AraSync: Precision Time Synchronization in Rural Wireless Living Lab

Md Nadim[§], Taimoor Ul Islam[§], Salil Reddy[¶], Tianyi Zhang[§], Zhibo Meng[§], Reshal Afzal[§], Sarath Babu[§], Arsalan Ahmad[§], Daji Qiao[§], Anish Arora[¶], and Hongwei Zhang[§]

{nadim,tislam,tianyiz,zhibom,reshal,sarath4,aahmad,daji,hongwei}@iastate.edu,{reddy.441,arora.9}@osu.edu

[§]Iowa State University, [¶]Ohio State University

ABSTRACT

Time synchronization is a critical component in network operation and management, and it is also required by Ultra-Reliable, Low-Latency Communications (URLLC) in next-generation wireless systems such as those of 5G, 6G, and Open RAN. In this context, we design and implement *AraSync* as an end-to-end time synchronization system in the ARA wireless living lab to enable advanced wireless experiments and applications involving stringent time constraints. We make use of Precision Time Protocol (PTP) at different levels to achieve synchronization accuracy in the order of nanoseconds. Along with fiber networks, *AraSync* enables time synchronization across the AraHaul wireless x-haul network consisting of long-range, high-capacity mmWave and microwave links. In this paper, we present the detailed design and implementation of *AraSync*, including its hardware and software components and the PTP network topology. Further, we experimentally characterize the performance of *AraSync* from spatial and temporal dimensions. Our measurement and analysis of the clock offset and mean path delay show the impact of the wireless channel and weather conditions on the PTP synchronization accuracy.

CCS CONCEPTS

• **Networks → Network measurement**; • **Computer systems organization → Real-time system architecture**.

KEYWORDS

Precision Time Protocol (PTP), ARA Wireless Living Lab, AraSync, Wired-PTP, Wireless-PTP

1 INTRODUCTION

Next-generation wireless networks are characterized by features to enable Ultra-Reliable, Low-Latency Communications (URLLC). As per the requirements of 6G [16], it is essential to ensure an end-to-end data-rate of hundreds of gigabits per second with guaranteed sub-millisecond latency and jitter. To meet such stringent low-latency requirements, time synchronization between interacting network components is of utmost priority.

Precision Time Protocol (PTP) [1–3, 6] and Network Time Protocol (NTP) [22] are the most widely used time synchronization protocols in various networks. PTP is specifically designed for industrial environments that involve control systems and is ideal for distributed systems because of its minimal bandwidth requirements and low processing overhead. In addition to lower overhead, PTP offers precise clock synchronization compared to NTP [15]. Depending on the network topology, PTP can achieve synchronization accuracy in the order of sub-microsecond and down to sub-nanosecond range in comparison with the millisecond-level accuracy in NTP. Since advanced wireless research platforms enable experimentation in next-generation communication systems requiring fine-grained synchronization, PTP is considered as the basis for system-wide synchronization.

In this paper, we present the design and implementation of PTP across the ARA wireless living lab [18, 26], i.e., *AraSync*. We discuss in detail the ARA network topology from the synchronization perspective, PTP hardware and software architecture, and the configuration considerations for the devices involved in synchronization. To illustrate the characteristics and research capabilities enabled by *AraSync*, we examine the behavior of PTP in ARA through extensive experimentation, measurement, and analysis. In our study, we include essential PTP metrics, such as offset from the master clock and Mean Path Delay (MPD), at various levels of PTP over both wired and wireless media and present a comparative analysis. Since long-range wireless links form the most feasible solution to ensure affordable broadband in rural areas, where fiber connectivity is scarce or absent, we measure the effectiveness of PTP over the 10.15 km long *AraHaul millimeter-wave* and *microwave* links, especially the impact of weather conditions on PTP in synchronizing

devices over wireless x-haul. It is observed that the accuracy of the synchronization varies more during high rain rates compared to moderate or light rain conditions. In short, PTP over long-range *AraHaul* [27] wireless links achieves sub-microsecond level accuracy that is similar to its wired counterpart, however incurring more fluctuations.

The rest of the paper is organized as follows. Section 3 provides an overview of the PTP operations and mechanisms. In Section 4, we discuss the PTP network topology for *AraSync*, including the capabilities and configurations of PTP-supported devices. Section 5 discusses our experiments, analysis, and observations. Finally, we conclude the paper in Section 6 with a discussion on the potential utility of *AraSync* in future research work.

2 RELATED WORK

Precision Time Protocol (PTP) has been extensively used for synchronizing nodes in time-sensitive systems. In fact, the use of PTP is inevitable in experimental platforms such as ARA [26], enabling high-precision measurement studies in time-sensitive communication frameworks using 5G/6G platforms such as OpenAirInterface and Open RAN.

In [25], authors explored the fundamental principles of the PTP and shared their experience of its implementation in power system networks to realize sub-microsecond-level accuracy. However, the experiments were confined to a small laboratory setup rather than a real-world environment. Moreover, instead of an external grand-master clock, the authors used the Oven Controlled Crystal Oscillator (OCXO) of the master clock as the time source. As far as the wireless networks are concerned, authors of [17] proposed the Wireless Precision Time Protocol (WPTP), an enhanced time synchronization protocol designed for over-the-air use. WPTP minimizes the number of PTP packet transmissions by consolidating timestamp information from *Sync* and *Delay_Request* messages into *DRPLY* message, thereby reducing the combined quantity of event and general PTP messages. Simulations demonstrated the effectiveness of WPTP in reducing network convergence time in a wireless multi-hop network.

From the perspective of real-world implementation of PTP, authors of [24] detailed the step-by-step installation of *LinuxPTP* on a testbed built on a 64-bit Raspbian operating system. From the measurements, it was observed that the TimeTransmitter or Master's offset had a median value of 0 ns with a standard deviation of 55 ns. In [23], authors proposed Simple PTP (SPTP), a streamlined implementation of the existing PTP using the unicast UDP profile. SPTP simplifies the two-step PTP clock mode by eliminating the handshake mechanism and transferring the packet exchange control from the master to the client/slave. Subsequently, SPTP operates in a single-step mode without slave-master negotiation and the message exchanges are driven by the slave, thereby resulting in reduced network overhead.

It is important to note that fine-grained synchronization is a critical component of any 5G/6G communication systems. For instance, the crucial role of PTP-based synchronization in the emerging Open RAN systems is highlighted in [20]. A dedicated synchronization plane (S-plane) in Open RAN is responsible for synchronization among different components to ensure timely scheduling information to ensure proper Carrier Aggregation (CA), avoid interference, and prevent call dropouts. Open fronthaul requires precise timing accuracy with a maximum allowable time error of 3 μs. Besides synchronization, the authors discussed possible security threats in the S-plane and proposed effective countermeasures to address these challenges. In this paper, we discuss the design and implementation of PTP-based synchronization in the ARA wireless living lab involving long-range, high-capacity wireless x-hauls for enabling advanced wireless research.

3 PRECISION TIME PROTOCOL

In this section, we present a brief overview of PTP, including the types of clocks and the delay mechanism.

3.1 PTP Clock Types and States

PTP uses four different types of clocks [8, 19] as well as different kind of messages between them for synchronizing the nodes. Different types of clocks include:

(1) **Grandmaster Clock (GMC)**: The clock has a high-precision oscillator with capability to synchronize it from a GNSS receiver. In PTP, the GMC always acts as the *master* clock.

(2) **Boundary Clock (BC):** Acts as the second level of clock and, serves as the *ordinary* clock for the GMC and *master* clock for the third level *ordinary* clocks. BCs have multiple PTP ports, each capable of establishing distinct paths to maintain the PTP clock hierarchy.

(3) **Slave or Ordinary Clock (OC):** Always the last level nodes (i.e., the devices that require synchronization) in the network hierarchy. OC has a single port which receives time from the *master*.

(4) **Transparent Clock (TC):** The function of a TC device is to forward messages while keeping track of the residence time of each message across the device (i.e., time taken for the message to traverse through the device from an ingress port to an egress port), allowing for the total time to be corrected at the destination. A TC has two modes: end-to-end mode and peer-to-peer mode. The end-to-end TC uses the end-to-end delay measurement mechanism between slave clocks and the master clock. The peer-to-peer TC involves link

delay correction using peer-to-peer delay measurement mechanism, in addition to the residence time correction.

On setting up the above-mentioned clocks at different levels, PTP achieves synchronization through exchanging messages of the following categories [5, 14]:

(1) **Event messages:** These messages are tagged with timestamps when data packets arrive or leave a port and are used to calculate the link-delay based on these timestamps. They are time-critical since the accuracy in the transmit and receive timestamping directly affects the clock distribution accuracy. Event messages include *Sync*, *Delay_Request*, *Pdelay_Request*, and *Pdelay_Response* messages.

(2) **General messages:** These messages are more conventional protocol data units, and the data in these messages is of importance to PTP, but their transmit and receive timestamps are not as important. They are used for purposes such as establishing Master-Slave hierarchies, and they include *Announce*, *Follow_Up*, *Delay_Response*, *Pdelay_Response_Follow_Up*, *Management*, and *Signaling* messages.

The ports associated with each clock in PTP can be in any of the two states: (1) *master* or (2) *slave*. In fact, the port state primarily depends on the BC [5, 9]. The *master* port synchronizes the clocks connected downstream, while the *slave* port synchronizes with the upstream *master* clock. A unique state, called the *passive* state, occurs when a BC receives PTP traffic from two or more different *master* clocks, and the *slave* clock selects only one *slave* port for its synchronization using the Best Master Clock Algorithm (BMCA) or Best TimeTransmitter Clock Algorithm (BTCA) [9].

BMCA is used to decide port states, *master* or *slave* [3], to form a Directed Acyclic Graph (DAG) for disseminating GMC time. When a clock is up, it enters the listening state to recieve the *Sync* messages from a *master* clock. Based on the received *Sync* message and the clock properties (such as the administrator-assigned clock priority, clock class, and clock precision), the clock adopts its *master* or *slave* role. When multiple BCs interconnect and form a cycle, certain ports are turned to *passive* state to break the cycle, thereby maintaining the DAG structure [5]. TCs are not involved in the execution of BMCA, but they forward the PTP messages between the associated *master* clocks and *slave* clocks by adding the residence time.

3.2 Delay Measurement Mechanism

To synchronize between a pair of clocks, PTP adopts two mechanisms, i.e., end-to-end [3, 4] and point-to-point [3, 12],

by measuring the delay between the clocks, as shown in Figure 1. They are collectively referred to as Delay Measurement Mechanisms.
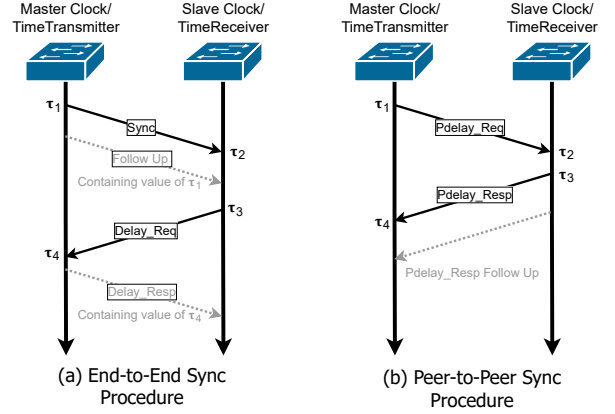
Figure 1: Delay measurement mechanisms.

End-to-End Delay Measurement: Master clocks (or Time-Transmitters) continuously send *Sync* messages to all their *slave* clocks (or TimeReceivers) at specified intervals. The *master* clock records the sending time τ_1 and *slaves* record the reception time τ_2 and, the time difference between them becomes $\tau_2 - \tau_1$. The *master* communicates τ_1 to the *slave* using *Sync* message itself or using a separate *Follow Up* message. Given that the time difference between *master* and *slave* is the sum of their clock offset (Δ) and the delay (δ) in message transmission, i.e., $\Delta + \delta$, we have $\tau_2 - \tau_1 = \Delta + \delta$. The *slave* clock, unaware of the delay, periodically sends *Delay Request* messages to the *master* clock, noting the sending time (τ_3). The *master* receives this message, records the arrival time (τ_4), and responds to *slave* with a *Delay Response* message containing τ_4. The time difference between the *slave* and *master* is $\tau_4 - \tau_3$, and for the *Delay Request* message, it can be expressed as $-\Delta + \delta$; that is, $\tau_4 - \tau_3 = -\Delta + \delta$. Assuming the delay is the same in both directions, the offset and Mean Path Delay (MPD) can be computed as $\Delta = \frac{(\tau_2 - \tau_1) - (\tau_4 - \tau_3)}{2}$ and $\delta = \frac{(\tau_2 - \tau_1) + (\tau_4 - \tau_3)}{2}$, respectively [4].

Peer-to-Peer Delay Measurement: In Ethernet networks, the assumption that round-trip time (RTT) being $\frac{\delta}{2}$ may not always be valid. To this end, peer-to-peer TCs measure the link peer delay to improve PTP accuracy. Unlike the end-to-end approach, peer-to-peer synchronization allows PTP devices to exchange peer-delay measurement messages with their direct neighbors. Periodically, each device initiates the peer-delay exchange and adjusts it by updating the correction field in either the *Sync* or *Follow Up* message as it enters the device. Peer link delay can be expressed as $\delta_{link} = \frac{(\tau_2 - \tau_1) + (\tau_4 - \tau_3)}{2}$, where τ_1 and τ_2 are the timestamps at the *master* and *slave* for *Pdelay Request* message, respectively, and τ_3 and τ_4 are
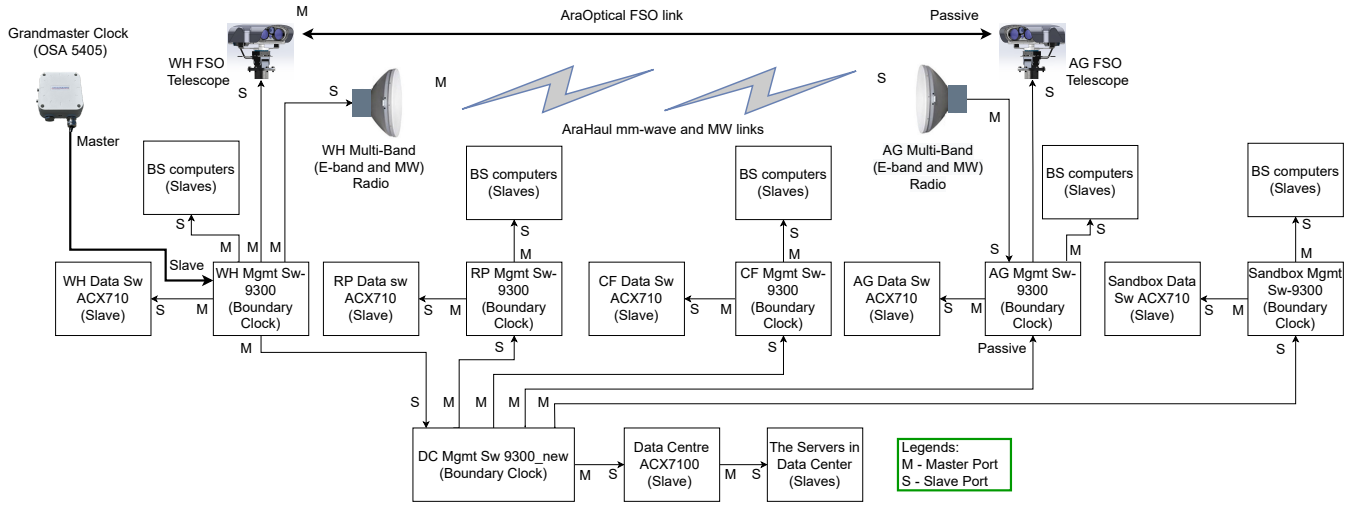
Figure 2: Network topology for *AraSync.*

the departure time at the *slave* and arrival time at the *master*, respectively.

4 ARASYNC FRAMEWORK

In ARA [18], we implemented PTP across the entire network using a tree/hierarchical topology [5] where the boundary clocks form the branch points in the tree depicted in Figure 2. A single outdoor grandmaster clock (GMC) is used to synchronize the network. The GMC is a smart Global Navigation Satellite System (GNSS) receiver leveraging signals from available satellites for synchronization. The GMC connects to a PTP-enabled switch, serving as the first-level BC in the hierarchical topology and acting as the *master* for all the downstream devices. The first-level BC forwards the PTP traffic through wireless x-haul and fiber links to the second-level BCs via Layer 2 Multicast. The second-level BCs then act as *masters* for their respective downstream devices forwarding PTP messages via Layer 2 multicast. Further, the second-level BCs act as the synchronization sources to the third-level BCs and the devices connected to them. In short, for *AraSync*, we use a 3-level hierarchical topology for synchronization.

Since the BC is a switch with a large number of ports, the tree topology allows it to serve as the *master* to a large number of OCs connected to it. The first-level BC becomes a *slave* only to the GMC through the port where GMC is connected, whereas the remaining ports act as the *master* for the devices connected to them. The second-level BCs serve as *slave* to the first-level BC and act as *master* for any devices connected to them. The switches connected directly with the second-level BC act as the last-level BCs and provide time synchronization to the end devices, i.e., the OCs. In case a BC receives PTP *Event* and *General messages* from two higher-level BCs, the BC with fewer number of hops toward

GMC is selected as the *master* in accordance with the Best Master Clock Algorithm (BMCA). Consequently, the port of the downstream device facing the BC with a lower hop-count will be configured as the *slave*, while the port connected to the BC with a higher hop-count is set to the *passive* state. For instance, one of the base station switches, i.e., Agronomy Farm Cisco Catalyst C9300, has wireless x-haul connectivity with the first-level BC (Wilson Hall Cisco Catalyst C9300) and fiber connectivity with the second-level BC (Data Center Cisco Catalyst C9300), illustrated in Figure 2. The wireless x-haul link spans approximately 6.31 miles between the two base stations, whereas the optical fiber link extends nearly 7.5 miles. The BMCA sets the port connecting first level BC (i.e., the one via wireless x-haul) as the *slave* due to the lower hop-count from GMC. On the other hand, the port connected to the fiber is set to *passive* state due to higher hop-count and pruned to avoid the possibility of cycle in the topology. In short, the BMCA determines the states of the PTP ports and constructs the synchronization tree.

4.1 PTP Devices in ARA

The Oscilloquartz OSA 5405 Outdoor SyncReach device, a GNSS antenna integrated with GNSS receivers and PTP stacks, is stationed at the Wilson Hall base station site (see Figure 3) as the GMC to serve as a precise time source [13]. The OSA 5405 GMC supports IEEE 1588 2008 L3/L2 and ITU-T 8265.1/8275.1/8275.2/power/broadcast/enterprise standards. It facilitates PTP over both IPv4 and IPv6 simultaneously and includes fallback options for PTP and Synchronous Ethernet (SyncE) inputs. In addition, the GMC supports both multicast and unicast modes.

The Cisco Catalyst C9300 switches function as BC at various levels of *AraSync* serving the downstream devices. These switches support both Layer-2 and Layer-3 multicast PTP

Figure 3: *AraSync* **GMC at the Wilson Hall rooftop.**

synchronization protocols, specifically conforming to the IEEE 1588/8275.1 profile, and enable precise clock synchronization with sub-microsecond accuracy [11]. The Juniper ACX7100 Cloud Metro Router [10] and ACX710 Universal Metro Router [7], which serve as the data traffic carriers, function as end-devices in *AraSync*. These routers support Layer-3 unicast PTP synchronization mode, exclusively adhering to the g.8275.2 profile, as well as SyncE. The integration of PTP and SyncE synchronization is referred to as Hybrid Mode.

The compute nodes at the base station, data center, and sandbox in ARA are Dell R750 PowerEdge servers with Intel X710 SFP+ Network Interface Cards (NICs) and Broadcom 5720 1 Gb Ethernet NIC, which support hardware and software timestamping required for PTP. The Ethernet port is used for the management network, including PTP sync, while the SFP+ ports are for the data network. The compute servers run Ubuntu 20.04 LTS with *LinuxPTP* [24] installed and configured on the servers for the PTP time Sync. *LinuxPTP* provides *ptp4l*, the daemon that synchronizes PTP Hardware Clock (PHC) from the NIC, and *phc2sys*, which synchronizes PHC and the system clock. *LinuxPTP* synchronize the servers to the respective switches that they are connected to. The compute nodes at the User Equipment (UE) have a high-precision GPS sensor, and the GPS data received by the sensor is fed as the source for NTP synchronization. We use the *gpsd* daemon to periodically poll the GPS signals.

4.2 PTP Configuration in ARA

The GMC is configured as Layer-2 PTP multicast to forward the PTP traffic to the downstream PTP instances. The PTP profile is configured as g8275.1, with the domain number being 24, the lowest domain number in the domain range of 24–43 for this PTP profile. The *Announce, Sync,* and *Delay Response* rates are set at 8 pps,16 pps, and 16 pps, respectively, to send out the PTP messages in a very frequent fashion, and these rates are the maximum as per the standard. The *Sync* messages are sent in one-step mode, i.e., the *Sync* messages

contain the initial timestamp to the TimeReceiver instances. The Global Navigation Satellite System (GNSS) is the primary way to synchronize the OSA-5405 GMC in GPS mode and tracking mode is enabled with the GNSS-tracking system. The GMC also provides SNR, elevation values, latitude, longitude, altitude, and the available satellites, from where the GMC receives the GPS signals for synchronization. Similar to the GMC, the Cisco Catalyst C9300 switches are configured to Layer-2 PTP multicast. As IEEE1588/8275.1 is the only profile available in Cisco C9300, we set the domain number to 24. The *Announce* message interval is 3 (i.e., 8 pps) while the *Sync* interval is -4 (i.e., 1/16 pps). We configure the end-to-end delay mechanism to compute the offset and mean path delay, and the mode of the switch is configured as a boundary clock delay request.

5 PERFORMANCE ANALYSIS

For analyzing the performance of *AraSync*, we first consider the offset of BCs at different levels with the GMC. For our analysis, we consider Wilson Hall Cisco C9300 as the first-level BC (connected directly to GMC), Data Center Cisco C9300 as the second-level *wired* BC (connected to Wilson Hall C9300 via Fiber cable), and Agronomy Farm Cisco C9300 as the second-level *wireless* BC (connected to Wilson Hall C9300 via the AraHaul link). The behavior of offset observed is shown in Figure 4. The range of offset from GMC to first-level BC (Wilson Hall Cisco C9300 switch) varies from -25 ns to 25 ns. The second-level BC, which is connected to the first-level BC via fiber, provides an offset from -45 ns to 45 ns, i.e., 20 ns higher than first-level BC. However, the offset measured at the Agronomy Farm BC fluctuates between -10,000 ns to 10,000 ns. This second-level BC relies on a first-level BC, which transmits PTP messages over *AraHaul*, and
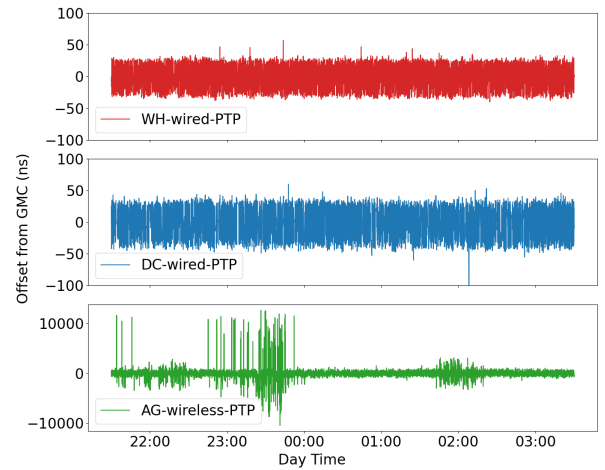
Figure 4: *Wireless*- and *wired*-**PTP offset from GMC at different levels of BCs.**
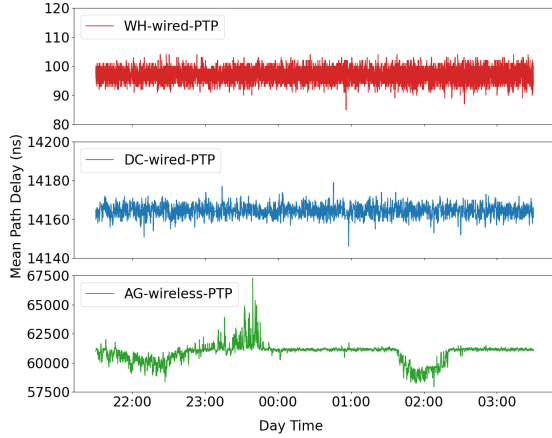
Figure 5: Mean Path Delays (MPDs) of *wireless*- and *wired*-PTP at different levels of BCs.

fluctuations in offset can be attributed to the wireless link's channel health and varying weather conditions.

In Figure 5, the Mean Path Delay (MPD) between the first-level BC, i.e., Wilson Hall, and the second-level *wired*-PTP BC (i.e., Data Center) is approximately 14,000 ns, which reflects the delay introduced during PTP message transmission via optical fiber. Conversely, the second-level *wireless*-PTP BC at Agronomy Farm shows a delay of nearly 60,000 ns, adding an extra 46,000 ns compared to the fiber medium.

The distribution of offset of *wireless*-PTP and *wired*-PTP at different levels of BC is illustrated in Figure 6. At the first-level BC (Wilson Hall), the offset for *wired*-PTP is tightly clustered around 0 ns, ranging from -35 ns to 35 ns. At the second-level BC (Data Center), the *wired*-PTP distribution remains stable, ranging from -50 to 50 ns, showing a minimal increase in time error. Conversely, the second-level *wireless*-PTP BC at Agronomy Farm shows a much broader distribution of time error, ranging from -6,000 ns to 6,000 ns. This significant increase in offset can be attributed to the inherent variability and less predictable nature of wireless channels as well as the impact of environmental factors on channel conditions. Figure 6 highlights the challenges associated with wireless-PTP propagation, where environmental factors and the less predictable nature of wireless channels lead to synchronization discrepancies. On the contrary, wired-PTP shows greater stability and lower time error associated with propagation across different levels of BCs.

The dependencies of *wireless*-PTP Mean Path Delay (MPD) on the health parameters of wireless link, such as Signal-to-Noise Ratio (SNR) and Received Signal Level (RSL), are illustrated in Figure 7. The Spearman correlation coefficient of MPD with RSL is -0.738, while with SNR, the coefficient is -0.754, indicating strong negative monotonic dependencies. The negative correlation coefficients underscore the
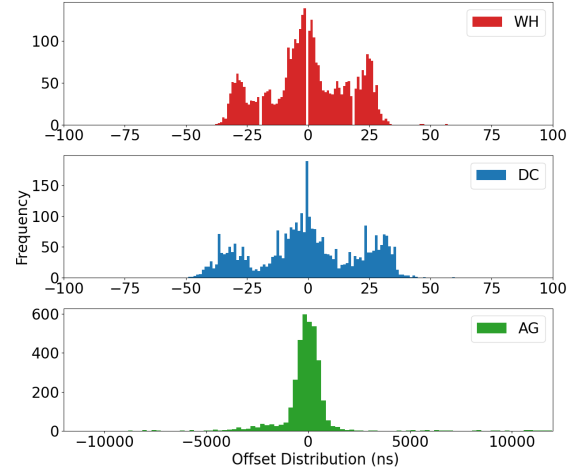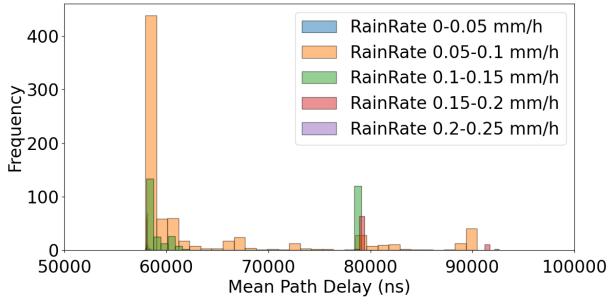

Figure 6: Offset distribution between *wireless*-PTP and *wired*-PTP at different levels of BCs.

sensitivity of PTP performance to wireless channel conditions. Figure 7 shows the characteristics of wireless channels for the duration of 14 hours. It is observed that higher SNR values, indicating better signal quality, correspond to lower MPD values. The RSL over the same period indicates a strong negative correlation with MPD.
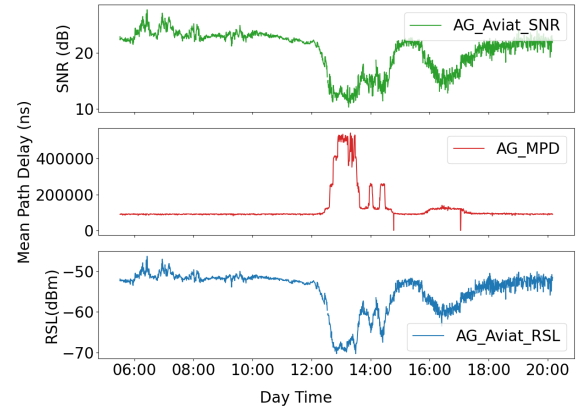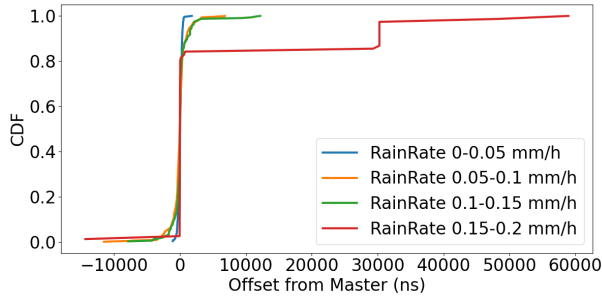

Figure 7: *Wireless*-PTP Mean Path Delay dependencies on wireless channel health.

Figure 8 indicates the distribution of MPD under varying rain conditions. The MPDs are predominantly clustered around 60,000 ns for rain rates of 0–0.05 mm/h and 0.05–0.1 mm/h, with a particularly high frequency for the latter. Another notable cluster is observed around 80,000 ns for the rain rates of 0.1–0.15 mm/h, 0.15–0.2 mm/h, and 0.2–0.25 mm/h. It is interesting to note that higher rain rates result in distributed and less frequent MPD values indicating more variability and less predictable PTP transmission,

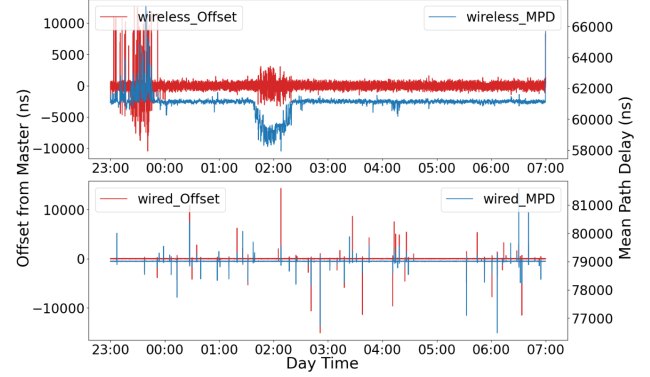**Figure 8: *Wireless*-PTP: MPD dependencies on rain-rate.**

Figure 9: *Wireless*-PTP offset from GMC vs. rain-rate.

Figure 10: *Wired*- and *Wireless*-PTP offset and MPD from GMC at Agronomy Farm.

synchronization. These comparisons highlight the superior stability and consistency of *wired*-PTP over *wireless*-PTP. We could achieve sub-microsecond level PTP performance, nearly as good as *wired*-PTP through the wireless link in this case; however, the *wireless*-PTP fluctuation is higher due to external factors.

Lessons Learned: The primary requirement to enable PTP synchronization is to select an optimal location for the GMC with sufficient satellite signal coverage. Since the data center cannot be considered the central point for *AraSync* to place the GMC due to its enclosed nature and surrounding buildings, we chose the Wilson Hall rooftop for the GMC, ensuring a view of the unobstructed sky. Another major challenge includes the identification and resolution of the mismatch between PTP-profiles and the layer (L2 or L3) on which PTP operates in switches. For example, the GMC supports g8275.1 for L2-PTP and g8275.2 for L3-PTP, while the Cisco switch BC supports only g8275.1-L2/L3 PTP. In fact, on a PTP configuration change, mid-level BCs enter into an *uncalibrated* state with their master, which was resolved by PTP rebooting multiple times on the first-level BC.

AraSync Use Cases: As an advanced research platform, PTP-based synchronization enables advanced wireless experiments in ARA involving time-critical network functions. For example, synchronization helps to precisely align data frames and time slots in 5G and beyond systems [21]In fact, PTP is a critical requirement in Open RAN to ensure the precise timing of the Enhanced Common Public Radio Interface (eCPRI) interface between Distributed Unit (DU) and Radio Unit (RU). Further, *AraSync* allows us to measure finer-level latency at different layers at both BS and UE to understand the impact of latency in URLLC applications. Finally, PTP is an essential requirement for massive MIMO and beamforming, network slicing, resource scheduling, and interference mitigation in Next-G wireless systems.

potentially leading to higher jitters and packet delays. In contrast, lower rain rates are associated with more frequent and stable MPD values, suggesting more consistent path delays.

In Figure 9, we show the Cumulative Distribution Function (CDF) illustrating the impact of rain on the second-level BC offset, i.e., the Agronomy Farm clock connected over Ara-Haul mmWave wireless channel operates at 80 GHz, from the GMC. Low rain-rates (0–0.05 mm/h, 0.05–0.1 mm/h) result in most offsets close to zero, indicating minimal offset and high synchronization accuracy. For moderate rain (0.15–0.2 mm/h) follows a similar trend as that of low rain-rate, however, with slightly more offset variability. Higher rain-rates (0.15–0.2 mm/h) show a gradual CDF increase, indicating more offset spread (up to 60,000 ns) and higher variability, i.e., heavy rain negatively impacts the precision of PTP-based synchronization.

The performance of PTP over wireless and wired connections at the Agronomy Farm BC is illustrated in Figure 10. *Wired*-PTP shows significantly more stability in offset and MPD compared to *wireless*-PTP, which experiences more fluctuations due to susceptibility to external factors, such as signal attenuation, interference, and environmental conditions, affecting synchronization precision. Additionally, the *wired*-PTP shows more consistent offset and MPD with fewer deviations, achieving better synchronization accuracy around 0 ns offset. However, the MPD is higher due to higher hop counts. In contrast, *wireless*-PTP has larger deviations, especially around 23:00 and 02:00, indicating less precise

6 CONCLUSION

In this paper, we discuss *AraSync*, an infrastructure-level framework for high-precision time synchronization to enable time-sensitive wireless experimentation. *AraSync* is primarily built on the top of PTP with a grand master clock and different levels of boundary clocks. Even though PTP is used over fiber to synchronize large-scale systems, *AraSync* is unique in synchronizing devices using PTP over long-range high-capacity wireless backhaul (i.e., *AraHaul*) involving mmWave and microwave links in addition to the optical fiber based wired links. Further, we studied the variations in offset and mean path delay of BCs at different levels toward the GMC. Our measurements shed light on the impact of dynamic wireless channel characteristics as well as weather conditions on the synchronization accuracy. Our framework and analysis can help in designing time-sensitive next generation wireless communication systems, such as Open RAN, involving wireless x-haul.

REFERENCES

[1] 2002. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2002* (2002), 1–154. https://doi.org/10.1109/IEEESTD.2002.94144

[2] 2008. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)* (2008), 1–269. https://doi.org/10.1109/IEEESTD.2008.4579760

[3] 2020. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2019 (Revision ofIEEE Std 1588-2008)* (2020), 1–499. https://doi.org/10.1109/IEEESTD.2020.9120376

[4] 2020. IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications. *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)* (2020), 1–421. https://doi.org/10.1109/IEEESTD.2020.9121845

[5] 2021. IEC/IEEE International Standard - Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEC/IEEE 61588-2021* (2021), 1–504. https://doi.org/10.1109/IEEESTD.2021.9456762

[6] 2021. IEEE Draft Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment: Addition of Precision Time Protocol (PTP) mapping for transport over Optical Transport Network (OTN). *IEEE P1588b/D1.0, August 2021* (2021), 1–14.

[7] 2022. ACX710 Universal Metro Router Datasheet. https://www.juniper.net/content/dam/www/assets/datasheets/us/en/routers/acx710-universal-metro-router-datasheet.pdf. [Online; accessed 14. July. 2024].

[8] 2023. Configuring Precision Time Protocol (PTP). https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst9500/software/release/16-11/configuration_guide/lyr2/b_1611_lyr2_9500_cg/configuring_precision_time_protocol__ptp_.pdf. [Online; accessed 11. July. 2024].

[9] 2023. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Amendment 2: Master-Slave Optional Alternative Terminology. *IEEE Std 1588g-2022* (2023), 1–14. https://doi.org/10.1109/IEEESTD.2023.10070440

[10] 2024. ACX7100 Cloud Metro Router Datasheet. https://www.juniper.net/content/dam/www/assets/datasheets/us/en/routers/acx7100-

cloud-metro-routers-datasheet.pdf. [Online; accessed 14. July. 2024].

[11] 2024. Cisco Catalyst 9300 Series Switches Data Sheet - Cisco. https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-9300-series-switches/nb-06-cat9300-ser-data-sheet-cte-en.html. [Online; accessed 14. July. 2024].

[12] 2024. Layer 2 Configuration Guide, Cisco IOS XE Dublin 17.10.x (Catalyst 9300 Switches). [Online; accessed 11. July. 2024].

[13] 2024. OSA 5405 Series. https://www.oscilloquartz.com/en/products-and-services/ptp-grandmaster-clocks/integrated-gnss-ptp-grandmasters/osa-5405-series [Online; accessed 07. July. 2024].

[14] Douglas Arnold. 2023. End-to-End Versus Peer-to-Peer. *Time Synchronization* (May 2023). https://blog.meinbergglobal.com/2013/09/19/end-end-versus-peer-peer [Online; accessed 08. July. 2024].

[15] Douglas Arnold. 2023. NTP vs PTP: Network Timing Smackdown! *Time Synchronization* (May 2023). https://blog.meinbergglobal.com/2013/11/22/ntp-vs-ptp-network-timing-smackdown [Online; accessed 10. July. 2024].

[16] Chamitha de Alwis, Quoc-Viet Pham, and Madhusanka Liyanage. 2023. *6G Requirements*. 21–33. https://doi.org/10.1002/9781119862321.ch3

[17] Ayush Garg, Akash Yadav, Axel Sikora, and Ashok Singh Sairam. 2018. Wireless Precision Time Protocol. *IEEE Communications Letters* 22, 4 (2018), 812–815. https://doi.org/10.1109/LCOMM.2017.2781706

[18] Taimoor Ul Islam et al. 2024. Design and Implementation of ARA Wireless Living Lab for Rural Broadband and Applications. *arXiv preprint arXiv:2408.00913* (2024).

[19] Renaud Lavoie. 2018. PTP: Transparent Versus Boundary Clock in Broadcast Environments. *The Broadcast Bridge* (Jan. 2018). https://www.thebroadcastbridge.com/content/entry/10287/ptp-transparent-versus-boundary-clock-in-broadcast-environments

[20] Assrar Maamary, Hyame Assem Alameddine, Mourad Debbabi, and Chadi Assi. 2024. Synchronization Plane in O-RAN: Overview, Security and Research Directions. *IEEE Communications Magazine* (2024), 1–7. https://doi.org/10.1109/MCOM.001.2300563

[21] Zhibo Meng and Hongwei Zhang. 2024. Joint Scheduling and Power Control for Predictable Per-Packet Reliability in URLLC. In *The 32nd IEEE International Conference on Network Protocols(ICNP)*.

[22] D.L. Mills. 1991. Internet time synchronization: the network time protocol. *IEEE Transactions on Communications* 39, 10 (1991), 1482–1493. https://doi.org/10.1109/26.103043

[23] Oleg Obleukhov, Alexander Bulimov, and Ahmad Byagowi. 2023. Simple Precision Time Protocol (SPTP). In *2023 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*. 1–6. https://doi.org/10.1109/ISPCS59528.2023.10296989

[24] John O'Raw, David Laverty, and Danny McFadden. 2023. Precision Time Protocol (PTP) Using LinuxPTP on Single Board Computers. In *2023 58th International Universities Power Engineering Conference (UPEC)*. 1–5. https://doi.org/10.1109/UPEC57427.2023.10294864

[25] Steve T. Watt, Shankar Achanta, Hamza Abubakari, Eric Sagen, Zafer Korkmaz, and Husam Ahmed. 2015. Understanding and applying precision time protocol. In *2015 Saudi Arabia Smart Grid (SASG)*. 1–7. https://doi.org/10.1109/SASG.2015.7449285

[26] Hongwei Zhang et al. 2022. ARA: A wireless living lab vision for smart and connected rural communities. In *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization*. 9–16.

[27] Guoying Zu, Md Nadim, Salil Reddy, Taimoor UI Islam, Sarath Babu, Tianyi Zhang, Daji Oiao, Hongwei Zhang, and Anish Arora. 2023. AraHaul: Multi-Modal Wireless X-Haul Living Lab for Long-Distance, High-Capacity Communications. In *2023 IEEE Future Networks World Forum (FNWF)*. 1–6. https://doi.org/10.1109/FNWF58287.2023.10520543