NTP Configuration Guide

First Published: 2025-09-15

Last Modified: 2026-02-17

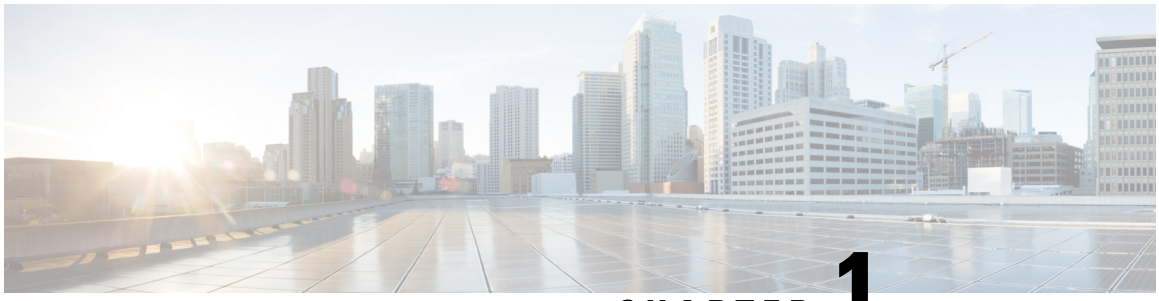
Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Read Me First

Only supported features are documented. To confirm or clarify all the supported features for a platform, go to Cisco Feature Navigator.

Read Me First

CONTENTS



CHAPTER 1

Network Time Protocol

Network Time Protocol

Network Time Protocol (NTP) is a networking protocol that synchronizes the clocks of devices in a network to Coordinated Universal Time (UTC). It plays a crucial role in ensuring that systems maintain accurate and synchronized time, which is essential for various applications such as time-stamping, logging, and network coordination.

NTP operates on UDP port 123 for accurate time synchronization, and even greater precision within a local area network (LAN). It uses a hierarchical stratum system to distribute time synchronization across devices.

A stratum refers to the level of hierarchy in the time synchronization process, indicating a device's distance from the primary reference clock (Stratum 0):

- **Stratum 0** : These are highly accurate reference clocks, such as atomic clocks, GPS clocks, or radio clocks. They are not network devices themselves but provide the most precise time source.
- **Stratum 1** : These are servers directly connected to a Stratum 0 reference clock. They act as the primary time sources for other devices in the network.
- **Stratum 2 and beyond** : These devices synchronize their clocks with a Stratum 1 server (or higher stratum server) over the network. As the stratum level increases, the device is further away from the reference clock, and the potential for reduced accuracy and increased latency grows.

- **Stratum 15** : The lowest level in the NTP hierarchy, indicating the furthest distance from the reference clock. Devices beyond Stratum 15 are considered unsynchronized.

The stratum system ensures a scalable and distributed approach to time synchronization, preventing overloading of primary time sources and maintaining a clear hierarchy.

Relevant RFCs for NTP

This topic provides an overview of the most important RFCs for NTP, highlighting their key contributions and recommendations.

Relevant RFCs for NTP

RFC 1119: Network Time Protocol (Version 2): This RFC describes NTP Version 2, providing a detailed specification of the protocol's operation and algorithms.

Key Contributions:

- Introduced algorithms for time synchronization, filtering, and selection.
- Discussed error analysis and precision modelling.
- Defined the hierarchical structure of stratum levels.

RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification: This RFC specifies NTP Version 4, the latest and most widely used version of the protocol.

Key Contributions:

- Enhanced security features.
- Improved synchronization accuracy and scalability.
- Support for IPv6.
- Backward compatibility with earlier versions.

RFC 7384: Security Requirements of Time Protocols in Packet Switched Networks: This RFC discusses the security requirements for time synchronization protocols, including NTP.

Key Contributions:

- Identifies potential vulnerabilities in time protocols.
- Provides recommendations for securing NTP against attacks such as spoofing and packet manipulation.

RFC 8633: Network Time Protocol Best Current Practices: Provides guidance and best practices for the deployment and operation of NTP.

Key Contributions:

- Recommendations for securing NTP deployments.
- Best practices for server configurations and client behavior.
- Guidance for mitigating attacks and ensuring robust time synchronization.

NTP Architecture and Protocol Fundamentals

NTP architecture consists of servers and clients that synchronize time across networked devices.

A NTP configured setup consists of the server and clients as part of the network.

NTP Server

- Provides accurate time to other devices.
- Can act as a Stratum 1, Stratum 2, or higher-level server depending on its source of synchronization.
- May obtain time from a higher-stratum server or a local reference clock (e.g., GPS).

NTP Client

- Requests time from an NTP server to synchronize its internal clock.
- Adjusts its clock based on the server's time, accounting for network delays.
- Can also operate in peer mode, where two devices synchronize with each other.

Peer Relationships

- Devices can be configured as peers to provide mutual backup and redundancy.
- If one peer loses access to a higher-stratum server, it can use the other peer as a fallback source.

How NTP Works

Time Synchronization Process: NTP synchronizes clocks by calculating and minimizing differences between a client's clock and the server's clock.

- NTP uses algorithms to adjust the client's clock, accounting for offset, delay, and jitter.

Summary

NTP involves several key actors and components in the time synchronization process.

- Client: Requests time synchronization from the server and adjusts its clock based on received data.
- Server: Provides accurate time information to clients and responds to polling requests.
- NTP Packet: Transports synchronization data, including offset, delay, and jitter measurements.

NTP synchronizes clocks by measuring offset, delay, and jitter, and dynamically polling servers to maintain accurate time.

Workflow

These stages describe how NTP synchronizes time between a client and a server.

1. **Clock Offset** : Measures the difference between the client's and server's clocks.
    - Offset is calculated by comparing timestamps from both client and server.

2. **Delay** : Calculates the time it takes for packets to travel between client and server.
   - Delay is determined by measuring round-trip time of NTP packets.
3. **Jitter** : Estimates variations in delay caused by network fluctuations.
   - Jitter is calculated by analyzing variations in delay over multiple NTP exchanges.
4. NTP applies algorithms to adjust the client's clock, accounting for offset, delay, and jitter.
   - The client's clock is corrected to match the server's time as closely as possible.
5. **Polling** : NTP regularly polls servers at configurable intervals to maintain synchronization.
   - Polling intervals are adjusted dynamically based on network conditions and clock stability.
6. **NTP Packet Structure** : NTP packets are compact and efficient, designed for minimal overhead.
   - **Leap Indicator** (LI): Indicates leap second adjustments.
   - **Version Number** (VN): Specifies the NTP version in use.
   - **Mode** : Determines whether the packet is from a client, server, or peer.
   - **Stratum** : Specifies the server's stratum level.
   - **Transmit Timestamp** : Contains the time the packet was sent by the server.
   - **Receive Timestamp** : Records the time the packet was received.

NTPv4

High Accuracy and Scalability

Accuracy:

- Achieves time synchronization within milliseconds over the Internet or microseconds in LAN environments.
- Implements advanced algorithms to improve precision and adapt to network latency.

Scalability:

- Supports large-scale networks with thousands of devices.
- Hierarchical structure minimizes load on primary servers.

Backward Compatibility with Earlier Versions

- NTPv4 is fully compatible with NTPv3 and earlier versions.
- Ensures seamless communication between devices using different NTP versions.

Advanced features of NTP

Orphan Mode

In some cases, an NTP subnet operates in isolation, disconnected from local reference clocks or Internet-based clock servers. During such periods of isolation, the servers and clients within the subnet must synchronize to a common time scale. Traditionally, this is achieved using a local clock driver, which simulates a UTC source to provide a shared time reference. A server connected directly or indirectly to this driver then synchronizes all other hosts in the subnet. However, relying on a local clock driver can result in critical, irrecoverable failures of the subnet. Additionally, maintaining redundancy using multiple servers may not always be feasible. To address these limitations, Orphan Mode provides a more robust solution, eliminating the need for a local clock driver. Orphan Mode enables multiple servers to simulate a single UTC source while ensuring seamless switching when servers recover from failures.

How Orphan Mode Works

In private networks, one or more core servers typically operate at the lowest stratum in the subnet hierarchy. These servers are configured as backups for one another using symmetric or broadcast modes. If one of these servers successfully reaches a UTC source, the entire subnet synchronizes to that server. If none of the servers can access a UTC source, Orphan Mode activates.

In Orphan Mode:

- A single server, known as the orphan parent, is selected to simulate the UTC source for the subnet.
- All other hosts in the subnet, referred to as orphan children, synchronize with this orphan parent.
- The mechanism ensures that the subnet continues to operate with a consistent time scale even in the absence of external time sources.

Configuring Orphan Mode

To enable a server for Orphan Mode, use the ntp orphan stratum command, where the stratum value must:

- Be less than 16.
- Be greater than the stratum values of any configured Internet time servers.

If no associations are configured for other servers or reference clocks, the orphan stratum value should be set to 1. It is important to ensure that all subnet hosts dependent on orphan children have stratum values less than 16, maintaining the proper hierarchy.

Behavior of Orphan Parents

- When an orphan parent operates at stratum 1 with no external sources, its reference ID is displayed as LOOP.
- If the orphan parent does not operate at stratum 1, it displays the UNIX loopback address 127.0.0.1.

Orphan Parent Selection

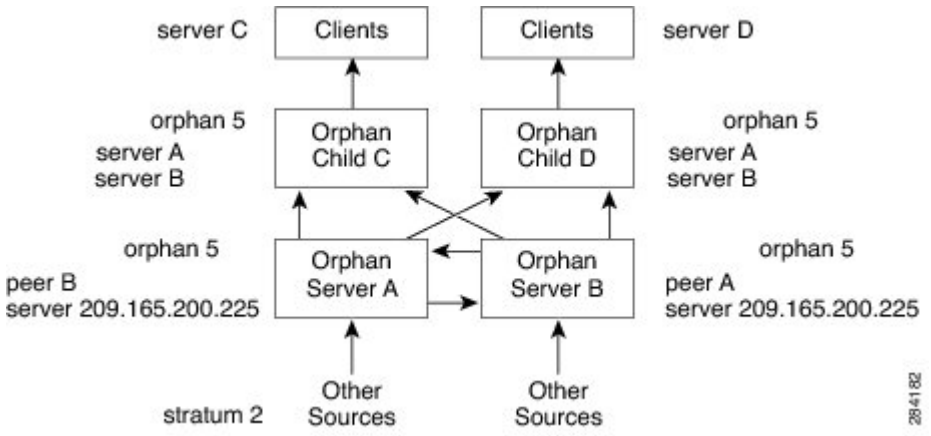
Unlike ordinary NTP clients, which select servers based on delay and dispersion metrics, orphan children use a unique metric based on the IP address of each core server in the subnet. The orphan parent is selected as the server with the smallest metric, ensuring a consistent root server for the subnet.

Redundancy and Continuous Synchronization

Even if a server loses all time sources, it continues to synchronize its local clock with other servers in the subnet. This approach ensures that the server remains backed up and aligned with the overall time scale.

The following figure showcases a typical Orphan Mode configuration in a peer network:

- Two primary or secondary (stratum 2) servers are configured with either reference clocks or public Internet primary servers.
- These servers use symmetric modes to back each other up and ensure seamless synchronization within the subnet.



Range for Trusted Key configuration

In NTP (Network Time Protocol), a trusted key is used for authentication purposes to ensure that time synchronization exchanges between an NTP client and server are secure and trustworthy. Trusted keys play an important role in preventing unauthorized devices or malicious actors from spoofing time synchronization data. The range for trusted key configuration refers to the key IDs that can be specified in the NTP configuration for authentication. Key IDs are typically integers, and their valid range depends on the specific NTP implementation being used.

Key ID Range for Trusted Keys

Standard Range:

- Key IDs are integers in the range of 1 to 65535.
- This range is consistent with most NTP implementations, including ntpd, the reference NTP implementation.

Configuration Context:

- Trusted keys are specified in the NTP configuration file using the trustedkey directive.
- You can list one or more key IDs as trusted keys.

NTP support for IPv6

Network Time Protocol (NTP) supports both IPv4 and IPv6, allowing time synchronization in networks using either protocol. Starting from NTPv4, which is specified in RFC 5905, NTP includes full support for IPv6. This is particularly important as more organizations adopt IPv6 to address the limitations of IPv4.

Key Features of NTP with IPv6 Support

1. Dual-Stack Support:
   - NTPv4 works in networks that use IPv4, IPv6, or both (dual-stack networks).
   - NTP servers and clients can operate over both IPv4 and IPv6 simultaneously.
2. Global Addressability: With IPv6, devices have unique global addresses, making it easier to configure NTP in large-scale and distributed environments.
3. Improved Network Efficiency: IPv6 multicast and anycast capabilities can be used to efficiently distribute time synchronization across a network.
4. Security: IPv6 improves overall network security through features like IPsec, which can also secure NTP traffic.

•

Advantages of Using NTP with IPv6

1. Scalability: IPv6 provides a virtually unlimited number of addresses, making it ideal for large-scale deployments.
2. Efficient Multicast: IPv6 multicast allows efficient time distribution, reducing the load on NTP servers.
3. Improved Network Efficiency: IPv6 multicast and anycast capabilities can be used to efficiently distribute time synchronization across a network.
4. Future-Proof: IPv6 adoption is increasing, and NTP's support for IPv6 ensures compatibility with future networks.

Challenges NTP IPv6

1. IPv6 Adoption: Not all networks are fully IPv6-enabled, and some devices may still rely on IPv4.
2. DNS Configuration: Ensure that DNS records (e.g., AAAA records) are correctly configured for hostnames used by NTP.

Guidelines for Configuring NTP

The Network Time Protocol (NTP) package contains a vulnerability that could allow an unauthenticated, remote attacker to cause a denial of service (DoS) condition. NTP versions 4.2.4p7 and earlier are vulnerable.

The vulnerability is due to an error in handling of certain malformed messages. An unauthenticated, remote attacker could send a malicious NTP packet with a spoofed source IP address to a vulnerable host. The host that processes the packet sends a response packet back to the transmitter. This action could start a loop of

messages between the two hosts that could cause both the hosts to consume excessive CPU resources, use up the disk space by writing messages to log files, and consume the network bandwidth. All of these could cause a DoS condition on the affected hosts.

Cisco software releases that support NTPv4 are not affected. All other versions of Cisco software are affected.

To display whether a device is configured with NTP, use the show running-config | include ntp command. If the output returns any of the following commands, then that device is vulnerable to the attack:

- ntp broadcast client
- ntp primary
- ntp multicast client
- ntp peer
- ntp server

There are no workarounds for this vulnerability other than disabling NTP on the device. Only packets destined for any configured IP address on the device can exploit this vulnerability. Transit traffic will not exploit this vulnerability.

Depending on your release, your feature will process NTP mode 7 packets and will display the message "NTP: Receive: dropping message: Received NTP private mode 7 packet " if debugs for NTP are enabled. Configure the ntp allow mode private command to process NTP mode 7 packets. This command is disabled by default.

Note NTP peer authentication is not a workaround and is a vulnerable configuration.

NTP services are disabled on all interfaces by default.

Networking devices running NTP can be configured to operate in a variety of association modes when synchronizing time with reference time sources. A networking device can obtain time information on a network in two ways: by polling host servers and by listening to NTP broadcasts.

Line Aux 0 option is disabled by default. When you configure both IP address and FQDN of the same NTP server in Cisco IOS XE, only the FQDN configuration is displayed in the show running-config command output after FQDN resolves to the same IP address.

Configuring Poll-Based NTP Associations

To configure poll-based NTP associations, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **ntp peer** *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**]
4. [**no**] **ntp server** [**vrf** *vrf-name*] *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**]
5. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **ntp peer** *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**] **Example:** `Device(config)# ntp peer 172.16.22.44 version 2`	Configures the device system clock to synchronize a peer or to be synchronized by a peer (peer association). • *ip-address*: The IP address of the peer providing or being provided, the clock synchronization. • *number*: NTP version number. The range is 1 to 4. By default, version 4 is selected. • *key-id*: Authentication key defined with the **ntp authentication-key** command. • *interface*: The interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. • **prefer**: Sets this peer as the preferred one that provides synchronization. This keyword reduces switching back and forth between peers. Use the **no** form of this command to remove a peer association.
Step 4	[**no**] **ntp server** [**vrf** *vrf-name*] *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**] **Example:** `Device(config)# ntp server 172.16.22.44 version 2`	Configures the device's system clock to be synchronized by a time server (server association). • *vrf-name*: The virtual routing and forwarding (VRF) address of the server providing the clock synchronization. **Note** Before you configure this command, the VRF must be configured. • *ip-address*: The IP address of the time server providing the clock synchronization.

	Command or Action	Purpose
		• *number*: NTP version number. The range is 1 to 4. By default, version 4 is selected. • *key-id*: Authentication key defined with the **ntp authentication-key** command. • *interface*: The interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. • **prefer**: Sets this peer as the preferred one that provides synchronization. This keyword reduces clock hop among peers. Use the **no** form of this command to remove a server association.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Broadcast-Based NTP Associations

To configure broadcast-based NTP associations, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. [**no**] **ntp broadcast** [**version** *number*] [**key** *key-id*] [*destination-address*]
5. [**no**] **ntp broadcast client**
6. **exit**
7. [**no**] **ntp broadcastdelay** *microseconds*
8. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Configures an interface and enters interface configuration mode.
Step 4	[**no**] **ntp broadcast** [**version** *number*] [**key** *key-id*] [*destination-address*] **Example:** `Device(config-if)# ntp broadcast version 2`	Enables the interface to send NTP broadcast packets to a peer. • *number*: NTP version number. The range is 1 to 4. By default, version 4 is used. • *key-id*: Authentication key. • *destination-address*: IP address of the peer that is synchronizing its clock to this switch. Use the **no** form of this command to disable the interface from sending NTP broadcast packets.
Step 5	[**no**] **ntp broadcast client** **Example:** `Device(config-if)# ntp broadcast client`	Enables the interface to receive NTP broadcast packets. Use the **no** form of this command to disable the interface from receiving NTP broadcast packets.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.
Step 7	[**no**] **ntp broadcastdelay** *microseconds* **Example:** `Device(config)# ntp broadcastdelay 100`	(Optional) Change the estimated round-trip delay between the device and the NTP broadcast server The default is 3000 microseconds. The range is from 1 to 999999. Use the **no** form of this command to disable the interface from receiving NTP broadcast packets.

	Command or Action	Purpose
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring NTP Authentication

To configure NTP authentication, perform this procedure:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. [**no**] **ntp authenticate**
4. [**no**] **ntp authentication-key** *number* {**md5** | **cmac-aes-128** | **hmac-sha1** | **hmac-sha2-256**} *value*
5. [**no**] **ntp trusted-key** *key-number*
6. [**no**] **ntp server** *ip-address* **key** *key-id* [**prefer**]
7. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **ntp authenticate** **Example:** `Device(config)# ntp authenticate`	Enables NTP authentication. Use the **no** form of this command to disable NTP authentication
Step 4	[**no**] **ntp authentication-key** *number* {**md5** | **cmac-aes-128** | **hmac-sha1** | **hmac-sha2-256**} *value*	Defines the authentication keys.

	Command or Action	Purpose
	Example: `Device(config)# ntp authentication-key 42 md5 aNiceKey`	• Each key has a key number, a type, and a value. • Keys can be one of the following types: • **md5**: Authentication using the MD5 algorithm. • **cmac-aes-128**: Authentication using Cipher-based message authentication codes (CMAC) with the AES-128 algorithm. The digest length is 128 bits and the key length is 16 or 32 bytes. • **hmac-sha1**: Authentication using Hash-based Message Authentication Code (HMAC) using the SHA1 hash function. The digest length is 128 bits and the key length is 1 to 32 bytes. • **hmac-sha2-256**: Authentication using HMAC using the SHA2 hash function. The digest length is 256 bits and the key length is 1 to 32 bytes Use the **no** form of this command to remove authentication key.
Step 5	[**no**] **ntp trusted-key** *key-number* **Example:** `Device(config)# ntp trusted-key 42`	Defines trusted authentication keys that a peer NTP device must provide in its NTP packets for this device to synchronize to it. Use the **no** form of this command to disable trusted authentication.
Step 6	[**no**] **ntp server** *ip-address* **key** *key-id* [**prefer**] **Example:** `Device(config)# ntp server 172.16.22.44 key 42`	Allows the software clock to be synchronized by an NTP time server. • *ip-address*: The IP address of the time server providing the clock synchronization. • *key-id*: Authentication key defined with the **ntp authentication-key** command. • **prefer**: Sets this peer as the preferred one that provides synchronization. This keyword reduces clock hop among peers. Use the **no** form of this command to remove a server association.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring an External Reference Clock

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line aux** *line-number*
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**line aux** *line-number* **Example:** `Device(config)# line aux 0`	Enters line configuration mode for the auxiliary port 0.
Step 4	**end** **Example:** `Device(config)# end`	Exits line configuration mode and returns to privileged EXEC mode.

Configuring Orphan Mode

Before you begin

To configure orphan mode, you would require at least two clients. The following task shows how to configure orphan mode on one client. Repeat the steps in the other client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ntp server** *ip-address*

4. **ntp peer** *ip-address*
5. **ntp orphan** *stratum*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ntp server** *ip-address* **Example:** `Device(config)# ntp server 10.1.1.1`	Forms a server association with another system.
Step 4	**ntp peer** *ip-address* **Example:** `Device(config)# ntp peer 172.16.0.1`	Forms a peer association with another system. **Note** Use an IP address that is different from the one you just configured, such as 172.16.0.2, while configuring the peer in the other client.
Step 5	**ntp orphan** *stratum* **Example:** `Device(config)# ntp orphan 4`	Enables orphan mode in the host.

Troubleshooting NTP

Proper time synchronization is critical for maintaining the integrity of network operations. However, issues with NTP can arise due to misconfigurations, network problems, or server issues. Below is a structured approach to troubleshooting NTP, including common issues, resolutions, and tools that can assist in identifying and resolving problems.

- Always ensure the system clock is reasonably close to the correct time before starting the NTP service (use date to check and set time manually if needed).
- Configure multiple upstream servers to provide redundancy and improve accuracy.
- Periodically monitor NTP performance and logs to identify potential issues early.