

# Time Transfer and Clock Synchronization Analysis over Spine-Leaf Networks

Alberto Ortega, Víctor Vázquez, Carlos Megías,

Jesús González, Manuel Rodríguez-Álvarez, Javier Díaz and Eduardo Ros

*Department of Computer Engineering, Automation, and Robotics, CITIC, University of Granada*  
Granada, Spain

{aoruz, victorvazrod, narg,  
jesusgonzalez, manolo, jda, eros} @ugr.es

**Abstract**—The integration of a time synchronization and time transfer system in data center networks is an increasingly common demand. Since moving the entire infrastructure of a data center network to switches with time-dedicated hardware is nowadays a considerable expense of resources, it is convenient to analyze and study it. In this paper we work on real hardware, presenting a characterization of the problem for a Spine-Leaf network by means of load variation tests proposed by the ITU, using the PTP protocol and common non-PTP aware switches. Through this study we manage to show, thanks to real measurements, the problem of dynamic asymmetry and how it affects the time distribution. Additionally, it is shown that if the network load is quadrupled, the PTP pk-pk offset through the network does not even double. Finally, thanks to long-term tests, we can conclude that time synchronization in a Spine-Leaf network shows considerable resilience to load changes.

**Index Terms**—Spine-Leaf, PTP, datacenter, clock synchronization

## I. INTRODUCTION

The design of a data center is complex and can be addressed with different approaches, with them being mainly influenced by its final utilization. A data center must be designed with specific characteristics that facilitate the execution of the applications that will be used in the infrastructure [28]. When designing a data center, some fundamental aspects must be taken into account. These include not only the nodes/servers and their computational capacities, but also the Data Center Network (DCN) [25]. In fact, the DCN is one of the most important components when considering the design of a data center: how to intercommunicate the different nodes, racks, and switching elements of the DCN.

Within the design of the DCN, there are many aspects to consider. Focusing on some current issues, such as the recent increase in distributed applications [21], [30] or the inherent time sensitivity of the data center traffic itself [6], [7], [19], both are important points to take into consideration in the design of a DCN. For this purpose, time synchronization systems are designed and implemented in DCNs so that all the equipment in the system can share the same time reference with a certain degree of accuracy.

Therefore, setting up a time distribution system that allows for accurate time in a data center offers several advantages: consistency of data in distributed systems (such as distributed

databases), performance in scheduling processes (such as in stock exchange systems), or preserving the actual order of events (such as the administration of the datacenters themselves).

Regarding time distribution, there are two main aspects to take into account in the design: the technology used and the DCN on which to apply that technology. Since, except for specific cases, most data center networks use Ethernet as the backbone network protocol, in this paper the focus will be on a time transmission protocol that works on Ethernet traffic.

One of the most widespread time distribution protocols today is the Precision Time Protocol (PTP), defined in the IEEE 1588-2002 standard [1], which allows specific time synchronization traffic to be introduced alongside standard network traffic. This protocol does not depend on the topology of the network itself, since it works directly over Ethernet networks. PTP works with time sources that serve their time via Ethernet packets to end clients, which update their offset with respect to the time source and adjust their internal clock. In the IEEE 1588-2008 [2] revision, several innovations were proposed to improve time distribution to the end devices of the network, such as transparent clocks, profiles, higher sampling rates, etc. In 2019, there was an update of PTP with the IEEE 1588-2019 [3], which included new features such as the calibration of asymmetries or the inclusion of the White Rabbit technology [29] as a high accuracy profile for PTP. These new PTP capabilities in switches allow the protocol to more accurately distribute time to end clients, but these advanced features require specific dedicated hardware on the network (the network switches need to include these features). Since PTP-aware switches have a higher price in the market due to that dedicated hardware, the cost of replacing the entire data center infrastructure with such switches is labor-intensive and costly.

Without this dedicated hardware, one of the most detrimental aspects of PTP, due to its internal calculation, are the asymmetries in the network. These asymmetries can be due to different causes: asymmetries in the routes, increases in traffic load, overloading of switches, etc. These asymmetries can be either static or dynamic: 1) static asymmetries, mainly due to the network topology, and can be usually calibrated; 2) dynamic asymmetries, mainly due to variations in latencies,

which are mostly due to network load.

In terms of network topologies [25], current DCN deployments eliminate static asymmetries in the network, facilitating the distribution of time across them. In recent years, DCNs have increasingly moved away from the standard three-level model to the Spine-Leaf model. This topology allows east-west communication between different Leafs always using the same number of hops, facilitating deterministic behavior. This eliminates any asymmetry inherent to the DCN topology itself.

Keeping an eye on all of the above, there are works on these aspects in the literature, but none with a concise DCN architecture, using standard traffic profiles and accurate measurement instruments. Thus, this paper proposes an accurate characterization of the time distribution in a Spine-Leaf network using standard tests proposed by the International Telecommunication Union (ITU) [17], using open software, PTP protocol, and non-capable standard PTP switches. Thanks to this study, it will be possible to appreciate the impact of network load on time synchronization and, therefore, to understand the effect of implementing a non-PTP compliant timing system in a DCN.

The rest of the paper is organized as follows: Section II presents the design to be taken for the characterization of the time transfer system in DCNs. Subsequently, Section III describes the adopted experimental methodology, and Section IV describes the implementation of the final design in a real testbed. Section V details the results obtained by performing these experiments. Finally, Section VI draws the final conclusions and a proposal for future work.

## II. DESIGN

Classic DCNs are usually organized into three levels. The first is the Top of the Rack (ToR): the switch that is in the racks and serves as access to the rest of the network. The top two levels are high-capacity switches that enable intercommunication between racks. In these classical networks, switches at the same level may be connected to each other, creating unpredictable routes between end nodes, even if the topology is known.

The Spine-Leaf model simplifies the classic topology so that there are only two basic levels: the Spine level (top) and the Leaf level (bottom). The Leaf level is the access level (it may be ToR or not), with it being the first switch to find an end node. The Spine level is the intermediary for any communication between different Leaf switches, so all Leaf switches are connected to all Spines, but never to each other. In these Spine-Leaf networks, the major workload is carried by the Spine switches, which are the intermediaries for all the information flowing through the network. Leaf switches, on the other hand, only process the information belonging to the end nodes connected to them. In Spine-Leaf networks, the routes between the end nodes are also not predictable, but the number of hops will always be known.

In order to create a Spine-Leaf network with which to test different network loads for a limited time, the simplest possible Spine-Leaf network has been chosen in this work: 2 Spine

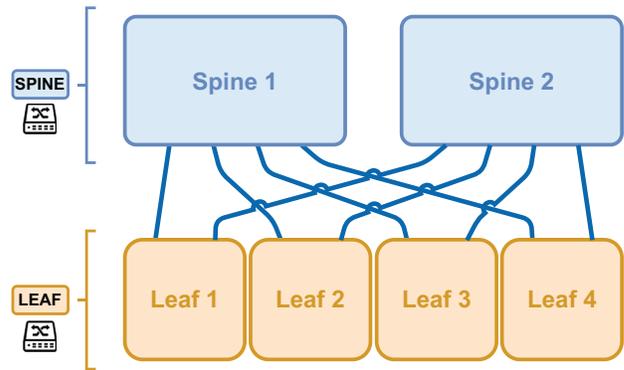


Fig. 1. Example of a Spine-Leaf network with 2 Spine switches and 4 Leaf switches.

switches and 4 Leaf switches, as Figure 1 shows. In this work, a Layer 2 [5] Spine-Leaf network has been chosen, i.e., the switches will only work with the MAC Address Table and no Layer 3 will be used. This Layer 2 implementation has dynamic load balancing and loop control through the IEEE 802.1AX standard [4].

Once the network topology is defined, the time synchronization system will be incorporated into it. A time reference is provided by a PTP grand master, which will be in charge of maintaining the time reference and distributing it to the PTP clients. Hence, PTP clients will be able to synchronize with it and have a local representation of the global time reference. Where to connect the time server within Spine-Leaf is neither standard nor trivial and depends on the data center itself. For example, it might be desirable to connect a time server on each Leaf or, rather, have a global time server connected on a Spine accessible to the entire data center.

For this work, in which the aim is to characterize the time distribution in Spine-Leaf networks, it has been decided to place the PTP grand master in one Leaf and the client in a different Leaf. So, a 2-hop east-west communication is generated, which simulates the farthest case of time synchronization, with it being the one with the highest number of hops and the one that allows better observability on how switch processing affects the PTP protocol. Finally, the PTP client is a device capable of maintaining its internal local clock adjusted according to the PTP grand master.

Thus, the Spine-Leaf topology has been defined, and the PTP system has been designed and placed within the topology. The methodology to be followed to characterize this design is defined below.

## III. EXPERIMENTAL METHODOLOGY

Before implementation, it is necessary to define how to characterize the network, as this may influence the physical testbed. It is important to choose a set of experiments relevant to the design described above. To do this, the key aspects to be measured must be known.

If one looks at previous studies [11], [24], when characterizing any network, one aspect that is prioritized for measurement is latency. The amount of time elapsed since a message departs from point A to point B is a measure that needs to be taken into account.

As for time synchronization, one clock is synchronized with respect to another by raising and lowering the frequency of its oscillator. These rises and falls of frequency in a normal working environment are never perfect, which designates a frequency error, and, therefore, a time error of the clock with respect to its reference.

Remembering the introduction, the aspect of a network that most affects PTP, and therefore its time error (offset), is when there is an asymmetry. When there is an asymmetry in the network, a variation in latency occurs, which is called Packet Delay Variation (PDV). Both static and dynamic asymmetries negatively affect the PTP offset. Static asymmetries are eliminated thanks to the Spine-Leaf topology, as explained above. However, dynamic asymmetries are produced by the load on the network, which produces queuing in the switches and therefore PDV. This PDV affects PTP in such a way that it internally makes incorrect latency estimates and, therefore, incorrect offset corrections.

Thus, we will focus on two aspects to characterize: synchronization time error and PDV. In order to observe how both values vary in a Spine-Leaf network, it is necessary to design several experiments that allow us to appreciate how they are affected by topology, switches, and workloads. For this purpose, this work has used as a basis for the experimental phase some test cases defined by the International Telecommunication Union (ITU) in Appendix VI of its document G.8261/Y.1361 [18]. Adopting these well-defined ITU experiments facilitates their reproducibility and serves as a benchmark with respect to other DCN approaches. These test cases are based on injecting traffic into the network in different ways, emulating real network use cases. In this work, they will be used to test a Spine-Leaf network and observe how both the PDV and the offset of a PTP client react to the network traffic.

### A. Traffic Model

First, the traffic to be injected into the network must be defined. The traffic model defined in Section VI.2.2 of [18] is used for this purpose, and different use cases are defined in that reference document. For a defined load on the network, the composition of the traffic is established as:

- 60% of the load are maximum size packets: 1518 Bytes.
- 10% of the load are medium size packets: 576 Bytes.
- 30% of the load are minimum size packets: 64 Bytes.

### B. Experiment 1

Experiment 1 (based on test case 1) models a static load on the network. Two tests have been designated: one at 20% static load for 1 hour and one at 80% for another hour.

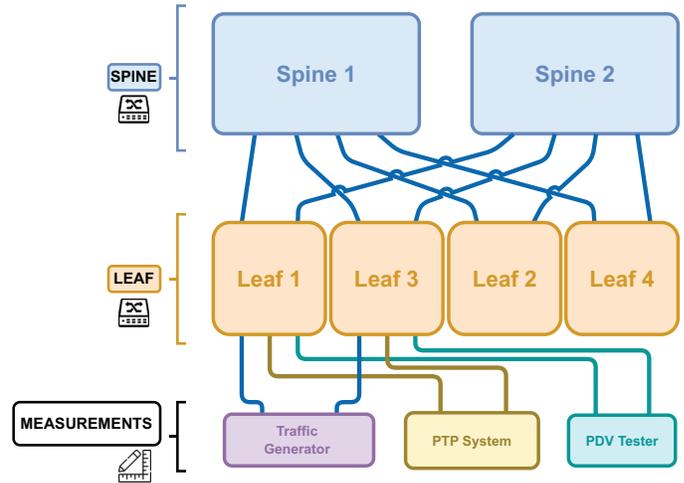


Fig. 2. Final design of the Spine-Leaf testbed. The traffic generator, the PTP system and the PDV tester have been added at the bottom.

### C. Experiment 2

Experiment 2 (based on test case 2) models large and persistent changes in network load and demonstrates stability for sudden large changes. The test will present the following load profile: the network starts at 80% for 1 hour, drops to 20% for 1 hour, 80% for 1 hour, 20% for 1 hour, 80% for 1 hour, and finally drops to 20% for 1 hour.

### D. Experiment 3

Experiment 3 (based on test case 3) models the slow change in network load over a long time scale, demonstrating stability with very slow changes in network conditions. This is done by smoothly varying the grid load from 20% to 80% and vice versa over a 24-hour period. It starts at 20% increasing by 1% every 12 minutes, until reaching 80% in 12 hours. From there, it starts decreasing the load by 1% every 12 minutes, returning to 20% after 24 hours.

Once the aspects to be measured and the experiments with which to perform the characterization have been described, the following Section IV will present the implementation of the design on a real experimental testbed.

## IV. IMPLEMENTATION

The implementation of the system designed in Section II is presented here with the addition of the necessary tools to perform the measurements proposed in Section III. The switches chosen for the Spine-Leaf testbed are: 2 *MikroTik CRS326-24S+2Q+RM* [23] in the Spine layer and 4 *MikroTik CRS310-1G-5S-4S+IN* [22] in the Leaf layer. For the links between the devices, 10 Gbps DAC wires have been used. Thus, the final testbed design is arranged as shown in Figure 2:

- Traffic Generator: an *Intel Ethernet E810* [15] NIC card has been used together with the *CISCO TRex* [8] traffic generator to run traffic profiles for each experiment and to be able to generate the desired amount of load on the network.

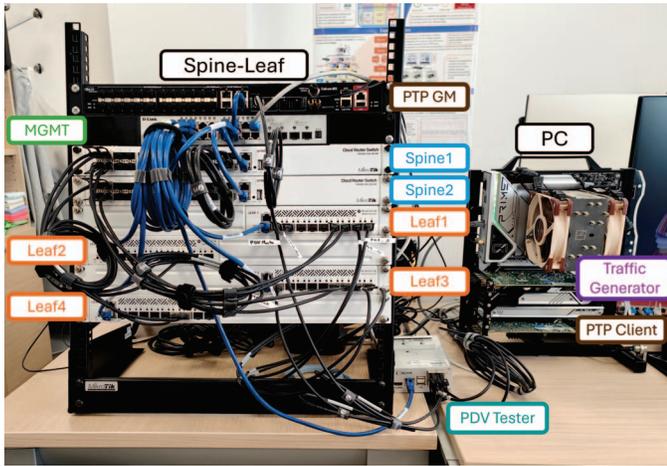


Fig. 3. Laboratory Spine-Leaf testbed.

- PDV Tester: a *Relyum RELY-TSN-LAB* [27] device was used to extract the timestamps of certain packets passing through it. Using a program that sends 1 probe packet per second, it is possible to measure the time through the timestamps it takes to go to and return from the network.
- PTP System: a *FibroLAN Falcon-MX/G* [12] PTP grandmaster and an *Intel Ethernet XXV710* [16] NIC card acting as PTP client have been used, all synchronized through the *Linux PTP Project* [9] implementation. A simple IPv4 Unicast PTP profile has been applied, with default values [10] for the `ptp4l` program parameters. To measure the offset, a frequency counter was used to compare the PPS pulses from the grandmaster and the client [13].

It should be pointed out that both the *Intel Ethernet E810* NIC and the *Intel Ethernet XXV710* NIC are installed on a computer with 64 GB of RAM, *Intel 13900K* CPU, running *Ubuntu 22.04*. On this computer, several applications are used/executed: the *CISCO TRex* traffic generator, the latency packet generator, and the *Linux PTP Project* program. Figure 3 shows the actual experimental testbed in the laboratory.

Both switches (Spines and Leafs) are capable of switching at such high speeds that it is almost impossible to saturate them with a simple traffic generator. Thus, in this implementation, the goal of increasing the network load is to saturate the capacity of the links. Since all links operate at 10 Gbps, the percentage of injected load will always be above this speed (e.g., 10% = 1Gbps). In addition, the traffic generator will work with bidirectional traffic, i.e., the links will be saturated in both transmission and reception.

## V. EXPERIMENTAL RESULTS

Figure 4 shows the offset distribution for 20% and 80% load for Experiment 1. It can be seen that even with a quadrupling of the load on the network, the distribution amplitude does not even double. This means that PTP withstands the load increase in a Spine-Leaf network quite well.

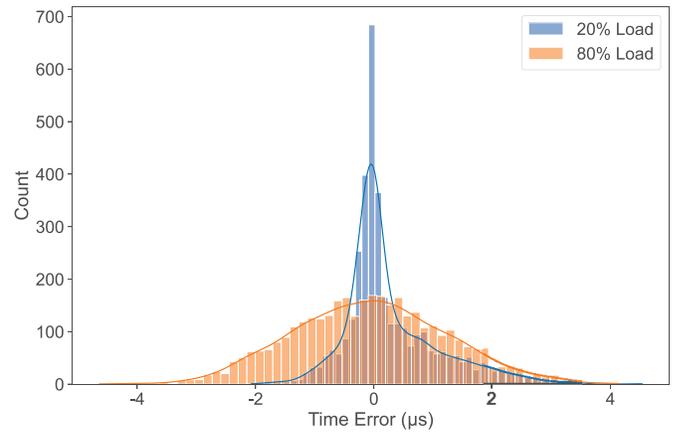


Fig. 4. Offset distribution obtained from the frequency counter for Experiment 1. Count histogram with 60 bins and *kde* included, 20% (blue) and 80% load (orange), duration of 1 hour for each load.

TABLE I  
PDV RESULTS FOR EXPERIMENT 1.

Network Load (%)	Max ( $\mu\text{s}$ )	Median ( $\mu\text{s}$ )	Std Dev ( $\mu\text{s}$ )
0 ( <i>Baseline</i> )	0.207	0.090	0.027
20	5.409	0.063	0.765
80	7.389	2.439	1.419

However, there is something remarkable to note: while the offset distribution at 80% load is a perfect Gaussian bell, the distribution at 20% load has a clear asymmetry: a positive skew. To find out why this happens, it is necessary to look at Table I. This table shows the values obtained for the PDV in experiment 1. Thus, at 20% load, a maximum PDV value similar to that of 80% is observed, but with half the standard deviation and with the median centered on the left. Thus, the distribution at 20% is skewed: not all packets take the same time to traverse the network. Meanwhile, at 80%, a normal distribution with the median centered is re-created: all packets take the same on average. The dynamic asymmetry in the PDV generates a problem in the calculation of the offset estimation for the PTP and, therefore, generates asymmetry in the offset. The normal distribution can only be confirmed when there is no traffic at all or the network is almost full of traffic, so all packets will be queued in the same way and there will be no asymmetries.

Figure 5 shows, thanks to the boxplots, how sudden changes in the network do not excessively affect the time distribution for Experiment 2. In fact, there is no cumulative error affecting PTP over time either.

Before showing Experiment 3, a smaller version of the experiment has been added in which the offset distribution is observed from 0% load to 20% load to see if there is a continuous progression or an abrupt change. Figure 6 shows how the offset distribution increases as the load increases from 0%. It can be seen how up to 7-9% the offset shifts in nanoseconds, and from this load there is a jump of order of magnitude to shift to microseconds. In fact, it can be seen

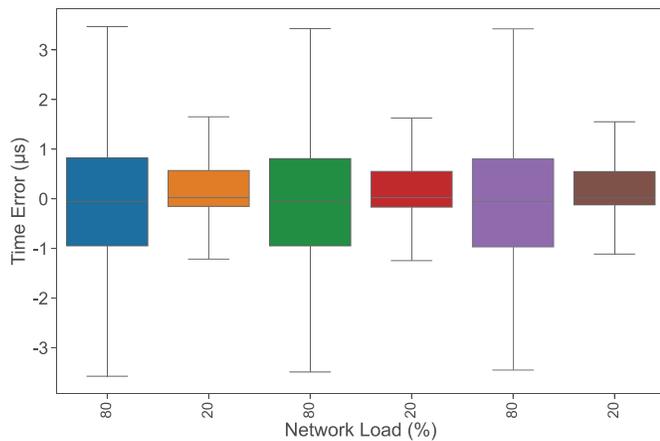


Fig. 5. Offset distribution obtained from the frequency counter for Experiment 2. Boxplot without outliers, duration of 1 hour for each load.

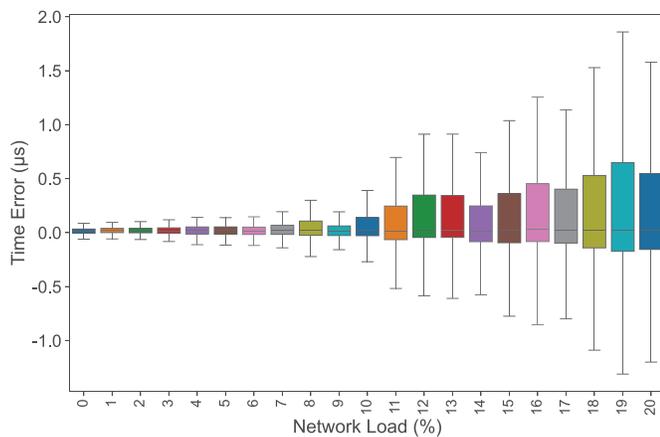


Fig. 6. Offset distribution obtained from the frequency counter for Experiment 3 (0-20%). Boxplot without outliers, duration of 12 minutes for each load.

how the bias seen in Experiment 1 is formed.

Finally, the complete Experiment 3 is presented in Figure 7. This experiment demonstrates what has already been seen in the previous experiments, but now in a progressive way, increasing the load by 1% every 12 minutes. The offset gradually increases, losing bias each time the network becomes more saturated, forming a normal distribution at peak load, and then returning to the initial values as the load decreases.

## VI. CONCLUSIONS AND FUTURE WORK

This work has allowed an experimental characterization of time transfer over a DCN. Starting with a design of the network itself and the synchronization system, followed by a well-defined experimental methodology, and ending with an actual implementation of the system and system testing. In fact, a feature that has been followed throughout the work has been to choose protocols and use cases standardized by organizations such as the IEEE or the ITU. All this facilitates a simple but, at the same time, effective and reproducible implementation of the characterization that was proposed

at the beginning of the work. Furthermore, obtaining real, physical measurements on a testbed with real hardware is something that yields valuable knowledge on the subject.

Observing the results confirms what was said at the beginning: Spine-Leaf networks are networks that withstand quite effectively the stress of a high load on the time distribution, managing to reduce the impact on time synchronization. One of the most valuable conclusions is how, by quadrupling the network load, time synchronization is not affected by the same factor. It has also been possible to study the load percentage threshold at which packets begin to queue on non-PTP aware switches. Both experiments of load changes in the network also demonstrate the resilience of the network, which does not disproportionately affect time synchronization. Whether there are progressive changes or sudden changes, the Spine-Leaf network is an optimal network for time distribution.

Still, the asymmetries observed with a medium load in the network are due to hardware queuing in the switches, which raises several ideas for future work. Reproducing this work with switches that include dedicated hardware for PTP functionalities such as boundary clocks or transparent clocks would be an even more interesting comparison. This comparison would provide answers to the question of whether PTP switches are really worth the investment of resources in a DCN using a Spine-Leaf configuration.

Looking at another aspect of the study, the network used here has been designed at Layer 2, but it would be interesting to reproduce these measurements with a Spine-Leaf network designed at Layer 3.

Since a standard PTP profile has been used in this analysis, another way to continue this study could be to try different PTP filters or profiles [26], trying to improve the accuracy of the final offset. Providing a comparison of all this with other proposed synchronization systems [14], [20] for datacenters would complete an intensive study on the subject.

## ACKNOWLEDGMENT

This work was supported partially under AMIGA-8 grant (PID2021-123930OB-C22) funded by the Spanish Ministry of Science, Innovation, and Universities; partially under DAIS grant (No. 101007273-2 within the EC-SEL Calls and PCI2021-121967) funded by MCIN/AEI /10.13039/501100011033 and EU NextGenerationEU/ PRTR; partially under INTARE grant (TED2021-131466B-I00), funded by MCIN/AEI /10.13039/501100011033 and the European Union NextGenerationEU/ PRTR; partially under Ayudas para contratos predoctorales para la formación de doctores/as 2022 (PRE2022-103110) funded by MCIN/AEI /10.13039/501100011033 and ESF+; and partially by the Formacion de Profesorado Universitario (FPU20/05842 and FPU20/01857) funded by MCIN/AEI /10.13039/501100011033.

## REFERENCES

- [1] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2002*, pages 1–154, 2002.

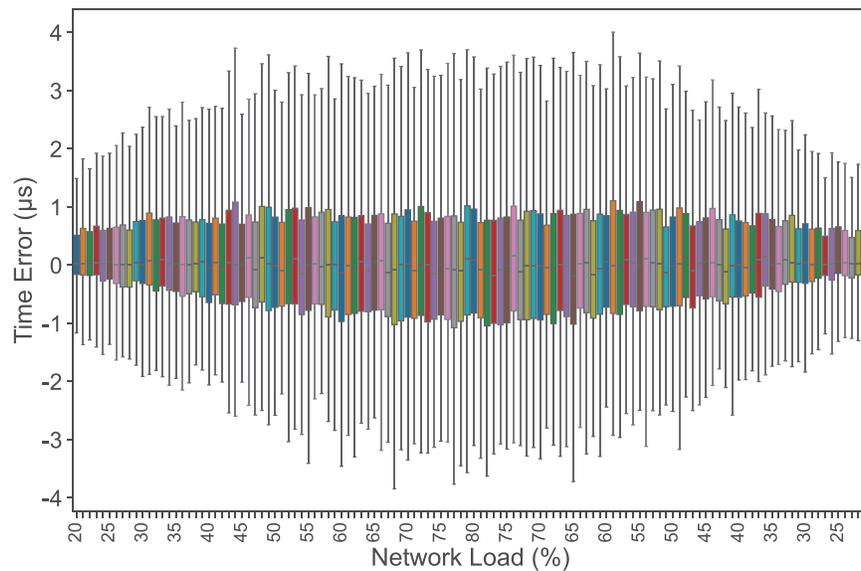


Fig. 7. Offset distribution obtained from the frequency counter for Experiment 3 (20-80-20%). Boxplot without outliers, duration of 12 minutes for each load.

- [2] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pages 1–269, 2008.
- [3] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008)*, pages 1–499, 2020.
- [4] IEEE Standard for Local and Metropolitan Area Networks—Link Aggregation. *IEEE Std 802.1AX-2020 (Revision of IEEE Std 802.1AX-2014)*, pages 1–333, 2020.
- [5] Mohammed M. Alani. *OSI Model*, pages 5–17. Springer International Publishing, 2014.
- [6] Theophilus Benson, Aditya Akella, and David A. Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, page 267–280. Association for Computing Machinery, 2010.
- [7] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. Understanding data center traffic characteristics. *SIGCOMM Comput. Commun. Rev.*, 40(1):92–99, 2010.
- [8] CISCO. TRex Realistic Traffic Generator. [trex-tgn.cisco.com](http://trex-tgn.cisco.com), accessed: 2024-05-04.
- [9] Richard Cochran. The Linux PTP Project. [linuxptp.sourceforge.net](http://linuxptp.sourceforge.net), accessed: 2024-05-04.
- [10] Richard Cochran. The Linux PTP Project Documentation: ptp4l. [linuxptp.nvtime.org/documentation/ptp4l](http://linuxptp.nvtime.org/documentation/ptp4l), accessed: 2024-08-29.
- [11] Luca De Vito, Sergio Rapuano, and Laura Tomaciello. One-Way Delay Measurement: State of the Art. *IEEE Transactions on Instrumentation and Measurement*, 57(12):2742–2750, 2008.
- [12] FibroLAN. FibroLAN Falcon-MX/G. [www.fibrolan.com/Falcon-MX-G](http://www.fibrolan.com/Falcon-MX-G), accessed: 2024-05-04.
- [13] Brandon Frech, Julian St. James, Armando Pinales, and Ahmad Byagowi. Characterizing the Advanced Synchronization Capabilities of PTM Enabled Hardware. In *2023 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, pages 1–5. IEEE, 2023.
- [14] Yilong Geng, Shiyu Liu, Zi Yin, Ashish Naik, Balaji Prabhakar, Mendel Rosenblum, and Amin Vahdat. Exploiting a Natural Network Effect for Scalable, Fine-grained Clock Synchronization. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, pages 81–94, 2018.
- [15] Intel. Intel Ethernet Network Adapter E810-XXVDA4T. [intel.com/content/www/us/en/products/sku/211608/intel-ethernet-network-adapter-e810xxvda4t/specifications.html](http://intel.com/content/www/us/en/products/sku/211608/intel-ethernet-network-adapter-e810xxvda4t/specifications.html), accessed: 2024-05-04.
- [16] Intel. Intel Ethernet Network Adapter XXV710-DA2. [www.intel.com/content/www/us/en/products/sku/95260/intel-ethernet-network-adapter-xxv710da2/specifications.html](http://www.intel.com/content/www/us/en/products/sku/95260/intel-ethernet-network-adapter-xxv710da2/specifications.html), accessed: 2024-05-04.
- [17] ITU. International Telecommunication Union. [itu.int/en](http://itu.int/en), accessed: 2024-05-04.
- [18] ITU-T. Timing and synchronization aspects in packet networks. Recommendation ITU-T G.8261/Y.1361, International Telecommunication Union, Aug 2019.
- [19] Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, and Ronnie Chaiken. The nature of data center traffic: measurements and analysis. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, page 202–208. Association for Computing Machinery, 2009.
- [20] Ki Suh Lee, Han Wang, Vishal Shrivastav, and Hakim Weatherspoon. Globally Synchronized Time via Datacenter Networks. In *Proceedings of the 2016 ACM SIGCOMM Conference*, page 454–467, 2016.
- [21] Haonan Lu, Shuai Mu, Siddhartha Sen, and Wyatt Lloyd. NCC: Natural Concurrency Control for Strictly Serializable Databases by Avoiding the Timestamp-Inversion Pitfall. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 305–323. USENIX Association, 2023.
- [22] MikroTik. CRS310-1G-5S-4S-IN. [mikrotik.com/product/crs310\\_1g\\_5s\\_4s\\_in](http://mikrotik.com/product/crs310_1g_5s_4s_in), accessed: 2024-05-04.
- [23] MikroTik. CRS326-24S+2Q+RM. [mikrotik.com/product/crs326\\_24s\\_2q\\_rm](http://mikrotik.com/product/crs326_24s_2q_rm), accessed: 2024-05-04.
- [24] Vinh Quang Nguyen and Jae Wook Jeon. EtherCAT network latency analysis. In *2016 International Conference on Computing, Communication and Automation (ICCCA)*, pages 432–436, 2016.
- [25] Mohammad Noormohammadpour and Cauligi S. Raghavendra. Datacenter Traffic Control: Understanding Techniques and Tradeoffs. *IEEE Communications Surveys and Tutorials*, 20(2):1492–1525, 2018.
- [26] Michel Ouellete and Thomas Kern. Precision Time Protocol Profile for Datacenter Applications. In *2023 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, pages 1–6, 2023.
- [27] Relyum. RELY-TSN-LAB. [relyum.com/web/rely-tsn-lab](http://relyum.com/web/rely-tsn-lab), accessed: 2024-05-04.
- [28] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. Inside the Social Network’s (Datacenter) Network. *SIGCOMM Comput. Commun. Rev.*, 45(4):123–137, 2015.
- [29] J Serrano, M Cattin, E Gousiou, E van der Bij, T Wlostowski, G Daniluk, and M Lipinski. The White Rabbit project. 2013.
- [30] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S. Rellermeier. A survey on distributed machine learning. *ACM Comput. Surv.*, 53(2), 2020.