2014 IEEE Military Communications Conference

An Efficient Method of Estimating the Ratio of Clock Frequency

Jun Liu (jliu@cs.und.edu)
Computer Science Department, University of North Dakota

***Abstract*—This paper describes an efficient method of estimating the ratio of clock frequency between two clocks used at the two end nodes of a one-way path. This estimation method relies on the local time measurements on the packets in a data stream going from a source node to a sink node. The time durations between adjacent packets are measured at both the source and the sink nodes according to the local clocks. The ratios of clock frequencies of two local clocks can be estimated on the suitable samples of the ratios of the time durations between two adjacent packets. The suitable samples of the ratios are obtained when two adjacent packets have experienced a zero value of the one-way variable delay. A heuristic has been developed to infer the cases when two adjacent packets experienced a zero one-way variable delay. The heuristic is used to select the suitable samples of the ratios of the time duration between two adjacent packets. Evaluation results show that the estimated values of the ratio of clock frequency very closely approximate the actual ratios. The accuracy of this method is robust to packet loss and the out-of-the-order transmission of packets. This estimation method is highly efficient without requiring complicated computations.**

***Index Terms*—One-Way Delay, Clock Synchronization, Clock Skew**

I. Introduction

It is important to make entities in a distributed environment to have the same understanding on time durations. An entity a distributed environment performs time measurement according to its local clock. The reading of a time duration measured according to a clock is proportional to the frequency of the clock. When multiple local clocks are not running at the same frequency, it is impossible to make the multiple entities to share a common understanding on a time duration. Hence, it is necessary to develop methods for unifying the readings of a time duration measured according to local clocks running at different frequencies.

The rates of the oscillators of physical clocks usually differ from each other, and the rates also vary with the environmental perturbations, like changes in temperature [1]. The timestamps reported by a physical clock are typically inconsistent to a global reference of time. A non-zero relative clock skew is resulted between two clocks if they are not running at the same frequency. The relative clock skew between two clocks is represented either as the difference or the ratio of the frequencies of the two clocks. The knowledge of the relative clock skew between two clocks is needed for unifying the local readings of a time duration. A number of methods [2], [3], [4], [5], [6] have been developed to estimate the relative clock skew between two clocks. These estimation methods mostly try to express the reading of a time duration measured according to one clock as a linear function of the the reading measured according to the other clock. Then, the relative clock skew is estimated from the coefficients of the linear function. The problems of these methods lie in two folds. First, these methods are with a low efficiency on estimating the relative clock skew. A sufficiently large number of measurements on time durations have to be performed in order to derive the coefficients of the linear function. Second, these methods lack the ability of keeping track of the fast-changing values of the relative clock skew.

This paper reports an efficient method of estimating the ratio of clock frequencies between two clocks which are equipped on two end nodes in a network. The timelines of the two clocks are modeled by piecewise linear functions. Each end node performs time measurements according to its local clock. The estimation method relies on the locally measured values of time durations according to the local clocks at the two end nodes. One of the two end nodes acts as the source node to send packets to the other end node which acts as the sink node. The time durations between sending and receiving two adjacent packets are measured at the source and sink nodes, respectively, according to the local clocks. The ratio of clock frequencies of the two clocks can be estimated on the suitable samples of the time duration between two adjacent packets. A fuzzy filtering algorithm is used to select the suitable samples of the time durations. The fuzzy filtering algorithm offers the ability of quickly following the fast-changing values of the ratio of clock frequencies. The accuracy of the estimation has been validated through network simulations. The estimated values of the ratio of clock frequencies very closely approximate the actual ratios.

This estimation method is developed from modeling the one-way delays (`OWD`s). `OWD`s provide extensive information about the state of networks and the application performance [7]. The one-way delays are more useful than the round-trip delays for applications whose performance is dependent on the one-way delays, *e.g.*, clock synchronization in a network, adjustment of sending rate in TCP [8], [9], real-time audio and video streaming services [10]. A `OWD` consists of a constant delay component and a variable delay component. Conventionally, the variable delay com-

978-1-4799-6770-4/14 $31.00 © 2014 IEEE
DOI 10.1109/MILCOM.2014.147

ponent includes the cumulative queuing, transmission, and processing delays along a one-way path, and the constant delay component includes the propagation delay along a one-way path. The measurements on `OWDs` along a one-way path involve two local clocks equipped on the two end nodes of the one-way path.

This paper is organized into the following sections. The related work is described in Section II. The clock model is described in Section III. The expression of the ratio of clock frequencies through modeling the one-way delays is described in Section IV. The method of estimating the ratio of clock frequencies between two clocks is described in Section V. A fuzzy filtering algorithm is described in Section VI for estimating the ratio of clock frequencies on the suitable samples of the ratios of time durations between adjacent packets. The application of the estimated ratios of clock frequencies on estimating the variable and constant components of `OWDs` experienced by packets is described in Section VII. Our work is summarized in Section VIII.

II. Related Work

Various methods for synchronizing clocks in a network have been developed to cope with diverse requirements in different application scenarios. These methods range from low-cost/low-precision solutions [11], [12], [13], to moderate-cost/improved-precision solutions [14], [15], to expensive/high-precision solutions [16]. The Network Time Protocol (NTP) [12], [13] is a low-cost, low-precision, purely software-based clock synchronization solution with the accuracy ranges from hundreds of microseconds to several milliseconds. NTP estimates the clock offsets between a local clock and a universal standard clock by halving the round-trip delays between the two clocks. IBM Coordinated Cluster Time (CCT) [14] aims to balance between accuracy and cost by achieving a better accuracy than NTP without additional hardware. IEEE 1588 Precision Time Protocol (PTP) [16] is a relatively expensive solution with a sub-microsecond or even nanosecond accuracy. IEEE 1588 PTP requires special hardware support and may not be fully compatible with legacy systems [17].

A number of methods [2], [3], [4], [5], [6] have been developed to estimate the relative clock skew between two clocks. An algorithm for removing a clock skew [2] has been proposed based on the input of a set of forward and reverse path delay measurements. A linear programming-based algorithm has been introduced in [3] for estimating the clock skew in network delay measurements. The relative clock skew can be estimated from delay measurements following the convex-hull approach [4]. A Piece-wise Reliable Clock Skew Estimation Algorithm (PRCSEA) [5] was introduced for the purpose of estimating the relative clock skews. A maximum likelihood estimator (MLE) [6] has been introduced to estimate the clock offset and skew. The general issue associated with these estimation methods is the computational complexity. In our work, we target to construct efficient estimation mathod.

III. The Clock Model

A physical clock has been characterized in [3] as a piecewise continuous function that is twice differentiable except on a finite set of points: $\mathcal{C} : \mathcal{R} \rightarrow \mathcal{R}$. For a data stream running in a network, we consider 3 clocks: a universal standard clock $\mathcal{C}_u$, a local clock used on the source side $\mathcal{C}_s$, and a local clock used on the sink side $\mathcal{C}_r$. The universal standard clock is used as a standard time reference and reports the "true" time. The two clocks $\mathcal{C}_s$ and $\mathcal{C}_r$ are assumed unsynchronized to each other. The timestamps with respect to the clocks $\mathcal{C}_u$, $\mathcal{C}_s$, and $\mathcal{C}_r$ are symbolized by u, t, and T, respectively. The clock characteristics include the following factors [3], [13].

- **Offset**: the difference between the readings of two clocks. The offset between clocks $\mathcal{C}_s$ and $\mathcal{C}_r$ at a universal timestamp u is $(C_r(u) - C_s(u))$.
- **Frequency**: the clock rate. The frequency of clocks $\mathcal{C}_s$ and $\mathcal{C}_r$ are $\beta_s(u) = \dfrac{d}{du}C_s(u)$ and $\dfrac{d}{du}C_r(u)$, respectively.
- **Skew**: the difference in the frequency of a clock and the universal standard clock [3]. The skew of clock $\mathcal{C}_s$ relative to clock $\mathcal{C}_r$ is $\left(\dfrac{d}{du}C_s(u) - \dfrac{d}{du}C_r(u)\right)$.
- **Drift**: the second derivative of the piecewise continuous function. The drifts of clocks $\mathcal{C}_s$ and $\mathcal{C}_r$ are $\dfrac{d^2}{du^2}C_s(u)$ and $\dfrac{d^2}{du^2}C_r(u)$, respectively. The drift of clock $\mathcal{C}_s$ relative to $\mathcal{C}_r$ is $\left(\dfrac{d^2}{du^2}C_s(u) - \dfrac{d^2}{du^2}C_r(u)\right)$.

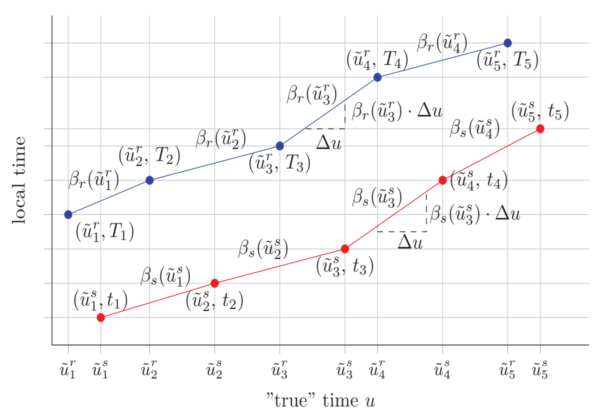


Fig. 1: Illustration of the piecewise linear functions for modeling the local clocks.

The timeline of a physical clock is characterized as a piecewise linear function as illustrated in Figure 1. The set of non-differentiable points on the timeline of clock $\mathcal{C}_s$ is denoted by

$$\{C_s(\tilde{u}_0^s), C_s(\tilde{u}_1^s), C_s(\tilde{u}_2^s), \cdots\}$$

where $\tilde{u}_j^s$ ($j \geq 0$) is the "true" time corresponding to $C_s(\tilde{u}_j^s)$. The clock frequency $\beta_s(u) = \frac{d}{du}C_s(u)$ stays constant for $u \in [\tilde{u}_j^s, \tilde{u}_{j+1}^s)$ ($j \geq 0$), and this constant value is denoted by $\beta_s(\tilde{u}_j^s)$. Similarly, the set

$$\{C_r(\tilde{u}_0^r), C_r(\tilde{u}_1^r), C_r(\tilde{u}_2^r), \cdots\}$$

is defined for the clock $\mathcal{C}_r$. The clock frequency $\beta_r(u) = \frac{d}{du}C_r(u)$ stays constant for $u \in [\tilde{u}_j^r, \tilde{u}_{j+1}^r)$ ($j \geq 0$), and this constant value is denoted by $\beta_r(\tilde{u}_j^r)$. The piecewise functions $C_s(u)$ and $C_r(u)$ are defined as (ref. Fig. 1)

$$C_s(u) = \beta_s(\tilde{u}_j^s) \cdot (u - \tilde{u}_j^s) + C_s(\tilde{u}_j^s) \text{ for } u \in [\tilde{u}_j^s, \tilde{u}_{j+1}^s), \quad (1)$$
$$C_r(u) = \beta_r(\tilde{u}_j^r) \cdot (u - \tilde{u}_j^r) + C_r(\tilde{u}_j^r) \text{ for } u \in [\tilde{u}_j^r, \tilde{u}_{j+1}^r). \quad (2)$$

Two functions $\tilde{u}_s$ and $\tilde{u}_r$ are defined to identify the starting point of a linear segment in which a send or receive time is included.

$$\begin{array}{lll} \tilde{u}_j^s = \tilde{u}_s(u) & \text{if} & u \in [\tilde{u}_j^s, \tilde{u}_{j+1}^s) \text{ and } \beta_s(u) = \beta_s(\tilde{u}_j^s); \\ \tilde{u}_j^r = \tilde{u}_r(u) & \text{if} & u \in [\tilde{u}_j^r, \tilde{u}_{j+1}^r) \text{ and } \beta_r(u) = \beta_r(\tilde{u}_j^r). \end{array} \quad (3)$$

In this paper, we treat the relative clock skew as the ratio of clock frequency. The ratio of clock frequency at the "true" time u is defined as

$$\phi(u) = \frac{\beta_r(u)}{\beta_s(u)}.$$

IV. Expression of The Ratio of Clock Frequencies Through Modeling The One-Way Delays

A sequence of packets is delivered from the source node to the sink node along a fixed one-way path. The sequence of packets are labeled by sequence numbers as $\{1, 2, \cdots, k, \cdots\}$. The notations used in the expression of one-way delays are shown in Table I. The `OWD` experienced by a packet consists of a cumulative propagation delay, a cumulative transmission delay, a cumulative processing delay, and a cumulative queueing delay along the one-way path. When the packets are with the same size, they share the same cumulative transmission delay. However, the cumulative processing delay and the cumulative queueing delay can vary for different packets. A `OWD` is the sum of the one-way constant delay (`cOWD`) and the one-way variable delay (`vOWD`). The values of `cOWD` and `vOWD` for the k-th packet are, respectively, expressed as

$$c = \tau + g + f^{\min}, \quad (4)$$
$$v_k = q_k + (f_k - f^{\min}). \quad (5)$$

The `OWD` for the k-th packet is expressed as

$$w_k = c + v_k. \quad (6)$$

Local time measurements are recorded on a per-packet basis according to the local clocks. The local timestamps of sending and receiving the k-th packet are shorthanded

τ	the one-way cumulative propagation delay along the path used by the data stream.
g	the one-way cumulative transmission delay for packets of equal size.
f_k	the cumulative processing delay for the k-th packet.
$f^{\min}$	the minimum cumulative processing delay.
q_k	the cumulative queuing delay experienced by the k-th packet.
w_k	the one-way delay for the k-th packet.
c	the constant component of w_k.
v_k	the variable component of w_k.

TABLE I: The notations of the OWD expressions.

$\beta_s(u)$	the frequency of the local clock at the source at "true" time u.
$\beta_r(u)$	the frequency of the local clock at the sink at "true" time u.
$\phi(u)$	the ratio of clock frequency at "true" time u.
u_k^s	the "true" time when the source starts to send the k-th packet.
t_k^s	the local time when the source starts to send the k-th packet.
u_k^r	the "true" time when the k-th packet is fully received at the sink.
T_k^r	the local time when the k-th packet is fully received at the sink.
a_k^s	the inter-arrival time between the k-th and the $(k-1)$-th packets at the source.
a_k^r	the inter-arrival time between the k-th and the $(k-1)$-th packets at the sink.

TABLE II: The notations used in modeling the variable and constant components of OWDs.

by $t_k^s = C_s(u_k^s)$ and $T_k^r = C_r(u_k^r)$, respectively. The notations used in modeling the OWD are shown in Table II. The OWD experienced by the k-th packet is expressed according to the "true" time as

$$w_k = u_k^r - u_k^s. \quad (7)$$

The difference between w_k and w_{k-1} is expressed as

$$\begin{aligned} w_k - w_{k-1} &= (u_k^r - u_k^s) - \left(u_{k-1}^r - u_{k-1}^s\right) \\ &= \left(u_k^r - u_{k-1}^r\right) - \left(u_k^s - u_{k-1}^s\right). \end{aligned} \quad (8)$$

The time durations between two adjacent packets measured according to the local clocks at the source and at the sink are expressed, respectively, as

$$a_k^s = t_k^s - t_{k-1}^s \quad \text{and} \quad a_k^r = T_k^r - T_{k-1}^r.$$

The timelines of the two local clokcs are modeled by piecewise linear functions (ref. Fig. 1). To simplify the analysis, we always assume that the local timestamps of two consecutive packets belong to the same linear region. Hence, we have that

$$a_k^s = \beta_s(\tilde{u}_s(u_k^s)) \cdot (u_k^s - u_{k-1}^s), \quad (9)$$
$$a_k^r = \beta_r(\tilde{u}_r(u_k^r)) \cdot (u_k^r - u_{k-1}^r). \quad (10)$$

The above expressions are only inaccurate when the timestamps of the $(k-1)$-th and the k-th packets belong to two adjacent linear regions. Combining Eqs. (8,9,10), we have that

$$w_k = w_{k-1} + \frac{a_k^r}{\beta_r(\tilde{u}_r(u_k^r))} - \frac{a_k^s}{\beta_s(\tilde{u}_s(u_k^s))}. \quad (11)$$

After removing the constant delay component from both sides of recursion (11), we also have that

$$v_k = v_{k-1} + \frac{a_k^r}{\beta_r(\tilde{u}_r(u_k^r))} - \frac{a_k^s}{\beta_s(\tilde{u}_s(u_k^s))}.$$

Reorganizing the above recursion, we further have that

$$\begin{aligned} \beta_r(\tilde{u}_r(u_k^r)) \cdot v_k &= \beta_r(\tilde{u}_r(u_k^r)) \cdot v_{k-1} + \\ & a_k^r - \frac{\beta_r(\tilde{u}_r(u_k^r))}{\beta_s(\tilde{u}_s(u_k^s))} \cdot a_k^s. \end{aligned} \quad (12)$$

The ratio of the clock frequency of the two clocks estimated at "true" time u_k^r is denoted by

$$\phi(u_k^r) = \frac{\beta_r(\tilde{u}_r(u_k^r))}{\beta_s(\tilde{u}_s(u_k^s))} \quad (13)$$

V. Estimating the Ratio of Clock frequency

The ratio of clock frequency can be estimated from the ratio of the time durations between adjacent packets measured at the sink and at the source nodes. The ratio of the time durations between the $(k-1)$-th and the k-th packets is defined as

$$\theta_k = \frac{a_k^r}{a_k^s}. \quad (14)$$

θ_k is used as the estimator of the ratio of clock frequency. Since both values of a_k^r and a_k^s can be directly measured according to the local clocks, the value of θ_k can be directly calculated from the local time measurements. Based on Eq. (12), the estimator θ_k can be expressed as

$$\theta_k = \beta_r(\tilde{u}_r(u_k^r)) \cdot \frac{v_k - v_{k-1}}{a_k^s} + \phi(u_k^r). \quad (15)$$

Since it is generally true that $\beta_r(\tilde{u}_r(u_k^r)) > 0$ and $a_k^s > 0$, we know that

$$\begin{cases} \theta_k > \phi(u_k^r), & \text{when } v_k > v_{k-1}, \\ \theta_k = \phi(u_k^r), & \text{when } v_k = v_{k-1}, \\ \theta_k < \phi(u_k^r), & \text{when } v_k < v_{k-1}. \end{cases} \quad (16)$$

The value of $\phi(u_k^r)$ can be estimated by θ_k when the condition $v_k = v_{k-1}$ is satisfied.

A. Validity of the Estimator

The satisfaction to the condition of $v_k = v_{k-1}$ can be tested by examining the difference between ϕ_k and ϕ_{k+1}. When assuming that $\beta_r(\tilde{u}_r(u_k^r)) = \beta_r(\tilde{u}_r(u_{k-1}^r))$, we know from Eq. (15) that

$$\begin{aligned} \theta_k - \theta_{k-1} &= [\phi(u_k^r) - \phi(u_{k-1}^r)] + \\ & \beta_r(\tilde{u}_r(u_k^r)) \cdot \left(\frac{v_k - v_{k-1}}{a_k^s} - \frac{v_{k-1} - v_{k-2}}{a_{k-1}^s} \right). \end{aligned}$$

When $\theta_k \approx \theta_{k-1}$, it is likely that the values of v_{k-2}, v_{k-1}, and v_k are close to each other. Furthermore, the similar values of v_{k-2}, v_{k-1}, and v_k lead to $\phi(u_k^r) \approx \phi(u_{k-1}^r)$. The common cases for v_{k-2}, v_{k-1}, and v_k to have similar values are that these values are all 0. Thus, the case of $\theta_k \approx \theta_{k-1}$ is the likely indication that v_{k-2}, v_{k-1}, and v_k are all 0. Meanwhile, it is still possible that v_{k-2}, v_{k-1}, and v_k have non-zero values when $\theta_k \approx \theta_{k-1}$. In order to filter out the cases that v_{k-2}, v_{k-1}, and v_k have non-zero values when $\theta_k \approx \theta_{k-1}$, we have developed a fuzzy filtering method to identify the suitable samples of θ_k which will lead to accurate estimation of $\phi(u_k^r)$.

Algorithm 1: Sending Packets At the Source.

$k \leftarrow 0$; $t_k^s \leftarrow 0$;
while *forever* **do**
  wait until a packet is ready;
  $k \leftarrow k + 1$;
  record in t_k^s *the current local time*;
  encapsulate k *and* t_k^s *in the packet and send the packet out to the sink*;

Algorithm 2: Estimating the Ratio of Time Durations between Adjacent Packets at the Sink.

input : A stream of packets received at the sink.
output: θ_k for each packet k.

$k' \leftarrow 0$; **# k' denotes the previous packet**
$t_{k'}^s \leftarrow 0$; $T_{k'}^r \leftarrow 0$;
for *every packet received* **do**
  read k and t_k^s from the packet;
  if $k > k'$ **then**
    record local clock time in T_k^r;
    $a_k^s \leftarrow t_k^s - t_{k'}^s$; $a_k^r \leftarrow T_k^r - T_{k'}^r$; $\theta_k \leftarrow \frac{a_k^r}{a_k^s}$;
    $k' \leftarrow k$; $t_{k'}^s \leftarrow t_k^s$; $T_{k'}^r \leftarrow T_k^r$;
  else **# out-of-the-order transmission**
    Ignore the packet.

B. The Algorithms for Estimating the Ratio of Time Duration between Adjacent Packets

The ratio of time duration between adjacent packets can be estimated at the sink node of a data stream. Only a unidirectional data stream is needed for estimating the ratio of time duration between adjacent packets. Both the source and the sink nodes of the data stream perform the local time measurements according to their local clocks. The source node encapsulates its time measurements in the data packets sent to the sink as shown in Algorithm 1. The sink estimates the ratio of clock frequency on a per-packet basis using Algorithm 2. The estimation of the ratio of time duration between adjacent packets is robust to packet losses and out-of-the-order transmission of packets.

C. Numerical Validation

Numerical validation through network simulations has been performed to examine the validity of the heuristic for inferring $\theta_k = \phi(u_k^r)$ when $\theta_k = \theta_{k-1}$. The DropTail queuing policy is adopted at all the routers. The propagation delays on all links are 10 ms. The bandwidths of an access link, a non-bottleneck link, and a bottleneck link are 50 Mbps, 30 Mbps (or 3.75 MBytes/s), and 25 Mbps (or 3.125 MBytes/s), respectively. Two data streams are used to deliver packets from their sources to their corresponding sinks along a simple network topology as shown in Figure 2. One data stream is used for the numerical validation with its arrival pattern of packets following a *Poisson* process. The other stream is used to generate the cross traffic. The cross traffic stream sends packet batches with the batch arrivals following a *Poisson* process and the batch size following a Normal distribution. The adoption of a Normal distribution of the batch size of a cross traffic tries to mimic the *Gaussian* traffic pattern seen at the major aggregation routers where a large number of flows are multiplexed [18], [19]. The traffic patterns of the traffic flows are shown in Table III. $\mathbb{E}[a^s]$ represents the average inter-arrival time between batches, $\mathbb{E}[N_P]$ represents the average batch size in packets, $\mathrm{Var}[N_P]$ represents the variance of the batch size in unit of square packets for the Normal distribution, and $\min N_P$ represents the minimum batch size in packets.

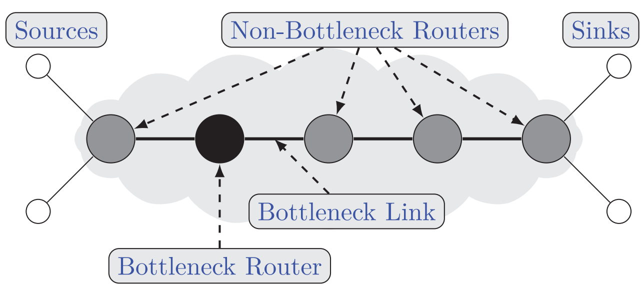


Fig. 2: The network topology used in the network simulation.

	Probe Traffic	Cross Traffic
$\mathbb{E}[a^s]$ (Seconds)	0.05	0.002
$\mathbb{E}[N_P]$ (Packets)	-	10
$\mathrm{Var}[N_P]$ (Packets^2)	-	2100
$\min N_P$ (Packets)	-	2
Packet Size (Bytes)	100	1000

TABLE III: Specification of the Traffic Streams.

A pair of skewing clocks are used at the two end nodes of the data stream used for numerical validation. The timelines of both clocks are modeled by piecewise linear functions. The timelines of the clocks are simulated using the specifications shown in Table IV. The initial offsets of the local clocks on the source and sink sides are denoted by δ^s and δ^r, respectively. Similarly, the initial clock frequency are denoted by $\beta_s(0)$ and $\beta_r(0)$, respectively.

	Source Clock	Sink Clock
δ^s or δ^r (Seconds)	0.5	1.0
$D^{\min}$ (Seconds)	5	10
$D^{\max}$ (Seconds)	10	20
$\beta_s(0)$ or $\beta_r(0)$	1.2	0.9
λ^s or λ^r	0.2	0.2
$\beta^{\min}$	0.2	0.2
$\beta^{\max}$	2.0	2.0

TABLE IV: Specification of the Skewing Clocks.

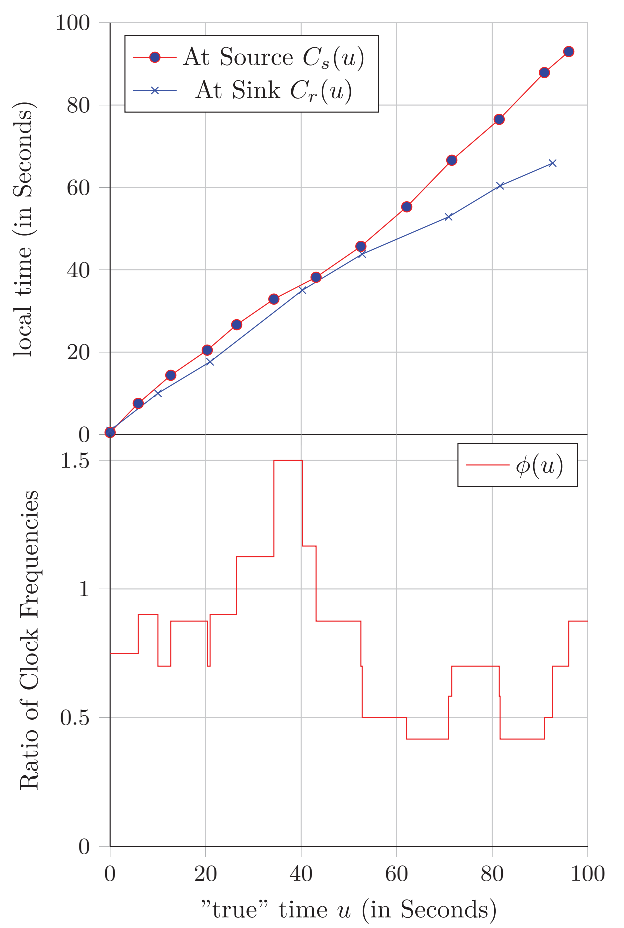


Fig. 3: Illustration of the timeline of the two local clocks and the ratio of clock frequencies. The ratios of clock frequencies are calculated from the piece-wise linear timelines of the two local clocks.

Each clock runs at its current clock frequency for a period of time before the current clock frequency is changed to a new value. This pattern repeats until the end of a network simulation. The duration of a time period uniformly goes between $D^{\min}$ and $D^{\max}$. The values of clock frequency oscillate between $\beta^{\min}$ and $\beta^{\max}$. The amount of a change on the current clock frequency on the source (sink) side is either $+\lambda^s$, 0, or -λ^s ($+\lambda^r$, 0, or -λ^r). The piecewise linear timelines of the local clocks and the ratio of clock frequencies collected in the simulations are shown in Fig. 3.

Algorithm 3: Estimating the Ratio of Clock Frequencies at the Sink.

input : A stream of packets received at the sink. Paramter ϵ and ϵ^{mid}.
output: $\hat{\phi}_k$ and n_k for each packet k.

$k' \leftarrow 0;$ # k' **denotes the previous packet**
$t^s_{k'} \leftarrow 0; \quad T^r_{k'} \leftarrow 0; \quad \theta_{k'} \leftarrow 0 \quad n_{k'} \leftarrow 0;$
$\hat{\phi}_k \leftarrow 1.0; \quad \rho^{\text{high}} \leftarrow 2.0; \quad \rho^{\text{mid}} \leftarrow 1.0; \quad \rho^{\text{low}} \leftarrow 0.5;$
for *every packet received* **do**
  read k and t^s_k from the packet;
  record local clock time in T^r_k;
  $a^s_k \leftarrow t^s_k - t^s_{k'}; \quad a^r_k \leftarrow T^r_k - T^r_{k'}; \quad \theta_k \leftarrow \dfrac{a^r_k}{a^s_k};$
  if $|\theta_k - \theta_{k'}| < \epsilon$ **then**
    if $\theta_k \geq \rho^{\text{high}}$ **then** # **case (i)**
      $\rho^{\text{high}} \leftarrow \dfrac{1}{2} \cdot (\theta_k - \rho^{\text{high}})$
    if $\theta_k \leq \rho^{\text{low}}$ **then** # **case (ii)**
      $\rho^{\text{low}} \leftarrow \dfrac{1}{2} \cdot (\rho^{\text{low}} - \theta_k)$
    if $\rho^{\text{low}} < \theta_k \leq \rho^{\text{mid}}$ **then** # **case (iii)**
      $\psi \leftarrow \dfrac{(\rho^{\text{mid}} - \theta_k)}{(\rho^{\text{mid}} - \rho^{\text{low}})};$
      $\rho^{\text{high}} \leftarrow \psi \cdot (\rho^{\text{mid}} - \theta_k);$
      $\rho^{\text{mid}} \leftarrow (1 - \psi) \cdot (\rho^{\text{mid}} - \theta_k);$
    if $\rho^{\text{mid}} < \theta_k < \rho^{\text{high}}$ **then** # **case (iv)**
      $\psi \leftarrow \dfrac{(\theta_k - \rho^{\text{mid}})}{(\rho^{\text{high}} - \rho^{\text{mid}})};$
      $\rho^{\text{high}} \leftarrow \psi \cdot (\theta_k - \rho^{\text{mid}});$
      $\rho^{\text{mid}} \leftarrow (1 - \psi) \cdot (\theta_k - \rho^{\text{mid}})$
    if $|\theta_k - \rho^{\text{mid}}| < \epsilon^{\text{mid}}$ **then**
      $\hat{\phi}_k \leftarrow \theta_k; \quad n_k \leftarrow n_{k'} + 1;$
      if $n_k > N^{\max}$ **then** $n_k = N^{\max}$;
    else
      $\hat{\phi}_k \leftarrow \hat{\phi}_{k'}; \quad n_k \leftarrow n_{k'} - 3;$
      if $n_k < 0$ **then** $n_k \leftarrow 0$;
  else $\{ \hat{\phi}_k \leftarrow \hat{\phi}_{k'}; \quad n_k \leftarrow n_{k'} \}$;
  Output $\hat{\phi}_k$, n_k;
  $k' \leftarrow k; \quad t^s_{k'} \leftarrow t^s_k; \quad T^r_{k'} \leftarrow T^r_k;$

and at the sink. The source encapsulates its time measurements in the data packets sent to the sink as shown in Algorithm 1. The sink estimates the ratios of clock frequencies on a per-packet basis using Algorithm 3. The 3 modes are updated in 4 cases as shown in Figure 6 and Algorithm 3. Cases (i) and (ii) are used to filter out and follow the tend of the outliers of θ_k. Cases (iii) and (iv) are used to update the middle range and the middle mode. The value of θ_k is treated as an estimated ratio of clock frequencies when it stays in the close neighborhood of the middle mode, *i.e.*, $|\theta_k - \rho^{\text{mid}}| < \epsilon^{\text{mid}}$. The quality of an estimated ratio of clock frequencies is associated with a confidence measure n_k. The value of n_k goes between 0 (denoting a poor estimation) and 10 (denoting a very good estimation). The value of n_k is increased by 1 when a value of θ_k falls in the close neighborhood of the middle mode, and it is decreased by 3 when falling outside the close neighborhood. An example of mode tracking driven by the empirical samples of θ_k obtained in the numerical validation is shown in Figure 7. It can be seen that the high and low modes can prevent the extreme values of θ_k from updating the value of the middle mode. Meanwhile, the quality measure n_k can also offer us with a numeric measure of the confidence that the value of θ_k is similar to the ratio of clock frequencies.

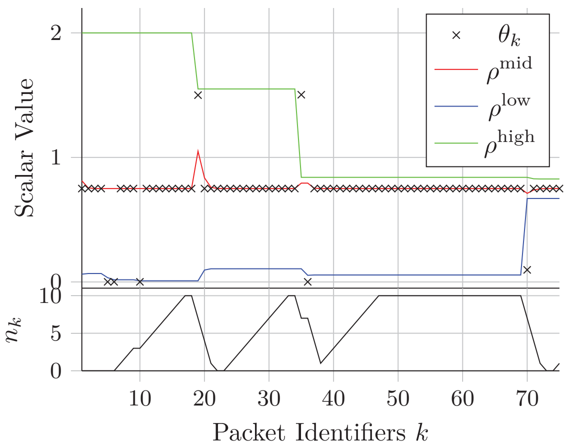
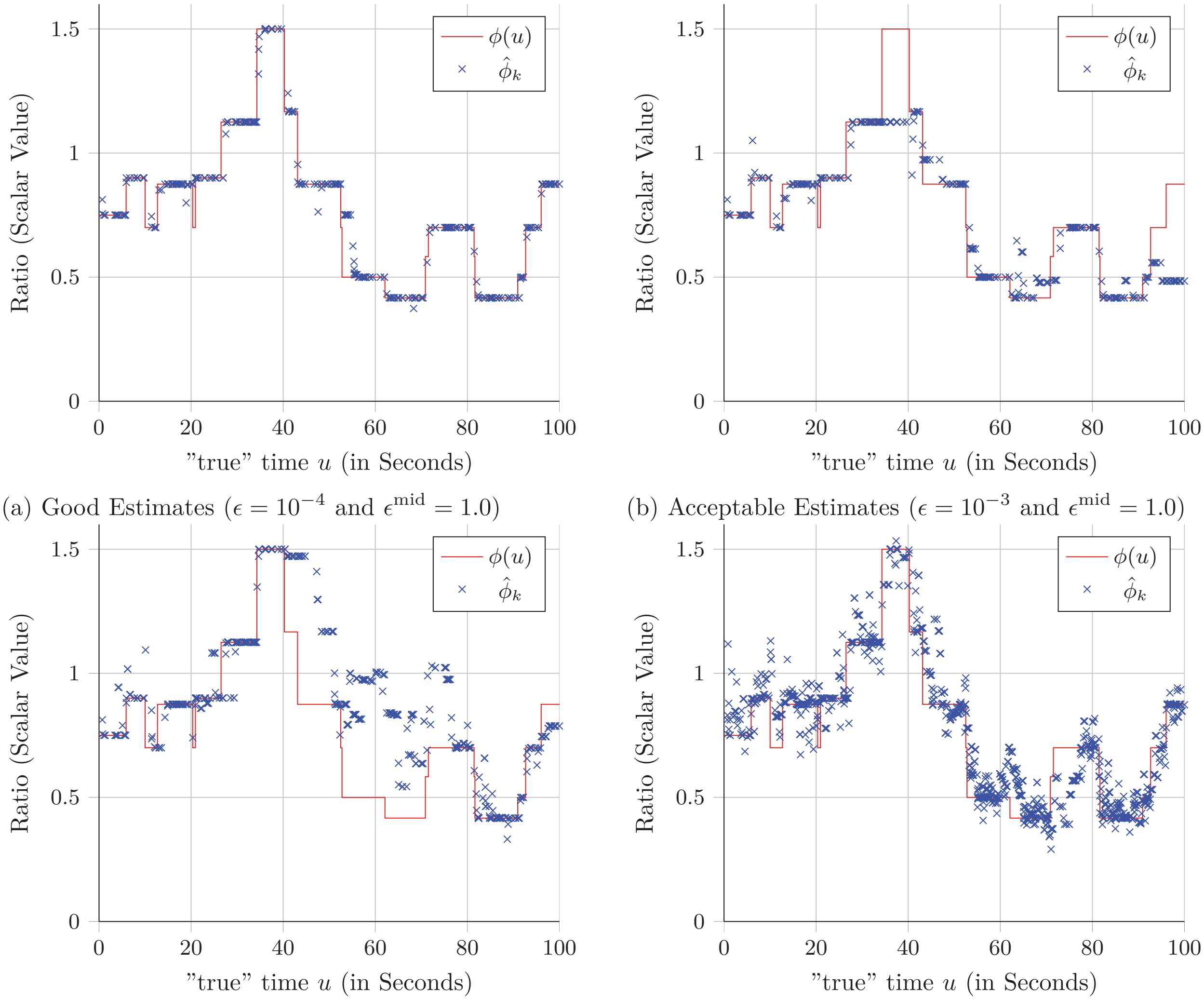


Fig. 7: Illustration of the mode tracking.

B. Numerical Validation

The effectiveness of the method of estimating the ratio of clock frequency has been validated using network simulations. The validation makes use of the same network simulations as described in Section V-C. The effectiveness of estimating the ratio of clock frequency using Algorithm 3 is shown in Fig. 8. The estimated values of the ratios of clock frequencies are compared to the true values of ratios of clock frequency known from the clock specifications. The accuracy of the estimation is only impacted by the parameter ϵ. The value of ϵ^{mid} has no direct impact on the accuracy of the estimation. Under the small value of the parameter ϵ (*e.g.*, $\epsilon = 10^{-2}$), the estimated ratios are very close to the actual values of the ratios at most of the samples of ϕ_k (ref. Fig. 8 (a)). As the value of the parameter ϵ increases, the accuracy of the estimated ratios deteriorates (ref. Fig. 8 (b)-(d)). The value of the quality indicator n_k is impacted by the parameter ϵ^{mid}. A lower value of ϵ^{mid} generally results in volatile values of the quality indicator n_k (ref. Fig. 9 (a)). Under volatile values of n_k, cautious estimation on $\phi(u^r_k)$ is made on high values of n_k. A higher value of ϵ^{mid} generally results in constantly



(a) Good Estimates ($\epsilon = 10^{-4}$ and $\epsilon^{\text{mid}} = 1.0$)

(b) Acceptable Estimates ($\epsilon = 10^{-3}$ and $\epsilon^{\text{mid}} = 1.0$)

(c) Mediocre Estimates ($\epsilon = 10^{-2}$ and $\epsilon^{\text{mid}} = 1.0$)

(d) Poor Estimates ($\epsilon = 10^{-1}$ and $\epsilon^{\text{mid}} = 1.0$)

Fig. 8: Comparison of the estimated and the "true" ratios of clock frequency. The accuracy of the estimation is only impacted by the value of ϵ.

high values of the quality indicator n_k (ref. Fig. 9 (b)). Under constantly high values of n_k, estimation on $\phi(u_k^r)$ by θ_k may not be accurate even on high values of n_k.

VII. Application of the Ratios of Clock Frequencies

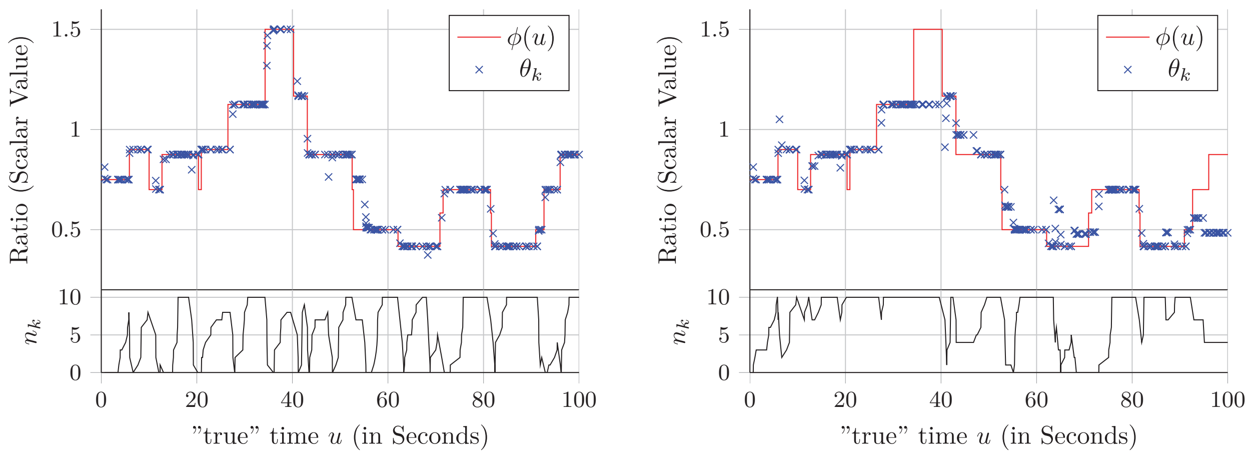
The knowledge of the ratio of clock frequencies can be used to remove the skew between two clocks if one of the clock runs at the same frequency as the universal standard clock (UTC). Such application has been reported in [3]. However, the ratio of clock frequencies can not be applied to remove the relative skew between two clocks if none of the two clocks is UTC. We report a new application of the ratio of clock frequencies for the estimation of the values of `vOWDs` and `cOWDs` experienced by each packet. The estimation is performed on the sink side. The estimated values of the `vOWDs` are distorted by a multiplicative factor which is the sink-side clock frequency. The estimated values of the `cOWDs` are distorted by the sink-side clock frequency and the relative clock offset. The knowledge of the clock frequency and relative clock offset are required to further estimate the actual values of `vOWDs` and `cOWDs` from the distorted values. The notations used in the estimation are shown in TABLE VI.

$\hat{v}_k$	the distorted value of `vOWD` for the k-th packet.
$\hat{c}_k$	the distorted value of the `cOWD` for the k-th packet.

TABLE VI: The notations used in estimating the distorted values of `vOWDs` and `cOWDs`.

A. Estimating the Distorted Values of `vOWDs`

It can be seen from recursion (12) that $v_k = v_{k-1}$ if $a_k^r = \phi(u_k^r)\cdot a_k^s$. This means that the $(k-1)$-th and k-th



(a) Cautious Estimates ($\epsilon = 10^{-5}$ and $\epsilon^{\text{mid}} = 10^{-6}$)

(b) Incautious Estimates ($\epsilon = 10^{-3}$ and $\epsilon^{\text{mid}} = 10^{-1}$)

Fig. 9: Comparison of the estimated and the "true" ratios of clock frequency. The value of the quality indicator of the estimation is impacted by the value of ϵ^{mid}.

packets have experienced the same amount of cumulative variable delays along the one-way path. Among the cases that adjacent packets experience the same amount of cumulative variable delays, we treat the case that the two packets experience a zero cumulative variable delay as the common case. When k_0 ($k_0 < k$) is denoted as the most recent packet before the k-th packet that yields $a^r_{k_0} = \phi(u^r_{k_0})\cdot a^s_{k_0}$, the estimated value of the vOWD experienced by the k-th packet is expressed as

$$\beta_r(\tilde{u}_r(u^r_k))\cdot v_k = \beta_r(\tilde{u}_r(u^r_{k_0}))\cdot v_{k_0} + \sum_{k_0 \le j \le k} \left[a^r_j - \phi(u^r_j)\cdot a^s_j\right].$$

If $v_{k_0} = 0$ is assumed, then the estimated value of the one-way cumulative variable delay can be estimated by

$$\hat{v}_k = \beta_r(\tilde{u}_r(u^r_k))\cdot v_k = \sum_{k_0 \le j \le k} \left[a^r_j - \phi(u^r_j)\cdot a^s_j\right]. \quad (17)$$

A heuristic that $v_{k_0} = 0$ when $a^r_{k_0} = \phi(u^r_{k_0})\cdot a^s_{k_0}$ is assumed in the estimation. The validity of this heuristic has been demonstrated in Fig. 4. The method for estimating the value of $\phi(u)$ has been described in Algorithm 3 in Section V. Hence, the value of $\tilde{v}_k$ can be estimated based on the time durations between adjacent packets according to local clocks. $\tilde{v}_k$ is distorted from v_k by a multiplicative factor of $\beta_r(\tilde{u}_r(u^r_k))$.

B. Estimating the Distorted Values of cOWD*s*

The cOWDfor the k-th packet is expressed based on Eqs. (6) and (7) as

$$c + v_k = u^r_k - u^s_k. \quad (18)$$

The distorted value of the cOWD for the k-th packet is estimated as

$$\hat{c}_k = \beta_r(\tilde{u}_r(u^r_k))\cdot(c + \delta_k) = T^r_k - \phi(u^r_k)\cdot t^s_k - \hat{v}_k, \text{ with} \quad (19)$$

$$\delta_k = \frac{C_r(\tilde{u}_r(u^r_k))}{\beta_r(\tilde{u}_r(u^r_k))} - \frac{C_s(\tilde{u}_s(u^s_k))}{\beta_s(\tilde{u}_s(u^s_k))} + \tilde{u}_s(u^s_k) - \tilde{u}_r(u^r_k). \quad (20)$$

The value of $\hat{c}_k$ can be calculated from the local time measurements and the estimated value of the ratio of clock frequencies. The value of $\hat{c}_k$ is distorted by both the value of $\beta_r(\tilde{u}_r(u^r_k))$ and δ_k. δ_k represents the relative offset between the two local clocks.

Algorithm 4: Estimating the Distorted Values of vOWDs and cOWDs at the Sink.

input : A stream of packets.
(packet number starts from 1)
output: $\{\hat{v}_k, k \ge 1\}$ and $\{\hat{c}_k, k \ge 1\}$.

$k' \leftarrow 0$; **# k' denotes the previous packet**
$t^s_{k'} \leftarrow 0$; $T^r_{k'} \leftarrow 0$; $\hat{v}_{k'} \leftarrow 0$;
for *every packet received* **do**
  record the current local time in T^r_k;
  read k and t^s_k from the packet;
  if $k > k'$ **then**
    $a^s_k \leftarrow t^s_k - t^s_{k'}$; $a^r_k \leftarrow T^r_k - T^r_{k'}$;
    # estimate ratio of clock frequency
    $\phi_k \leftarrow$ `freq_ratio_estimation`(a^s_k, a^r_k);
    if $a^r_k = \phi_k\cdot a^s_k$ **then** $\hat{v}_k \leftarrow 0$;
    else $\hat{v}_k \leftarrow \hat{v}_{k'} + (a^r_k - \phi_k\cdot a^s_k)$;
    $\hat{c}_k \leftarrow [T^r_k - \phi_k\cdot t^s_k - \hat{v}_k]$;
    $k' \leftarrow k$; $t^s_{k'} \leftarrow t^s_k$; $T^r_{k'} \leftarrow T^r_k$; $\hat{v}_{k'} \leftarrow \hat{v}_k$;
  else **# out-of-the-order transmission**
    Ignore the packet.

C. The Estimation Algorithms

The algorithms running at the source and at the sink for estimating the vOWDs and cOWDs are shown in Algorithms 1 and 4, respectively. The estimation on the ratio of clock frequencies (`freq_ratio_estimation` in Algorithm 4) is performed by the body of the loop in Algorithm 3. The estimation on the distorted values of `vOWDs` and `cOWDs` is also robust to packet losses and the out-of-the-order transmission of packets.

VIII. Conclusions

This paper describes an efficient method of estimating the ratios of clock frequencies between two skewing clocks. A pair of skewing clocks are used as the local clocks at the two end nodes of a one-way path. This estimation method relies on the local time measurements on the packets in a data stream going from a source node to a sink node. The time durations between adjacent packets are measured at both the source and the sink nodes according to the local clocks. Under the appropriate condition, the ratio of clock frequencies equals to the ratio of the time duration between two adjacent packets at the sink node to the time duration between the same two packets at the sink node. The appropriate condition is that both of the two adjacent packets experienced a zero one-way variable delay. A heuristic has been developed to infer the cases when two adjacent packets experienced a zero one-way variable delay. The heuristic is used to select the suitable samples of the ratios of the time duration between two adjacent packets. The effectiveness of the estimation method is robust to packet losses and out-of-the-order transmission of packets. This estimation method is capable of following the changed values of the ratios of clock frequencies if the clock frequencies vary over time. The effectiveness of the estimation method has been validated through network simulations. The estimated ratios of clock frequencies have been shown to very closely approximate the actual ratios. The estimated ratios of clock frequencies can be applied to estimate the distorted values of the one-way variable and constant delays experienced by packets.

References

[1] Julien Ridoux and Darryl Veitch. Principles Of Robust Timing Over The Internet. In *Communications of the ACM*, 53(5):54–61, May 2010.

[2] Vern Paxson. On Calibrating Measurements Of Packet Transit Times. In *ACM SIGMETRICS Performance Evaluation Review*, 26(1):11–21, 1998.

[3] Sue B. Moon, Paul Skelly, and Donald F. Towsley. Estimation And Removal Of Clock Skew From Network Delay Measurements. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 227–234, IEEE, New York, NY, March 21-25 1999.

[4] Li Zhang, Zhen Liu, and Cathy Honghui Xia. Clock Synchronization Algorithms For Network Measurements. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 160–169, New York, NY, June 23-27 2002.

[5] Jingping Bi, Qi Wu, and Zhongcheng Li. On Estimating Clock Skew For One-way Measurements. In *Computer Communications (Elsevier)*, 29(8):1213–1225, May 2006.

[6] Qasim M. Chaudhari, Erchin Serpedin, and Khalid Qaraqe. On Maximum Likelihood Estimation Of Clock Offset And Skew In Networks With Exponential Delays. In *IEEE Transactions on Signal Processing*, 56(4):1685–1697, 2008.

[7] Ana Hernandez and Eduardo Magana. One-way Delay Measurement And Characterization. In *Proceedings of the International Conference on Networking and Services (ICNS)*, page 114, Athens, Greece, June 19–25 2007.

[8] Ren Wang, Giovanni Pau, Kenshin Yamada, M. Y. Sanadidi, and Mario Gerla. tcp Startup Performance In Large Bandwidth Networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 796–805, IEEE, Hong Kong, China, March 7-11 2004.

[9] Kathleen Nichols and Van Jacobson. Controlling Queue Delay. In *ACM Queue*, 10(5):1–15, May 2012.

[10] Omer Gurewitz, Israel Cidon, and Moshe Sidi. One-way Delay Estimation Using Network-wide Measurements. In *IEEE Transactions on Information Theory*, 52(6):2710–2724, June 2006.

[11] David L. Mills. Improved Algorithms For Synchronizing Computer Network Clocks. In *Proceedings of the ACM International Conference on Communications Architecture and Protocols (SIGCOMM)*, London, United Kingdom, August 31-Semptember 2 1994.

[12] David L. Mills. Network Time Protocol Version 4 Reference And Implementation Guide. University of Delaware, Technical Report 06-6-1, June 2006.

[13] David L. Mills. Network Time Protocol (version 3) Specification, Implementation And Analysis. RFC 1305 (Draft Standard), 1305, IETF, March 1992.

[14] Steven Froehlich, Michel Hack, Xiaoqiao Meng, and Li Zhang. Achieving Precise Coordinated Cluster Time In A Cluster Environment. In *Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS)*, pages 54–58, Ann Arbor, MI, September 24-26 2008.

[15] Darryl Veitch, Julien Ridoux, and Satish Babu Korada. Robust Synchronization Of Absolute And Difference Clocks Over Networks. In *IEEE Transactions on Networking*, 17(2):417–430, April 2009.

[16] Ieee Standard For A Precision Clock Synchronization Protocol For Networked Measurement And Control Systems. IEEE, 1588-2008, pages 1–269, 2008.

[17] Enrique Mallada, Xiaoqiao Meng, Michel Hack, Li Zhang, and Ao Tang. Skewless Network Clock Synchronization. In *Proceedings of the International Conference on Network Protocols (ICNP)*, IEEE Computer Society, Göttingen, Germany, October 07-10 2013.

[18] Jinwoo Choe and Ness B. Shroff. New Bounds And Approximations Using Extreme Value Theory For The Queue Length Distribution In High-speed Networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 364–371, San Francisco, CA, March 29-April 2 1998.

[19] Ronald G. Addie and Moshe Zukerman. An Approximation For Performance Evaluation Of Stationary Single Server Queues. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 28-April 1 1993.