Clock Offset Estimation for Systems with Asymmetric Packet Delays

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

14-01-2022 / 19-01-2022

CITATION

Ha, Youngmok; Pak, Eunji; Park, Jongkil; Kim, Taeho; Yoon, Ji Won (2022): Clock Offset Estimation for Systems with Asymmetric Packet Delays. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.18393800.v1>

DOI

[10.36227/techrxiv.18393800.v1](https://doi.org/10.36227/techrxiv.18393800.v1)

Clock Offset Estimation for Systems with Asymmetric Packet Delays

Youngmok Ha, Eunji Pak, Jongkil Park, Taeho Kim, and Ji Won Yoon

Abstract—This paper proposes a new clock offset estimation that mitigates unwanted link asymmetry for precise clock synchronization. The main contribution is to address the design issue of the IEEE 1588 standard precision time protocol (PTP), which estimates clock offset under the assumption that the delays of exchanged packets are symmetric. To mitigate the issue, we focus on the fact that PTP measures asymmetry variation through the derivatives of its timestamps with respect to the time step. By exploiting the measurement of the variation, the proposed approach defines the asymmetry in the form of a linear differential equation (LDE) and leverages the LDE to define and exclude asymmetry-induced errors. Additionally, we clearly derive the state transition of the asymmetry. Subsequently, we derive a novel state-space model from our approach. The model describes PTP clock offset estimation perfectly, allowing optimal clock offset estimation. We verify the theoretical validity of the proposed method with real data. Our approach improves PTP accuracy by more than thousand times and achieves an accuracy at the level of tens to hundreds of nanoseconds on an asymmetric communication link. Our approach realizes an accuracy comparable to that of PTPv2, without the cost of specialized hardware.

Index Terms—Clocks, Synchronization, Precision time protocol, State space model, Kalman filtering

I. INTRODUCTION

The IEEE 1588 precision time protocol (PTP) [1], a packet exchange based protocol, is one of the most accurate clock synchronization protocols. The protocol is capable of maintaining timing differences between clocks at the submicrosecond level. Its performance is an order of magnitude improvement from the network time protocol (NTP) [2], which guarantees time differentiation at the submillisecond level. With the high accuracy, PTP has been widely adopted in networked systems for automation, control, testing, and measurement.

PTP has a well-known design issue in that the protocol uses an approximately calculated packet delay for its clock offset estimation. The protocol assumes that the delays of the exchanged packets in a master-slave architecture are symmetric, i.e., the transit time of a packet communicated from the master

to the slave is equal to the transit time of the packet conveyed from the slave to the master. However, in real environments, this assumption does not hold true as delays are asymmetric. Further, the asymmetry can nonlinearly vary over time and be magnified with congestion or increasing distance from the communication link. The disagreement between assumption and reality causes an error in the clock offset estimation. This practical error limits the use of PTP and increases the cost of expanding the PTP network.

Several studies [3]–[13] have attempted to address the aforementioned design issue of PTP. For instance, PTPv2 [3] introduced the transparent clock (TC), which is implemented inside specialized PTPv2-enabled networking devices to record the residence times of the PTP packets in the devices. PTPv2 applies these residence times to the offset estimations to reduce the errors due to asymmetry. The most popular PTP stacks, Linux PTP [10] and PTPd [11], implement packet delay filters such as moving median, averaging, or infinite impulse response (IIR) to minimize the effects of the fluctuating asymmetry. Moreover, Karthik et al. [4] proposed a method to mitigate the asymmetry of queueing delays of packets by approximating these delays using the Gaussian mixture probabilistic model and a variant of the expectation-maximization (EM) algorithm. The authors also showed that their method converges to a lower bound of the clock synchronization, which is derived using the invariant decision theory [13]. Exel et al. [5] reported that the asymmetry of a link delay occurs by the different twist rates of the Ethernet cable pairs used for transmission and reception; thus, they configured links to ensure that the transmission path of the PTP packets used the same cable pairs. Giorgi et al. [6] proposed software-based packet sampler with an oversampling strategy, which filters out packets with delays exceeding a certain value to reduce the possibility of asymmetry. The reliable range of packet delay is derived with the soft-min machine learning function and weight function defined by the Boltzmann distribution. However, these previous works partially resolve the issue or introduce their own weaknesses. They require additional hardware supports [3], [7], address only specific components of the total packet delay [3]–[5], [9], require additional packet exchanges [6]–[8], or do not provide high accuracy at the nanosecond level [6], [8], [9].

In this paper, we therefore propose a new offset estimation method that mitigates the negative impact of asymmetry. Based on the fact that the PTP timestamps represent the nonlinear asymmetry variations, we define the mathematical model of the asymmetry. Specifically, to achieve robustness against errors due to asymmetry, we first define the asymmetry

This work was supported by the IITP grants funded by the Korean government (No.B0101-16-0663, Safety-critical Distributed Modular SW Platform, and No.2018-0-00769, Neuromorphic Computing Software Platform for Artificial Intelligence Systems)

Y. Ha, E. Pak, and T. Kim are with Electronics and Telecommunications Research Institute (ETRI), Daejeon, 34129, Republic of Korea, e-mail: {ymha,pakeunji,taehokim}@etri.re.kr.

J. Park is with Korea Institute of Science and Technology (KIST), Seoul, 02792, Republic of Korea, email: jongkil@kist.re.kr.

J. Yoon is with Graduate School of Information Security, Korea University, Seoul, 02841, Republic of Korea, email: jiwon_yoon@korea.ac.kr.

in the form of a linear differential equation (LDE) using the derivatives of the timestamps with respect to the time step in the PTP architecture. Then, to improve the clock synchronization accuracy, we leverage the LDE to define and exclude errors due to the asymmetry. The novelty of our proposed approach is that it directly models the asymmetry itself, rather than estimates the asymmetry based on the packet delay values or probability distributions as in [12], [4], [13]. On the comprehension that a linear combination of the derivatives represents the *measurement* of asymmetry variation, our approach exploits the derivatives to explore the asymmetry and to track asymmetry-induced errors; the use of the derivatives ensures the ability to accurately describe the dynamically changing asymmetry due to the nature of measurement. Thus, the approach achieves accuracy of the order of tens or hundreds of nanoseconds in a software-based manner even under heavy traffic environments.

We also propose a novel state-space model that is completely consistent with PTP. The virtue of the state-space analysis is the integrated allowance of optimal estimations through stochastic models, such as the Kalman [14] or ARIMA [15]. Previous studies [16]–[21] have proposed state-space models for clock synchronization. However, the effect of asymmetry has not been fully considered or resolved in these approaches. For example, the Giorgi state-space model [17] simply assumes that the error from asymmetry constitutes a zero-mean Gaussian variable and no longer deals with the asymmetry. Unlike other studies, we clearly derive the state transition of the asymmetry and extend it to the impact of asymmetry. Furthermore, a complete state-space model of PTP is derived by considering its clock offset estimation as an observation. To the best of our knowledge, this is a pioneering work on the PTP that considers the state-space of the asymmetry.

The strength of our approach is threefold. First, this approach is a purely software-based method and does not assume any application-specific condition or require specialized hardware support; thus, it can be deployed on any system without significant effort. Second, our approach maintains the PTP architecture without requiring additional packet exchanges. In clock synchronization scenarios, it is important to reduce unexpected packet exchange overheads to ensure periodicity of the clock offset estimation. Therefore, our approach retains the PTP architecture to minimize any additional overhead while maintaining all the advantages of PTP and the ability to be combined with any PTP-based approach. Finally, the proposed approach is designed to be capable of handling the asymmetries of total packet delays. The packet delay is composed of nodal processing, queuing, transmission, and propagation delays. The packet delay considered in our model includes all these delays occurring between the timestamps on the master and slave nodes.

The remainder of this paper is organized as follows. Section II discusses the terminologies and elaborates on PTP. Section III describes the limitations of the PTP. Section IV presents the proposed approach, state-space model, and an optimal estimator. Section V provides the evaluation setups and results. Finally, Section VI concludes our paper.

II. BACKGROUND

A. Terminology

Clock offset refers to the time difference between different clocks. Suppose t denotes a referenced master clock time¹ and $C(t)$ denotes a local slave clock time. The clock offset $\theta(t)$ between these clocks at time t is represented as

$$\theta(t) = C(t) - t. \quad (1)$$

The goal of clock synchronization is to estimate and reduce such clock offsets. In general, a clock synchronization process is periodically repeated because the offset estimation is not error-free, and the clock offset varies with time.

Clock skew refers to a phenomenon in which clocks operate at different frequencies, even if their designs and settings are the same. Suppose that both clocks are set to a nominal frequency f_n . In practice, their frequencies are not the same as f_n , and vary over time. The dimensionless clock skew, $\gamma(t)$, between the frequencies is defined by

$$\gamma(t) = \frac{f_s(t) - f_m(t)}{f_m(t)} \quad (2)$$

where f_m and f_s denote the clock frequencies of the master and slave, respectively.

Clock drift is the accumulated discrepancies between the clocks, which originates from clock skew. The drift caused in a time duration from t_i to t_f , $\tau|_{t_i}^{t_f}$, is defined as

$$\tau|_{t_i}^{t_f} = \int_{t_i}^{t_f} \gamma(t) dt. \quad (3)$$

The relationship between the clock offset $\theta(t)$, and the clock drift $\tau|_0^t$ is given as

$$\theta(t) = \theta(0) + \int_0^t \gamma(t') dt' = \theta(0) + \tau|_0^t \quad (4)$$

where $\theta(0)$ denotes an initial offset.

One-way packet delay (OWPD) is the the total time required for a packet to be transmitted from a source and arrived at the destination; it includes all possible delays in the network, such as nodal processing, queuing, transmission, and propagation delays. Suppose D_s and D_d denote the OWPDs of packets from a source to a destination and vice versa, respectively. These OWPDs are given as

$$D_s = t_2 - t_1 \text{ and } D_d = t_4 - t_3 \quad (5)$$

where t_1 and t_3 are the transmitted times and t_2 and t_4 are the received times of the packets, as shown in Fig. 1.

Degree of asymmetry between two packet delays is the difference between the packet delays transmitted in opposite directions:

$$\lambda = D_d - D_s. \quad (6)$$

¹The referenced master clock is usually adopted in the literature for the clarity. In n -depths hierarchy with the grandmaster or multi-master architectures, the noise contribution to the clock is usually taken into account through policy or engineering.

Fig. 1. The basic IEEE 1588 PTP end-to-end clock synchronization process

Estimation error is the difference between an estimated value and a true value. For example, the error of an offset estimation is given as

$$\varepsilon_{\hat{\theta}}(t) = \hat{\theta}(t) - \theta(t) \quad (7)$$

where $\hat{\theta}(t)$ denotes an estimated value of $\theta(t)$.

B. IEEE 1588 PTP

PTP is a packet-exchange-based protocol for synchronizing local slave clocks with their referenced master clocks. We herein summarize the core aspects of the end-to-end synchronization of PTP, which can be applied without any hardware constraints.

The synchronization between the master and slave clocks is achieved as shown in Fig. 1. To remove the clock offset at t_1 , $\theta(t_1)$, the master and slave exchange timestamping packets, using which, the slave estimates its offset from the master clock and adjusts the value along the following steps:

- **Step 1:** At time t_1 , the master sends a *sync* message to the slave. After transmitting the *sync* message, the master measures timestamp t_1 and subsequently sends a *follow-up* message with a timestamp t_1 .
- **Step 2:** After receiving *sync* and *follow-up* messages, the slave measures the local timestamp $C(t_2)$ and calculates $C(t_2) - t_1$, which is equal to the sum of the clock offset and packet delay between the master and slave, $\theta(t_1) + D_s$.
- **Step 3:** The slave sends a *delay-req* message to the master, measures the local clock timestamp $C(t_3)$, and subsequently sends a *follow-up* message.
- **Step 4:** After receiving the *delay-req*, the master measures the arrival time t_4 and sends a *delay-resp* message with a timestamp t_4 .
- **Step 5:** Once the slave receives a *delay-resp*, it estimates $\hat{D}_s$ and $\hat{\theta}(t_1)$ using (8) and adjusts its clock by $\hat{\theta}(t_1)$. Note that the PTP estimates $\hat{D}_s$ using a simple averaging under the assumption that $D_s = D_d$.

$$\hat{\theta}(t_1) = (C(t_2) - t_1) - \hat{D}_s \quad (8a)$$

$$\begin{aligned} \hat{D}_s &= \frac{1}{2}(D_s + D_d) \\ &= \frac{1}{2}\{(t_4 - t_1) - (C(t_3) - C(t_2))\} \end{aligned} \quad (8b)$$

III. PROBLEM STATEMENT

One of the primary and traditional problems of PTP is that it uses a simple approach such as averaging to estimate $\hat{D}_s$ and $\hat{\theta}(t_1)$ under the assumption that the OWPDs are symmetric ($D_s = D_d$) as in (8b). In the real world, the assumption is often violated because the transit times of packets in the master-slave and reverse slave-master paths can be different ($D_s \neq D_d$ then $\lambda \neq 0$). Inside this disagreement between assumption and reality, a nonzero asymmetry, λ , occurs and causes critical estimation error $\varepsilon_{\hat{\theta}}$ in (8).

Fig. 2 shows the asymmetry, $\lambda_k (= D_{d,k} - D_{s,k})$, and offset estimation error, $\varepsilon_{\hat{\theta},k} (= \varepsilon_{\hat{\theta}}(t_{1,k}))$, of the IEEE 1588 PTP in a real environment (the base environment in Section V), where the rightmost subscript k denotes the repetitions of clock offset estimation. As shown, λ_k can extend up to several tens of milliseconds. Moreover, the asymmetry is nonlinear and changes dynamically over time. As presented, λ_k directly affects $\varepsilon_{\hat{\theta},k}$. The asymmetry-induced estimation error may extend up to tens of milliseconds, even though there is only one hop in the communication link and no network congestion. The asymmetry and the induced error are intractable. This is because packet delay has multiple components, and their asymmetries are caused by different factors. For example, the *sync* and *delay-req* messages can experience different nodal processing delays because of the inconsistent time consumption of processes including routing table lookup. Their queuing delays also can be different because they are affected by network conditions such as the congestion caused when messages are communicated. Transmission and propagation delays may not be symmetric owing to the different data rate or cable twist rate. In actual environments, various factors, including these, may occur in combination, and continue to change over time. As a result, the asymmetry and error become nonlinear and unpredictable variables with respect to the time.

This design issue, i.e., asymmetry-induced error, has not been fully addressed since PTP was first introduced in 2002. The reasons why this issue is challenging are as follows: i) it is difficult to mathematically analyze the asymmetry due to the nonlinearities and ii) the relationship between the asymmetry and error is unknown.

IV. PROPOSED APPROACH

To address the design issue of PTP definitively, we propose an enhanced clock offset estimation method and focus on three questions,

- 1) What model can accurately represent the variations of the nonlinear asymmetry?
- 2) How can we define the relation between the estimation error of (8a) and defined asymmetry?
- 3) How can precise clock offset estimation be achieved using the relation?

To explore answers to these questions, we define a discrete linear differential equation (LDE) with respect to k for the asymmetry and relation between asymmetry and offset estimation error to design a clock offset estimation robust to delay asymmetry (Section IV-A). Later, we provide a concrete example of the estimation by constructing a state-space model that is

Fig. 2. Real asymmetry λ_k (upper) and clock offset estimation error $\varepsilon_{\hat{\theta},k}$ of PTP (lower); Environmental setup: closed one hop network, only the PTP packets are exchanged between master and slave nodes, execute PTP clock offset estimation at every second, oscilloscope monitoring (Refer to Section V-B2 for more information on the oscilloscope monitoring)

completely consistent with PTP (Section IV-B). Finally, based on the presented state-space model, we obtain an optimal clock offset estimation using the Kalman filter (Section IV-C).

A. Asymmetry, Error, and Estimation

First, we derive an LDE for the asymmetry; Equation (6) implies that an LDE for the degree of asymmetry λ_k can be derived from the LDEs of the OWPDS of the *sync* and *delay-req* signals. Therefore, we start by building the LDEs for $D_{s,k}$ and $D_{d,k}$ as

$$\begin{aligned} D_{s,k} &= D_{s,k-1} + \Delta t_{2,k} - \Delta t_{1,k} \\ &= D_{s,k-1} + \Delta C(t_{2,k}) - \Delta t_{1,k} - \tau|_{t_{2,k-1}}^{t_{2,k}} \end{aligned} \quad (9a)$$

$$\begin{aligned} D_{d,k} &= D_{d,k-1} + \Delta t_{4,k} - \Delta t_{3,k} \\ &= D_{d,k-1} + \Delta t_{4,k} - \Delta C(t_{3,k}) + \tau|_{t_{3,k-1}}^{t_{3,k}} \end{aligned} \quad (9b)$$

where Δ denotes the numerical differential, e.g. $\Delta t_{i,k} = t_{i,k} - t_{i,k-1}$, $i \in \{1, 2, 3, 4\}$, and $\Delta t_{2,k}$ and $\Delta t_{3,k}$ are equal to $(\Delta C(t_{2,k}) - \tau|_{t_{2,k-1}}^{t_{2,k}})$ and $(\Delta C(t_{3,k}) - \tau|_{t_{3,k-1}}^{t_{3,k}})$, respectively. Note that reference timestamps $t_{2,k}$ and $t_{3,k}$ are not available in practice, irrespective of whether the clock skew exists, because $D_{s,k}$ and $D_{d,k}$ are unknown.

Theorem IV.1. Using (6) and the LDEs of (9), we can derive an LDE for λ_k as

$$\lambda_k = \lambda_{k-1} + \left(\tau|_{t_{2,k-1}}^{t_{2,k}} + \tau|_{t_{3,k-1}}^{t_{3,k}} \right) + [1 \quad 1] \alpha_k + [-1 \quad -1] \beta_k \quad (10)$$

where α_k and β_k are respectively given by $[\Delta t_{1,k} \quad \Delta t_{4,k}]^T$ and $[\Delta C(t_{2,k}) \quad \Delta C(t_{3,k})]^T$.

Corollary IV.1. In an ideal case without clock skew, the LDE of λ_k can be directly determined using

$$\lambda_k = \lambda_{k-1} + [1 \quad 1] \alpha_k + [-1 \quad -1] \beta_k \quad (11)$$

with only the given timestamps. From (11), in cases with negligible clock skew, we can approximate the LDE for λ_k as

$$\lambda_k \approx \lambda_{k-1} + [1 \quad 1] \alpha_k + [-1 \quad -1] \beta_k. \quad (12)$$

For example, if we use atomic clocks, (12) can be adopted.

In cases with nonnegligible clock skews, for example, if we use oven-controlled or temperature-compensated crystal oscillators, we cannot directly process (10) because the drift terms are not approximated to zero and reference timestamps $t_{2,k}$ and $t_{3,k}$ are not available. Therefore, for these cases, we introduce another variable, ζ_k , to describe the impact of λ_k as

$$\begin{aligned} \zeta_k &= \lambda_k - \left(2\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{2,k}}^{t_{3,k}} \right) \\ &= \lambda_k - \left(\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{2,k}}^{t_{3,k}} \right) \\ &= \lambda_k - \left(\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{1,k}}^{t_{4,k}} - \tau|_{t_{3,k}}^{t_{4,k}} \right) \\ &= \lambda_k - \left\{ \left(\tau|_{t_{1,k}}^{t_{2,k}} - \tau|_{t_{3,k}}^{t_{4,k}} \right) + \tau|_{t_{1,k}}^{t_{4,k}} \right\}. \end{aligned} \quad (13)$$

Here, $\left(\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{2,k}}^{t_{3,k}} \right) = \left(\tau|_{t_{1,k}}^{t_{4,k}} - \tau|_{t_{3,k}}^{t_{4,k}} \right)$ because $\tau|_{t_{1,k}}^{t_{4,k}} = \left(\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{2,k}}^{t_{3,k}} + \tau|_{t_{3,k}}^{t_{4,k}} \right)$. Equation (13) implies that ζ_k includes a difference between the clock drifts $\tau|_{t_{1,k}}^{t_{2,k}}$ and $\tau|_{t_{3,k}}^{t_{4,k}}$, which respectively occur in the time durations of $D_{s,k}$ and $D_{d,k}$. The manner in which these terms include the impacts is determined by the design of the clock skew, as shown in an example later.

Here, we derive the LDE for ζ_k .

Corollary IV.2. By combining (10) and (13), an LDE for ζ_k can be represented as

$$\zeta_k = \zeta_{k-1} + 2\tau|_{t_{1,k-1}}^{t_{1,k}} + [1 \quad 1] \alpha_k + [-1 \quad -1] \beta_k \quad (14)$$

Proof. Equation (14) is equivalent to (10) because the drift terms in (10) are represented by $\tau|_{t_{2,k-1}}^{t_{2,k}} = \left(\tau|_{t_{1,k-1}}^{t_{1,k}} - \tau|_{t_{1,k-1}}^{t_{2,k-1}} \right) + \tau|_{t_{1,k}}^{t_{2,k}}$ and $\tau|_{t_{3,k-1}}^{t_{3,k}} = \left(\tau|_{t_{1,k-1}}^{t_{1,k}} - \tau|_{t_{1,k-1}}^{t_{3,k-1}} \right) + \tau|_{t_{1,k}}^{t_{3,k}}$. In addition, $\left(\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{1,k}}^{t_{3,k}} \right)$ is equal to $\left(2\tau|_{t_{1,k}}^{t_{2,k}} + \tau|_{t_{2,k}}^{t_{3,k}} \right)$.

Second, we derive the estimation error of (8a) related to the asymmetry.

Theorem IV.2. The estimation error $\varepsilon_{\hat{\theta},k}$ caused by the asymmetry can be derived as

$$\varepsilon_{\hat{\theta},k} = -\frac{1}{2}\zeta_k. \quad (15)$$

Proof. The derivation uses the relation between $D_{s,k}$ and $\hat{D}_{s,k}$ in (8b). If we replace $C(t_{2,k})$ and $C(t_{3,k})$ in (8b) with $C(t_{2,k}) = (t_{1,k} + \theta_k) + D_{s,k} + \tau|_{t_{1,k}}^{t_{2,k}}$ and $C(t_{3,k}) = (t_{1,k} + \theta_k) + \{(t_{4,k} - t_{1,k}) - D_{d,k}\} + \tau|_{t_{1,k}}^{t_{3,k}}$, we obtain the relation

$$\hat{D}_{s,k} = D_{s,k} + \frac{1}{2}\zeta_k + \tau|_{t_{1,k}}^{t_{2,k}}.$$

Based on (8a), $\hat{\theta}_k$ is equal to $\theta_k + D_{s,k} - \hat{D}_{s,k} + \tau|_{t_{1,k}}^{t_{2,k}}$. Therefore, $\varepsilon_{\hat{\theta},k} (= \hat{\theta}_k - \theta_k)$ is given by (15).

Finally, we design a clock offset estimation robust to the asymmetry for the PTP based on the above. The new clock offset estimation, which takes the error of (15) and LDE of (10) into account, is designed as (16).

$$\hat{\theta}_{new,k} = \hat{\theta}_k - \varepsilon_{\hat{\theta},k} \quad (16)$$

The intuitive estimation (16), however, cannot be directly used in real implementation because $\varepsilon_{\hat{\theta},k}$ includes drift terms

that should be estimated through a stochastic model. Therefore, we propose to expand (16) to (17)

$$\hat{\theta}_{new,k} = f(\hat{\theta}_k, \varepsilon_{\hat{\theta},k}) \quad (17)$$

where $f(\bullet)$ denotes an estimator. The use of a general function $f(\bullet)$ implies that our LDEs and $\varepsilon_{\hat{\theta},k}$ can be incorporated into various stochastic models. Next, to provide a concrete and practical example, we construct a state-space model.

B. State-Space Model of PTP

The presented model suggests a method to address the state-space related to asymmetry and is completely consistent with PTP. Consequently, the proposed states-space model establishes the basis for optimal clock offset estimations on asymmetric communication links via stochastic models, as shown in the next section.

In the design of the observation model, our approach considers the estimate of (8a) as an observation of PTP and uses (15) to describe the observation error of (8a). In the design of the state and error transitions, the model adopts a predefined clock skew model [22] and the LDE of (14). For the transition noise, the model obtains an analyticity from the properties of an unbiased clock skew estimator analyzed in [22].

To avoid repeated explanations, we consider environments with nonnegligible clock skews. Therefore, we use ζ_k rather than λ_k without loss of generality.

Eventually, we can have a state-space model for the proposed approach as follows:

$$\mathbf{y}_k = \mathbf{x}_k + \begin{bmatrix} -\frac{1}{2}\zeta_k & 0 \end{bmatrix}^T + \mathbf{v}_k \quad (18a)$$

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k \quad (18b)$$

$$\boldsymbol{\epsilon}_k = \mathbb{E}[\mathbf{y}_k] - \bar{\mathbf{y}}_k \quad (18c)$$

where the transition of ζ_k is derived by

$$\zeta_k = \zeta_{k-1} + 2[\gamma_{k-1} - u_{\gamma,k}]\Delta t_{1,k} + u_{\zeta,k} + \varepsilon_{\hat{\zeta},k} \quad (18d)$$

from (14) for $u_{\zeta,k} = \begin{bmatrix} 1 & 1 \end{bmatrix} \boldsymbol{\alpha}_k + \begin{bmatrix} -1 & -1 \end{bmatrix} \boldsymbol{\beta}_k$, $\boldsymbol{\epsilon}_k$ is the model error, and the bar operator $\bar{\bullet}$ represents the real-world observation of $\bullet$. In this state-space model, $\mathbf{y}_k$ represents the observational model of $[\theta_k \ \gamma_k]^T$. The real-world observations and observation errors, $\bar{\mathbf{y}}_k$ and $\mathbf{v}_k$, are defined as

$$\bar{\mathbf{y}}_k = [\bar{\theta}_k \ \bar{\gamma}_k]^T, \text{ and} \quad (19a)$$

$$\mathbf{v}_k = \begin{bmatrix} -\frac{1}{2}\varepsilon_{\Delta\hat{\zeta},k} & 0 \end{bmatrix}^T + \begin{bmatrix} \varepsilon_{\bar{\theta},k} & \varepsilon_{\bar{\gamma},k} \end{bmatrix}^T \quad (19b)$$

where $\varepsilon_{\bar{\theta},k}$ describes the remaining observation errors except for $-\frac{1}{2}\varepsilon_{\Delta\hat{\zeta},k}$. Note that $\varepsilon_{\Delta\hat{\zeta},k}$ can be replaced with $\varepsilon_{\Delta\bar{\zeta},k}$ when ζ_k is observable. The hidden states, $\mathbf{x}_k$, state inputs, $\mathbf{u}_k$, and transition noise, $\mathbf{w}_k$ are defined as

$$\mathbf{x}_k = [\theta_k \ \gamma_k]^T, \quad (20a)$$

$$\mathbf{u}_k = [\hat{\theta}_{k-1} \ \hat{\gamma}_{k-1}]^T, \text{ and} \quad (20b)$$

$$\mathbf{w}_k = \begin{bmatrix} \varepsilon_{\hat{\theta},k} & \varepsilon_{\hat{\gamma},k} \end{bmatrix}^T. \quad (20c)$$

Note that concrete models of the state transition matrices, $\mathbf{A}_k$ and $\mathbf{B}_k$, and the transition noises $\varepsilon_{\hat{\theta},k}$, $\varepsilon_{\hat{\gamma},k}$, and $\varepsilon_{\hat{\zeta},k}$ are determined by clock or clock skew models. For example, if one of the models in [22], [23], [24] is adopted, the transition

noises are postulated to follow the normal distribution. In this paper, because we use the models in [22], the matrices and noises are derived by

$$\mathbf{A}_k = \begin{bmatrix} 1 & \Delta t_{1,k} \\ 0 & 1 \end{bmatrix} \text{ and } \mathbf{B}_k = \begin{bmatrix} -1 & -\Delta t_{1,k} \\ 0 & -1 \end{bmatrix}, \quad (21)$$

and

$$\varepsilon_{\hat{\theta},k} \sim N\left(0, \frac{1}{3}(\sigma_\eta)^2(\Delta t_{1,k})^3\right), \quad (22a)$$

$$\varepsilon_{\hat{\gamma},k} \sim N\left(0, (\sigma_\eta)^2 \Delta t_{1,k}\right), \text{ and} \quad (22b)$$

$$\varepsilon_{\hat{\zeta},k} \sim N\left(0, \frac{4}{3}(\sigma_\eta)^2(\Delta t_{1,k})^3\right), \quad (22c)$$

respectively, under the assumptions that $\int \eta(t)dt$ is a standard Wiener process [25] and the density function $\eta(t)$ follows the independently and identically distributed (i.i.d.) Gaussian of $N(0, (\sigma_\eta)^2)$ where $(\sigma_\eta)^2$ is the step variance of random walk skew; for the detailed proof, see Appendix A.1-(i), (ii), and (iii) in Zhong et al. [22].

C. Optimal Clock Offset Estimation

We provide the optimal clock offset estimation robust to delay asymmetry based on our state-space model. The optimal estimation uses the linear Kalman filter (LKF) [14] for state estimation and a Bayesian algorithm for hyperparameter inference.

In order to use the LKF, adopting the symbols $\{\mathbf{Q}, \mathbf{R}, \mathbf{S}\}$ commonly used in state-space models, we define the relationships of the noises $\mathbf{w}_k$ and $\mathbf{v}_k$ as

$$\text{Cov}(\mathbf{w}_k) = \mathbf{Q}_k - \mathbf{S}_{k-1} \mathbf{R}_{k-1}^{-1} \mathbf{S}_{k-1}^T \quad (23a)$$

$$\mathbf{Q}_k = \text{Cov}(\begin{bmatrix} \varepsilon_{\hat{\theta},k} & \varepsilon_{\hat{\gamma},k} \end{bmatrix}^T) \quad (23b)$$

$$\mathbf{R}_k = \text{Cov}(\mathbf{v}_k) \quad (23c)$$

$$\mathbf{S}_k = \text{Cov}(\begin{bmatrix} \varepsilon_{\hat{\theta},k} & \varepsilon_{\hat{\gamma},k} \end{bmatrix}^T, \mathbf{v}_k) \quad (23d)$$

where $\text{Cov}(\bullet)$ denotes the covariance function of $\bullet$, which ensure the independence between the noise terms $\mathbf{w}_k$ and $\mathbf{v}_k$ [26]. The independence condition is a fundamental requirement for optimal estimation based on the state-space model, such as the Kalman-filter-based estimation. To guarantee the required condition, (23) handles the effects of $\mathbf{S}_k$ in the state-space model by introducing the terms $\mathbf{S}_{k-1} \mathbf{R}_{k-1}^{-1} \boldsymbol{\epsilon}_{k-1}$, $\mathbf{S}_{k-1} \mathbf{R}_{k-1}^{-1} \mathbf{v}_{k-1}$, and $\mathbf{S}_{k-1} \mathbf{R}_{k-1}^{-1} \mathbf{S}_{k-1}^T$ to (18b), (20c), and (23a), respectively: In our model, $\mathbf{S}_k \neq \mathbf{0}$ owing to the term ζ_k in (18a). Moreover, the uncertainty of ζ_k introduced at k -th step for the given ζ_{k-1} stems from $\Delta\zeta_k$ where $\varepsilon_{\Delta\hat{\zeta},k}$ includes the uncertainties of γ_{k-1} and $\varepsilon_{\hat{\zeta},k}$ related to clock skew, as seen in (18d). Hence, the total offset observation error, which is the first component of $\mathbf{v}_k$, is correlated with $\varepsilon_{\hat{\theta},k}$ and $\varepsilon_{\hat{\gamma},k}$ because the transition errors are also related to clock skew. (23) removes the effects of the correlation.

Algorithm 1 describes the optimal estimation, where the set of hyperparameters denoted by $\boldsymbol{\Theta}$ comprises σ_η , $\sigma_{\varepsilon_{\bar{\theta},k}}$,

Algorithm 1: Optimal Clock Offset Estimation

Input : $p([\theta_{k-1}, \gamma_{k-1}, \zeta_{k-1}]^T | \mathbf{y}_{1:k-1}, \Theta)$,
 $\hat{\theta}_k, \hat{\gamma}_k, u_{\theta,k}, u_{\gamma,k}, \epsilon_{k-1}, \mathbf{R}_{k-1}, \mathbf{S}_{k-1}$.
Output: $p([\theta_k, \gamma_k, \zeta_k]^T | \mathbf{y}_k, \Theta), \epsilon_k, \mathbf{R}_k, \mathbf{S}_k$.
Function `Optimal_Estimation (Input) :`

```

 $\Delta \hat{\zeta}_{k|k-1} \leftarrow 2[\hat{\gamma}_{k-1} - u_{\gamma,k}] \Delta t_{1,k} + u_{\zeta,k}$ ;
 $\epsilon_{\Delta \hat{\zeta},k|k-1} \leftarrow 2\epsilon_{\hat{\gamma},k|k-1} + \epsilon_{\hat{\zeta},k}$ ;
 $\hat{\zeta}_k \leftarrow \hat{\zeta}_{k-1} + \Delta \hat{\zeta}_{k|k-1}$ ;
 $\mathbf{x}_{k-1} \leftarrow [\hat{\theta}_{k-1}, \hat{\gamma}_{k-1}]^T$ ;
 $\mathbf{u}_k \leftarrow [u_{\theta,k}, u_{\gamma,k}]^T$ ;
 $\mathbf{w}_k \leftarrow \begin{bmatrix} \epsilon_{\hat{\theta}_k} & \epsilon_{\hat{\gamma}_k} \end{bmatrix} - \mathbf{S}_{k-1} \mathbf{R}_{k-1}^{-1} \mathbf{v}_{k-1}$ ;
 $\mathbf{y}_k \leftarrow [\hat{\theta}_k \quad \hat{\gamma}_k]^T$ ;
 $\mathbf{v}_k \leftarrow \begin{bmatrix} -\frac{1}{2}\epsilon_{\Delta \hat{\zeta},k|k-1} & 0 \end{bmatrix}^T + \begin{bmatrix} \epsilon_{\hat{\theta}_k} & \epsilon_{\hat{\gamma}_k} \end{bmatrix}^T$ ;
/* Run two LKFs */
estimate  $p(\mathbf{x}_k | \mathbf{y}_{1:k}, \Theta)$  using (24);
estimate  $p(\Delta \zeta_k | \mathbf{y}_{1:k}, \Theta)$  using (24);
/* Determine the state density */
 $p(\zeta_k | \mathbf{y}_{1:k}, \Theta) \leftarrow \int p(\zeta_k, \Delta \zeta_k | \mathbf{y}_{1:k}, \Theta) d\Delta \zeta_k$ ;
calculate  $p([\theta_k, \gamma_k, \zeta_k]^T | \mathbf{y}_{1:k}, \Theta)$ ;
return
```

and $\sigma_{\epsilon_{\gamma,k}}$. The estimation guarantees convergence of the state error (or state covariance) in the LKF. The Riccati difference equation (RDE) [27] sets the *detectability* criterion on the convergence [26]; to satisfy the detectability requirement, we use two LKFs for $[\theta_k, \gamma_k]^T$ and $\Delta \zeta_k$, and determine the density of states, $p([\theta_k, \gamma_k, \zeta_k]^T)$.

We provide a reference to the LKF process and its complexity analysis. Extensive literary works are available for the LKF since it has been successfully used in many fields, including networks, measurements, and robotics. The LKF process and analysis are summarized in [28].

In brief, in the LKF, the posterior for given historical observations, $p(\mathbf{x}_k | \mathbf{y}_{1:k}, \Theta)$, is given as

$$p(\mathbf{x}_k | \mathbf{y}_{1:k}, \Theta) \propto p(\mathbf{y}_k | \mathbf{x}_k, \Theta) p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \Theta). \quad (24)$$

The LKF produces optimal estimates adaptively to minimize the mean-squared error (MSE) under the Gauss–Markov assumption. The LKF is lightweight in terms of computation and memory. The complexity of the LKF is dominated by one matrix inversion and it does not require storing estimates or data records owing to the Markovian property.

Our hyperparameter inference is inspired by the integrated nested Laplace approximation (INLA) [29] and a sequential Bayesian inference [30]. The inference uses an approximation of the marginal posterior of Θ , $\tilde{p}(\Theta | \mathbf{y}_{1:k}) \propto p(\mathbf{x}_{1:k}, \mathbf{y}_{1:k}, \Theta) / \tilde{p}(\mathbf{x}_{1:k} | \mathbf{y}_{1:k}, \Theta)$, and a grid search.

We first use the posterior of the LKF in (24) to construct the denominator $\tilde{p}(\mathbf{x}_{1:k} | \mathbf{y}_{1:k}, \Theta)$. Then, we use the Newton–Raphson method [31] to determine the mode $\Theta^* = \arg\max_{\Theta} \tilde{p}(\Theta | \mathbf{y}_{1:k})$. Moreover, to avoid the local optima problem, we search a grid of Θ around the mode via $\Theta(\mathbf{z}) = \Theta^* + U\Lambda^{\frac{1}{2}}\mathbf{z}$, where $U\Lambda U^T$ is the eigen-decomposition for the inverse of Hessian and $\mathbf{z}$ is a standardized Gaussian

variable. We test whether the mode is a local optimum by evaluating $\tilde{p}(\Theta(\mathbf{z}) | \mathbf{y}_{1:k})$.

Note that the grid search is performed only once in the startup phase for the evaluation noted in Section V because it involves large computational costs. In the evaluation, we focus on a lightweight design to ensure that the approach is practical. Therefore, after the startup phase, the inference is performed using the approximation based on the LKF and Newton–Raphson method for recent N observations, where N is empirically derived value. The use of the Newton–Raphson method is allowed because the LKF has a closed form. The Hessian matrix is computed from the model error $\epsilon_{1:k}$ defined in (18c) and numerator of (24). If the grid search is executed in each step in the synchronization phase, performance will be improved.

V. EVALUATION

This section presents the evaluation of the proposed approach for clock offset estimation. The main purpose of the evaluation is to determine whether the approach addresses the design issue of PTP and clock offset estimation error $\epsilon_{\hat{\theta},k}$ caused by the asymmetry λ_k of the OWPDs, $D_{s,k}$ and $D_{d,k}$. Because our main approach proposed in Section IV-A has a general form, the evaluations are conducted with the novel optimal estimation of Algorithm 1 presented above.

We compare the performances of our approach with those of PTP, previous approaches [8], [10], [17], and PTPv2. Comparisons with PTP and previous approaches show how well our approach addresses the asymmetry-induced error. Comparison with PTPv2 shows the potential of our approach as an alternative to TC, which is currently the most powerful means to improve PTP. TC has the disadvantage of requiring a compatible network switch and is hence expensive. Thus, if our algorithmic method can replace PTPv2 that requires TC-enabled hardware, it can offer considerable benefits to the industrial field.

The comparisons are conducted in eleven different network environments, including baseline environment as a stable closed network. The other ten environments are configured with different network loads, traffic patterns, and packet size distributions.

A. Evaluation Suite

We build an evaluation suite, as illustrated in Fig. 3, consisting of two development boards (Freescale P2020RDB-PCA [32]), one each for the master and slave nodes, a PTPv2-enabled switch (CISCO IE-3000-4TC-E [33]) implementing TC, an oscilloscope (Tektronix MSO 5034b [34]) for measurement, and a desktop computer (PC) to control the network load.

The master and slave nodes are connected via a wired network to form a closed local area network (LAN). For clock synchronization, the master and slave nodes exchange PTP packets once every second. The PTPv2-enabled switch records the residence times of *sync* and *delay-req* packets in the correction field of the respective *follow-up* packets by the default end-to-end TC configuration [3]. The oscilloscope

Fig. 3. Evaluation suite. (a) Picture, left: CISCO IE-3000-4TC-E, mid: Tektronix MSO 5034b, right: Freescall P2020RDB-PCAs. (b) Block diagram.

monitors clock offset $\theta(t)$ and clock skew $\gamma(t)$, and reports $\bar{\gamma}_k$ to the slave node. For the measurements, the probes of the oscilloscope are connected to the master and slave to capture the periodic pulses generated from the clocks. The PC sends network traffics to control noise on the packet delays between master and slave nodes.

B. Evaluation Setup

1) *Timestamping*: We use MAC and PHY hardware timestamps supported by the P2020RDB-PCA [35], [36]. For the timestamps, the GNU/Linux kernel v3.12.37 of QorIQ SDK v1.9 [37] is installed. The device driver *gianfar* enables the timestampings by default, which refer to the value of a 64-bit timer counter composed of two 32-bit registers, ETSEC_1588_TMR_CNT_H/L. The timer counter is set to operate at 200 MHz and increments by five (as set by TCLK_PERIOD) to ensure granularity on nanoseconds level.

2) *Monitoring*: We capture the periodic pulses generated from the counters of the master and slave to monitor clock offset and clock skew. The P2020RDB-PCA contains two fixed-period event registers (ETSEC_1588_TMR_FIPERa, $a=\{1,2\}$) for two separate periodic pulse generations. For clock offset monitoring, we set the ETSEC_1588_TMR_FIPER1s of both the master and slave to 10^9 to generate periodic pulses per second (PPS). For clock skew monitoring, we set the ETSEC_1588_TMR_FIPER2s of both the master and slave to 2.5×10^4 to generate a periodic pulse per twenty-five microseconds. To capture the pulses, the probes are connected to the output pins of the J6 on the P2020RDB-PCAs. The TekScope preinstalled on MSO 5034b is set to measure the delay between the master and slave pulses as well as the momentary frequencies of the pulses. The monitoring requires a minimum sampling rate, 1 GS/sec (gigasamples per second), for nanosecond precision, and the MSO 5034b is capable of sampling at most 20 GS/sec.

3) *Reporting*: Our approach uses two observations, $\bar{\theta}_k$ and $\bar{\gamma}_k$. As described in Section IV, $\bar{\theta}_k$ is reported by PTP. For

(a) PDFs of the asymmetry in SV settings

(b) PDFs of the asymmetry in ATM1 settings

Fig. 4. Distribution of the absolute asymmetry $|\lambda_k|$ in various network environment setups

$\bar{\gamma}_k^2$, we use the instrument control toolbox of MATLAB on the MSO 5034b and the secure shell protocol (SSH) to automatically report the measurements to the slave. The MSO 5034b is set to report $\bar{\gamma}_k$ at least three times per second with a sampling rate of 5 GS/sec. The slave is programmed to use the most recent observations; the observation errors due to the query time and reporting latency are dealt with by $\varepsilon_{\bar{\gamma},k}$.

4) *Network Load*: In addition to the base environment with only PTP packets, we setup ten network environments to evaluate how our approach responds to different load conditions. We used Ostinato v1.0 [38] to generate and transmit packet streams to the network. The load configurations are composed of two categories: Symmetric load with Variable packet size (SV) and Asymmetric load with realistic Traffic Model (ATM). In the case of SV, two different streams of the same load rate r , $(r_{m2s}, r_{s2m}) \in \{(25, 25), (50, 50), (75, 75), (100, 100)\}\%$, and uniformly random packet size l , $64 \leq l \leq 1518$ bytes, are introduced in the forward (master-to-slave, $m2s$) and reverse (slave-to-master, $s2m$) directions. In the case of ATM, different load rates, $(r_{m2s}, r_{s2m}) \in \{(40, 20), (60, 20), (80, 20)\}\%$, are adopted for the forward and reverse directions. In this case, three packet sizes, $l \in \{64, 576, 1518\}$ bytes, are used and their portions, $(p_{64}, p_{576}, p_{1518})$, are set following TM1 and

²The use of instruments to observe $\bar{\gamma}_k$ enables a reliable evaluation on the clock offset estimation by minimizing uncertainties from clock skew estimations. Stochastic skew estimators such as [24] can also be adopted.

TABLE I
THE STATISTICAL RESULT OF CLOCK OFFSET ESTIMATION ERRORS FOR ONE HOUR MEASURED WITH FIFTEEN REPETITIONS.
(AVERAGE ($E[E[\bullet]]$), VARIANCE ($E[Var[\bullet]]$), MAXIMUM ($E[Max[\bullet]]$) OF $|\varepsilon_{\hat{\theta},k}|$ IN SECONDS.)

Metric	Method	0% (Baseline)	SV-100	ATM1-8020	ATM2-8020
Average $E[E[\bullet]]$	PTP [1]	2.59×10^{-4}	3.05×10^{-4}	2.94×10^{-4}	2.90×10^{-4}
	Lee's Approach [8]	2.80×10^{-4}	3.25×10^{-4}	3.19×10^{-4}	3.13×10^{-4}
	Linux PTP, MM+PI [10]	1.91×10^{-4}	2.17×10^{-4}	1.93×10^{-4}	1.94×10^{-4}
	Giorgi's LKF [17]	7.29×10^{-4}	8.48×10^{-4}	8.21×10^{-4}	8.17×10^{-4}
	PTPv2 [3]	4.76×10^{-8}	8.21×10^{-7}	6.02×10^{-8}	6.35×10^{-8}
	Proposed Approach	3.76×10^{-8}	1.75×10^{-7}	1.26×10^{-7}	7.64×10^{-8}
Variance $E[Var[\bullet]]$	PTP [1]	1.35×10^{-6}	1.43×10^{-6}	1.45×10^{-6}	1.38×10^{-6}
	Lee's Approach [8]	1.35×10^{-6}	1.43×10^{-6}	1.47×10^{-6}	1.37×10^{-6}
	Linux PTP, MM+PI [10]	4.69×10^{-10}	2.14×10^{-9}	1.37×10^{-9}	5.44×10^{-10}
	Giorgi's LKF [17]	6.48×10^{-6}	6.87×10^{-6}	6.98×10^{-6}	6.74×10^{-6}
	PTPv2 [3]	1.26×10^{-15}	4.77×10^{-13}	1.85×10^{-15}	1.89×10^{-15}
	Proposed Approach	1.71×10^{-31}	4.91×10^{-28}	1.19×10^{-29}	1.77×10^{-30}
Maximum $E[Max[\bullet]]$	PTP [1]	1.53×10^{-2}	3.05×10^{-2}	1.58×10^{-2}	1.52×10^{-2}
	Lee's Approach [8]	1.53×10^{-2}	1.54×10^{-2}	1.58×10^{-2}	1.52×10^{-2}
	Linux PTP, MM+PI [10]	6.21×10^{-4}	1.10×10^{-3}	8.12×10^{-4}	9.01×10^{-4}
	Giorgi's LKF [17]	3.19×10^{-2}	3.07×10^{-2}	3.37×10^{-2}	3.13×10^{-2}
	PTPv2 [3]	2.02×10^{-7}	2.47×10^{-6}	2.44×10^{-7}	2.35×10^{-7}
	Proposed Approach	5.29×10^{-8}	2.78×10^{-7}	2.49×10^{-7}	1.27×10^{-7}

TM2 of the ITU-T documents [39], [40]; ATM1: (80, 5, 15)%, and ATM2: (30, 10, 60)%. Moreover, when sending the maximum sized packets, $l = 1518$ bytes, the transmission is set to occur in burst lasting for a uniformly and randomly chosen period between 0.1 sec and 3 sec.

The profiled probability density functions (PDFs) of the absolute asymmetry under an ideal network environment and the ten different network load configurations are plotted in Fig. 4. Note that in an ideal environment, which is marked as Baseline in Fig. 4, only the PTP packets are exchanged and thereby the asymmetry is minimized. Fig. 4(a) shows the distribution of the asymmetry in the SV settings. As shown, the distribution of asymmetry value is widened as the network load increases. Under a heavily loaded network, the asymmetry values do not fall within a certain level, thus making it difficult to effectively eliminate it. Fig. 4(b) shows the asymmetry distribution in the ATM1 settings. The distribution of asymmetry values is also wider compared with the ideal environment. Contrary to our expectations, the asymmetry distribution is much narrower than that of SV settings. This implies that asymmetry is more affected by the network traffic profile (e.g., packet size) at the moment the PTP packets are exchanged, rather than the long-term network load. The results were similar in the ATM2 setting, then the ATM2 result is omitted for the sake of clarity.

5) *Parameter Configurations*: In our approach, in the startup phase of the clock synchronization, the initial values of the hyperparameters σ_η , $\sigma_{\varepsilon_{\hat{\theta},k}}$, and $\sigma_{\varepsilon_{\hat{\gamma},k}}$ are inferred by the method described in Section IV-C. In our experiment, the inference is performed for 30 min. During the clock synchronization phase, the hyperparameters are periodically updated every 5 min based on the previous 5 min of data with approximation of the marginal posterior. In the startup phase, the initial states are configured using PPS-based offline

measurements. The manual configuration of the initial states is thus feasible because PTP is mostly implemented in LANs.

C. Evaluation Results

1) *Representative results*: We first provide the results evaluated in the baseline network environment and the harshest network environments, SV-100, ATM1-8020, and ATM2-8020. Table I lists the statistical results of the one-hour clock offset estimation (3600 repetitions of the clock synchronization) of PTP [1], previous approaches [8], [10], [17], PTPv2 [3], and our approach. We measured the absolute offset estimation error $|\varepsilon_{\hat{\theta},k}|$. To ensure reliability of our results, one-hour clock offset estimation experiment was repeated fifteen times, and we calculated the overall means of the average, variance, and maximum values.

Our clock offset estimation approach improved the accuracy of PTP by more than a thousand times in all the network environments. The level of accuracy was of the order of tens and hundreds of nanoseconds (3.76×10^{-8} , 1.75×10^{-7} , 1.26×10^{-7} , 7.64×10^{-8}) in the (Baseline, SV-100, ATM1-8020, ATM2-8020) network environments, while the PTP estimation error showed errors of the order of hundreds of microseconds (2.59×10^{-4} , 3.05×10^{-4} , 2.94×10^{-4} , 2.90×10^{-4}). The results indicate that our approach significantly improved the accuracy of PTP while maintaining the PTP architecture, without any additional packet exchanges.

Our approach also outperformed Lee's approach [8], Linux PTP [10], and Giorgi's LKF [17]. For the latest Linux PTP v3.1, we set the default configuration to use the built-in moving median (MM) filter and proportional & integral (PI) control servo. The Lee's approach and Giorgi's LKF were implemented without any modifications. All the previous approaches without supports from the PTPv2-enabled switch

Fig. 5. Absolute estimation error $|\varepsilon_{\hat{\theta},k}|$ in eleven different network environments

did not provide accuracy at a nanoseconds level, unlike our approach. Because the Linux PTP implements techniques to react against practical conditions along with the filter and servo, it showed better performance than the other naively implemented previous approaches. The results of Lee's approach and Giorgi's LKF have a room for the improvement through professional engineering. Especially, the Giorgi's LKF produced large errors because its observational parameters $\varepsilon_{\hat{\theta},k}$ and $\varepsilon_{\hat{\gamma},k}$ were designed under the assumption that the PTP error is zero mean Gaussian. However, as shown in the Table I, the PTP error was not zero mean and distributed up to 15.3 milliseconds. Due to this huge discrepancy, the Giorgi's LKF was not suitable for our environments. We discuss this more in Section V-C4.

Moreover, the performance of our approach was comparable to that of PTPv2. Our approach and PTPv2 both showed tens and hundreds of nanoseconds level accuracy. The accuracy of our approach was similar (Baseline, ATM1-8020, ATM2-8020) with or slightly better (SV-100) than that of PTPv2. In terms of variance and maximum values, our approach had more robust performance than PTPv2. Overall, our approach had the capability to achieve a competitive advantage over PTPv2 without incurring additional costs from specialized hardware. The reasons why this was possible are stated in Section V-C3

2) *Results with various network environments:* Next, we evaluated the estimation errors in eleven different network environments. Based on the order of error, we divided the evaluated methodologies into two groups; {PTPv2, Proposed approach} with nanoseconds (under 1×10^{-6} sec) and {PTP, Lee's Approach, Linux PTP} with microseconds order (under

1×10^{-6} sec). Results of two groups are shown in Figs. 5(a) and 5(b). The Giorgi's LKF is excluded here because of the discrepancy. It is covered later in Section V-C4.

In Fig. 5(a), three interesting aspects are observed. First of all, in all the cases of SV, proposed approach provides better and stable performance (less estimation error) than PTPv2. Meanwhile, the PTPv2 error was relatively large and it rapidly increases with network load. In the SV cases, as the network load increases, asymmetry of total packet delay other than the residence time also increases. While PTPv2 only mitigates the asymmetry of residence time, our approach deals with the asymmetry of *total* packet delays in a stochastic manner, thus mitigate the disturbance and outperformed PTPv2.

Second, our approach offered comparable performance to PTPv2 in the ATM cases. The traffic models of ATM cases have realistic properties, as stated in [39], [40]. In such realistic environments, our approach achieved tens of nanoseconds of accuracy in a software-based manner. In the ATM cases, PTPv2 slightly outperformed our approach, unlike in Baseline and SV cases. We analyzed that it comes from the noise introduced to the observational parameters when the burst traffic occurs.

Last, variations in packet sizes may cause harsh conditions for the precise clock synchronization. In the ATM cases, as the network load rate increases, the error of PTPv2 slightly increased. (5.08×10^{-8} , 5.34×10^{-8} , 5.52×10^{-8}) and (4.79×10^{-8} , 5.34×10^{-8} , 6.35×10^{-8}) in (4020, 6020, 8020) of ATM1 and ATM2, respectively. On the other hand, in the SV cases, the error was rapidly increased and amplified up to 8.21×10^{-7} . For reconfirmation, we conducted additional experiments. We simply fixed the packet size to 512 bytes and configured the rest of the environments are the same as

(a) Distribution of $|\varepsilon_{\theta,k}|$, proposed approach, Baseline (b) Distribution of $|\varepsilon_{\theta,k}|$, PTPv2, Baseline

Fig. 6. Absolute estimation error $|\varepsilon_{\theta,k}|$ of the proposed approach and PTPv2, in Baseline. On each box, the central mark indicates the median, and the bottom and top edges indicates the 25th and 75th percentiles, respectively. Outliers are plotted individually using the '+' symbol.

the SV. In this additional experiments, the PTPv2 errors were $(4.92 \times 10^{-8}, 4.58 \times 10^{-8}, 4.53 \times 10^{-8}, 6.10 \times 10^{-8})$. Remind that the PTPv2 errors were $(1.45 \times 10^{-7}, 3.49 \times 10^{-7}, 5.32 \times 10^{-7}, 8.21 \times 10^{-7})$ in the original SV cases with varying packet sizes. From these results, we concluded that streams composed of various packet sizes can be negatively critical to the offset estimation.

In the Fig. 5(b), the Linux PTP performed well in all of the environments. It produced its worst performance, $(1.93 \times 10^{-4}, 1.99 \times 10^{-4}, 2.05 \times 10^{-4}, 2.17 \times 10^{-4})$, in the cases of SV. In the same cases, the error of PTP and Lee's approach were $(2.87 \times 10^{-4}, 2.96 \times 10^{-4}, 2.92 \times 10^{-4}, 3.05 \times 10^{-4})$ and $(3.06 \times 10^{-4}, 3.19 \times 10^{-4}, 3.13 \times 10^{-4}, 3.26 \times 10^{-4})$. These results imply that the asymmetry-induced estimation errors of these methodologies were microseconds level in the harsh conditions.

3) *Detailed comparison with PTPv2:* Here, we present comparisons with PTPv2 based on the results in the Baseline and SV-100 network environment. For comparison of long-term performance, additional 15 (total 30) evaluations were conducted.

Baseline: The results of 30 trials of one-hour clock offset

(a) $|\varepsilon_{\theta,k}|$ over time in Baseline (b) $|\varepsilon_{\theta,k}|$ over time in SV-100

Fig. 7. Absolute estimation error $|\varepsilon_{\theta,k}|$ over time of the proposed approach and PTPv2, in Baseline (chosen from Fig. 6, trial index = 22) and SV-100 (trial index = 5). Note that the scale of the y-axis is the same.

estimations are illustrated with the box plot in Fig. 6. As shown in Figs. 6(a) and 6(b), our approach showed good overall performance compared to PTPv2. In most trials, the averages of the estimation errors were less than 5.00×10^{-8} sec, as indicated by the central values of the boxes, and the variances were low, as indicated by the short heights of the boxes. Moreover, in most cases, there were a small number of outliers and maximum estimation errors of no more than 1×10^{-7} sec. As shown in Fig. 6(b), PTPv2 showed performance results similar to those reported since 2008. PTPv2 had stable performance in terms of average estimation errors with low values of less than 5.00×10^{-8} sec. However, the variances of the estimation errors were relatively high, there were too many outliers, and the maximum estimation errors were close to or greater than 2.00×10^{-7} sec. Overall, in most trials, our approach showed more robust performance than PTPv2. While the average estimation errors in both approaches were less than 5.00×10^{-8} sec in terms of variance, outliers, and maximum values, our approach slightly outperformed PTPv2.

SV-100 : To evaluate the performance under a heavily loaded network, we compared the results under the Baseline and SV-100 network environments. We chose the best examples among the results, then illustrated the absolute

estimation errors $|\varepsilon_{\hat{\theta},k}|$ over time in Fig. 7.

As shown in Fig. 7(a), our approach and PTPv2 showed different response patterns in Baseline. In our approach, the estimation errors along the time axis were sequentially correlated. However, in PTPv2, the estimation errors were independently distributed regardless of the time. As shown in Fig. 7(b), this difference was magnified in SV-100. Our approach had a small number of outliers and relatively narrow peak-to-peak performance. However, PTPv2 had a large number of outliers with severely magnified estimation errors.

These differences in patterns come from the principles of the offset estimation methods. Our approach uses the LDE of ζ_k to handle asymmetry-induced errors. The LDE of ζ_k is highly effective because it includes $u_{\zeta,k}$ that measures variations of the asymmetry. Note that $u_{\zeta,k}$ dramatically reduces the uncertainties of the variations and retains only the uncertainty from the clock skew of the measurement, which is managed by the unbiased skew estimator in our model. Moreover, in the estimation of the hidden states, we adopted an adaptive maximum a posteriori (MAP) scheme with the LKF to minimize the MSE. Estimation of the MAP is based on prior knowledge and results; therefore, our approach does not suddenly produce large erroneous results. Consequently, our approach tends to have a small number of outliers. However, PTPv2 newly estimates the offset in each clock synchronization phase independently from the previous estimation, and the results have little correlation with the order of estimations. Therefore, the range of errors changes immediately according to the current network condition. In an idle network, as shown in Fig. 7(a), the range is narrow. However, when traffic is high on the network, the range is enlarged as shown in Fig. 7(b), even up to tens of microseconds.

In summary, our approach and PTPv2 showed similar performances, as shown in Table I and Fig. 6. However, as shown in Fig. 7, our approach exhibited more stable performance than PTPv2 in a dynamically changing network environment.

4) *Effect of ζ_k* : We provide more information about the key factor of our approach, ζ_k . We analyzed the effect of ζ_k using the model error ϵ_k of (18c). The model error ϵ_k reflects the reliability of the state-space model and enables approximation of the marginal posterior $p(\Theta|y_{1:k})$ as mentioned in Section IV-C.

Fig. 8 illustrates the model error ϵ_k . Without taking account of ζ_k , as shown in the left panels, the model error has a number of nonlinear components of the millisecond level. Considering of ζ_k , as shown in the right panels, the model error only had components with a magnitude of less than 2.00×10^{-8} sec. This performance improvement was achieved using $u_{\zeta,k}$ and the unbiased skew estimator to address the impact of asymmetry. Moreover, in the latter case, the model error approximately followed a zero-mean Gaussian. This is important for optimal estimation based on the LKF under a Gaussian assumption.

The results in Fig. 8 indirectly represent a comparison of our state-space model with the previously proposed Giorgi state-space model [17] of PTP. Without ζ_k , our model is the same as that of Giorgi. The only difference is the design of the noise terms, where Giorgi's model adopts the Allan

Fig. 8. Model errors ϵ_k over time (upper panels) and logarithmic PDFs of model error ϵ_k (lower panels) without ζ_k (left panels) and with ζ_k (right panels)

variance [41], while we use ζ_k and the unbiased clock skew estimator. The Giorgi's LKF however showed low performance in our evaluation, as seen in Table I. This is because the Giorgi state-space model assumes that its ϵ_k is zero mean Gaussian when setting the observation noise parameters. The assumption does not hold true unless the PTPv2-enabled device is allowed, as seen in the left panels of Fig. 8. Because the Kalman filtering [14] is highly sensitive to the noise parameters and requires strict parameter settings in general, this disagreement lowers the performance in a critical manner. On the other hand, our approach fills the gap, as seen in the right panels of Fig. 8. As shown, using ζ_k , our state-space model improved upon Giorgi's model.

VI. CONCLUSION

In this paper, we proposed a clock offset estimation robust to delay asymmetry for the PTP on asymmetric communication links. The PTP has a critical design flaw for estimating clock offsets under the assumption that the packet delays are symmetric. To address this, we derived an LDE to define the variation of the asymmetry and applied it to define and exclude the asymmetry-induced errors in the PTP. Moreover, to verify the approach, we implemented an optimal clock offset estimator using the proposed scheme and conducted performance evaluations. The estimator was constructed using an LKF and a Bayesian optimizer. Our approach was compared with the PTP and PTPv2 in environments with and without network loads. In the experiments, our approach mitigated most of the asymmetry-induced errors and improved the PTP by several orders of magnitude in terms of accuracy. Compared to the PTPv2, our approach showed performance improvements without incurring additional costs from specialized hardware.

ACKNOWLEDGMENT

Patent No. 10404393(US), ZL201710659956.6(CN), 10-2013617(KR)

REFERENCES

- [1] PNCS, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Standard 1588-2002*, Oct. 2002. [Online]. Available: <https://doi.org/10.1109/IEEESTD.2002.94144>
- [2] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1483–1493, Oct. 1991. [Online]. Available: <https://doi.org/10.1109/26.103043>
- [3] PNCS, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Standard 1588-2008*, Jul. 2008. [Online]. Available: <https://doi.org/10.1109/IEEESTD.2008.4579760>
- [4] A. K. Karthik and R. S. Blum, "Robust Clock Skew and Offset Estimation for IEEE 1588 in the Presence of Unexpected Deterministic Path Delay Asymmetries," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 5102–5119, Aug. 2020. [Online]. Available: <https://doi.org/10.1109/TCOMM.2020.2991212>
- [5] R. Exel, T. Bigler, and T. Sauter, "Asymmetry mitigation in IEEE 802.3 ethernet for high-accuracy clock synchronization," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 3, pp. 729–736, Mar. 2014. [Online]. Available: <https://doi.org/10.1109/TIM.2013.2280489>
- [6] G. Giorgi and G. Narduzzi, "Precision Packet-Based Frequency Transfer Based on Oversampling," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 7, pp. 1856–1863, Jul. 2017. [Online]. Available: <https://doi.org/10.1109/TIM.2017.2672478>
- [7] S. Lv, Y. Lu, and Y. Ji, "An enhanced IEEE 1588 time synchronization for asymmetric communication link in packet transport network," *IEEE Communications Letters*, vol. 14, no. 8, pp. 764–766, 2010.
- [8] S. Lee, S. Lee, and C. Hong, "An accuracy enhanced IEEE 1588 synchronization protocol for dynamically changing and asymmetric wireless links," *IEEE Communications Letters*, vol. 16, no. 2, pp. 190–192, Feb. 2012. [Online]. Available: <https://doi.org/10.1109/LCOMM.2011.092011.110582>
- [9] R. Exel, "Mitigation of asymmetric link delays in IEEE 1588 clock synchronization systems," *IEEE Communications Letters*, vol. 18, no. 3, pp. 507–510, Mar. 2014. [Online]. Available: <https://doi.org/10.1109/LCOMM.2014.012214.132540>
- [10] "The Linux PTP Project." [Online]. Available: <https://github.com/openil/linuxptp>
- [11] "PTP daemon." [Online]. Available: <https://github.com/ptpd/ptpd>
- [12] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock Synchronization of Wireless Sensor Networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2010. [Online]. Available: <https://doi.org/10.1109/MSP.2010.938757>
- [13] A. K. Karthik and R. S. Blum, "Optimum Full Information, Unlimited Complexity, Invariant, and Minimax Clock Skew and Offset Estimators for IEEE 1588," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3624–3637, Feb. 2019. [Online]. Available: <https://doi.org/10.1109/TCOMM.2019.2900317>
- [14] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. [Online]. Available: <https://doi.org/10.1115/1.3662552>
- [15] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day, 1970.
- [16] B. R. Hamilton, X. Ma, Q. Zhao, and J. Xu, "ACES: Adaptive Clock Estimation and Synchronization Using Kalman Filtering," in *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*. ACM, 2008. [Online]. Available: <https://doi.org/10.1109>
- [17] G. Giorgi and G. Narduzzi, "Performance Analysis of Kalman-Filter-Based Clock Synchronization in IEEE 1588 Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 8, pp. 2902–2909, Aug. 2011. [Online]. Available: <https://doi.org/10.1109/TIM.2011.2113120>
- [18] Z. Yang, L. Cai, Y. Lui, and J. Pan, "Environment-Aware Clock Skew Estimation and Synchronization for Wireless Sensor Networks," in *2012 IEEE INFOCOM*. IEEE, 2012. [Online]. Available: <https://doi.org/10.1109/INFCOM.2012.6195457>
- [19] H. Kim, X. Ma, and B. R. Hamilton, "Tracking Low-Precision Clocks With Time-Varying Drifts Using Kalman Filtering," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 257–270, Feb. 2012. [Online]. Available: <https://doi.org/10.1109/TNET.2011.2158656>
- [20] Z. Chaloupka, N. Alsindi, and J. Aweya, "Clock Skew Estimation Using Kalman Filter and IEEE 1588v2 PTP for Telecom Networks," *IEEE Communications Letters*, vol. 19, no. 7, pp. 1181–1184, Jul. 2015. [Online]. Available: <https://doi.org/10.1109/LCOMM.2015.2427158>
- [21] J. Cano, S. Chidami, and J. L. Ny, "A Kalman Filter-Based Algorithm for Simultaneous Time Synchronization and Localization in UWB Networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019. [Online]. Available: <https://doi.org/10.1109/ICRA.2019.8794180>
- [22] Z. Zhong, P. Chen, and T. He, "On-Demand Time Synchronization with Predictable Accuracy," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011. [Online]. Available: <https://doi.org/10.1109/INFCOM.2011.5935071>
- [23] C. Zucca and P. Tavalla, "The Clock Model and Its Relationship with the Allan and Related Variances," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 52, no. 2, pp. 289–296, 2005. [Online]. Available: <https://doi.org/10.1109/TUFFC.2005.1406554>
- [24] F. Guo, B. Zhou, and M. C. Vuran, "CFOSynt: Carrier Frequency Offset Assisted Clock Synchronization for Wireless Sensor Networks," in *IEEE INFOCOM 2017 – IEEE Conference on Computer Communications*. IEEE, 2017. [Online]. Available: <https://doi.org/10.1109/INFCOM.2017.8057075>
- [25] D. Revuz and M. Yor, *Continuous Martingales and Brownian Motion*. Berlin, Heidelberg: Springer, 1999.
- [26] S. W. Chan, G. C. Goodwin, and K. S. Sin, "Convergence properties of the Riccati difference equation in optimal filtering of nonstabilizable systems," *IEEE Transactions on Automatic Control*, vol. 29, no. 2, pp. 110–118, 1984. [Online]. Available: <https://doi.org/10.1109/TAC.1984.1103465>
- [27] J. Riccati, "Animadversiones in aequationes differentiales secundi gradus," *Acta Eruditorum Lipsiae*, pp. 66–73, 1724.
- [28] A. Valade, P. Acco, P. Grabolosa, and J.-Y. Fourniols, "A Study about Kalman Filters Applied to Embedded Sensors," *Sensors (Basel)*, vol. 17, no. 12, p. 2810, Dec. 2017. [Online]. Available: <https://doi.org/10.3390/s17122810>
- [29] H. Rue and S. Martino, "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations," *Journal of the Royal Statistical Society: Series B*, vol. 71, no. 2, pp. 319–392, 2009. [Online]. Available: <https://doi.org/10.1111/j.1467-9868.2008.00700.x>
- [30] A. Bhattacharya and S. P. Wilson, "Sequential Bayesian inference for static parameters in dynamic state space analysis," *Computational Statistics and Data Analysis*, vol. 127, pp. 187–203, Jun. 2018. [Online]. Available: <https://doi.org/10.1016/j.csda.2018.05.018>
- [31] J. Raphson, *Analysis aequationum universalis, seu, Ad aequationes algebraicas resolvendas methodus generalis, expedita, ex nova infinitarum serierum methodo, deducta ac demonstrata*. Londini: Typis T. Braddyll, prostant venales apud Johannem Taylor, 1697.
- [32] F. Semiconductor, "P2020 QorIQ Integrated Processor Reference Manual," Dec. 2012. [Online]. Available: <http://www.nxp.com/>
- [33] Cisco, "Cisco Industrial Ethernet 3000 Layer 2/Layer 3 Series Switches Data Sheet," Mar. 2019.
- [34] Tektronix, "Mixed Signal Oscilloscopes MSO5000, DPO5000 Series Data sheet," Apr. 2013.
- [35] F. Semiconductor, "IEEE 1588 for Power System Applications, FTF-SEG-F0101," Jun. 2012. [Online]. Available: <http://www.nxp.com/>
- [36] —, "Support for IEEE 1588 Protocol in PowerQUICC and QorIQ Processors, AN3423, Rev. 0," Sep. 2010. [Online]. Available: <http://www.nxp.com/>
- [37] —, "QorIQ SDK 1.9 Documentation," Dec. 2015. [Online]. Available: <http://www.nxp.com/>
- [38] S. P., "Ostinato User Guide v1.0," Sep. 2019. [Online]. Available: <https://ostinato.org/>
- [39] "Timing and synchronization aspects in packet networks, document Rec. ITU-T G.8261/Y.1361 Amendment 2 (10/2020)," *ITU, Geneva, Switzerland*, 2020.
- [40] "Timing characteristics of packet-based equipment clocks, document Rec. ITU-T G.8263/Y.1363 (08/2017)," *ITU, Geneva, Switzerland*, 2017.
- [41] D. W. Allan, "Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators," *IEEE Transactions on Automatic Control*, vol. 34, no. 6, pp. 647–654, Nov. 1987. [Online]. Available: <https://doi.org/10.1109/TUFFC.1987.26997>