HOW-TOs

Troubleshooting Network Problems

by Mike Diehl on March 25, 2009

Back when I worked in the Network Operations department at one of my previous jobs, we used to chuckle when a customer would call us reporting that “the Internet is down.” Now, I realize that there are otherwise intelligent computer users out there who don't understand why that might cause a technician to chuckle, and I'm not trying to make fun of them. But you've got to know that help desks get this type of trouble report more frequently than they should.

I've known quite a few hardware technicians or software developers who were very good at what they did, but who didn't have the first clue as to how to diagnose a network problem. To them, the Internet, and networking in general, was just “Pure Freaking Magic”. Actually, it is magic, but it's not a magic that we can't learn to troubleshoot. So, let's begin.

Recent Articles

Typically, these problem reports come in because a customer can't reach a Web site, e-mail or printer. Before raising the red flag, you should check the basics. “Has this worked before?” “Did you make any changes recently?” As a technician, you should check to see whether the cable actually is plugged in; you'd be amazed at how many times that really happens! So, once you've confirmed that you really can't access the network resource, the fun begins.

My gut reaction is to try to ping the server that provides the service we are trying to access. First, try to ping it by its fully qualified hostname, server.example.com as an example. If that doesn't work, try to ping it by its IP address. If that DOES work, the problem is with DNS. Domain Name Service, or DNS, is a network service that resolves hostnames to IP addresses and vise versa. In this case, either the DNS server is down, or the client is configured improperly.

If you can ping the server but the service isn't available, network connectivity is there and the service is actually down. Many times, you can confirm this with the telnet command. For a Web server that listens on port 80, you simply can issue a command like the following:

```
telnet www.example.com 80
```

What you see next will tell you how healthy the server on the other end is. If you see a timeout error message, or nothing at all, it implies that the server either is brain dead or is so heavily loaded that it can't process your request. This also could indicate that a firewall is blocking access, but we're assuming that this worked before and that the firewall policy hasn't changed. The other response you might get from the telnet command is a connection-refused message; this indicates that the service has crashed and is no longer accepting incoming connection requests. Of course, you need to know the port number for the service you are trying to access. As a quick reminder, mail is usually on 25. IMAP is on 143. SSH is on 22. Secure HTTP is on 443. These are the common ones. Anyway, at this point, it's not a network problem; it's a server issue.

But, what if you weren't even able to ping the server? Now it's looking like a network problem, but where? In general, when faced with a problem like this, I usually like to split the problem in half. So, the first thing I do is see whether I can at least ping my default gateway or router. Usually, when I'm on the phone with customers troubleshooting a problem, this needs a bit more explanation. Invariably, they don't know what their default gateway is. To find out what your default gateway is, you have to open up a command window in Windows or a shell window in Linux. Sometimes, you have to guide customers through this process.

Linux Journal Ceases Publication: An Awkward Goodbye

Kyle Rankin

Oops! Debugging Kernel Panics

Petros Koutoupis

Loadsharers: Funding the Load-Bearing Internet Person

Eric S. Raymond

Documenting Proper Git Usage

Zack Brown

Understanding Python's asyncio

Reuven M. Lerner

RV Offsite Backup Update

Kyle Rankin

Let's tackle the Windows client first. In Windows, most of the IP configuration information is obtained with the `ipconfig /all` command.

From the output of this command, you should be able to see your client's IP address first. If not, the client probably has lost its DHCP lease, and you should try to renew it with the `ipconfig /renew` command. If you're not running DHCP and your customer doesn't have an IP address, it's time for a desktop visit. Otherwise, renewing the lease may fix the problem. If the customer can't get a DHCP lease, it means that the network connectivity to the DHCP server isn't working, or the DHCP server is down. Typically, when a DHCP server dies, the help desk phone goes nuts with people who can't access network resources.

So at this point, our Windows client has an IP address. We need to make note of the default gateway, which also is displayed by the `ipconfig /all` command.

On a Linux client, we can get the IP address with the ifconfig -a command. You simply have to pick out the correct network interface and see whether it has an IP address. If not, the earlier comments about DHCP apply. If you've confirmed that your Linux client has an IP address, you need to find the default gateway. This can be done by issuing the `route -n command`.

That output looks something like this:

```
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.5.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
69.254.152.0 0.0.0.0 255.255.252.0 U 0 0 0 eth5
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
0.0.0.0 69.254.152.1 0.0.0.0 UG 0 0 0 eth5
```

You are interested in the IP address from the Gateway column where the Destination is indicated as 0.0.0.0, in this case, 69.254.152.1. This is the default gateway.

So, we now have the default gateway for either a Linux or Windows client. If we can ping the default gateway, but still can't ping the network resource that we are trying to access, we probably have a bona fide networking problem, which needs to be reported.

At this point, we've discovered that the problem is somewhere between our workstation and the default router.

This probably is a good time to discuss briefly how IP communication actually works, as this will help explain why knowing the default route was so important. Essentially, there are two cases. In one case, two machines want to talk to each other and are on the same network. In the other case, two machines want to talk, but they are on different networks. We'll discuss each case shortly. But first, it's important to know what constitutes a “network”. Certainly, every machine on the Internet and every machine connected at home and at work are on “a network”. But in this case, what we are talking about more aptly should be referred to as “a local network”. A local network is a group of networked machines that use the same default router; it's that simple. Now by changing the subnet mask, we can adjust how big a given network can be, but the default router is the actual boundary between local networks.

That leaves us with the two communications cases mentioned earlier. Let's deal with the case where two machines, A and B for example, want to talk on the same local network. It turns out that on a local network, a device's IP address isn't as important as you might think. Device A can't simply send a network packet to device B if all it knows is B's IP address. So, device A shouts out, or broadcasts, “What is B's MAC address?” When B hears this request, that device will broadcast back, “B's MAC address is...” This process is known as “arping” and uses the Address Resolution Protocol, or ARP. Now A can use both B's IP address and MAC address to address a network packet to B and send it. When A sends this packet out, every machine on the local network receives that packet, but only the device with the MAC address to which the packet is addressed will process the packet.

This ARP process gives us another diagnostic tool. All networked devices maintain a list of IP and MAC addresses that they've encountered. This is known as an ARP Table and can be viewed with the `arp -n` command. First, try to ping the default router and a few machines on the local network that you know are working. Then, see if you find any entries in the ARP Table, like this:

```
# arp -n
Address HWtype HWaddress Flags Mask Iface
10.0.1.51 ether 00:04:F2:12:78:77 C eth1
10.0.1.59 ether 00:1B:2F:34:6E:B7 C eth1
69.254.152.1 ether 00:1E:BE:FE:F8:05 C eth5
10.0.1.4 ether 00:0E:A6:87:BA:2B C eth1
10.0.1.200 (incomplete) eth1
```

As you can see, we've got several IP and MAC addresses. Note that the last entry is “incomplete”. What this means is that I tried to communicate with that address and nothing responded to my ARP request. That machine is down. On the other hand, if our ARP Table were empty, this would tell us that we have a problem somewhere between the network port on the back of our computer and the port on the switch or hub we're plugged in to. In many cases like this, we can use the mii-tools command to see if our network card has “link”. If it doesn't, we've got a cable problem or the switch port has been disabled. In the rare case where you find that you have link but still can't communicate, you might have a hardware problem.

So now, let's consider the case where our two devices, A and B, are on different local networks. In this situation, the two devices can't talk to each other directly. So instead, what they do is talk to their default router and essentially ask it to “relay”, or “route”, the message to the other device. What ends up happening is that device A will arp for its default router's MAC address and send a packet addressed to the router's MAC, but device B's IP address. The router receives the packet, notices that it's not addressed to one of its IP addresses and forwards it on the the next hop. Each hop brings the packet closer to device B's default router. At this end, B's router receives the packet and notices that it's addressed to an IP address on one of its local networks. So, the router arps for B's MAC address and sends the packet to B with both B's IP address and MAC address. Now, how the routers know how to route packets and get them to their destination can be a complicated topic, so I won't discuss it here.

So there you have it. I think I've covered most of common problems people encounter. There are certainly more esoteric situations that come up occasionally. I'm hoping that with a few common software tools and some basic understanding of networking, that you can diagnose most network problems whether they're at work or home.

Sign up to get all the good stuff delivered to your inbox every week.

Enter your email. Get the newsletter.

SIGN UP

I give my consent to be emailed

Connect With Us

Linux Journal, currently celebrating its 25th year of publication, is the original magazine of the global Open Source community.