error counters have nonzero values.)

### Input/Output Rates

The five-minute input and output rates can be useful for troubleshooting ❸. Although they are averaged over the last five minutes, you can check the interface several times in succession to see how they change. Is the average tending toward zero or climbing rapidly?

Remember, a T1 only handles 1.54MB/sec, or 1,540,000 bits per second. If your average throughput is close to that, that's why your network feels slow. A distributed denial-of-service attack or a sudden rush on your website can make your users feel like the Internet is down when actually it's just a massive flood of traffic making your circuit useless. The totals of all packets processed since the counters were cleared can be useful in a similar way ❹.

### Types of Errors

The interface then gets specific on the types of errors it sees ❺. *Broadcasts* are standard network broadcasts and generally are not a cause for alarm. *Runt* packets are smaller than the router's minimum packet size, and *giants* are larger than the maximum packet size. Neither should be on a T1 circuit. If they appear on an Ethernet segment, some network device is sending them out.

*Input errors* cause packets to be rejected ❻. The interface presents a total of all the input errors and then breaks them down by category. Although a certain number of errors is normal, if input errors comprise over one percent of your incoming packets, you have a problem. In this example, we have 36,000 input errors. We've input about 6.4 million packets, though, so this is far less than one percent of all packets, an acceptable error rate.

*CRC* and *frame input errors* probably indicate some sort of line noise; if these are happening frequently, call your ISP or telco and have them troubleshoot the circuit.

*Overrun*, *ignored*, and *abort input errors* indicate that the router cannot process incoming packets quickly enough and is forced to drop surplus packets. You can adjust the router's internal buffers, but this is very tricky to do correctly. I strongly encourage you to use Cisco's SmartNet support and have a technician help you adjust the buffers to fit your particular situation.

*Output errors* are most common on an Ethernet interface ❼, where an extremely high collision rate can interfere with the network's performance. On a serial line, output errors are most commonly the result of running out of buffers for outgoing packets (though this is quite rare on small routers). Again, if you're consistently running out of buffers, contact Cisco for help.

### Using Carrier Transition to Detect a Bad Serial Connection

One of the more useful ways to detect a bad serial connection is the carrier transitions line ❽. A *carrier transition* is when the interface either goes up or comes down. It's entirely possible for a serial line to bounce up and down so quickly that you won't catch it by successive `sho int` commands. The carrier transition increments whenever the line goes down, and again whenever it comes back up. By watching this field, you can detect an unstable circuit.

Finally, a serial line will list the signals that it is receiving over the line ❾. For the line to be functional, all of these need to be up; if any are not up even after rebooting everything, call your ISP or telco. Your router cannot process traffic it does not receive!

## Extended Pings and Circuit Troubleshooting

Cisco routers can use a variety of ping tests to check IP connectivity to other networks. This is very useful for testing troublesome serial circuits that are not down but that are not actually working properly. At times, a misprovisioned circuit will cause problems that do not appear on `sho int`, but that cause no end of headaches.

On more than one occasion, I've dealt with a circuit with a perfect-looking interface that passed every standard ping test with flying colors and yet was useless. GIF files could not be moved across these circuits and Windows networking infrastructure protocols would not work over them. In each case, the problem was exactly the same: the circuit was provisioned incorrectly (with AMI encoding instead of the modern standard of B8ZS). This meant that the circuit could not pass large all-zero packets. The .gif image format uses a lot of large all-zero packets, as do the Windows network protocols.

**NOTE** *Other physical problems generate different symptoms; this particular example is just the one that seems destined to haunt me. Your data circuit should accept whatever sort of data you want to transmit, even if it is a whole lot of nothing!*

To perform an extended ping test, enter enable mode and just type **ping**.

```
router# ping
Protocol [ip]:
Target IP address: 192.168.5.3
```

The default protocol is IP, so just press ENTER. Then give the IP address you want to ping. Generally, this is the IP address of the router on the other side of your serial link.

Don't ping Internet sites to test your circuit's status. That tests every link between your router and the remote Internet site! The other end of the circuit is the only reasonable place to ping when testing your own circuit's behavior.

```
Repeat count [5]: 1000
```

When using extended ping, use more than five pings. A few hundred is the bare minimum for this sort of test, but a thousand is usually reasonable. Either your test will be over in just a few seconds, or you'll interrupt it. If you sweep the range of sizes (as recommended later), the router will decide how many pings are necessary to properly cover the range, and the value you enter here will be ignored anyway.

```
Datagram size [100]:
```

You could specify a size here if you wished, but we'll be doing something slightly more advanced in the extended commands.

```
Timeout in seconds [2]:
```

Two seconds is a very reasonable timeout. Giving a longer timeout is generally useless for most IP circuits because either the packet will return in much less time or you'll wait that much longer for failed packets. With a shorter timeout, you may miss perfectly legitimate return packets that are delayed by a busy remote router.

```
Extended commands [n]: y
```

If you take the default `n` (no) here, the router will run the ping as you have specified it. But most of the truly nifty options are in the extended commands, so enter **y** for yes.

```
Source address or interface:
```

You can change your source IP address and send packets from a particular interface, which is useful to do on backbone routers but not for most small offices. I highly recommend that you don't try this because the results will only confuse you. (Tracking down problems is difficult enough without using an easily misunderstood tool!)

```
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
```

These three flags manipulate or verify TCP/IP data but aren't generally useful for day-to-day work. If you're not familiar with the innards of TCP/IP, just take the defaults.

```
Data pattern [0xABCD]: 0xffff
```

A ping packet contains four bytes, and you can choose what data to put in them. The hexadecimal 0xABCD is the default, but two common choices are 0x0000 (all zeros) and 0xFFFF (all ones). Many misconfigured circuits fail with certain sizes of all-zero or all-one packets.

```
Loose, Strict, Record, Timestamp, Verbose[none]:
```

You can choose to use loose or strict source rerouting, record the route taken by the ping packets, datestamp each ping packet, or use verbose output. If you don't know what loose or strict source routing are, don't try to use them. For most situations, almost none of these are useful, but you might try verbose output sometime for your own edification; just enter **v** at the prompt.

```
Sweep range of sizes [n]: y
```

By sweeping the range of sizes, you send packets ranging from very small to quite large. Definitely set this when testing a troublesome circuit. The smaller packets will appear quickly, larger packets much more slowly. If you sweep the range of sizes, the `ping` command will ignore the count you entered earlier; you will need to interrupt it with CTRL-^.

```
Sweep min size [36]:
Sweep max size [18024]:
```

These are the minimum and maximum packet sizes when sweeping. The defaults range from quite tiny to reasonably large.

```
Sweep interval [1]: 50
```

The sweep interval is the increment to the packet size when sweeping. For example, with Cisco's default, the first packet sent will be 36 bytes, the

second 37 bytes, the third 38, and so on, until a packet size of 18,024 is reached. Picking a larger increment than 1 will accelerate the test. Also, you may not want to let your router send 17,092 pings to perform the test with every possible sized packet in the range.

Once you have chosen all the values, the ping will run.

```
Type escape sequence to abort.
Sending 1800, [36..18024]-byte ICMP Echos to 198.88.118.11, timeout is 2
seconds:
Packet has data pattern 0xFFFF
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
...
```

Each exclamation point indicates a successful ping, while a period indicates a missed one. This ping test will continue until the 1800 packets have been sent. Reducing the increment will increase the number of pings the router sends. Again, you can interrupt the test at any time by pressing CTRL-^.

**NOTE** *Small packets appear more quickly than large packets, and the largest packets in the ping sweep test may make you think that the router has stopped responding. Note which sorts of packets the router has trouble with.*

For circuits that are behaving poorly or exhibiting "weird" behavior, I suggest running the extended ping test three times: once with the default data pattern, once with an all-zero data pattern, and once with all ones. If your circuit has trouble with certain sorts of packets, this process will probably uncover it. If not, your circuit is probably running cleanly.

## Phoning the ISP

So you've opted to gather all your debugging information before calling. You know which sorts of errors are appearing on your circuit, and what sort of packets your circuit will and will not pass. Now what?

You got your T1 circuit from either a telco or an ISP. Phone them. The first words out of your mouth should be, "Hi, I have a down circuit." They will ask you for your circuit ID, customer number, or other identifying information. When you reach a technical support person, explain:

- The state of the line itself (both line and protocol down, just protocol down, or both line and protocol up)
- That you have rebooted your router and CSU/DSU (this will almost certainly be their first suggestion)
- The types of errors you are seeing: CRC, framing, overruns, aborts, and so on
- The results of any ping tests you have run

The technician should check the ISP's end of the circuit and phone the telco who originally provisioned the line. Most telephone companies will return a call to the circuit owner (usually the ISP) within an hour with an initial status report.

**NOTE** *While your ISP should handle circuit issues, especially when they own the circuit, if the problem is actually a failed circuit, it may take hours to repair.*

### Circuit Loopback Tests

T1 circuits include all sorts of intelligent equipment such as repeaters, switches, and smartjacks. The telco can talk to this equipment using a series of successive loopback tests to quickly identify where a problem lies and what piece of equipment is at fault.

The telco has a diagram listing every piece of equipment along a circuit. A loopback test is where they attempt to communicate with each piece of equipment in succession. For example, the telco central office will try to "loop up" the first piece of equipment outside their office—say, the repeater in the little gray box up the street. If they can easily communicate with that device, and if that device can return data it receives to the central office, they will loop up the second closest device and test it. When something fails to respond, they send a technician out to examine and repair that device.

Eventually, the telco will work their way down to looping up the smartjack at your demarc. If the telco can communicate correctly with the smartjack, they will probably try to loop up your CSU/DSU. The telco/ISP's responsibility technically ends at the smartjack, but they will frequently go the extra step simply to demonstrate that it isn't their problem. (This tactic should be painfully familiar to anyone who has worked a help desk!) If the telco says that they can cleanly loop up the smartjack but cannot loop up the CSU/DSU, the problem lies somewhere between the CSU/DSU and the smartjack and is unquestionably your problem.

This sort of exhaustive test takes time, especially on a long circuit, but it is the quickest way to identify a problem. The telco may have to send a technician out to your facility to perform testing. You may well have to have someone stay after hours to let the technician in and out of the building.

**WARNING** *A loopback test will completely disable your circuit! If your circuit is already down, then they can loopback all they want, but if you are merely suffering degraded performance, you may want to ask the ISP or telco to perform the test after hours. It's all a question of what your environment requires.*

Once you know without a doubt that the problem is with the telco or the ISP, the only skill required to fix the issue is a willingness to make a nuisance of yourself at the ISP's help desk.

### If It's Your Problem

If the ISP can loop up the smartjack but not your router, it's your problem. Congratulations! The good news is, as you have very few components in your section of the circuit, testing your equipment won't take very long. The bad news is, many components you do have will require outside help.

Check to make sure everything in your system is tightly attached. A loose connection can cause no end of problems. "Is it plugged in?" is still a good question in the comms closet.

A common culprit in circuit failures is wiring. Replace the wiring between the smartjack and your CSU/DSU. A T1 circuit uses a standard Cat 5 cable, so many people plug in a cable that they had in a drawer somewhere. A T1 is less forgiving than Ethernet, however. Use a good premade cable with sturdy connectors on both ends.

If you've replaced all your wiring, everything is firmly seated, and reboots and resets have not solved your problem, phone your CSU/DSU's technical support line. On a modern Cisco system, this would be Cisco itself.

**NOTE** *If your vendors start each blaming the other, as is common among lower-level technical support staff, do not be afraid to get them all on the phone at once and let them fight it out.*

Now that you know how to troubleshoot your router, you can learn how to upgrade it.