10.5.7 Network Time Protocol (NTP)

Click one of the buttons to take you to that part of the video.

Network Time Protocol (NTP) 0:00-1:59

In this lesson we're going to discuss synchronizing time with the Network Time Protocol, or NTP. NTP is currently the best option available for synchronizing network time among multiple network hosts.

You can use the netdate command to do the same thing, but NTP is a better option. Like netdate, the NTP protocol gets its time from a network time provider and then it sets this local system time to whatever time it got from that time provider.

However, NTP is better because it manages time in a much cleaner fashion than netdate does. NTP will adjust the time in a very gentle manner. If there's a big differential between the time provider's time and the time on your local system, NTP will gradually adjust the time on your local system in small increments until the time eventually becomes synchronized.

The netdate command on the other hand, will just say 'whatever time it is on the time provider, that's the time you're getting' and it immediately sets the system time to that time that it gets from the time provider. Now, in some situations that works fine; it's okay. But there are situations when it can cause a lot of problems. We call this an unclean time synchronization.

For example maybe the time on my local system is a day ahead of the time on the time provider. If I look in the netdate command my system time just got set back one whole day. As a result, I have new files and new time stamps being created that are actually older than the older timestamps on files that were created the day before. That causes all kinds of problems on the system. In addition, NTP protocol has wider support, it's supported really by most major computer operating systems. It works on Linux systems, it works on Windows systems, it works on Macintosh systems and so on. So essentially, NTP also allows time to be synchronized among systems in a heterogeneous network environment which most networks are these days.

NTP Time Synchronization 2:00-2:36

In order to synchronize time using NTP, we install and configure the NTP daemon (ntpd). Notice that we use the NTP daemon on both the time provider and on the time consumer. That's a very important thing to remember. When you run the NTP daemon, a given system can function as both a time provider and as a time consumer at the same time.

Because of this, the system on the right can get its time from the system on the left, but then some other system somewhere else could get its time from the system on the right, because it can function both as a time consumer and as a time provider for other systems. The NTP protocol operates over IP port 123.

NTP IP Port 123 2:37-3:41

It's a really easy port to remember, one, two, three. What happens is that the NTP daemon on the time consumer sends a time synchronization request to the NTP daemon on the time provider. The time provider then sends its time to the consumer, and then the system time on the consumer is adjusted according to the NTP algorithm.

Remember that because of the way NTP works, it does not immediately set the local time to the same time as the time provider. Instead, it gradually adjusts it until they both come, eventually, into synchronization. Unlike netdate, this does not happen with one single command. Instead, the NTPD daemons on these two systems will communicate back and forth with each other multiple times per day to keep time in sync.

That's another benefit of using ntpd over netdate. Remember, netdate just does one quick time change, and that's it. Unless you run it with a cron job, it never checks to see that the systems are in synchronization again.

Using ntpd, on the other hand, these two systems continually talk to each other saying, "What time is it? Okay, that's the time I've got. I'll set my time to that." Because of this, it prevents drift from happening. These two systems will stay fairly well synchronized, with respect to their system times.

NTP Stratum 3:42-5:47

Essentially, the NTP protocol is a much more sophisticated way of synchronizing time than that provided by the netdate command and the time service. Because of that, there are several key NTP concepts that you have to be familiar with in order to effectively implement and manage the NTP protocol on your network.

The first concept that you have to understand is that of stratum. NTP uses the concept of stratum to define a hierarchy of NTP servers. At the top of the hierarchy are our Stratum 1 time servers. Stratum 1 time servers get their time from a reference time source, such as the Naval Atomic Clock.

If you want to learn more about how the Naval Atomic Clock works, you can visit the URL shown here to learn more about it. In addition to Stratum 1 time providers, we also have Stratum 2 time providers. As you can see here, Stratum 2 time providers get their time from a Stratum 1 time provider.

And you guessed it, Stratum 3 time providers get their time from a Stratum 2 time provider, not from a Stratum 1 time provider. We can continue on down this NTP stratum to a depth of 256 layers.

Understand that Stratum 1 servers are not publicly available. They are on the internet, but you are not allowed to access them. However, there are a whole bunch of Stratum 2 publicly accessible NTP time providers on the internet, and you can configure your system to connect to these systems to get your time.

If you have just one or two systems in your network, it's no problem at all for you to configure your systems to get their time directly from a Stratum 2 NTP time provider.

However, if you have a large network, you really don't want to have all of your computers, say, 500 or 700 or 1,000 computers all going out and beating up these Stratum 2 time providers to get the exact same information.

Instead, what you should do is configure a limited number of systems on your internal network, maybe two or three, to go out and synchronize time with a Stratum 2 time provider on the internet, and then configure the rest of your internal systems to get their time from those hosts.

Remember, that the NTP daemon allows a system to be both a time consumer, as what happened in this with this arrow, and as a time provider to the other systems, as happens with this arrow.

NTP Concepts 5:48-8:19

There are several other NTP concepts you need to be familiar with. You need to be familiar with the concepts of stepping and slewing. Understand the NTP protocol initially syncs time between the time consumer and the time provider about once every minute. However, this interval gradually increases to once every 17 minutes once the time between the two systems is more closely synced.

When the time is more closely synced, you don't need to synchronize every minute--once every 17 minutes is enough. Essentially, this means that large adjustments occur relatively quickly, but then only small adjustments are made across a longer time interval.

What does this have to do with stepping and slewing? The thing you need to remember is that if the time difference between the provider and the consumer is small, such as less than about 128 milliseconds, then the NTP protocol is going to adjust the time on the time consumer very gradually. This is called slewing.

If, on the other hand, the time difference between the provider and the consumer is large, then it's going to have to make time adjustments a little more quickly. That's called stepping.

Another concept you have to be familiar with if you're going to work with NTP is the concept of insane time. Of all the issues that cause problems when implementing and maintaining an NTP deployment, this is the one that does it.

You need to understand that if the time difference between the provider and the consumer is more than 17 minutes--in other words, they are more than 17 minutes out of sync--then the NTPD daemon (ntpd) is going to consider time to be insane. As a result, it's not going to adjust it, and that can cause all kinds of problems.

I can't tell you how many times I've seen NTP set up, we're trying to synchronize two systems, and they never come into synchronization, and we can't figure out why. Well, every single time that has happened has been because of insane time.

The system time on the two systems was so far apart that the ntpd says, "Huh, I'm not doing anything with this. This is crazy. I'm not touching it." And it doesn't do anything. You have to get the time on the provider and the consumer within 17 minutes of each other initially. Once you do that, then the NTP protocol can start stepping until it gets time close, and then it can start slewing until it gets time really, really close.

Watch for this issue. You also need to be familiar with drift. NTP measures and corrects for incidental clock frequency errors, which is a fancy way of talking about drift--where the system time on one system may not run at exactly the same frequency as the system time on

another system. Over time, the two get out of sync, which is why we synchronize time with NTP in the first place.

We also need to be familiar with the concept of jitter. Jitter refers to the time difference between the time consumer and the time provider since the last time polling.

/etc/ntp.conf 8:20-9:59

With this background in mind, let's discuss how you configure a system to use the NTP protocol to get time from a time provider. Before you can do this, you do have to install the NTP package on your Linux system. Some distributions install it by default, others do not. If yours does not have the NTP protocol installed, then you need to use yum, Zypper or apt-get--whichever is appropriate for your distribution--to go out and get it.

Then once done, you need to edit this file right here to configure how NTP is going to work: /etc/ntp.conf. There are a variety of different parameters within this configuration file, but the key one that you need to be familiar with is this one right here: server.

The server directive tells the NTP daemon where it's supposed to go to get time. You enter server, space, and then the IP address or DNS name of the time provider. The time provider that you specify could be another computer on your network.

Remember the diagram that we drew earlier, where we have one system that goes out on the internet to get time, and then all the systems internally get their time from just that one internal server, instead of everybody going out on the internet at one time. It could also point to an NTP time provider on the internet.

If you want to, you can use what is called an NTP pool time server. The pool.ntp.org domain uses a type of DNS round robin mechanism to make a random selection from a pool of time providers who have volunteered to be in that pool out on the internet. That way, no one public NTP server is overloaded with a whole bunch of different time synchronization requests.

If you want to use this pool, which is a really good idea, you just add the directive that you see here to your ntp.conf file: serverpool.ntp.org. Remember, this URL does not point to a server; it points to a pool of servers, and you never know which one you're going to get, but they are all synchronized so it doesn't really matter. They all have the same time.

Initial Time Synchronization 10:00-11:50

Here's an absolutely critical thing that you have to remember. Remember the issue of insane time? Before you actually start the NTP daemon, you can prevent insane time issues by performing a quick one-time synchronization with your NTP time provider. This makes sure that your system time is not more than 17 minutes off from the NTP time provider.

We do this by using the ntpdate command. The ntpdate command gets these two systems within this 17-minute window. It doesn't work like netdate. It does not bring the two systems into direct synchronization; it just gets them close, within that 17-minute window, and thus eliminates insane time issues.

We enter ntpdate, followed by the IP address or DNS name of the time provider that we want to get time from.

Because ntpdate does not immediately synchronize your system time with the time provider's time, you may actually need to run this command multiple times until you get your system time within that 17-minute window. The further apart the times are on the two systems, the more times you're going to have to run this command to get them within that time window.

An example of running the ntpdate command is shown here, where we synchronize the local time on the fs5 system, with the time on a server that is a member of pool.ntp.org. If you look at the output, you can see how far off you were over here, and the time on this system was actually pretty close to the time on the time provider--.002459 seconds off.

That is, for all intents and purposes, synchronized already. I do need to point out that before you can run the ntpdate command, you need to make sure that NTPD, the daemon itself, is off. ntpdate will not work while NTPD is running.

Basically, you're going to have two time synchronization mechanisms running simultaneously, and they don't get along with each other. I shut down ntpd, the daemon, first, run ntpdate until you get your time fairly close, and then turn NTPD back on.

Time Monitoring 11:51-14:25

Now, if you're running an older Linux distribution, then you would use the NTPD 'init' script. If you're using a newer Linux distribution that is based on system D then use the system control command to both enable and start the NTPD daemon. That's a really good idea to make sure

the NTPD daemon is enabled at system boot, and use this command right here to make sure that happens: system control enable NTPD. By running this command the NTPD daemon will automatically start whenever the system starts up.

Until that happens, though, the naval command will not actually start the daemon itself so you'll have to do that manually. So, you type system control start NTPD, and it's not bad idea to use the status option with the system control command to make sure that everything started normally; in this case, you can see that the daemon is active, it's running. Therefore, we know that every things working properly. In fact, if we look down here in the output, you can see the different network interfaces that NTPD is using to synchronize time on.

Once the NTPD daemon has been started, there are several commands you can use to monitor how well it's working; how closely it's keeping time synchronized. The first one is shown here. You can run ntpq -p at the shell prompt to query the status of the NTPD daemon. The columns and the output include those shown here.

First of all, we have the remote column, which specifies the hostname or IP address of the time provider that we're getting time from. We have the refid column, which specifies the type of time reference source that we're connecting to. The st column specifies the stratum of that time provider.

The when column specifies the number of seconds since the last time poll occurred. The poll column specifies the number of seconds between two time polls. The reach column displays whether or not the time server was reached, the last time it was polled.

A successful poll increments this field by one, so as you can see, these servers were hit 377 times. The delay specifies how much time, in milliseconds, that it took for the time provider to respond to the time request that was sent from the local system.

The offset column specifies the time difference between the local system time and the time on the time provider. This is in milliseconds, not seconds. This confuses a lot of folks initially. They see this and think, "Oh, wow, I'm 13 seconds off. I'm 12 seconds off. I'm 11 seconds off." No, you're actually 13 milliseconds off, or 12 milliseconds off, or whatever it shows here, which for all intents and purposes is synchronized.

And then jitter specifies the size of the time discrepancies. Again, this is also measured in milliseconds. You can also use the ntptrace command to monitor time synchronization. The ntptrace utility traces how the time consumer is getting its time from the time provider. It specifies the time provider stratum, it also lists the time offset between the local system and the time provider, and so on.

Summary 14:26-14:38

That's it for this lesson. In this lesson you learned how to use the NTPD daemon to synchronize time with other systems over a network connection. We discussed how the NTP protocol works, and then we discussed how to configure the NTPD daemon on a Linux system to get network time.
