Summary: The Center for Alternative Synchronization and Timing (CAST) promotes resilient, terrestrial based time synchronization through protocols such as Precision Time Protocol (PTP) and Network Time Protocol (NTP) to maintain accuracy across the electric grid. This technical bulletin focuses specifically on secure NTP implementation, covering best practices, closed-loop timing architecture, and step-by-step configuration for both general-purpose and dedicated NTP servers, as well as clients. CAST reduces reliance on public time sources and ensures resilient, secure timekeeping across power system operations.

Secure NTP Implementation for Power System Synchronization

INTRODUCTION

Network Time Protocol (NTP) [1] , originally developed in the 1980s, remains one of the most widely adopted protocols for synchronizing clocks over Internet Protocol (IP)-based networks. It distributes time with millisecond-level accuracy across Ethernet-based systems and continues to be a standard in both enterprise and operational technology environments.

In the power sector, synchronized time is essential for coordinating activities across geographically distributed assets—from sensors and relays to control systems. Accurate timing supports critical functions such as real-time grid monitoring, event logging, and fault detection. NTP plays an important role in the execution of these functions by providing a reliable time reference across substations, control centers, and field equipment, ensuring consistent data correlation and system-wide operation coordination.

NTP BEST PRACTICES

Despite its widespread adoption, NTP has known security vulnerabilities, including man-in-the-middle attacks, distributed denial-of-service amplification, and masquerading, where malicious actors spoof legitimate time sources. These threats pose significant risks to the integrity and availability of time synchronization in critical infrastructures.

To mitigate these risks, best practices for secure NTP deployment include the following [2]:

- Using multiple, independent, and geographically diverse time sources to reduce reliance on a single provider.
- Regularly updating NTP software to incorporate security patches and protocol improvements.
- Monitoring and validating NTP time sources to detect anomalies and prevent tampering.
- Implementing authentication mechanisms (e.g., NTPv4 with symmetric keys) where possible.

By following these guidelines, organizations can enhance the resilience of their timing infrastructure and support the secure operation of power systems that depend on accurate and trustworthy time.

CLOSED-LOOP TIMING ARCHITECTURE

One important issue is enterprise timing architecture. Many organizations rely on public internet-based NTP sources—such as the National Institute of Standards and Technology, Google, or the NTP Pool Project—for time synchronization. The organizations configure firewalls to allow for NTP traffic (typically on UDP port 123) to and from these trusted providers. This configuration often routes many enterprise internal NTP clients through UDP 123 to maintain two-way connections with the external NTP servers for time synchronization. Such an enterprise timing architecture approach may cause timing delays and security vulnerabilities induced by traffic congestion and/or an open port to the internet providing an access vulnerability for malicious intent.

Alternatively, and as a best practice, rather than depending on eternal time sources over the internet, organizations can adopt a more secure and resilient closed-loop timing architecture, as illustrated in Figure 1. This architecture includes an internal server that acts as the local NTP source, syncing with a trusted internal time source and distributing accurate time across the internal network. This design centers around an internal NTP server (stratum-1) that derives time from a trusted, on-premises reference, such as a stratum-0 grandmaster clock (GMC), cesium clock, or a boundary clock (BC).

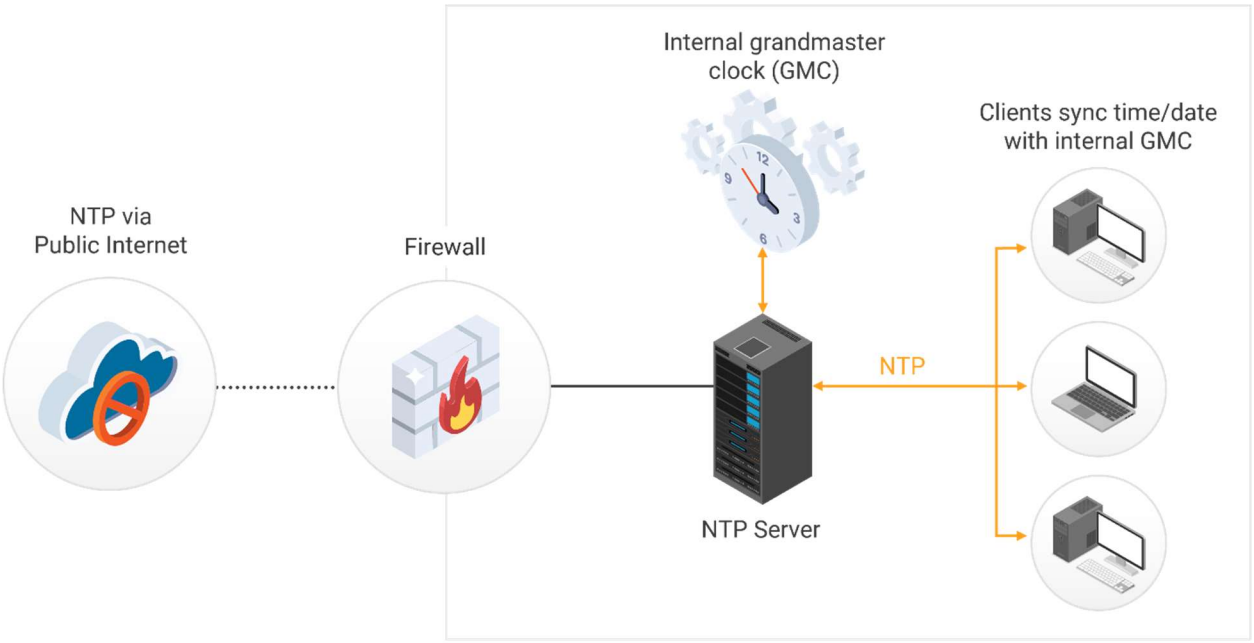

Figure 1. Closed-loop timing architecture with an internal NTP server.

By integrating an authoritative internal time source, organizations effectively bring timing and synchronization fully in control within their network perimeter, minimizing exposure to external threats and internet-based vulnerabilities. A GMC typically syncs to UTC via the Global Navigation Satellite System (GNSS) such as GPS. If GNSS synchronization is disrupted (e.g., due to jamming or spoofing), the GMC can fall back to a cesium atomic clock to maintain accurate timing independently. This approach provides a robust and deterministic timing infrastructure required for critical applications such as power system control, substation automation, and other operational technology systems where trust and continuity in time synchronization are paramount.

There are cost-effective options to establish an enterprise internal NTP server architecture. For example, lower end NTP servers might cost hundreds of dollars [3] [4], while higher end ADVA enterprise, miniature, or embedded servers range from hundreds up to thousands of dollars [5]. Standing up an internal NTP timing infrastructure is not necessarily resource intensive, and it could offer improvements to enterprise timing performance as well as reduce cybersecurity threats.

NTP Server Failover

NTP failover design allow for enterprise clients switching over to a redundant or standby NTP server seamlessly upon the failure or abnormal termination of one server in a

computer network. As illustrated in Figure 2, one approach to implement this fail-over enterprise timing architecture is to utilize two NTP servers, both synced to the same GMC, while sharing one single virtual IP for enterprise client access.

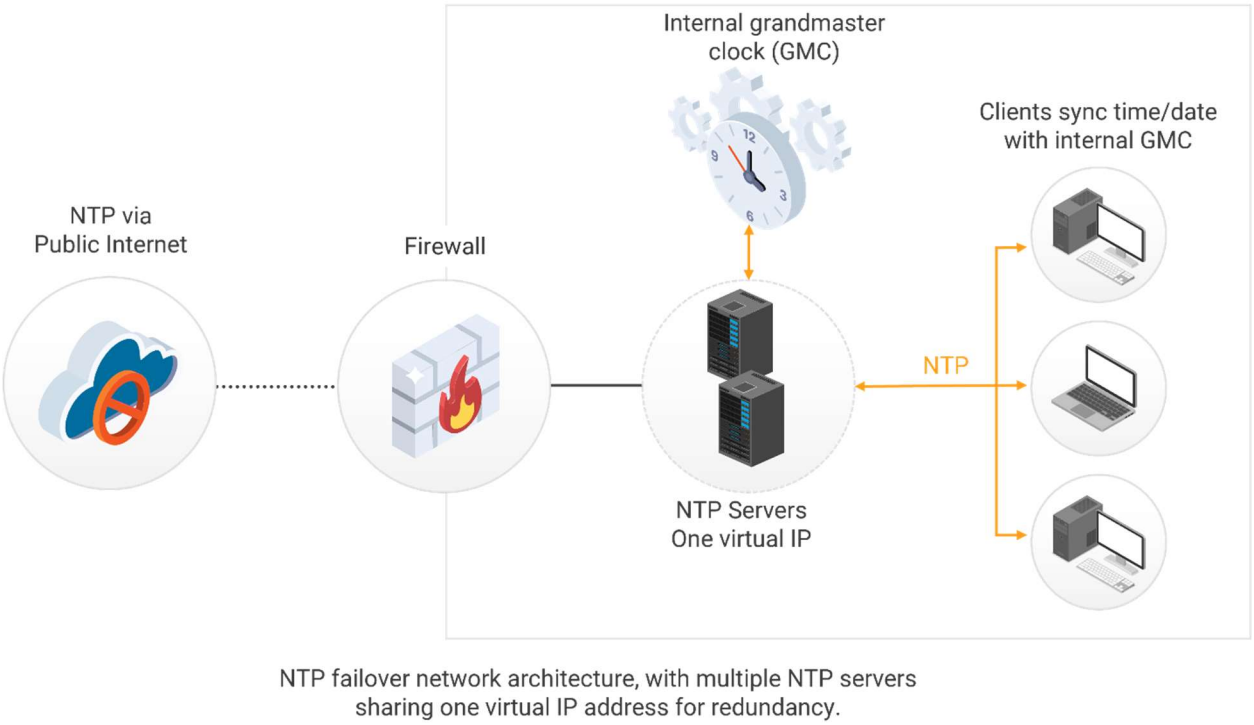

Figure 2. NTP server failover architecture

NTP CONFIGURATION

NTP configuration comprises both configuring NTP servers (which serve time) and configuring NTP clients (which consume time). This section describes the common and basic procedure for configuring NTP servers and clients. The basic configuration logic is straightforward. First, trusted timing sources for servers and clients must both be identified. Second, NTP daemon software must be installed and activated. It should be noted that NTP servers also require static IP addresses for client access.

NTP Server Configuration Fundamentals

Based on the underlying server computer and hardware, two scenarios are common when setting up an NTP server. In the first scenario, an NTP server is built on top of a **generic computer** (Linux or Windows). The second scenario involves using a commercial appliance—**a dedicated machine or hardware** custom-built to function as a timing device or a clock (GMC/BC). Such devices may come with built-in with GPS/GNSS receivers or a

Cesium clock so they can function as a GMC. The differences among these dedicated commercial base platforms lead to different configuration processes. However, the basic configuration logic is the same.

Generic NTP Server Configuration

A generic computer (Linux or Windows) can be configured as an NTP server. The computer may or may not come with built-in NTP software. To function as an NTP server, the computer needs to have properly configured NTP software running and be connected to a network with a static IP address. Using **Debian/Ubuntu** as an illustration for Linux systems, the following steps describe the NTP server configuration process for both **Linux/UNIX** [6] and **Windows** operating systems [7].

1. Connect the machine to the network with a **static IP address**.
2. **For Linux**:

   Update apt packages, **install NTP daemon software**, and validate.

   ```
   $ sudo apt update
   $ sudo apt install ntp
   $ sudo ntp –version
   ```

   For Windows:

   Open command prompt as administrator. Windows has a built-in NTP service called "Windows Time," so NTP software installation is not needed for NTP server or client configuration.

3. **Configure internal trusted timing source**. When following the closed-loop enterprise timing architecture and when an internal GMC/BC is available as the trusted timing source as shown in Figure 1, configure the NTP server to recognize the GMC/BC.

   For Linux:

   Edit NTP configuration file **/etc/ntp.conf** (alternatively, /etc/ntpsec/ntp.conf).

   ```
   $ sudo emacs /etc/ntp.conf
   ```

   Specify the trusted timing source by adding/updating the following configuration line into ntp.conf. The iburst option instructs the ntpd daemon to send a burst of time queries (eight packets) at start-up for faster synchronization with the GMC/BC server.

server <GMC/BC hostname or IP> iburst

For Windows:

Use **w32tm**, a command-line tool in Windows to configure, diagnose, and trouble-shoot the Windows Time service (W32Time), to configure internal timing source as follows.

w32tm /config /manualpeerlist:"<GMC/BC hostname or IP >,0x8" /syncfrom-flags:manual /reliable:YES /update

4. **For Linux**:

   Control access to the NTP server.

   Use the **restrict** directive in ntp.conf to protect the server from unauthorized or malicious activity. For example, the following restriction defines the default client restrictions during connection to the NTP server:

   restrict default nomodify notrap nopeer noquery

   Here, **nomodify** prevents the NTP configuration from being modified by external clients; **notrap** disables NTP control messages that could potentially be used for manipulation or denial-of-service attacks; **nopeer** prevents peer relationships with other NTP servers for security or to limit bandwidth consumption; **noquery** prevents queries such as ntpq or ntpdc from extracting server status and configuration information.

5. **Restart** NTP service daemon.

   For Linux:

   $ sudo systemctl restart ntp

   For Windows:

   net stop w32time

   net start w32time

6. **Verify** configuration.

   For Linux:

   $ ntpq -p

   For Windows:

```
w32tm /query configuration
```

7. **Monitor** NTP logs for potential issues.

   For Linux:

   ```
   $ sudo tail -f /var/log/syslog | grep ntpd
   ```

   For Windows:
   Using the "w32tm /query /status" command to show the status of the Windows Time service, and using Event Viewer (Eventvwr.msc) to navigate "Applications and Services Logs" → "Microsoft" → "Windows" → "Time-Service," the "Operational" channel contains events related to the Windows Time service, including NTP synchronization attempts and failures.

Dedicated Clock with Built-In NTP Server Configuration

Dedicated commercial timing/clock devices typically come with varied software and hardware configurations. The differences are wide, and most products come with built-in and customized NTP/PTP daemon software implementations. As a result, NTP software installation is typically not required. Most dedicated devices also have built-in dashboard GUIs for easy parameter adjustments. Furthermore, many dedicated timing devices can be configured with an attached GNSS/GPS receiver and/or cesium clock.

1. **Configure trusted time source**. The dashboard GUI typically provides an interface to select the internal timing sources (GPS/internal oscillator). The device may require creating an NTP clock instance where the clock stratum level can be specified.

2. **NTP server address**. A static IP address is required for an NTP server. Dedicated devices commonly provide an option within the dashboard GUI to set up the static NTP server IP address.

3. **Physical and virtual ports**. Some dashboards also allow for specification of the NTP physical port layout. Terminology might be different from product to product. For example, one vendor calls physical ports "**flow points.**" Some devices also allow users to specify virtual network configuration elements such as "**virtual ports**" and how they map to the physical ports as well as the NTP interfaces.

NTP Client Configuration Fundamentals

NTP client configuration is similar to that of the NTP server. Both need to have an NTP daemon installed, configured, and activated. Both also need to specify the trusted timing sources. However, there are also some differences.

1. A **static IP address** is not necessarily required for NTP clients as long as the NTP server enforces an IP-based subscription control.
2. For a Linux client, follow the NTP server configuration process to download and install **NTP daemon software**.
3. Follow the NTP server configuration process to configure the **trusted time source**.
4. Configure the NTP client to communicate only with **authorized servers** by using the **restrict** directive in the ntp.conf file. Optionally, enable **authentication** for Linux machines and use the following command for Windows machines.

   w32tm /config /syncfromflags:manual /manualpeerlist:"server1,0x8 server2,0x8"

CONCLUSION

By specifying NTP configuration, the behavior of the NTP server and its interactions with clients and other NTP servers—within and outside of the enterprise network—can be defined. Properly configuring these options is crucial for maintaining not only accurate time synchronization but also security and efficiency of the enterprise network.

By following the configuration process, users can set up an NTP server on the enterprise network while ensuring that the devices can synchronize their time with it. Regularly checking and maintaining the NTP server is crucial to ensure consistent and accurate timekeeping across the network.

REFERENCES

[1] D. Mills, "Network Time Protocol (Version 1) Specification and Implementation,," Network Wroking Group, RFC 1059, 1988. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1059.txt.

[2] H. S. &. D. S. D. Reilly, "Network Time Protocol Best Current Practices, RFC 8633," Internet Engineering Task Force (IETF), 2019. [Online]. Available: https://datatracker.ietf.org/doc/rfc8633/.

[3] TIME MACHINES, "GPS NTP Network Time Server," [Online]. Available: https://timemachinescorp.com/product/gps-time-server-tm1000a/.

[4] Veracity, "TIMENET Pro," [Online]. Available: https://www.veracityglobal.com/transmission/timenet-pro.

[5] OSCILLOQUARTZ, "NTP network time servers," [Online]. Available: https://www.oscilloquartz.com/en/products-and-services/ntp-network-time-servers.

[6] TimeTools, "How to Install and Configure NTP on Linux," Feb 2019. [Online]. Available: https://timetoolsltd.com/ntp/how-to-install-and-configure-ntp-on-linux/.

[7] Dean, "Setting up a NTP Server and Client," Mar 2023. [Online]. Available: https://adroittechnologiesautomation.com/t/setting-up-a-ntp-server-and-client/1470.

The Center for Alternative Synchronization and Timing (CAST) at Oak Ridge National Laboratory (ORNL) performs research, development, testing, evaluation, and technical assistance to enabled resilient timing and synchronization for the power grid. Working closely with power utilities, timing hardware and software vendors, network operators, and federal stakeholders, CAST helps develop and validate alternative timing architectures to augment GPS time. CAST also translates and transfers ORNL's research and development (R&D) advances in secure timing and grid communications to power sector applications, and engages across the broader timing community to develop best practices to ensure the resilience of US critical infrastructure. CAST is sponsored by DOE's Office of Electricity. Visit https://cast.ornl.gov for more information.