

Introduction to NTP (Network Time Protocol)

What is NTP?

- All computers have an internal clock, including network devices
- It's important that these devices have an accurate clock that is synchronized with other devices
- That's the purpose of NTP

CCNA Exam Topic

- NTP is covered in exam topic 4.2, which says that you must be able to configure and verify NTP operating in client and server mode
- This video on NTP is the first in the section 4.0, IP Services

Video Outline

- First: briefly explain why time is important for network devices
- Second: show how to manually configure the time on a device, without using NTP
- Third: cover the basics of NTP, Network Time Protocol
- Fourth: show how to configure NTP on Cisco devices

Why Time is Important

Internal Clocks

- All devices have an internal clock, including routers, switches, PCs, and smartphones

Primary Importance

- From a CCNA perspective, the most important reason to have accurate time on a device is to have accurate logs for troubleshooting

Device Logs

- Devices keep logs of various events that occur, such as:
 - An interface being enabled or disabled
 - OSPF neighbor relationships being formed or broken
 - Device restarts
- Syslog, the protocol used to keep device logs, is a CCNA exam topic and will be covered in a later video

Viewing Time on Cisco Devices

Command: `show clock`

```
R1#show clock
*00:16:00.857 UTC Sat Dec 26 2020
```

The default time zone is UTC (Coordinated Universal Time).

- Used to view the current time on a Cisco IOS device
- Example output:
 - Time: 12:16 AM, 0 seconds, 857 milliseconds
 - Date: Saturday, December 26th 2020
 - Time zone: UTC (default)

Command: `show clock detail`

```
R1#show clock detail
*00:19:49.411 UTC Sat Dec 26 2020
Time source is hardware calendar
```

* = time is not considered authoritative

The hardware calendar is the default time source.

- Used to see the time source of the device
- Example output:
 - Time source: hardware calendar
- The hardware calendar is the built-in internal clock of the device
- This is the default time source

Asterisk in Output

- An asterisk (*) next to the time means the time is not considered authoritative
- This indicates the device is not confident that the time is accurate
- Reason: The internal hardware clock of a device will drift over time, so it is not the ideal time source

Viewing Device Logs

Command: `show logging`

- Used to view a device's logs

Example: Troubleshooting with Logs (R2 and R3)

Scenario

- A network admin receives a phone call around 1 AM reporting that the network connection has been unstable
- The admin logs into one of the devices to investigate by checking the logs

Investigation on R2

```
R2#show logging
!output abbreviated!
*Dec 27 00:50:20.005: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.122.192 on GigabitEthernet0/0 from LOADING to FULL,
Loading Done
*Dec 27 01:06:38.653: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:07:07.311: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done
*Dec 27 01:08:29.924: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from FULL to DOWN, Neighbor
Down: Dead timer expired
*Dec 27 01:09:10.714: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.6 on GigabitEthernet0/1 from LOADING to FULL,
Loading Done

R2#show clock
*01:17:06.706 UTC Sun Dec 27 2020
```

- Admin uses `show logging` on R2
- Output shows a series of messages about OSPF neighbor relationships
- Multiple messages about neighbor 10.0.0.6 moving to the **FULL** state and to the **DOWN** state
- Timestamps indicate these events occurred from about 1:06 AM to 1:09 AM

- Admin then checks R2's clock with `show clock`, which shows the current time is 1:17 AM on December 27th, 2020

Investigation on R3 (10.0.0.6)

```
R3#show logging
!output abbreviated!
May 23 16:24:17.320: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done
May 23 16:25:08.758: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from FULL to DOWN, Neighbor
Down: Interface down or detached
May 23 16:25:10.714: %LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to down
May 23 16:25:11.716: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to down
May 23 16:26:14.976: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
May 23 16:26:15.977: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
May 23 16:26:20.618: %OSPF-5-ADJCHG: Process 1, Nbr 10.0.0.5 on GigabitEthernet0/0 from LOADING to FULL, Loading
Done

R3#show clock
16:30:37.020 UTC Fri May 23 2008
```

- Admin logs into R2's neighbor 10.0.0.6, which is R3, to investigate further
- Admin uses `show logging` on R3
- Output shows:
 - The same messages about an OSPF neighbor moving between the `FULL` and `DOWN` states
 - Additional messages about an interface going `UP` and `DOWN`
- However, the timestamps show a completely different time than on R2
- Admin checks R3's clock and realizes R3 thinks it's 4:30 PM on May 23rd, 2008

Why This Is a Problem

- Mismatched timestamps make it much more difficult to correlate the logs of these devices
- In this case, it's a fairly simple issue — an interface going down caused problems, so the admin can investigate why that happened
- But when troubleshooting more complex issues on devices with logs full of thousands of messages, it gets much more difficult and accurate time is very important

Manually Configuring the Software Clock

Clock vs. Calendar

- The software clock and the hardware calendar are separate and can be configured separately

- The hardware calendar is the built-in internal clock of the device

Command: `clock set`

```
R2#clock set ?  
hh:mm:ss Current Time  
  
R2#clock set 14:30:00 ?  
<1-31> Day of the month  
MONTH Month of the year  
  
R2#clock set 14:30:00 27 ?  
MONTH Month of the year  
  
R2#clock set 14:30:00 27 Dec ?  
<1993-2035> Year  
  
R2#clock set 14:30:00 27 Dec 2020 ?  
<cr>  
  
R2#clock set 14:30:00 27 Dec 2020  
R2#show clock detail  
14:30:05.887 UTC Sun Dec 27 2020  
Time source is user configuration
```

- Used to manually configure the time on the device (the software clock)
- Syntax: `clock set hh:mm:ss <day> <month> <year>`
- Note: You can enter the day and month in either order
- Note: This command is done from **privileged-exec mode**, not global config mode
- Note: These clock configurations are not part of the device's running configuration; they are separate

Example: Setting the Clock

```
R2#clock set ?  
hh:mm:ss Current Time  
  
R2#clock set 14:30:00 ?  
<1-31> Day of the month  
MONTH Month of the year  
  
R2#clock set 14:30:00 27 ?  
MONTH Month of the year  
  
R2#clock set 14:30:00 27 Dec ?  
<1993-2035> Year  
  
R2#clock set 14:30:00 27 Dec 2020 ?  
<cr>  
  
R2#clock set 14:30:00 27 Dec 2020  
R2#show clock detail  
14:30:05.887 UTC Sun Dec 27 2020  
Time source is user configuration
```

- The professor entered `clock set` and used the `?` to see options
- He entered the time in hours:minutes:seconds format
- The next option is either the day of the month or the month itself — you can enter them in either order
- The professor entered the day first, then the month
- The next option is the year
- The professor entered the year 2020 and checked the options
- **CR** (carriage return) appeared — this basically means press the Enter key to execute the command
- There were no other options after the year

Verification

```
R2#clock set ?
hh:mm:ss Current Time
R2#clock set 14:30:00 ?
<1-31> Day of the month
MONTH Month of the year
R2#clock set 14:30:00 27 ?
MONTH Month of the year
R2#clock set 14:30:00 27 Dec ?
<1993-2035> Year
R2#clock set 14:30:00 27 Dec 2020 ?
<cr>
R2#clock set 14:30:00 27 Dec 2020
R2#show clock detail
14:30:05.887 UTC Sun Dec 27 2020
Time source is user configuration
```

- After setting the clock, the professor used `show clock detail`
- Output showed the newly configured time
- The time source changed to **user configuration**

Configuring the Hardware Clock (Calendar)

Terminology

- From now on, the term "clock" refers to the clock in the software of the device
- The term "calendar" refers to the hardware clock

Command: `calendar set`

```
R2#calendar set 14:35:00 ?  
  <1-31> Day of the month  
  MONTH Month of the year  
  
R2#calendar set 14:35:00 27 ?  
  MONTH Month of the year  
  
R2#calendar set 14:35:00 27 Dec ?  
  <1993-2035> Year  
  
R2#calendar set 14:35:00 27 Dec 2020 ?  
  <cr>  
  
R2#calendar set 14:35:00 27 Dec 2020  
R2#show calendar  
14:35:07 UTC Sun Dec 27 2020
```

- Used to manually configure the hardware clock
- The command syntax is the same as `clock set`, only the keyword `calendar` is different
- The order of options is:
 1. Time
 2. Either the day of the month or the month itself (you can enter them in either order)
 3. The year

Example: Setting the Hardware Clock

- The professor set the day, then the month
- Then set the year

Verification

- Use `show calendar` to view the hardware clock time

Synchronizing the Clock and Calendar

Why Synchronize?

- Typically you want to synchronize the clock and calendar
- The professor states he can't think of a good reason not to sync them

Command: `clock update-calendar`

- Syncs the calendar to the clock's time
- The calendar will update its time to match the clock

Command: `clock read-calendar`

- Syncs the clock to the calendar's time
- The clock will update its time to match the calendar

Demonstration: `clock update-calendar`

```
R2#show clock
14:38:14.301 UTC Sun Dec 27 2020
R2#show calendar
00:00:03 UTC Sun Dec 27 2020
R2#clock update-calendar
R2#show clock
14:38:22.181 UTC Sun Dec 27 2020
R2#show calendar
14:38:23 UTC Sun Dec 27 2020
```

- The professor viewed R2's clock with `show clock`, and the time was about 2:38 PM on December 27th, 2020 (this is the correct time).
- The calendar was about 12:00 AM (incorrect).
- The professor used the `clock update-calendar` command.
- The calendar was updated to match the time of the clock.

Demonstration: `clock read-calendar`

```
R2#show clock
00:00:15.788 UTC Mon Sep 6 1993
R2#show calendar
14:55:07 UTC Sun Dec 27 2020
R2#clock read-calendar
R2#show clock
14:55:12.522 UTC Sun Dec 27 2020
R2#show calendar
14:55:15 UTC Sun Dec 27 2020
```

- The professor showed the opposite situation where the calendar is correct but the clock is incorrect.
- The clock showed about 12:00 AM on September 6th, 1993 (incorrect).
- The calendar showed 2:55 PM on December 27th, 2020 (correct).
- The professor used the `clock read-calendar` command.
- The clock updated its time to match the calendar's time.

Configuring the Time Zone

Command: `clock timezone`

- Used to configure the time zone on the device
- This command is done from **global configuration mode** (unlike `clock set` and `calendar set` which are from privileged-exec mode)
- The time zone configuration becomes part of the device's running configuration

Syntax Walkthrough

```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
WORD name of time zone

R2(config)#clock timezone JST ?
<-23 - 23> Hours offset from UTC

R2(config)#clock timezone JST 9 ?
<0-59> Minutes offset from UTC
<cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

- First, the professor used `do show clock` to view the current time from global config mode
- The time was about 3:13 PM in the default time zone of UTC
- Note: The professor used the `do` prefix because he was in global config mode
- The first option is the name of the time zone — this is just a word, the device doesn't check if it's a real time zone name
- The next option is the hours offset from UTC
- There is an optional minutes offset for time zones that are offset by partial hours

Example: Configuring JST (Japan Standard Time)


```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
WORD name of time zone

R2(config)#clock timezone JST ?
<-23 - 23> Hours offset from UTC

R2(config)#clock timezone JST 9 ?
<0-59> Minutes offset from UTC
<cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

- The professor configured JST, which is 9 hours ahead of UTC
- He entered the command `clock timezone JST 9`
- No minutes offset was needed for JST

What Happened After Configuring

```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
WORD name of time zone

R2(config)#clock timezone JST ?
<-23 - 23> Hours offset from UTC

R2(config)#clock timezone JST 9 ?
<0-59> Minutes offset from UTC
<cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

- The time zone changed from UTC to JST

```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
WORD name of time zone

R2(config)#clock timezone JST ?
<-23 - 23> Hours offset from UTC

R2(config)#clock timezone JST 9 ?
<0-59> Minutes offset from UTC
<cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

- The time itself jumped 9 hours ahead — this is because the previous time configurations were done in UTC, so when changing to JST (9 hours ahead), the clock automatically adjusted

```
R2(config)#do show clock
15:13:33.985 UTC Sun Dec 27 2020
R2(config)#clock timezone ?
WORD name of time zone

R2(config)#clock timezone JST ?
<-23 - 23> Hours offset from UTC

R2(config)#clock timezone JST 9 ?
<0-59> Minutes offset from UTC
<cr>

R2(config)#clock timezone JST 9
R2(config)#do show clock
00:13:45.414 JST Mon Dec 28 2020
R2(config)#do clock set 15:15:00 Dec 27 2020
R2(config)#do show clock
15:15:02.129 JST Sun Dec 27 2020
```

- The actual current time was about 3:15 PM JST, so the professor adjusted the clock once more to display the correct time in the correct time zone

Note:

- The time zone is important because NTP only uses UTC, so you must adjust each device to the correct local time zone

Daylight Saving Time (Summer Time)

Introduction

 Canada	Northern America	Northern	Second Sunday March at 02:00 local standard time (for most of Canada)	First Sunday November at 02:00 local daylight saving time (for most of Canada)
--	------------------	----------	---	--

- Daylight saving time is also known as summer time, depending on the country
- Not all countries do this, but in many countries clocks are adjusted forward once a year and then back once a year
- You can configure Cisco devices to do this automatically

Command: `clock summer-time`

```

R2(config)#clock summer-time ?
WORD name of time zone in summer
R2(config)#clock summer-time EDT ?
date Configure absolute summer time
recurring Configure recurring summer time
R2(config)#clock summer-time EDT recurring ?
<1-4> Week number to start
first First week of the month
last Last week of the month
<cr>
R2(config)#clock summer-time EDT recurring 2 ?
DAY Weekday to start
R2(config)#clock summer-time EDT recurring 2 Sunday ?
MONTH Month to start
R2(config)#clock summer-time EDT recurring 2 Sunday March ?
hh:mm Time to start (hh:mm)
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 ?
<1-4> Week number to end
first First week of the month
last Last week of the month
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 ?
DAY Weekday to end
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday ?
MONTH Month to end
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday November ?
hh:mm Time to end (hh:mm)
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday November 02:00 ?
<1-1440> Offset to add in minutes
<cr>
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday November 02:00

```

 Canada	Northern America	Northern	Second Sunday 02:00 local (for most of)
--	------------------	----------	---

- Used to configure automatic daylight saving time adjustments
- This command is done from **global configuration mode**

Example: Canada Daylight Saving Time

- In most parts of Canada:
 - Clocks are set forward one hour on the second Sunday of March at 2 AM
 - Clocks are set back one hour on the first Sunday of November at 2 AM

Syntax Walkthrough

```

R2(config)#clock summer-time ?
WORD name of time zone in summer
R2(config)#clock summer-time EDT ?
date Configure absolute summer time
recurring Configure recurring summer time

```

 Canada	Northern America	Northern	Second Sunday 02:00 local
--	------------------	----------	------------------------------

- The first option is the name of the time zone:
 - The professor's time zone in Canada is EST (Eastern Standard Time)
 - During daylight saving time, the name changes to EDT (Eastern Daylight Time), so that's what he configured
- Next, you can either set a specific date with the keyword **date**, or use **recurring** (the more useful option)

```
R2(config)#clock summer-time EDT recurring ?
<1-4>  Week number to start
first  First week of the month
last   Last week of the month
<cr>
```

- **recurring** makes summer time start and end on the specified days every year
- After **recurring**, specify the start of daylight saving time:

```
R2(config)#clock summer-time EDT recurring 2 Sunday March ?
hh:mm  Time to start (hh:mm)
```

- Week of the month: 2 (second week)
- Weekday: Sunday
- Month: March
- Time: 2 AM
- Then specify the end of daylight saving time in the same order:

```
R2(config)#clock summer-time EDT recurring 2 Sunday March 02:00 1 Sunday November ?
hh:mm  Time to end (hh:mm)
```

- Week of the month: 1 (first week)
- Weekday: Sunday
- Month: November
- Time: 2 AM
- Optionally you can specify the offset, but the default is 60 minutes (which is what most countries use)

Example Command Output

```
R2(config)#$ recurring 2 Sunday March 02:00 1 Sunday November 02:00 ?  
  <1-1440> Offset to add in minutes  
  <cr>  
R2(config)#$ recurring 2 Sunday March 02:00 1 Sunday November 02:00
```

- The full command is too long for one line in the terminal
- Dollar signs (\$) in the output indicate that some of the output is cut off and can't be displayed on a single line
- The full command:

```
clock summer-time EDT recurring 2 Sunday March 2:00 1 Sunday November 2:00
```

Summary of Manual Time Configuration Commands

```
R1# show clock  
R1# show clock detail  
R1# clock set hh:mm:ss {day|month} {month|day} year  
R1# show calendar  
R1# calendar set hh:mm:ss {day|month} {month|day} year  
  
R1(config)# clock timezone name hours-offset [minutes-offset]  
R1(config)# clock summer-time recurring name start end [offset]
```



- Here is a summary of the manual time commands we just covered
- The `clock summer-time` command is long, so the format is simplified in review
- Note: For the start and end of daylight saving time, you must specify the week, weekday, month, and time

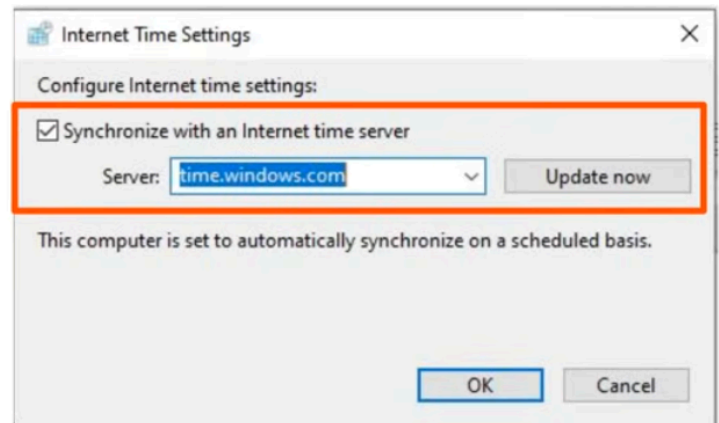
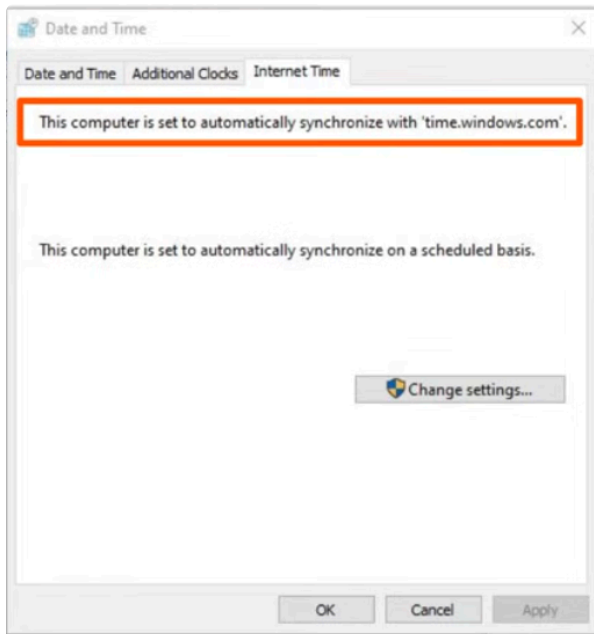
Quick Reference

Command	Purpose
<code>show clock</code>	Displays the current software time.
<code>show clock detail</code>	Displays detailed clock information (time source, time zone, DST).
<code>clock set</code>	Manually sets the software clock.
<code>show calendar</code>	Displays the hardware calendar.
<code>calendar set</code>	Manually sets the hardware calendar.
<code>clock timezone</code>	Configures the router's time zone.
<code>clock summer-time recurring</code>	Configures automatic daylight saving time.

Why NTP?

- Manually configuring the time on devices is not scalable.
- In a large network, manually configuring the time on every router, every switch, firewall, PC, phone, etc., would be a huge task and a huge waste of time.
- Manually configured clocks will drift, resulting in inaccurate time.
- NTP allows automatic syncing of time over a network to NTP servers.

NTP Example (Windows PC)



- The device you are using to watch this video is almost certainly using NTP.
- Example: On the professor's Windows 10 PC, the computer is set to automatically synchronize with `time.windows.com`.
- This is an NTP server somewhere on the Internet.
- This is a configurable setting.
- Example: If you wanted the PC to synchronize to a different NTP server (e.g., Google's server), you can change it in the settings.

DNS Preview (Separate Topic)

- This point is separate from the main NTP topic
- Domain names like `time.windows.com` are resolved to IP addresses using the protocol **DNS**
- DNS is an important CCNA topic and will be covered in a future video

Example: NSLOOKUP on Windows CLI

```
C:\Users\user>nslookup time.windows.com
Server:  dns.google
Address:  8.8.8.8
```

- The professor used the command `nslookup time.windows.com` in the CLI of his Windows PC
- The PC contacted the DNS server being used, which is Google's DNS server with IP address 8.8.8.8

```
Non-authoritative answer:
Name:      time.microsoft.akadns.net
Address:   20.43.94.199
Aliases:   time.windows.com
```

- Google's server gave the answer: the actual IP address of the Windows NTP server is 20.43.94.199

Example: NSLOOKUP for Google's NTP Server

```
C:\Users\user>nslookup time.google.com
Server:  dns.google
Address:  8.8.8.8
```

- The professor used `nslookup time.google.com`

```
Non-authoritative answer:
Name:      time.google.com
Addresses: 2001:4860:4806::
           2001:4860:4806:c::
           2001:4860:4806:8::
           2001:4860:4806:4::
           216.239.35.12
           216.239.35.8
           216.239.35.4
           216.239.35.0
```

- The PC asked the Google DNS server for the IP address of `time.google.com`
- The answer showed four IPv6 addresses and four IPv4 addresses used by `time.google.com`
- Note: DNS will be covered in detail in another video; this was just a quick preview

How NTP Works

NTP Clients and Servers

- NTP clients request the time from NTP servers
- Clients then synchronize their time to the time of the server
- It's possible for a device to be an NTP server and client at the same time
 - Example: A device syncs its time to a server, and then other devices sync their time to it

NTP Accuracy

- These are rough numbers, they can vary
- NTP allows accuracy of time within about 1 millisecond if the NTP server is in the same LAN
- NTP allows accuracy of time within about 50 milliseconds if connecting to the NTP server over a WAN or the Internet

NTP Stratum

- Some NTP servers are "better" than others
- The "distance" of an NTP server from the original reference clock is called **stratum**
- The farther away from the reference clock, the higher the stratum
- If the stratum level of a server is high, it is considered less accurate

NTP Port

- NTP uses UDP port 123 to communicate
- Remember this, in addition to the ones covered in the TCP & UDP Lecture

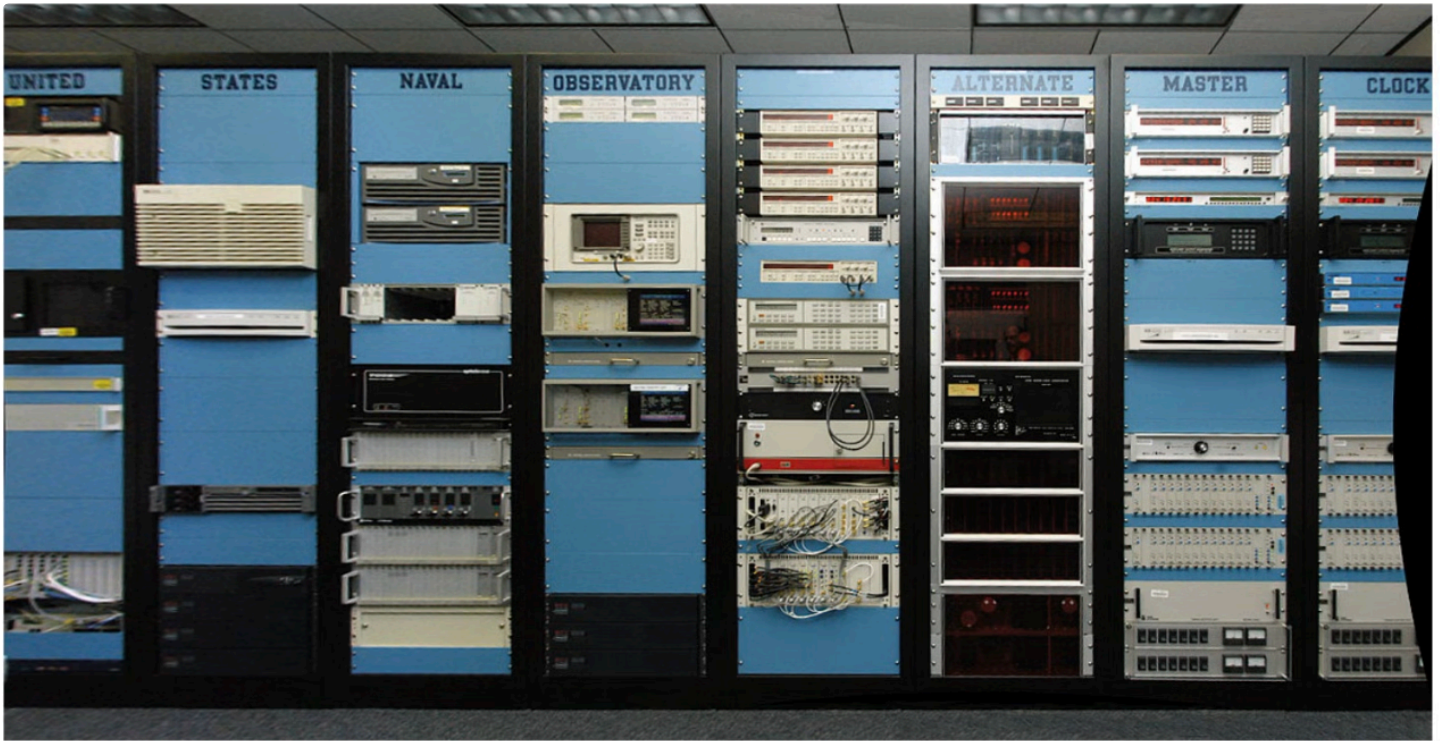
NTP Reference Clocks

What is a Reference Clock?

- A reference clock is usually a very accurate time device like an atomic clock or a GPS clock
- Reference clocks are **stratum 0** within the NTP hierarchy

- They are as close to the time source as possible, because they are the time source
- NTP servers directly connected to reference clocks are **stratum 1**

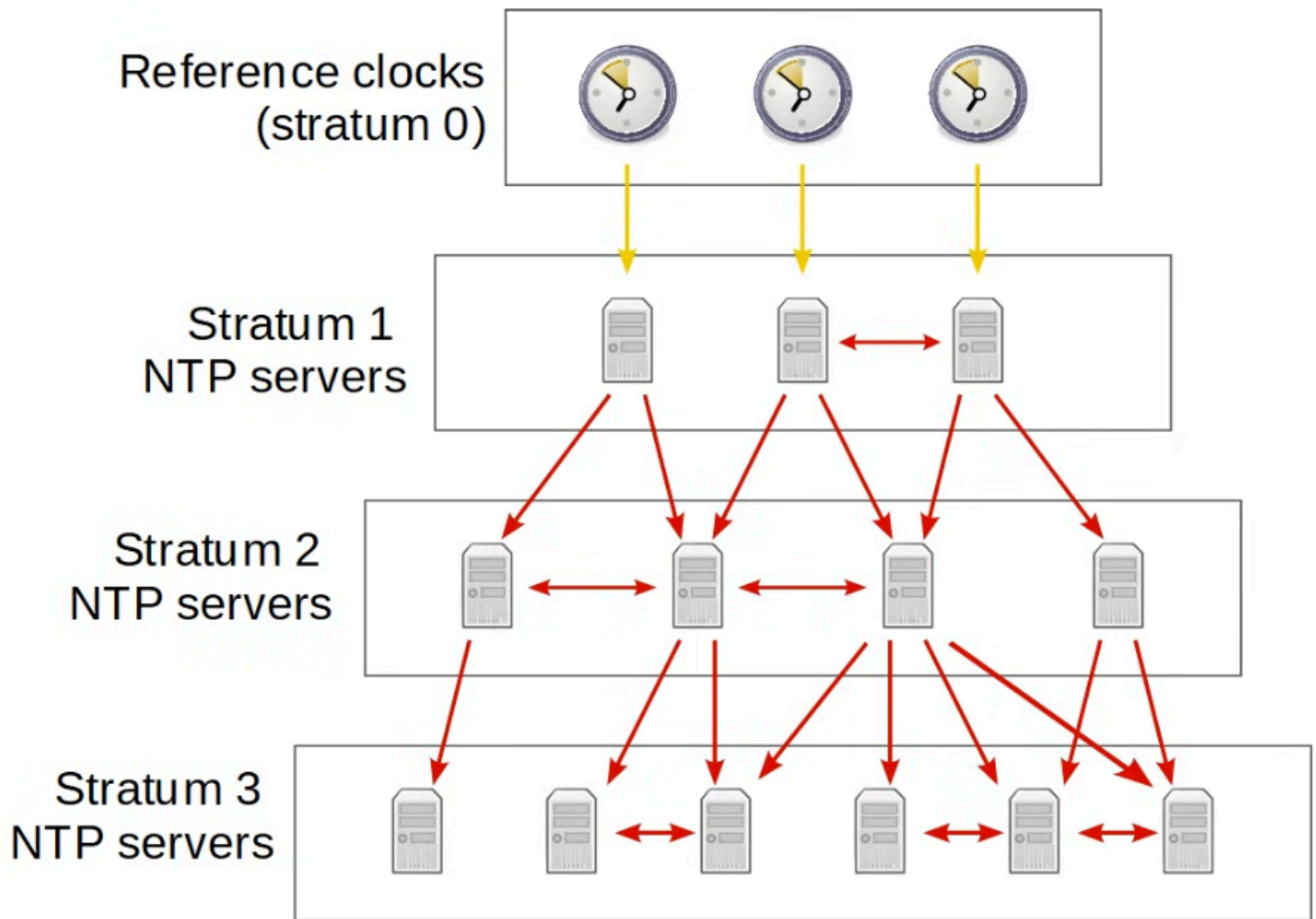
Example: Reference Clock



- The professor showed an example of a reference clock from a United States Naval Observatory
- Cisco devices aren't able to get their time directly from a stratum 0 reference clock like this
- However, they can get their time from a stratum 1 NTP server

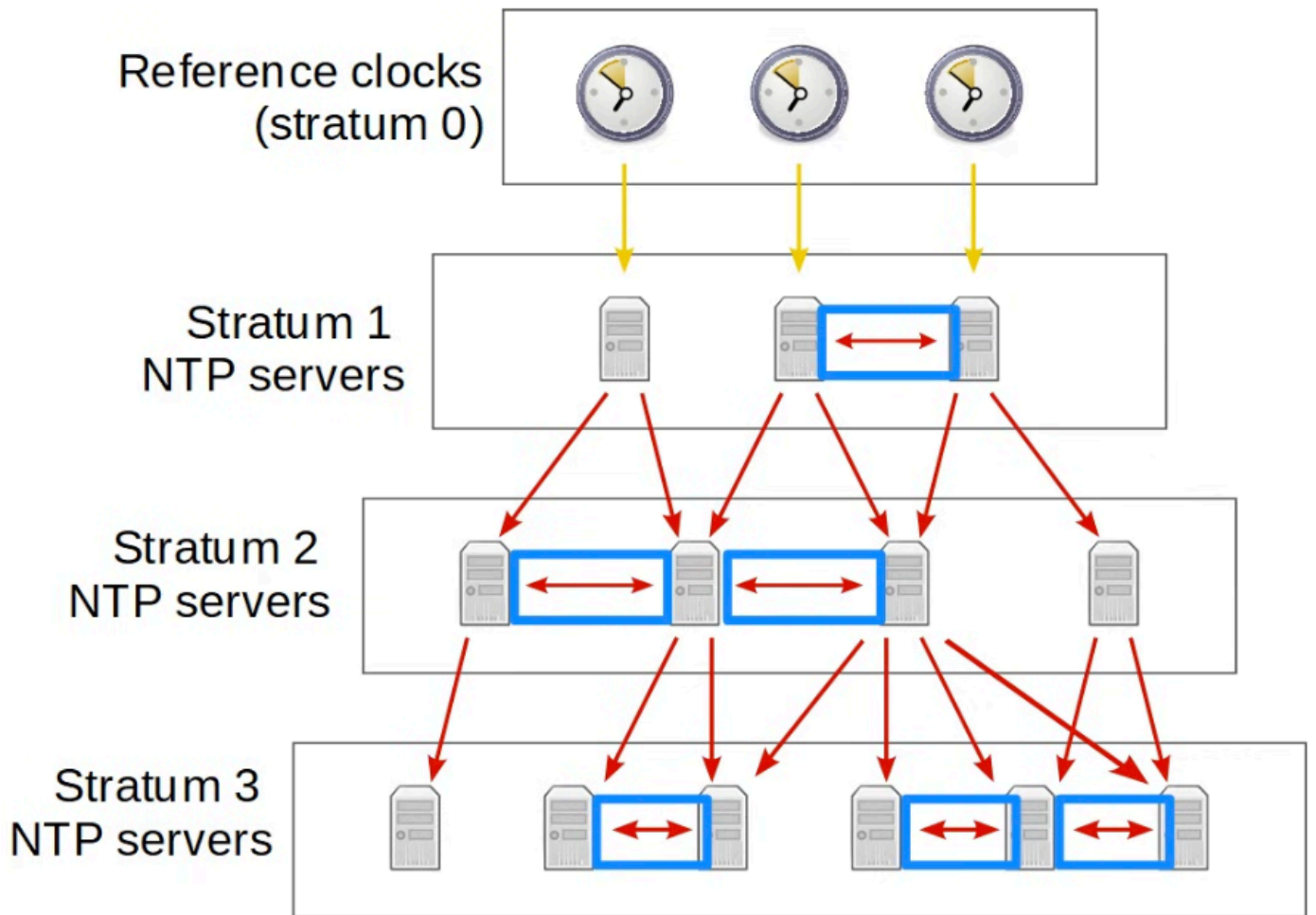
NTP Hierarchy and Peering

The NTP Hierarchy



- Reference clocks are **stratum 0** — they are the original time sources (e.g., atomic clocks, GPS clocks, such as the US Navy clock)
- NTP servers directly connected to reference clocks are **stratum 1** servers
- **Stratum 2** NTP servers get their time from stratum 1 servers
- **Stratum 3** servers get their time from stratum 2 servers
- This pattern continues down the hierarchy
- **Stratum 15** is the maximum — anything above that is considered unreliable and the device will not synchronize to it

NTP Peering (Symmetric Active Mode)



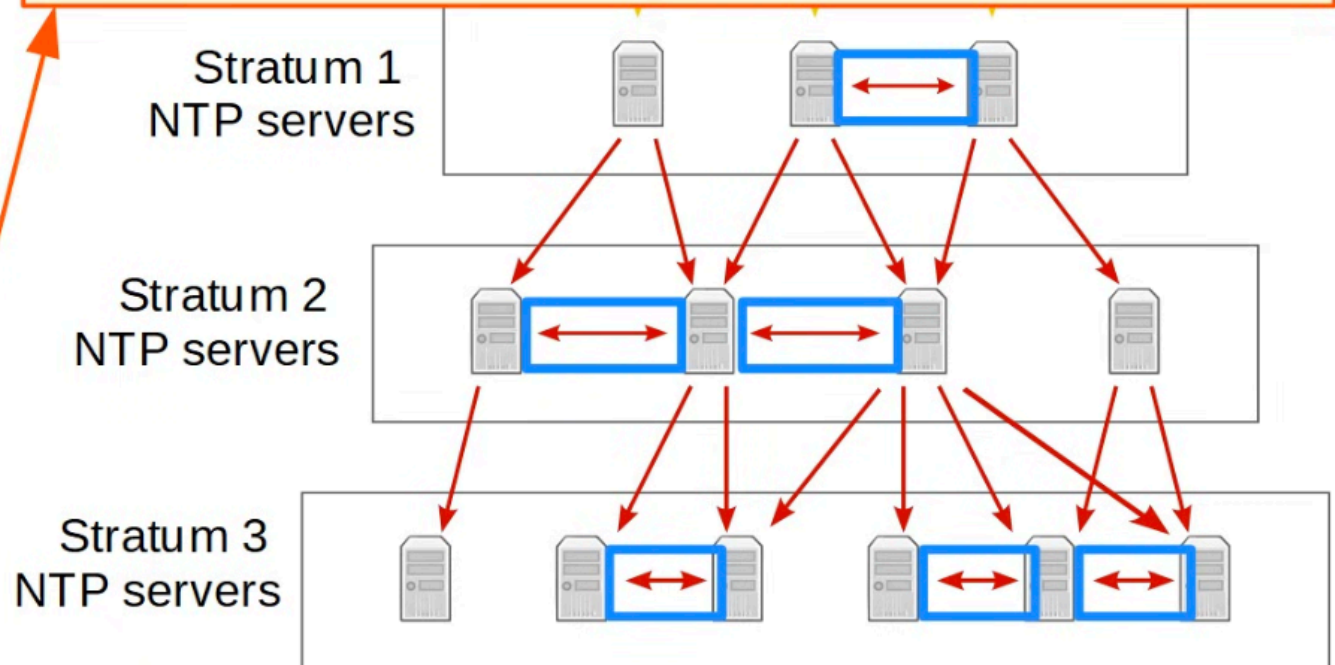
- Devices can peer with devices at the same stratum to provide more accurate time
- This also acts as a backup in case they lose access to the lower-stratum NTP server
- This mode is called **symmetric active mode**

Cisco NTP Modes

- Cisco devices can operate in three NTP modes:

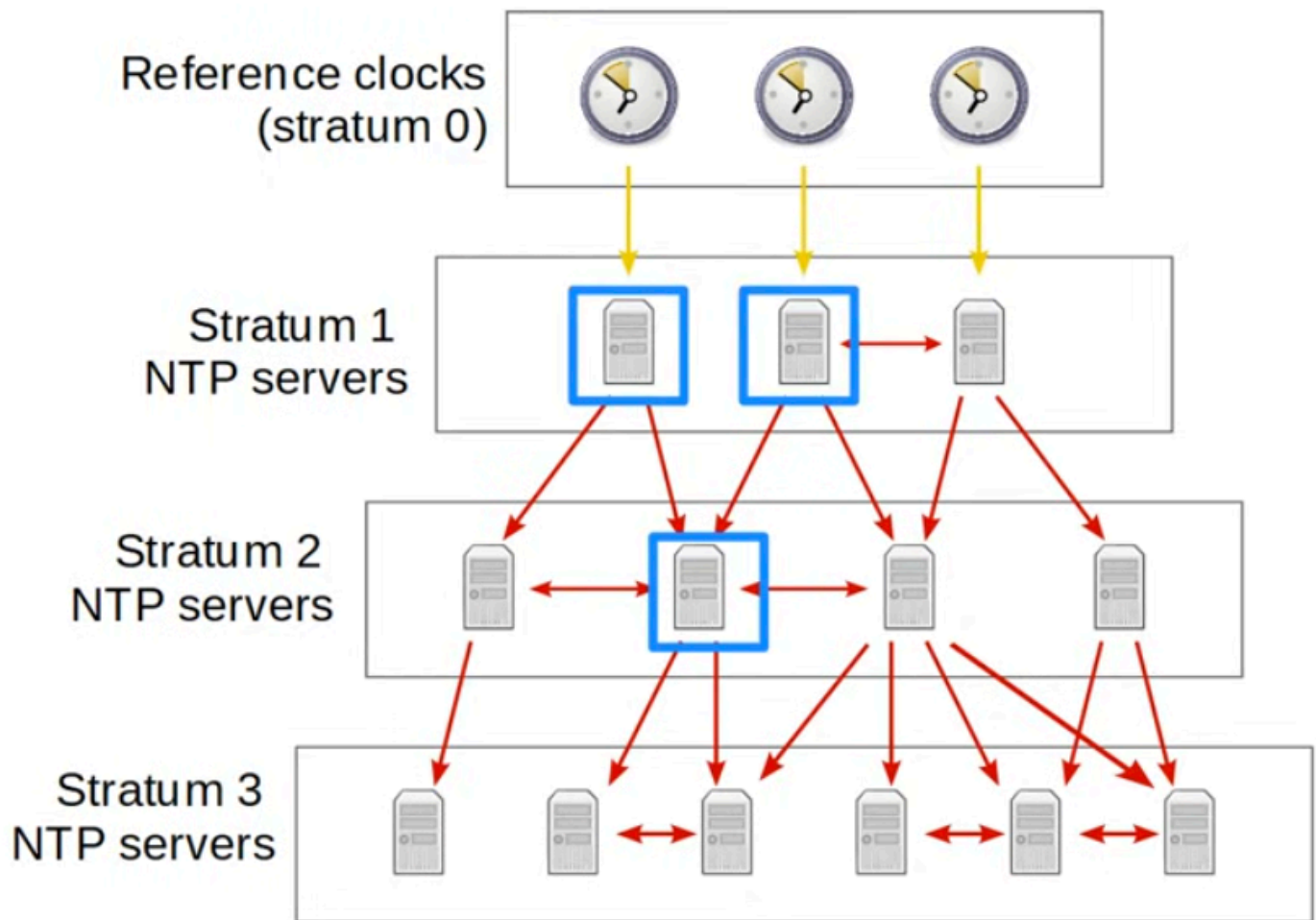
This is called 'symmetric active' mode.
Cisco devices can operate in three NTP modes:

- Server mode
- Client mode
- Symmetric active mode



1. Server mode
 2. Client mode
 3. Symmetric active mode
- A device can be in all three of those modes at the same time

Syncing to Multiple Servers

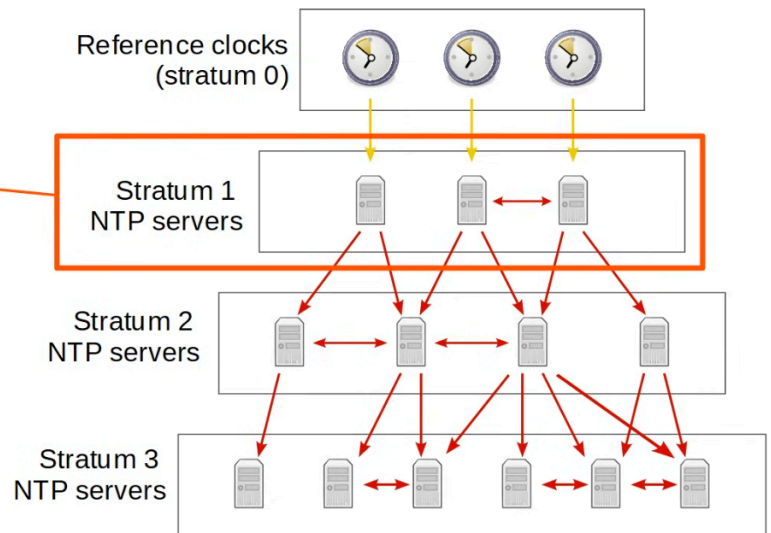


- An NTP client can sync to multiple servers
- Example: In the diagram, a stratum 2 server is getting its time from two stratum 1 servers

NTP Terminology

Primary Servers

NTP servers which get their time directly from reference clocks are also called **primary servers**.

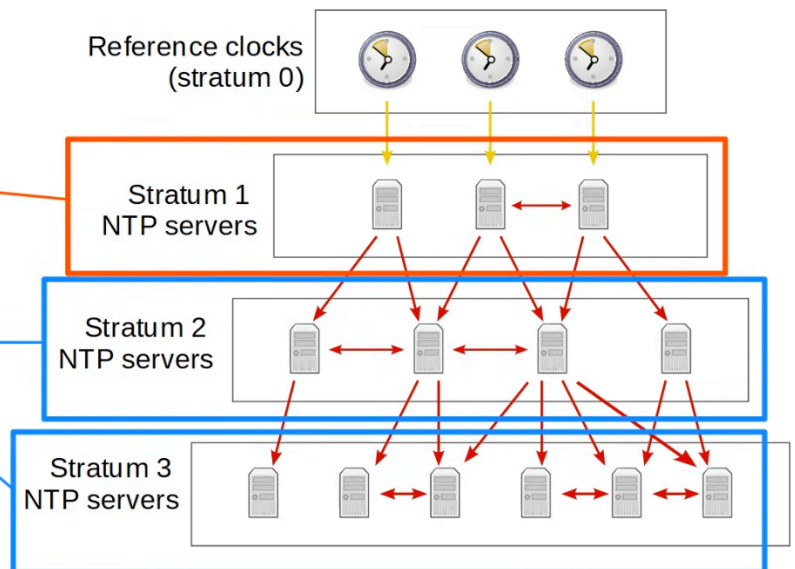


- NTP servers which get their time directly from reference clocks
- They sync their time directly to a reference clock
- They are stratum 1 servers

Secondary Servers

NTP servers which get their time directly from reference clocks are also called **primary servers**.

NTP servers which get their time from other NTP servers are called **secondary servers**. They operate in server mode and client mode at the same time.

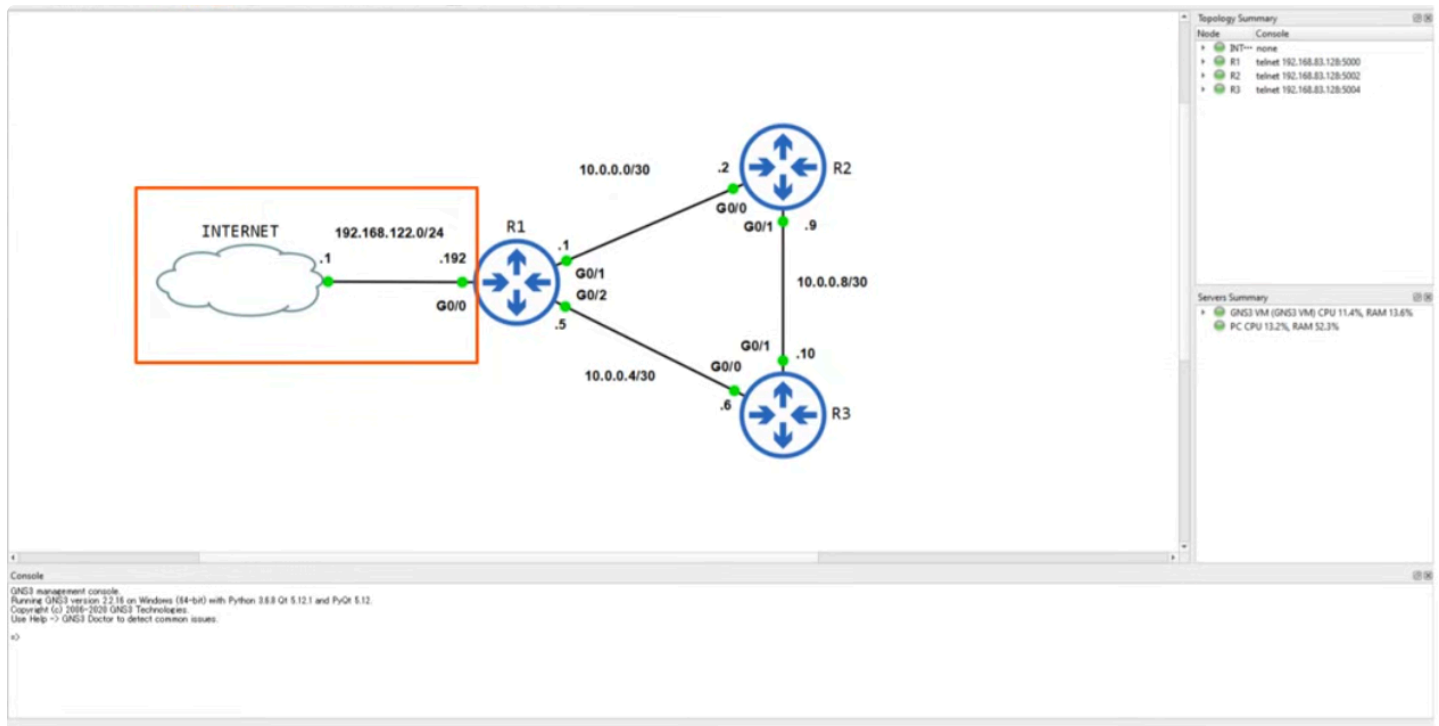


- NTP servers which get their time from other NTP servers
- They operate in server mode and client mode at the same time

- Servers with stratum 2 or above are secondary servers

Configuring NTP Client Mode (Syncing to Google's Servers)

Topology Setup



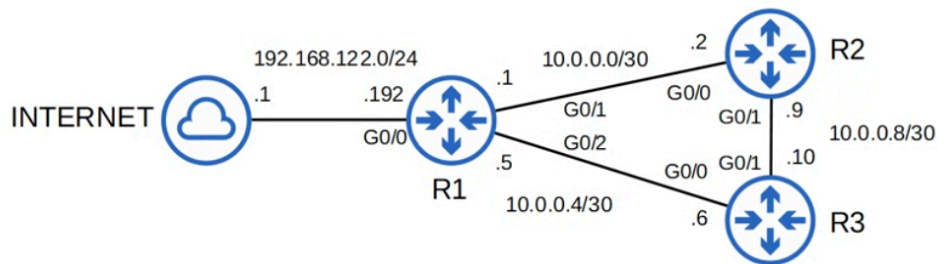
- The professor uses a GNS3 topology with an Internet cloud
- The virtual routers are actually connected to the real Internet via GNS3 (this is not available in simulators like Packet Tracer)
- Note: The point-to-point connection between R1 and the Internet uses a /24 prefix length because that is the default configuration for that cloud in GNS3. For a real connection to an ISP, a /30 or /31 would be used.

Configuration

```
C:\Users\user>nslookup time.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:     time.google.com
Addresses: 2001:4860:4806::
           2001:4860:4806:c::
           2001:4860:4806:8::
           2001:4860:4806:4::
           216.239.35.12
           216.239.35.8
           216.239.35.4
           216.239.35.0
```

```
R1(config)#ntp server 216.239.35.0
R1(config)#ntp server 216.239.35.4
R1(config)#ntp server 216.239.35.8
R1(config)#ntp server 216.239.35.12
```



- R1 will be configured to sync to Google's NTP servers over the Internet
- The professor uses the four IPv4 addresses resolved via `nslookup time.google.com`
- The command to configure an NTP server is `ntp server`, followed by the server IP address
- The order of the commands does not matter
- R1 will ask all servers for the time and select the one that gives the best, quickest responses
- The selected server may change if the current one slows down or stops responding
- It is best to specify multiple NTP servers for reliability

The PREFER Keyword

```
C:\Users\user>nslookup time.google.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:     time.google.com
Addresses: 2001:4860:4806::
           2001:4860:4806:c::
           2001:4860:4806:8::
           2001:4860:4806:4::
           216.239.35.12
           216.239.35.8
           216.239.35.4
           216.239.35.0
```

```
R1(config)#ntp server 216.239.35.0
R1(config)#ntp server 216.239.35.4
R1(config)#ntp server 216.239.35.8
R1(config)#ntp server 216.239.35.12
```

- You can add **prefer** to the end of the **ntp server** command to manually force the device to prefer a specific server
- Example: **ntp server 216.239.35.0 prefer** makes that server the preferred one, with others as backups
- The professor notes that he did not use the **prefer** keyword for this demonstration

Commands (from Global Config Mode)

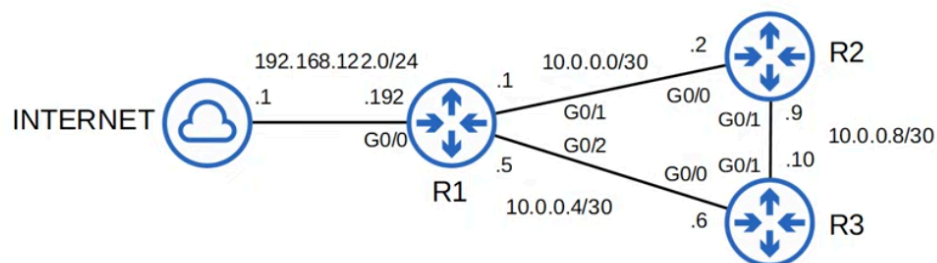
```
R1(config)# ntp server 216.239.35.0
R1(config)# ntp server 216.239.35.1
R1(config)# ntp server 216.239.35.4
R1(config)# ntp server 216.239.35.12
```

Verifying NTP Associations

```
R1#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~216.239.35.0	.GOOG.	1	43	64	17	62.007	1401.54	0.918
+~216.239.35.8	.GOOG.	1	43	64	17	64.220	1416.65	0.939
+~216.239.35.4	.GOOG.	1	47	64	17	57.669	1402.11	0.916
+~216.239.35.12	.GOOG.	1	39	64	17	62.229	1409.03	0.960

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured



Command: **show ntp associations**

- This is a very important show command for NTP.

- It displays all configured NTP servers.

Understanding the Output Symbols:

```
R1#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~216.239.35.0	.GOOG.	1	43	64	17	62.007	1401.54	0.918
+~216.239.35.8	.GOOG.	1	43	64	17	64.220	1416.65	0.939
+~216.239.35.4	.GOOG.	1	47	64	17	57.669	1402.11	0.916
+~216.239.35.12	.GOOG.	1	39	64	17	62.229	1409.03	0.960

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

- **Asterisk (*)**: Next to an IP address (e.g., 216.239.35.0) and labeled `sys.peer`. This indicates the NTP server that the device is currently syncing to.
- **Plus Sign (+)**: Next to other servers. These are **candidates**, but the device is not currently syncing to them.
- **Tilde (~)**: Next to all configured servers. This simply means they were manually configured using the `ntp server` command.
- **Other States**: If a server is marked as an `outlyer` or `falseticker`, the device will not sync to it (e.g., it believes the time is inaccurate). The details of these states are beyond the CCNA scope.

Understanding Stratum:

```
R1#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~216.239.35.0	.GOOG.	1	43	64	17	62.007	1401.54	0.918
+~216.239.35.8	.GOOG.	1	43	64	17	64.220	1416.65	0.939
+~216.239.35.4	.GOOG.	1	47	64	17	57.669	1402.11	0.916
+~216.239.35.12	.GOOG.	1	39	64	17	62.229	1409.03	0.960

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

- The `reference clock` column shows the source for each NTP server.
- All configured servers in the example use Google's reference clock, which is stratum 0.
- In the `st` (stratum) column, all four servers display a stratum level of 1.

Changing Associations:

```
R1#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
~216.239.35.0	.GOOG.	1	22	64	1	50.637	1.087	939.58
~216.239.35.8	.GOOG.	1	19	64	1	60.279	-4.402	939.83
*~216.239.35.4	.GOOG.	1	20	64	1	63.205	1351.20	938.52
~216.239.35.12	.GOOG.	1	20	64	1	49.130	1343.72	938.34

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

- When the command is run again **show ntp associations**, the output may change.
- Example: R1 may select 216.239.35.4 as the server it wants to sync to.
- This selection changes constantly as R1 interacts with the servers and determines which one is the best to sync to at that moment.

Verifying NTP Status

Command: `show ntp status`

- This is another useful show command for NTP.
- There is a lot of information in the output that you don't need to focus on, but the top line is important.

Output Breakdown:

```
R1#show ntp status
Clock is synchronized, stratum 2, reference is 216.239.35.12
nominal freq is 1000.0003 Hz, actual freq is 999.5003 Hz, precision is 2**14
ntp uptime is 295800 (1/100 of seconds), resolution is 1001
reference time is E393F0A9.1F758C5B (05:50:33.122 UTC Mon Dec 28 2020)
clock offset is 1343.7280 msec, root delay is 49.13 msec
root dispersion is 2275.31 msec, peer dispersion is 3.44 msec
loopfilter state is 'SPIK' (Spike), drift is 0.000499999 s/s
system poll interval is 64, last update was 173 sec ago.
```

- **Clock is synchronized:** This indicates that the synchronization was successful. It means at least one of the configured NTP servers was good and R1 was able to sync to it.

- **Stratum 2:** This is R1's stratum level. Because R1 is synchronizing its time to Google's NTP servers (which are stratum 1), R1 automatically becomes an NTP server itself with a stratum level 1 higher (stratum 2).
- **Reference Clock:** You can see the address of the reference clock here. Note that in the professor's example, the address shown is `.12`, which may be different from the server addresses seen in the `show ntp associations` output (like `.0` or `.4`).

Verifying Time and Calendar Status

```
R1(config)#do show clock detail
06:56:32.315 UTC Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
05:23:06 UTC Mon Dec 28 2020
R1(config)#clock timezone JST 9
R1(config)#ntp update-calendar
R1(config)#do show clock detail
15:57:33.078 JST Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
15:57:36 JST Mon Dec 28 2020
```

Checking the Software Clock

- The professor used the `do show clock detail` command.
- The time was correct, but the time zone was not.
- NTP uses only the UTC time zone, so you must manually configure the appropriate time zone on each device.
- The professor noted he hadn't configured R1's time zone yet.

Checking the Hardware Calendar

- The professor used the `do show calendar` command.
- Notice that the time was not synced up with the software clock.
- By default, NTP does not update the hardware calendar.

Synchronizing Calendar with NTP

```
R1(config)#do show clock detail
06:56:32.315 UTC Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
05:23:06 UTC Mon Dec 28 2020
R1(config)#clock timezone JST 9
R1(config)#ntp update-calendar
R1(config)#do show clock detail
15:57:33.078 JST Mon Dec 28 2020
Time source is NTP
R1(config)#do show calendar
15:57:36 JST Mon Dec 28 2020
```

Configuration

- The professor configured the time zone (JST) on R1.
- Then he used the `ntp update-calendar` command.
- This command configures the router to update the hardware clock (the calendar) with the time learned via NTP.

Verification

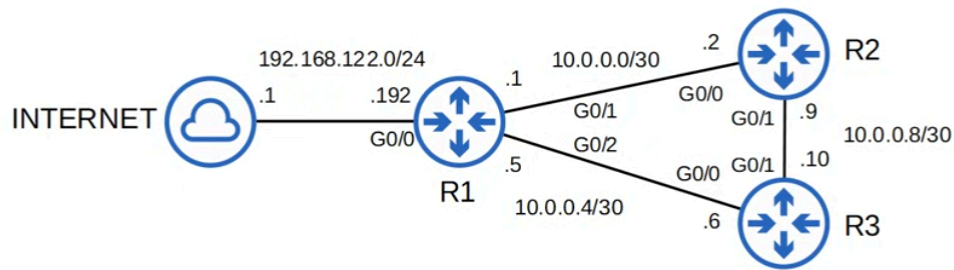
- The professor checked the clock and the calendar again, and now they are both synced.

Why Sync the Hardware Clock?

- The hardware clock tracks the date and time on the device even if it restarts or power is lost.
- When the system is restarted, the hardware clock is used to initialize the software clock.
- It is a good idea to keep the hardware clock accurate, although it is not totally essential.

Configuring NTP on R2

Topology Setup



- Usually in a small network like this you'd just configure all devices to sync to public NTP servers like Google's
- However, for this lesson, R2 will be configured to use R1 as an NTP server
- OSPF is configured in this network so all routers are sharing routes, including the route to R1's loopback interface

Configuration on R1

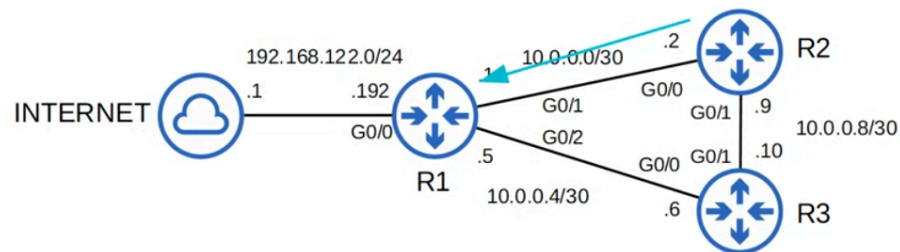
```
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
R1(config)#ntp source loopback0
```

- A loopback interface (Loopback0) is configured on R1 with address 10.1.1.1
- The `ntp source loopback0` command is configured on R1
- This tells R1 to use 10.1.1.1 as the source IP address for all NTP messages it sends

Why Use a Loopback Interface for NTP?

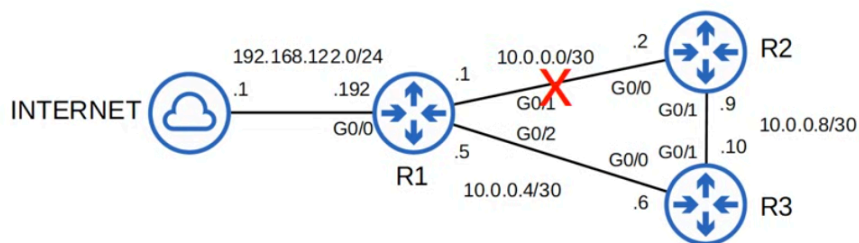
- If R2 was configured to use 10.0.0.1 (R1's physical G0/1 interface) as an NTP server:

```
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
R1(config)#ntp source loopback0
```



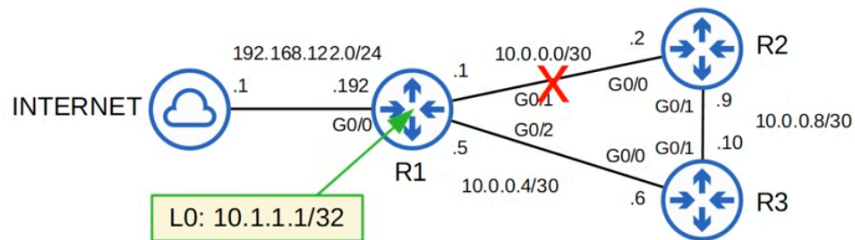
- In normal situations it would be able to send NTP request messages and get replies from R1

```
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#exit
R1(config)#ntp source loopback0
```

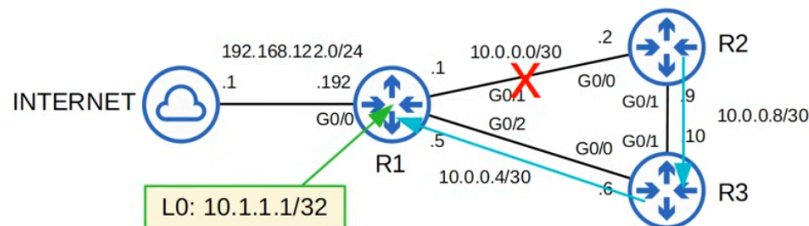


- But what if the interface failed for some reason?** R2 would suddenly lose its NTP server
- The availability of address 10.0.0.1 is dependent on the status of R1's G0/1 interface

- But what if we configure this loopback interface on R1 and tell R2 to use it's address? (e.g., 10.1.1.1 as it's NTP server):



- Even if the closest path to R1, via R2's G0/0 interface, is down, R3 will still advertise a route to 10.1.1.1 to R2

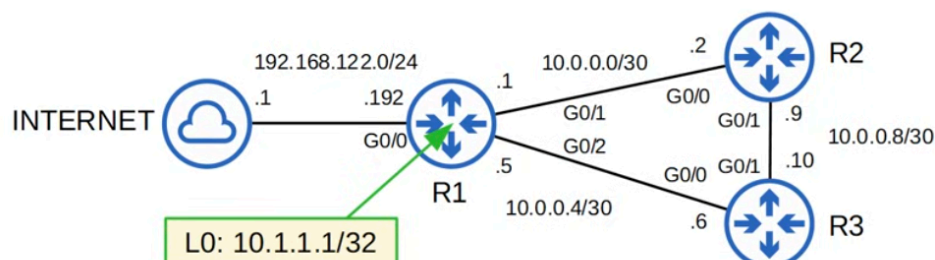


- R2 will still be able to communicate with its NTP server, R1
- Loopback interfaces are useful because they provide the router an address that isn't dependent on the status of any particular physical interface

Configuration on R2

```
R2(config)#ntp server 10.1.1.1
R2(config)#do show ntp associations

address      ref clock      st   when   poll reach  delay  offset  disp
*~10.1.1.1    216.239.35.12  2    0      64     1    7.038 -13.128 3937.5
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R2(config)#do show ntp status
Clock is synchronized, stratum 3, reference is 10.1.1.1
...
```



- The command `ntp server 10.1.1.1` is configured on R2 to sync to R1

Verification on R2

```
R2(config)#ntp server 10.1.1.1
R2(config)#do show ntp associations

  address      ref clock      st  when  poll reach  delay  offset  disp
*~10.1.1.1     216.239.35.12  2    0    64    1  7.038 -13.128 3937.5
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R2(config)#do show ntp status
Clock is synchronized, stratum 3, reference is 10.1.1.1
...
```

- The `do show ntp associations` command was used on R2
- Output showed an asterisk (*) next to 10.1.1.1, meaning R2 has selected R1 to sync to
- **R1's reference clock** is displayed as one of Google's NTP server addresses
- **R1's stratum** level is displayed as 2

Stratum Calculation

- Google's reference clock is stratum 0
- Google's NTP servers are stratum 1
- R1 is stratum 2 (syncd to stratum 1)
- R2's stratum is 3 (verified with `do show ntp status`), because it got its time from a stratum 2 server (R1)

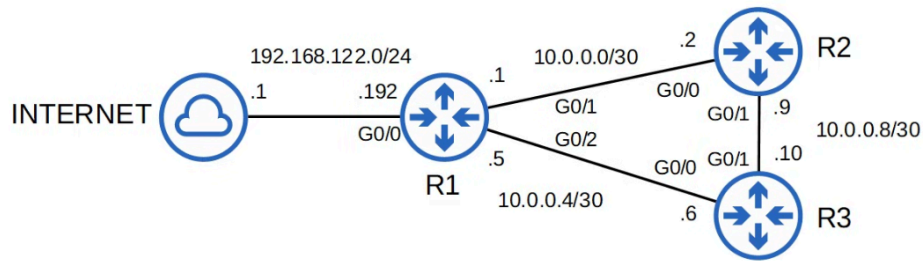
Note:

- Remember to use the `ntp source` command to specify the loopback interface as the source of NTP packets on R1; if you don't do that, R2 and R3 won't want to sync with R1.

Configuring NTP on R3

Configuration

```
R3(config)#ntp server 10.1.1.1
R3(config)#ntp server 10.2.2.2
```



- NTP is configured on R3
- A loopback interface is also configured on R2 with address 10.2.2.2
- Both R1 (10.1.1.1) and R2 (10.2.2.2) are configured as NTP servers for R3

Question: Which Server Will R3 Prefer?

- Between R1 (10.1.1.1) and R2 (10.2.2.2), R3 will prefer R1

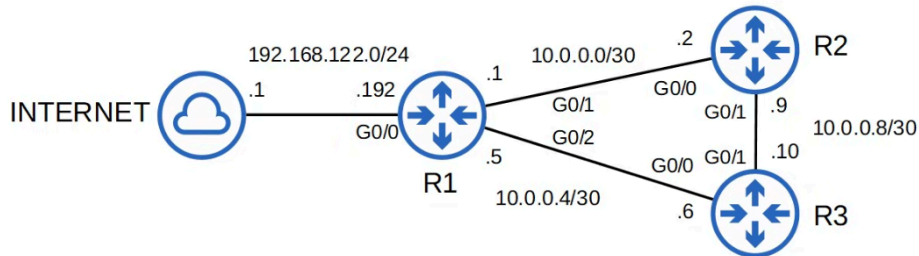
Why R1 is Preferred

```
R3(config)#ntp server 10.1.1.1
R3(config)#ntp server 10.2.2.2
R3(config)#do show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~10.1.1.1	216.239.35.0	2	1	64	0	0.000	0.000	15937.
~10.2.2.2	10.1.1.1	3	1	64	0	0.000	0.000	15937.

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

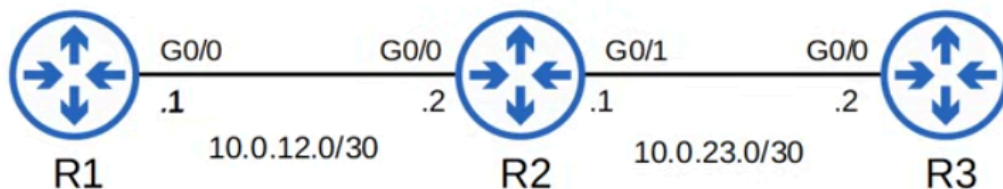
Servers with lower stratum levels are preferred.



- R1 is the preferred NTP server because of the stratum levels
- NTP servers with lower stratum levels are preferred, because they are closer to the source
- Therefore, they are considered more accurate

Configuring NTP Master Mode

Different Network Topology



- For the next few concepts, the professor uses a different network topology than the one used for syncing to Google's servers

When to Use NTP Master Mode

- If a device is already syncing to an NTP server (it's an NTP client), it automatically acts as an NTP server too
- But what if there is no NTP server to sync to?
- You probably still want the devices in the network to have the same time, even if it is slightly inaccurate compared to the actual time
- So how can you manually configure a Cisco device to operate as an NTP Server?
 - You can manually configure a Cisco device to operate as an NTP server even though it isn't synced to another NTP server

Command: `ntp master`

```
R1(config)#ntp ?
access-group      Control NTP access
allow             Allow processing of packets
authenticate      Authenticate time sources
authentication-key Authentication key for trusted time sources
broadcastdelay    Estimated round-trip delay
clock-period      Length of hardware clock tick
logging           Enable NTP message logging
master            Act as NTP master clock
max-associations  Set maximum number of associations
maxdistance       Maximum Distance for synchronization
mindistance       Minimum distance to consider for clockhop
orphan            Threshold Stratum for orphan mode
panic             Reject time updates > panic threshold (default 1000Sec)
passive           NTP passive mode
peer              Configure NTP peer
server            Configure NTP server
source            Configure interface for source address
trusted-key       Key numbers for trusted time sources
update-calendar   Periodically update calendar with NTP time
```

- Makes the device act as an NTP master clock
- You can optionally specify the stratum level
- If no stratum is specified, the default stratum level is used

Example: Configuring R1 as NTP Master

```

R1(config)#ntp master ?
<1-15> Stratum number
<cr>

R1(config)#ntp master
R1(config)#do show ntp associations

  address      ref clock      st   when   poll reach  delay  offset  disp
*~127.127.1.1  .LOCL.          7     2     16   377  0.000  0.000  0.292
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1(config)#do show ntp status
Clock is synchronized, stratum 8, reference is 127.127.1.1
...

```

- The professor used the `ntp master` command on R1 without specifying a stratum level
- He chose to just enter the command, which means it will use the default stratum level

Verification: `show ntp associations`

```

R1(config)#ntp master ?
<1-15> Stratum number
<cr>

R1(config)#ntp master
R1(config)#do show ntp associations

  address      ref clock      st   when   poll reach  delay  offset  disp
*~127.127.1.1  .LOCL.          7     2     16   377  0.000  0.000  0.292
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1(config)#do show ntp status
Clock is synchronized, stratum 8, reference is 127.127.1.1
...

```

- The address of R1's NTP server is now 127.127.1.1
- This is a loopback address
- R1 is using itself as its reference clock
- The stratum level of this server is 7

Loopback Addresses vs. Loopback Interfaces

- The entire 127.0.0.0/8 address range is reserved for loopback addresses
- Loopback addresses and loopback interfaces are different concepts, although the names are similar
- Loopback interfaces are virtual interfaces in the router and their addresses can be advertised to other devices using OSPF, etc.
- Loopback addresses are totally internal to the local device and can't be reached by other devices

Verification: `show ntp status`

```
R1(config)#ntp master ?
<1-15> Stratum number
<cr>

R1(config)#ntp master
R1(config)#do show ntp associations

  address      ref clock      st   when   poll reach  delay  offset  disp
*~127.127.1.1  .LOCL.         7     2    16   377  0.000  0.000  0.292
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
R1(config)#do show ntp status
Clock is synchronized, stratum 8, reference is 127.127.1.1
...
```

The default stratum of the **ntp master** command is 8.

- The actual stratum level of R1 is 8
- Remember that the default stratum of the `ntp master` command is 8

R2 and R3

```
R2(config)#ntp server 10.0.12.1
R2(config)#do show ntp associations

  address      ref clock      st   when   poll reach  delay  offset  disp
*~10.0.12.1    127.127.1.1    8     2    64    1  5.263  62.494 187.64
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

R3(config)#ntp server 10.0.12.1
R3(config)#do show ntp associations

  address      ref clock      st   when   poll reach  delay  offset  disp
*~10.0.12.1    127.127.1.1    8    45    64   17 21.534 -21.440 0.976
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

- R2 and R3 are configured to use R1 as their NTP server
- (Details of this configuration are the same as previously covered)

Configuring Symmetric Active Mode

When to Use Symmetric Active Mode

- R2 and R3 both have a stratum level of 9, so they are equals in terms of NTP

- They can become peers and help each other sync their time
- They also act as backups in case they lose contact with R1

Command: `ntp peer`

- Used to configure symmetric active mode
- Followed by the peer's address

Example: Configuring Peers on R2

```
R2(config)#ntp peer 10.0.23.2
```

```
R2(config)#do show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~10.0.12.1	127.127.1.1	8	60	64	17	24.040	206.682	0.987
~10.0.23.2	10.0.12.1	9	33	64	0	0.000	0.000	15937.

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```
R3(config)#ntp peer 10.0.23.1
```

```
R3(config)#do show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~10.0.12.1	127.127.1.1	8	11	64	37	12.605	-7.406	63.575
~10.0.23.1	10.0.12.1	9	1	64	0	0.000	0.000	15937.

* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

- On R2, the professor configured R3 as its peer using `ntp peer` with R3's address

- The entry for R3 in R2's NTP association table shows:

```
R2(config)#ntp peer 10.0.23.2
R2(config)#do show ntp associations

address          ref clock      st   when   poll reach  delay  offset  disp
*~10.0.12.1      127.127.1.1    8    60     64   17 24.040 206.682 0.987
~10.0.23.2      10.0.12.1     9    33     64    0 0.000  0.000 15937.
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R3(config)#ntp peer 10.0.23.1
R3(config)#do show ntp associations

address          ref clock      st   when   poll reach  delay  offset  disp
*~10.0.12.1      127.127.1.1    8    11     64   37 12.605 -7.406 63.575
~10.0.23.1      10.0.12.1     9     1     64    0 0.000  0.000 15937.
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

- R3's reference clock is R1 (10.0.12.1)
- R3's stratum level is 9 (because R1's is 8)

Example: Configuring Peers on R3

- The same configuration was done on R3, specifying R2 as the peer
- Again, the reference clock is R1 and the stratum level is 9

NTP Authentication

Is It on the Exam?

- It is not certain if this is on the exam; some study resources include it and some don't
- The professor recommends learning these few commands just in case

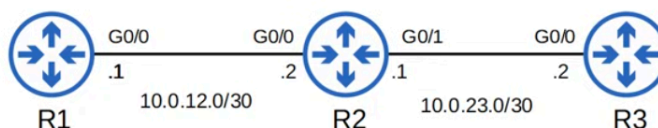
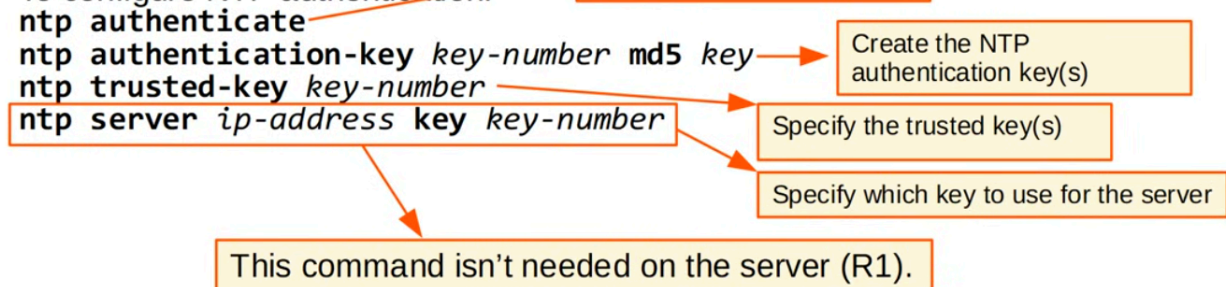
Overview

- NTP authentication is optional; you don't need to configure it
- It allows NTP clients to ensure that they only sync to the intended servers
- The clients check that the server is using the same password as them, and only sync if the passwords match

Configuration Steps (Cisco IOS)

- NTP authentication can be configured, although it is optional.
- It allows NTP clients to ensure they only sync to the intended servers.

- To configure NTP authentication:



- Ensure the server and clients are configured the same
- **Step 1:** Enable NTP authentication with the command `ntp authenticate`
- **Step 2:** Create the authentication key using `ntp authentication-key`
 - For the CCNA level, you only need to worry about one key (key number 1)
 - The `key-number` field is a number (use 1)
 - The `key` field is the password itself
- **Step 3:** Specify which key(s) are trusted using `ntp trusted-key`
 - Don't forget this step
- **Step 4:** Specify which key to use for each server using `ntp server <ip> key <key-number>`
 - This command is needed on the clients only
 - This command isn't needed on the server itself

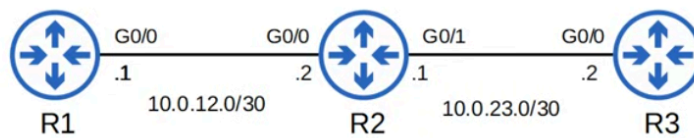
NTP Authentication Example Configurations

Common Configuration (R1, R2, and R3)

```
R1(config)#ntp authenticate
R1(config)#ntp authentication-key 1 md5 jeremysitlab
R1(config)#ntp trusted-key 1
```

```
R2(config)#ntp authenticate
R2(config)#ntp authentication-key 1 md5 jeremysitlab
R2(config)#ntp trusted-key 1
R2(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.2 key 1
```

```
R3(config)#ntp authenticate
R3(config)#ntp authentication-key 1 md5 jeremysitlab
R3(config)#ntp trusted-key 1
R3(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.1 key 1
```



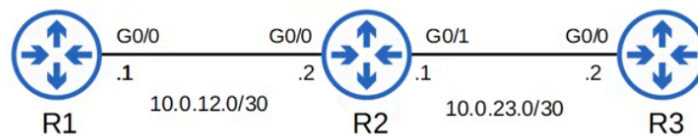
- On all three routers, the following commands were configured:
 - ntp authenticate**: Enables NTP authentication
 - ntp authentication-key 1 MD5 jeremysitlab**: Creates key number 1 with the password **jeremysitlab**
 - ntp trusted-key 1**: Specifies that key number 1 is trusted

Client Configuration (R2 and R3 only)


```
R1(config)#ntp authenticate
R1(config)#ntp authentication-key 1 md5 jeremysitlab
R1(config)#ntp trusted-key 1
```

```
R2(config)#ntp authenticate
R2(config)#ntp authentication-key 1 md5 jeremysitlab
R2(config)#ntp trusted-key 1
R2(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.2 key 1
```

```
R3(config)#ntp authenticate
R3(config)#ntp authentication-key 1 md5 jeremysitlab
R3(config)#ntp trusted-key 1
R3(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.1 key 1
```



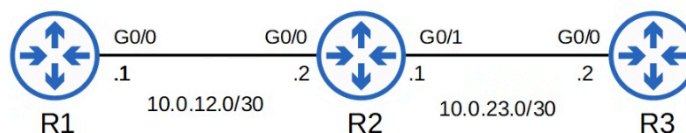
- On the clients R2 and R3 only, the following command was configured:
 - `ntp server 10.0.12.1 key 1`
 - This tells R2 and R3 to use key number 1 (the password `jeremysitlab`) to authenticate R1

Peer Authentication (R2 and R3)

```
R1(config)#ntp authenticate
R1(config)#ntp authentication-key 1 md5 jeremysitlab
R1(config)#ntp trusted-key 1
```

```
R2(config)#ntp authenticate
R2(config)#ntp authentication-key 1 md5 jeremysitlab
R2(config)#ntp trusted-key 1
R2(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.2 key 1
```

```
R3(config)#ntp authenticate
R3(config)#ntp authentication-key 1 md5 jeremysitlab
R3(config)#ntp trusted-key 1
R3(config)#ntp server 10.0.12.1 key 1
R2(config)#ntp peer 10.0.23.1 key 1
```



- One extra command was configured on R2 and R3:
 - `ntp peer <peer_address> key 1`
 - This adds authentication to the peer relationship between R2 and R3
- That's all you need to configure NTP authentication for the CCNA

Summary of Commands

!Basic Configuration Commands

```
R1(config)# ntp server ip-address [prefer]
R1(config)# ntp peer ip-address
R1(config)# ntp update-calendar
R1(config)# ntp master [stratum]
R1(config)# ntp source interface
```

!Basic Show Commands

```
R1# show ntp associations
R1# show ntp status
```

!Basic Authentication Commands

```
R1(config)# ntp authenticate
R1(config)# ntp authentication-key key-number md5 key
R1(config)# ntp trusted-key key-number
R1(config)# ntp server ip-address key key-number
R1(config)# ntp peer ip-address key key-number
```



- In addition to the manual time configuration commands, many NTP commands were covered
- The professor provided a review list of all the commands covered in the video
- If you don't remember any of them, go back in the video to review

Command	Purpose
<code>ntp server <ip></code>	Configures an NTP server.
<code>ntp server <ip> prefer</code>	Makes the server the preferred NTP source.
<code>ntp peer <ip></code>	Configures an NTP peer.
<code>ntp update-calendar</code>	Updates the hardware calendar from the software clock.
<code>ntp master [stratum]</code>	Makes the router an NTP server.
<code>ntp source <interface></code>	Sets the source interface for NTP packets.
<code>show ntp associations</code>	Displays NTP servers/peers and synchronization.
<code>show ntp status</code>	Displays NTP synchronization status.
<code>ntp authenticate</code>	Enables NTP authentication.
<code>ntp authentication-key</code>	Creates an MD5 authentication key.
<code>ntp trusted-key</code>	Marks a key as trusted.
<code>ntp server <ip> key <#></code>	Configures an authenticated NTP server.
<code>ntp peer <ip> key <#></code>	Configures an authenticated NTP peer.