International Journal of P2P Network Trends and Technology
ISSN: 2249–2615 / https://doi.org/10.14445/22492615/IJPTT-V14I2P401
Volume 14 Issue 2, 1-4, May-Aug 2024


*Original Article*

# Enhancing Time Precision in White-Box Switches Using Software-Based Precision Time Protocol (PTP)

Rakesh Datta

*Zscaler Inc., San Jose, USA*

*Corresponding Author : rakesh.datta6@gmail.com*

Received: 19 June 2024 Revised: 28 July 2024 Accepted: 14 August 2024 Published: 30 August 2024

***Abstract*** - *White-box switches have emerged as a pivotal component in modern networking architectures due to their flexibility, cost-effectiveness, and programmability. Within a data center network, achieving precise time synchronization within these switches is imperative for various critical functions such as packet scheduling, network security, and compliance with regulatory requirements. Conventional time synchronization protocols, like the Network Time Protocol (NTP), fail to meet the stringent time synchronization demands of the white-box switches because they are software-based. In contrast, the Precision Time Protocol (PTP), a hardware-based solution, offers sub-microsecond accuracy and low-latency time synchronization. However, despite the advantages of PTP, integrating it into white-box switches may pose challenges due to additional hardware requirements. This paper elaborates on a study conducted to understand the role of white-box switches and their precise time synchronization in data center networking, as well as a comparative measurement between NTP and PTP. Ultimately, this research contributes to a nuanced understanding of the role of PTP in modern white-box switches and lays down decision-making processes regarding time synchronization protocols in networking deployments.*

***Keywords*** - *Network performance, Network Time Protocol (NTP), Precision Time Protocol (PTP), Packet processing, White-box switches.*

## 1. Introduction

In recent years, white-box switches have gained significant traction in networking environments due to their flexibility, cost-effectiveness, and programmability [1]. Unlike traditional proprietary switches, white-box switches offer users the ability to customize and optimize their networking infrastructure according to specific requirements, making them increasingly popular among enterprises and service providers alike [2]. In a data center network, the most appropriate placement of a white-box switch is Top-Of-Rack (TOR). As line rate packet transmission has growingly become a critical aspect of any modern networking infrastructure, the necessity of precise time synchronization within TOR white-box switches has become paramount [3]. In modern networking architectures, accurate timekeeping is essential for various critical functions, including packet scheduling, network security, performance monitoring, and compliance with regulatory requirements. However, achieving precise time synchronization is challenging, particularly with conventional time synchronization protocols such as the Network Time Protocol (NTP) [4]. While NTP has long been the de facto standard for time synchronization in networked systems, it falls short of meeting the stringent timing requirements of modern networking systems [5]. NTP, being primarily designed for general-purpose timekeeping, lacks the precision and determinism necessary for real-time applications and high-performance networking environments [6]. Its reliance on software-based clock adjustments and variable network latency introduces jitter and inaccuracies, which can significantly impact the reliability and performance of time-sensitive applications [7]. In contrast, the Precision Time Protocol (PTP) offers a compelling solution to the challenges of time precision in white-box switches [8]. PTP is specifically designed to provide sub-microsecond accuracy and low-latency time synchronization in networked systems, making it well-suited for demanding applications such as financial trading, industrial automation, and telecommunications [4]. By leveraging hardware-based timestamping and synchronized clock distribution mechanisms, PTP enables precise coordination of time-sensitive operations across distributed network devices, ensuring consistent timing accuracy and reliability. However, despite its technical advantages, integrating PTP into white-box switches may not be cost-effective compared to NTP [7]. While NTP implementations typically rely on software-based synchronization algorithms and can be easily deployed on commodity hardware, PTP requires specialized hardware support for accurate timestamping and clock synchronization [6]. This additional hardware cost may present a barrier to adoption for organizations with budget constraints or legacy infrastructure [8]. In light of these considerations, this research paper aims to elucidate a study on the growing popularity of PTP for white-box switches and the cost implications of integrating PTP into white-box switches compared to NTP. Through comprehensive analysis and evaluation, this paper seeks to provide valuable insights into the role of time synchronization protocols in modern networking deployments and inform strategic decision-making processes for network architects, operators, and researchers.

## 2. Necessity of Time Precision in White-Box Switches

White-box switches have gained popularity recently due to their flexibility, cost-effectiveness, and compatibility. Unlike proprietary switches, white-box switches offer customizable configurations and can be installed on ASICs from various ODMs, making them attractive for organizations looking to optimize networking infrastructure. Their hardware-agnostic nature allows deployment on commodity hardware, enabling cost-effective solutions and leveraging existing investments. Moreover, open-source white-box software grants users' greater control over their network environment. It facilitates customization and integration with Software-Defined Networking (SDN) or Network Functions Virtualization (NFV) solutions from multiple vendors. Accurate time precision is crucial for white-box switches, as it impacts performance, reliability, and security:

- Packet Scheduling: Inaccurate synchronization leads to packet scheduling issues, causing latency, packet loss, and degraded performance.
- Network Security: Security protocols rely on precise timekeeping; inaccuracies can compromise authentication and encryption, leaving the network vulnerable to attacks.
- Performance Monitoring: Precise timestamps are necessary for effective performance analysis, troubleshooting, and anomaly detection.
- Compliance: Industries face regulatory requirements for accurate timekeeping; failure to comply can result in legal and financial consequences.
- Network Coordination: Time synchronization ensures effective coordination and control in distributed networks, preventing synchronization errors and operational inefficiencies.

## 3. Network Time Protocol

For synchronizing the time of computer systems over packet-switched, variable-latency data networks. It operates by synchronizing the time of day among distributed time servers and clients. NTP uses a hierarchical, semi-layered system of time sources, which enables it to maintain accuracy and reliability even in the presence of unreliable reference clocks. NTP synchronizes time by exchanging timestamps between servers and clients, adjusting the system clock frequency, and implementing various algorithms to mitigate network jitter and latency.

NTP is critical in maintaining time consistency across networked systems, facilitating various essential functions such as logging, authentication, and data consistency. It is commonly deployed in various applications, including network infrastructure, financial trading platforms, and telecommunications networks.

NTP, despite its widespread adoption, encounters several hurdles when meeting the timing requirements of contemporary white-box switches:

- Accuracy and Precision: NTP, designed for general-purpose timekeeping, may lack the necessary precision demanded by modern white-box switches. These switches operate in high-performance settings, where sub-microsecond accuracy is critical for financial trading and industrial automation applications.
- Variable Network Latency: NTP's reliance on packet-based communication over data networks exposes it to variable network latency. Fluctuations due to congestion, packet loss, and route changes introduce jitter and uncertainty, compromising time synchronization's accuracy and reliability.
- Software-Based Clock Adjustments: Traditional NTP implementations adjust clocks via software, which may not meet the stringent requirements of high-performance environments. This method can limit precision and determinism, especially where hardware-based timestamping and synchronization are preferred.
- Scalability and Resilience: As networked systems expand in size and complexity, NTP's scalability and resilience become critical. However, conventional NTP deployments may struggle to maintain synchronization accuracy and reliability in large-scale, distributed environments with diverse network conditions and architectures.

NTP's dependency on software-based clock adjustments and variable network latency presents significant challenges for modern white-box switches. Software-based adjustments may constrain accuracy, precision, and determinism, particularly in environments requiring sub-microsecond synchronization. Moreover, variable network latency can disrupt synchronization consistency and reliability, resulting in jitter and inaccuracies.

While NTP remains a prevalent time synchronization protocol, its challenges may hinder its effectiveness for modern white-box switches. Overcoming these hurdles demands innovative solutions leveraging hardware-based timestamping, low-latency communication protocols, and advanced synchronization algorithms to meet the precision, accuracy, and reliability standards of high-performance networking environments.

## 4. Precision Time Protocol

The Precision Time Protocol (PTP) stands as a revolutionary solution in the realm of time synchronization, offering unparalleled precision and accuracy, particularly suited for high-performance networking environments. PTP is designed to deliver sub-microsecond accuracy and low-latency synchronization, making it an ideal choice for applications demanding precise timing, such as financial trading, industrial automation, and telecommunications. Compared to the Network Time Protocol (NTP), PTP boasts several advantages that set it apart as the preferred choice for stringent timing requirements. While NTP is renowned for its versatility and widespread adoption, it falls short in meeting modern networking infrastructures' demanding timing precision needs. Unlike NTP, which relies on software-based clock adjustments and packet-based communication, PTP leverages hardware-based timestamping and precise clock distribution mechanisms, significantly enhancing accuracy, reliability, and determinism. At the core of PTP's capabilities are its hardware-based timestamping and synchronized clock distribution mechanisms, contrasting with NTP's software-centric operation. PTP utilizes specialized hardware components for precise time synchronization, enabling sub-microsecond accuracy across networked devices. PTP operates on a master-slave synchronization model, where a master clock distributes synchronized time to slave devices. The master clock generates accurate timestamps and transmits them via network messages to synchronize slave clocks, ensuring consistent timing across the network. Moreover, PTP incorporates advanced clock filtering and correction algorithms to mitigate network latency and packet delay variations, further enhancing synchronization accuracy and reliability. This combination of hardware-based timestamping and sophisticated algorithms enables PTP to achieve unparalleled precision and low-latency synchronization, making it ideal for applications prioritizing timing accuracy. PTP represents a groundbreaking solution for achieving sub-microsecond accuracy and low-latency synchronization in modern networking environments. It's hardware-based approach distinguishes it from conventional protocols like NTP, offering superior performance and reliability for applications requiring precise timing.

## 5. Software-based PTP Solution - LinuxPTP

The Linux PTP, aka Linux Precision Time Protocol, is an open-source implementation of the Precision Time Protocol (PTP) designed specifically for Linux-based operating systems. Often referred to as Linux PHP or ptp4l, it provides a user-space daemon called ptp4l, responsible for synchronizing the system clock with a PTP master clock over the network. It operates by exchanging timing messages with other PTP-enabled devices in the network, allowing the clocks to be synchronized within a common time reference. Linux PTP supports both hardware-based and software-based clock synchronization methods, offering flexibility in deployment. Hardware-based synchronization relies on specialized Network Interface Cards (NICs) that support hardware timestamping, allowing for more accurate and precise time measurements. Software-based synchronization, on the other hand, relies solely on software algorithms for clock synchronization and is generally less accurate but more widely compatible. Overall, Linux PTP provides a reliable and efficient solution for achieving precise time synchronization in Linux environments, contributing to the stability and performance of networked systems.

## 6. Materials and Methods

### *6.1. Software-based PTP Setup for a White-Box*

For the POC and experiments, linuxptp [9] is installed on top of a standard white-box switch like an open switch [10]. The linuxptp package is installed on top of the white-box switch's Linux-based OS. The versions need to be compatible. During the setup, the ptp4l demon is to be correctly configured to operate in the slave mode while specifying the network interface for PTP communication and the address of the PTP master clock. On another locally connected white box, Linux PHP is installed and started as a PTP master clock. Using the ptp4l tool, logs would indicate the successful clock synchronization with the Grand Master Clock. Using the '*ptp4l -s*' command, the status of both slave and master nodes is monitored. This way, a software-based PTP is implemented on top of the white-box switches.

### *6.2. NTP Setup for a White-Box*

For this purpose, the in-built NTP feature of the white-box switch OS is leveraged. Log in to the management console of the white-box switch and simply enable the NTP feature, providing the NTP server address and clock time zone. Connect two white-box switches with NTP enabled to test the clock synchronization between them.

### *6.3. Performance testing*

For the experiments, the software-based solutions for both NTP and PTP are chosen in order to draw comparisons on the same set of parameters. Tests include multiple parameters, including time precision, latency and resilience to network fluctuation.

## 7. Results and Discussion

### *7.1. Time Offset Precision*

In either protocol, the client or slave uses the offset value to adjust its clock and synchronize with the server or the master clock. By comparing the offsets for both protocols, it is determined which one is more accurate or precise.

Time Offset = [(T2 - T1) + (T3 - T4)] / 2

In this formula, the timestamps are defined as follows:

- **T1**: The client timestamp on the request packet
- **T2**: The server timestamp upon arrival

- **T3**: The server timestamp on the departure of the reply packet
- **T4**: The client timestamp upon arrival

In LinuxPTP, the time offset between the master and slave was 81 microseconds on average. While for NTP the time offset between the client and server was at 98 microseconds on average.

| Scenario | Software PTP | NTP |
|---|---|---|
| **Average of 10 request/responses** | 101 | 120 |
| **Average of 50 request/responses** | 92 | 108 |
| **Average of 100 request/responses** | 79 | 100 |
| **Average of 1000 request/responses** | 81 | 98 |

**Fig 1. Time offset (microseconds) between clocks**

| Scenario | Software PTP | NTP |
|---|---|---|
| **Failover the Master clock network interface** | 10 | 60 |
| **Failover the Slave clock network interface** | 15 | 65 |

**Fig 2. Clocks Reconciliation Speed (seconds) Post Failover**

### *7.2. Reconciliation after Network Failover*

A network failover is performed by restarting the network interface of either white-box switches connected to the link used for the clock synchronization and testing the convergence speed between the client and server clocks. For LinuxPTP, the clocks are synchronized 5-6 times faster than in NTP.

## 8. Conclusion

In conclusion, the research compared software-based Precision Time Protocol (PTP) and Network Time Protocol (NTP) implementations for time synchronization. While software PTP showed efficiency over NTP, the margin wasn't significant. This suggests that while software PTP offers advantages in accuracy and reliability, a complete transition from NTP may not always be warranted, especially in environments with moderate precision requirements. The investigation emphasized scalability and precision needs when choosing a synchronization solution. For high precision at scale, hardware PTP is preferred due to its superior accuracy and performance. However, adopting hardware PTP involves additional costs for compatible hardware and infrastructure upgrades. The study compares software PTP and NTP and showcases the real-time performance of these protocols. It helps in considering the parameters to measure while choosing the clock synchronization protocol and also warrants thoughtful consideration of trade-offs between software PTP, NTP, and hardware PTP. Factors such as precision needs, scalability, budget, and existing infrastructure should guide decision-making. By weighing these factors, organizations can optimize time synchronization effectively.

## Funding Statement

This paper is the production of individual research, and it is neither funded by Zscaler Inc. nor involves Zscaler Inc. resources.

## References

[1] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat, "A Scalable, Commodity Data Center Network Architecture," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 63-74, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[2] Craig Labovitz et al., "Internet Inter-domain Traffic," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 75-86, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[3] David L. Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482-1493, 1991. [CrossRef] [Google Scholar] [Publisher Link]

[4] K. Balakrishnan et al., "Clock Synchronization in Industrial Internet of Things and Potential Works in Precision Time Protocol: Review, Challenges and Future Directions," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 205-219, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Vishal Shrivastav et al., "Globally Synchronized Time via Datacenter Networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1401-1416, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] PTP: Timing Accuracy and Precision for the Future of Computing. [Online]. Available: https://engineering.fb.com/2022/11/21/production-engineering/future-computing-ptp/

[7] Julien Ridoux, and Darryl Veitch, "Principles of Robust Timing over the Internet: The Key to Synchronizing Clocks over Networks is Taming Delay Variability," *ACM Queue*, vol. 8, no. 4, pp. 30-43, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[8] Muhammad Aslam et al., "Hardware Efficient Clock Synchronization Across Wi-Fi and Ethernet-Based Network Using PTP," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3808-3819, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] The Linux PTP Project, Linux PTP. [Online]. Available: http://linuxptp.sourceforge.net/

[10] The Open-Switch Project, Github. [Online]. Available: https://github.com/open-switch/