East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2022-330.120623

Vol. **14**, No. 1, pp. 179-194
February 2024

# A Round Trip Time Weighting Model for One-way Delay Estimation

Wei Zhang[1], Zhenyu Ming[2,*], Liping Zhang[1] and Yanwei Xu[2]

[1]*Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China.*
[2]*Theory Lab, Center Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Hong Kong 999077, SAR, China.*

*Received 23 November 2022; Accepted (in revised version) 12 June 2023.*

**Abstract.** A new delay estimation model using a round trip time as a weight of the asymmetry of each host pair is developed. It improves the estimation accuracy and is suitable for complex wide area network architecture. For large-scale scenarios in practice, we design a symmetric Gauss-Seidel alternating direction method of multipliers. It significantly reduces memory consumption and computational cost. Numerical experiments demonstrate the accuracy and efficiency of the model and algorithm.

**AMS subject classifications**: 68M12, 90B18, 90C20

**Key words**: One-way delay estimation, clock synchronization, ADMM.

## 1. Introduction

In this work, we present a novel optimization model with the weighted minimum norm principle for one-way delay (OWD) estimation in wide-area networks (WANs), whose mathematical formulation is as follows:

$$\min_{\boldsymbol{x}\in\Omega} \quad \sum_{i\neq j} \frac{|x_{ij}-x_{ji}|^2}{\mathrm{RTT}_{ij}^{\alpha}}, \quad \Omega=\left\{A_{loops}\boldsymbol{x}=\boldsymbol{b}_{loops},\ \boldsymbol{x}\geq \mathbf{0}\right\}. \tag{1.1}$$

Here $x_{ij}$ is the delay from node $i$ to another node $j$ in a particular network, and $\mathrm{RTT}_{ij}$ is the round trip time between them. The constraints are the observations of loop measurements and natural non-negativity of delays, inheriting from [16]. Note that model (1.1) is actually a loop estimation method using RTT weighting (LERW), in which $\alpha$ controls the effect of RTT in weighting.

With the continuous complexity and scale of computer networks, network performance analysis becomes intractable, bringing challenges to downstream tasks, including robust

*Corresponding author. *Email addresses:* mingzhenyu1@huawei.com (Z. Ming), xuyanwei1@huawei.com (Y. Xu), lipingzhang@mail.tsinghua.edu.cn (L. Zhang), zhang-w20@mails.tsinghua.edu.cn (W. Zhang)

network design, scheduling, and congestion control. LERW provides an improved methodology for accurate OWD estimation, which is desired extremely in characterizing network performance. Informally, OWD refers to the time it takes for a data packet to be sent from one network node to another. It plays a vital role in presenting the real-time status of the network [2]. One should note that the sending is only one-way, which means OWD focuses only on unidirectional characteristics. In contrast, RTT, another standard delay metric, provides a clear view of bidirectional characteristics. It is generally believed that the measurement of OWD is more critical than that of other delay metrics, including RTT. This is because the performance of an application may greatly depend on unidirectional characteristics [2]. For example, the quality of video on demand mostly depends on the performance of the links from servers to clients. File transfer only relies on the path from sender to receiver [26]. For this reason, service level agreements (SLAs) that aim at ensuring QoS in real-time applications, such as Voice over IP (VoIP) [4], use OWD as a parameter.

However, measuring OWD directly is impossible because the clocks of hosts in the network are not synchronized [27]. This asynchrony stems from the different frequencies of the quartz crystal oscillators of hosts, which is severe in WANs. Generally, two timestamps representing the transmission time at the transmitter and the reception time at the receiver are stamped when a probe packet is sent. The clock offset causes their difference to deviate from the real OWD. The immediate idea is to achieve OWD estimation through high-precision clock synchronization, but the common methods do not work well due to fundamental limits [12]. NTP [21] is one of the oldest clock synchronization protocols, which calculates the clock offset by simply combining the four timestamps obtained from a pair of packets sent in opposite directions. It has an unsatisfying accuracy of tens of milliseconds in WANs [22], thanks to inaccurate timestamps and the unrealistic assumption that the forward and reverse delays are symmetrical. PTP, also known as IEEE 1588 [11], is another usual method for synchronization, adopting hardware timestamps to counter stack delays occurring in time stamping. It is superior to NTP and suitable for high-precision scenes. If deployed properly, it will reach an accuracy within $3.2ms$. Nevertheless, it suffers from asymmetry like NTP. GPS is the most reliable clock synchronization method, providing the highest clock synchronization accuracy [27]. Unfortunately, unique hardware and high expense hinder its application on the Internet, making it impractical to synchronize using GPS.

A novel method called loop estimation for OWD measurement without clock synchronization was proposed by Gurewitz *et al*. [15–17], who suggested considering the OWD measurement as an optimization problem. They offered to perform measurements along loops to form an underdetermined system of equations with OWDs as variables. Then a convincing optimization model aiming to minimize the asymmetries of all node pairs was proposed to help select a solution in the colossal solution space, taking the form of LERW with $\alpha = 0$. For the sake of distinction, we shall hereafter call it LE. Benefiting from a great deal of information provided by loop measurements, LE works remarkably in multiple network architectures, far superior to previous methods. Considering its excellence, one did some transformations later and then applied it to the last step of the Huygens algorithm, a recognized clock synchronized method for local area networks (LANs) in industry [13].

Nevertheless, LE was initially designed for symmetrical scenes, or rather, slightly asymmetrical scenes. It does not examine the diversity of asymmetries of different pairs that exist in WANs. As a result, LE is shortsighted and performance constrained in WANs.

LERW is proposed in this paper exactly to overcome LE's limitations. In contrast, it penalizes delay asymmetries of node pairs with small RTTs, thus allowing those with large RTTs to dominate significant asymmetries. We insist on its advance, for it agrees with the law that asymmetry in WANs is correlated positively with RTT presented in [23]. Moreover, we show its rationality theoretically by proving to be an upper bound of the ideal objective. We have to admit that this upper bound is relaxed, making the theory not essential. Given the sheer difficulty of the problem, we still emphasize its value and inspiration. Analogous to LE, LERW benefits from the increasing scale of networks so that one can artificially introduce extra nodes to improve estimation accuracy. Unfortunately, the traditional interior point method's high memory consumption and low computation speed for optimization cause a bottleneck in engineering. A crude way to deal with it is to remove inequality constraints to reduce the difficulty of solving, as did in [16,17], which makes the loss of rigor. For this, we propose a new solution algorithm named symmetric Gauss-Seidel alternating direction method of multipliers (SGS-ADMM) to solve LERW. The algorithm breaks the problem down into multiple iterations involved with a univariate optimization model solution, thus reducing computation complexity and memory consumption significantly. Theoretically, it is proven to converge globally to the optimal solution. We assert its potential applications in engineering.

The rest of the paper is organized as follows. After stating the OWD estimation problem in Section 2, we introduce the LERW model and illustrate its reasonability in Section 3. In Section 4, we cover the SGS-ADMM algorithm and discuss its convergence. Section 5 is our numerical experiments, and finally, the summary of our paper.

## 2. Problem Statement

### 2.1. Probes and timestamps

The communication between hosts is usually performed by sending probing packets in networks. In the context of delay estimation, packets are timestamped at the sender and the receiver, respectively, as shown on the left of Fig. 1. The two timestamps, denoted as $t_1$ and $t_2$, are established according to the local clocks. It is biased to regard their difference as the actual delay because there is an offset between the clocks. Suppose that the clocks of hosts $A$ and $B$ offset $\tau_a$ and $\tau_b$ related to the reference clock. The relationship between the delay, clock offsets, and timestamps is formulated as

$$x_{ab} + \tau_b - \tau_a = t_2 - t_1,$$

where $x_{ab}$ is the OWD from $A$ to $B$. If swapping the sender and the receiver and denoting corresponding timestamps as $t_3$ and $t_4$, we have

$$x_{ba} + \tau_a - \tau_b = t_4 - t_3.$$

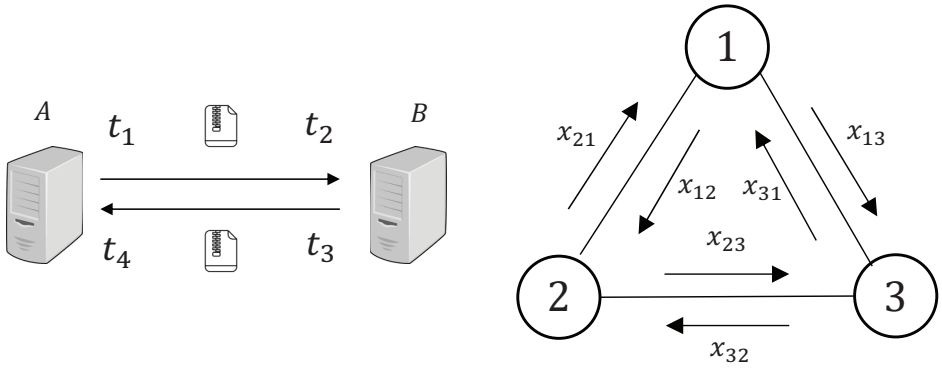



Figure 1: Left: Two hosts send probe packets to each other. Right: Connected graph of three hosts.

Adding the above two equations gives that

$$x_{ab} + x_{ba} = t_4 - t_3 + t_2 - t_1 \overset{\text{def}}{=} \text{RTT}_{ab} = \text{RTT}_{ba}.$$

If the delays are symmetric, i.e, $x_{ab} = x_{ba}$, it exists that

$$x_{ab} = x_{ba} = \frac{\text{RTT}_{ab}}{2}. \tag{2.1}$$

Thanks to the blocks' complex network architectures and different network protocols, enormous delay asymmetries exist almost everywhere in WANs. Consequently, the Eq. (2.1) is no longer applicable.

## 2.2. Host network

This paper uses a completely connected graph with $N$ nodes and $m = (N-1)N$ directed edges to characterize communication between multiple hosts in WANs. The numbered node set denotes hosts, and a pair of directed edges connect any two nodes. An example of a three-host network is shown on the right of Fig. 1. Note the graph represents just a logical network instead of a physical one. The directed edges are not actual network links but only refer to communication relationships. Routers responsible for forwarding packets are hidden. One can send packets along arbitrary paths network-wide and get all timestamps. Denoting $x_{ij}$ the delay from $i$ to $j$, we combine all $x_{ij}$ into a vector $\boldsymbol{x} \in \mathbb{R}^m$. Assume its $k$-th entry $x^{(k)}$ corresponds to the delay from node $i_k$ to $j_k$. There are two rules about $\boldsymbol{x}$:

1. For all $k \geq 0$, $i_{2k+1} = j_{2k+2},\ j_{2k+1} = i_{2k+2}$.

2. For all $k_1, k_2 \geq 0$, if $k_1 > k_2$, then $i_{2k_1+1} \geq i_{2k_2+1},\ j_{2k_1+1} \geq j_{2k_2+1}$. Our goal is to estimate $\boldsymbol{x}$ given the observations provided by probing packets.

## 3. LERW Model

In this section, we develop the LERW model gradually, first its constraints and then the objective function.

### 3.1. Loop measurement

Loop measurement refers to sending probe packets to propagate along a simple loop of a graph. Since the source and destination are the same, its observation (RTT is a special case), the difference between the receive timestamp and the transmit timestamp, exactly equals the delay of the loop and is not affected by clock offset. Assume that $b_{loop:ijki}$ is the observation from the loop measurement on loop $i \to j \to k \to i$ of a host network (note $b_{loop:iji} = \mathrm{RTT}_{ij}$). It follows that

$$x_{ij} + x_{jk} + x_{ki} = b_{loop:ijki}.$$

The above equation can be viewed as an equation of $\boldsymbol{x}$. If $l$ loops are measured, we will obtain a system of linear equations of $\boldsymbol{x}$, formulated as

$$A_{loops}\boldsymbol{x} = \boldsymbol{b}_{loops}. \tag{3.1}$$

Here $A_{loops} \in \mathbb{R}^{l\times m}$ whose entries are 1 and 0, and $\boldsymbol{b}_{loops} \in \mathbb{R}^{l}$. The above system is underdetermined as the following theorem states.

**Theorem 3.1** (cf. Gurewitz & Sidi [17]). *The maximal number of independent equations obtained by measuring loop delays in an $N$-host connected network is less than the number of variables by $(N-1)$.*

Therefore, it is impossible to solve delays directly from the system of equations (3.1).

In addition to the equations, there are some substantial non-negative constraints for OWDs

$$\boldsymbol{x} \geq 0. \tag{3.2}$$

It is natural because OWDs measure the time taken to send a data packet from one node to another, and therefore are always a nonnegative quantity.

As an example, the network with three hosts on the right of Fig. 1 has a total of five simple loops. Thus, the following constraints for OWD hold:

$$x_{12} + x_{21} = b_{loop:121}, \tag{3.3a}$$
$$x_{13} + x_{31} = b_{loop:131}, \tag{3.3b}$$
$$x_{23} + x_{32} = b_{loop:232}, \tag{3.3c}$$
$$x_{12} + x_{23} + x_{31} = b_{loop:1231}, \tag{3.3d}$$
$$x_{13} + x_{32} + x_{21} = b_{loop:1321}, \tag{3.3e}$$
$$x_{ij} > 0, \quad i,j = 1,2,3, \quad i \neq j. \tag{3.3f}$$

It is clear that Eq. (3.3e) is redundant, as it can be derived by (3.3a)+(3.3b)+(3.3c)-(3.3d). In other words, the probing on loop $1 \to 3 \to 2 \to 1$ is unnecessary.

In the rest of the paper, we may assume $A_{loops}$ has full row rank — i.e. $l = m-(N-1) = (N-1)^2$. In this sense, only measurements need to be made on specific $l$ loops instead of all. In fact, these $l$ loops are easy to determine. Apart from the obvious $m/2$ two-node loops that are associated with the RTTs, a fundamental cycle basis of the undirected complete graph reduced from the host network provides the remaining $m/2-(N-1)$ loops. To find them, one needs first to calculate a spanning tree. Then each edge outside the tree and the corresponding two paths from the root to the two ends of the edge, form a loop that we require. In the context of the complete graph, they each contain only three different nodes. As a result, $A_{loops}$ has either only two or three elements of 1 in a row, thus sparse.

### 3.2. RTT weighting

Equality constraints provided by loop measurements and non-negativity determine a feasible region for delays. We consider designing a reasonable optimization objective to help select the solution from infinite candidates.

As the forerunner, Gurewitz *et al*. [15] present the LE model

$$\min_{\boldsymbol{x}\in\Omega} \quad \sum_{i\neq j} |x_{ij}-x_{ji}|^2, \quad \Omega = \left\{A_{loops}\boldsymbol{x} = \boldsymbol{b}_{loops},\ \boldsymbol{x} \geq \boldsymbol{0}\right\}$$

for slight asymmetrical scenarios, which aims to minimize the total asymmetry to emphasize the symmetric nature. Inspired by this, for modern WAN with abundant asymmetry this paper concerns, we propose the LREW model

$$\min_{\boldsymbol{x}\in\Omega} \quad \sum_{i\neq j} \frac{|x_{ij}-x_{ji}|^2}{\mathrm{RTT}_{ij}^{\alpha}}, \quad \Omega = \left\{A_{loops}\boldsymbol{x} = \boldsymbol{b}_{loops},\ \boldsymbol{x} \geq \boldsymbol{0}\right\}.$$

Here $\alpha > 0$ is a hyperparameter. In comparison to LE, LERW assigns a weight inversely proportional to the $\alpha$ power of RTT for each delay asymmetry of the network so that pairs of node pairs with large RTTs have significant delay asymmetries. We emphasize its positive because it follows the relationship between delay asymmetry and RTT observed by Pathak *et al*. [23]. Their results on the PlanetLab testbed [8] show that the deviation of the forward delay from one-half of the RTT increases with RTT — i.e. delay asymmetry is positively correlated with RTT. In fact, this conclusion fits with intuition. In WANs, the communication between hosts is bridged by a number of routers. A large RTT is often associated with a multi-hop routing path, in which case a great deal of equal-cost paths exist. Selected from these paths under a certain stochastic strategy, forward and reverse paths differ more, at least in terms of probability. Moreover, multi-hop means that the packet passes through more routers to be forwarded before being received. It occurs likely that there is much congestion in one direction, while less congested relatively in the others, which also contributes to asymmetry.

The rationality of LERW can be demonstrated further theoretically, to a certain extent. Suppose $\boldsymbol{x}^*$ is the true delay and $\boldsymbol{x}$ is any estimate satisfying $\boldsymbol{x} \in \Omega$.

Let

$$I_1=\sum_{i\neq j}\frac{|x_{ij}-x_{ij}^*|^2}{\mathrm{RTT}_{ij}^{\alpha}},\quad I_2=\sum_{i\neq j}\frac{|x_{ij}-x_{ji}|^2}{\mathrm{RTT}_{ij}^{\alpha}},\quad I_3=\sum_{i\neq j}\frac{|x_{ij}^*-x_{ji}^*|^2}{\mathrm{RTT}_{ij}^{\alpha}}.$$

We give the following theorem.

**Theorem 3.2.**

$$2I_1\leq I_2+I_3.$$

*Proof.* Considering $\boldsymbol{x},\boldsymbol{x}^*\in\Omega$, for all $(i,j)$ we obtain $x_{ij}+x_{ji}=x_{ij}^*+x_{ji}^*=\mathrm{RTT}_{ij}$. It is straightforward to show that

$$\begin{aligned}\frac{(x_{ij}-x_{ij}^*)^2+(x_{ji}-x_{ji}^*)^2}{\mathrm{RTT}_{ij}^{\alpha}}&=\frac{(x_{ij}-x_{ji})^2}{\mathrm{RTT}_{ij}^{\alpha}}+\frac{(x_{ij}^*-x_{ji}^*)^2}{\mathrm{RTT}_{ij}^{\alpha}}+\frac{2(x_{ij}-x_{ij}^*)(x_{ji}-x_{ji}^*)}{\mathrm{RTT}_{ij}^{\alpha}}\\&\leq\frac{(x_{ij}-x_{ji})^2}{\mathrm{RTT}_{ij}^{\alpha}}+\frac{(x_{ij}^*-x_{ji}^*)^2}{\mathrm{RTT}_{ij}^{\alpha}}+\frac{(x_{ij}-x_{ji}^*+x_{ji}-x_{ij}^*)^2}{\mathrm{RTT}_{ij}^{\alpha}}\\&=\frac{(x_{ij}-x_{ji})^2}{\mathrm{RTT}_{ij}^{\alpha}}+\frac{(x_{ij}^*-x_{ji}^*)^2}{\mathrm{RTT}_{ij}^{\alpha}}.\end{aligned}$$

Accumulating the inequality for all $(i,j)$ gives that

$$2I_1\leq I_2+I_3.$$

The proof is complete. □

In view of the unknown $\boldsymbol{x}^*$, $I_1$ is an ideal but unrealistic objective function. Theorem 3.2 indicates that solving $I_2$ as an alternative of $I_1$ is practical since $I_3$ is an invariant.

## 4. SGS-ADMM Algorithm

LERW is a convex quadratic programming problem with linear constraints

$$\begin{aligned}\min_{\boldsymbol{x}}\quad&\frac{1}{2}\boldsymbol{x}^TQ\boldsymbol{x}\\\text{s.t.}\quad&A\boldsymbol{x}=\boldsymbol{b},\quad\boldsymbol{x}\geq\boldsymbol{0},\end{aligned}\tag{4.1}$$

where $Q$ is a positive semidefinite and tridiagonal matrix, and $A$ is highly sparse. A general algorithm to solve it is the interior point method (IPM) [14]. However, because of the prohibitive complexity and high memory consumption, IPM does not meet the requirement of coping with large-scale scenarios†. Instead, an alternating direction method of multipliers algorithm is welcomed [5–7, 18–20].

In this part, we propose an efficient symmetric Gauss-Seidel alternating direction method of multipliers (SGS-ADMM) algorithm [20] to solve model (4.1), which fully exploits

†In the follow-up experiments, we find that the expansion of network size can improve the mode performance.

the special structure and sparsity of $Q$ and $A$ to accelerate the iteration. Its global convergence is guaranteed. The tridiagonal matrix $Q$ takes the form

$$\begin{bmatrix} 1/\mathrm{RTT}_{i_1,j_1}^{\alpha} & -1/\mathrm{RTT}_{i_1,j_1}^{\alpha} & 0 & 0 & \cdots \\ -1/\mathrm{RTT}_{i_1,j_1}^{\alpha} & 1/\mathrm{RTT}_{i_1,j_1}^{\alpha} & 0 & 0 & \cdots \\ 0 & 0 & 1/\mathrm{RTT}_{i_3,j_3}^{\alpha} & -1/\mathrm{RTT}_{i_3,j_3}^{\alpha} & \cdots \\ 0 & 0 & -1/\mathrm{RTT}_{i_3,j_3}^{\alpha} & 1/\mathrm{RTT}_{i_3,j_3}^{\alpha} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}.$$

We decompose it as $Q = H^T H = H^2$, $H = \hat{Q}/\sqrt{2}$ where $\hat{Q}$ has the same form as $Q$ except for replacing $\alpha$ with $\alpha/2$. Then we can reformulate model (4.1) as

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \quad & \frac{1}{2}\|\boldsymbol{y}\|^2 + \delta_{\mathbb{R}_+}(\boldsymbol{z}) \\ \text{s.t.} \quad & H\boldsymbol{x} = \boldsymbol{y}, \quad A\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} = \boldsymbol{z}. \end{aligned} \tag{4.2}$$

The indicator function $\delta_{\mathbb{R}_+}$ is defined as

$$\delta_{\mathbb{R}_+}(\boldsymbol{z}) = \begin{cases} 0, & \boldsymbol{z} \geq \boldsymbol{0}, \\ +\infty, & \boldsymbol{z} < \boldsymbol{0}. \end{cases}$$

The dual of model (4.2) is given by

$$\max_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}} \min_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z};\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}),$$

where the Lagrangian function

$$\begin{aligned} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z};\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}) &= \frac{1}{2}\|\boldsymbol{y}\|^2 + \delta_{\mathbb{R}_+}(\boldsymbol{z}) - \langle \boldsymbol{u}, H\boldsymbol{x} - \boldsymbol{y}\rangle - \langle \boldsymbol{v}, A\boldsymbol{x} - \boldsymbol{b}\rangle - \langle \boldsymbol{w}, \boldsymbol{x} - \boldsymbol{z}\rangle \\ &= \frac{1}{2}\|\boldsymbol{y}\|^2 - \langle H^T\boldsymbol{u} + A^T\boldsymbol{v} + \boldsymbol{w}, \boldsymbol{x}\rangle + \big(\delta_{\mathbb{R}_+}(\boldsymbol{z}) + \langle \boldsymbol{w}, \boldsymbol{z}\rangle\big) + \langle \boldsymbol{u}, \boldsymbol{y}\rangle + \langle \boldsymbol{v}, \boldsymbol{b}\rangle. \end{aligned}$$

Let $(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*, \boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*)$ be the optimum solution of the dual model. From $\partial L/\partial \boldsymbol{x} = \boldsymbol{0}$ and $\partial L/\partial \boldsymbol{y} = \boldsymbol{0}$, we obtain that

$$H^T\boldsymbol{u}^* + A^T\boldsymbol{v}^* + \boldsymbol{w}^* = \boldsymbol{0}, \quad \boldsymbol{y}^* + \boldsymbol{u}^* = \boldsymbol{0}.$$

Thus, the dual model actually reads

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{u}\|^2 - \langle \boldsymbol{v}, \boldsymbol{b}\rangle \\ \text{s.t.} \quad & H^T\boldsymbol{u} + A^T\boldsymbol{v} + \boldsymbol{w} = \boldsymbol{0}, \quad \boldsymbol{w} \geq \boldsymbol{0}. \end{aligned} \tag{4.3}$$

Its augmented Lagrangian function is

$$L_\sigma(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w};\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{u}\|^2 - \langle \boldsymbol{v}, \boldsymbol{b}\rangle + \langle \boldsymbol{x}, H^T\boldsymbol{u} + A^T\boldsymbol{v} + \boldsymbol{w}\rangle + \frac{\sigma}{2}\|H^T\boldsymbol{u} + A^T\boldsymbol{v} + \boldsymbol{w}\|^2.$$

We consider solving model (4.3) instead of (4.1) by using Algorithm 4.1.

**Algorithm 4.1** SGS-ADMM Algorithm

**Input:** $\gamma, \sigma$, initial $\boldsymbol{u}^{(0)}, \boldsymbol{v}^{(0)}, \boldsymbol{w}^{(0)}, \boldsymbol{x}^{(0)}$.
**Output:** $\boldsymbol{x}^{(k)}$.

k=0.
**repeat**
  $\boldsymbol{u}^{(k+\frac{1}{2})} = \arg\min_{\boldsymbol{u}} \quad \frac{1}{2}\|\boldsymbol{u}\|^2 + \langle H\boldsymbol{x}^{(k)}, \boldsymbol{u}\rangle + \frac{\sigma}{2}\|H^T\boldsymbol{u} + A^T\boldsymbol{v}^{(k)} + \boldsymbol{w}^{(k)}\|^2;$
  $\boldsymbol{v}^{(k+\frac{1}{2})} = \arg\min_{\boldsymbol{v}} \quad \langle A\boldsymbol{x}^{(k)} - \boldsymbol{b}, \boldsymbol{v}\rangle + \frac{\sigma}{2}\|H^T\boldsymbol{u}^{(k+\frac{1}{2})} + A^T\boldsymbol{v} + \boldsymbol{w}^{(k)}\|^2;$
  $\boldsymbol{w}^{(k+1)} = \arg\min_{\boldsymbol{w}\geq 0} \quad \langle \boldsymbol{x}^{(k)}, \boldsymbol{w}\rangle + \frac{\sigma}{2}\|H^T\boldsymbol{u}^{(k+\frac{1}{2})} + A^T\boldsymbol{v}^{(k+\frac{1}{2})} + \boldsymbol{w}\|_F^2;$
  $\boldsymbol{v}^{(k+1)} = \arg\min_{\boldsymbol{v}} \quad \langle A\boldsymbol{x}^{(k)} - \boldsymbol{b}, \boldsymbol{v}\rangle + \frac{\sigma}{2}\|H^T\boldsymbol{u}^{(k+\frac{1}{2})} + A^T\boldsymbol{v} + \boldsymbol{w}^{(k+1)}\|^2;$
  $\boldsymbol{u}^{(k+1)} = \arg\min_{\boldsymbol{u}} \quad \frac{1}{2}\|\boldsymbol{u}\|^2 + \langle H\boldsymbol{x}^{(k)}, \boldsymbol{u}\rangle + \frac{\sigma}{2}\|H^T\boldsymbol{u} + A^T\boldsymbol{v}^{(k+1)} + \boldsymbol{w}^{(k+1)}\|^2;$
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \gamma\sigma(H^T\boldsymbol{u}^{(k+1)} + A^T\boldsymbol{v}^{(k+1)} + \boldsymbol{w}^{(k+1)});$
  $k = k + 1.$
**until** convergence

In each iteration, a series of subproblems concerning only one variable are solved. In this way, the optimization problem is intensely simplified. Whereafter, we give the closed forms of the optimums related to all the subproblems.

The subproblem of $u$ is equivalent to the following linear equations:

$$\big(I + \sigma HH^T\big)\boldsymbol{u} = -H\big(\sigma(A^T\boldsymbol{v} + \boldsymbol{w}) + \boldsymbol{x}\big).$$

Given $H$ is tridiagonal, it can be efficiently solved by the well-known Thomas algorithm [9]. The subproblem of $v$ is also equivalent to a system of linear equations

$$\sigma AA^T\boldsymbol{v} = \boldsymbol{b} - A\big(\boldsymbol{x} + \sigma(H^T\boldsymbol{u} + \boldsymbol{w})\big).$$

The potential large scale, e.g., the number of rows and columns of $A$ attains more than ten thousand, makes it impractical to solve with direct methods. By exploiting the high sparsity and the full row rankness of $A$, we instead pursue an effective strategy that is to implement the inexact conjugate gradient algorithm [3]. It is rather fast and reliable when the coefficient matrix is symmetric positive definite. For the subproblem of $\boldsymbol{w}^*$, with simple calculation, we have that

$$\boldsymbol{w} := \max\left\{\boldsymbol{0}, -\left(H^T\boldsymbol{u} + A^T\boldsymbol{v} + \frac{\boldsymbol{x}}{\sigma}\right)\right\}.$$

The global convergence of the proposed SGS-ADMM is provided as follows [20].

**Theorem 4.1.** *Suppose the iteration sequence of the proposed SGS-ADMM is* $\{(\boldsymbol{u}^{(k)}, \boldsymbol{v}^{(k)}, \boldsymbol{w}^{(k)}, \boldsymbol{x}^{(k)})\}$, $k = 1, 2, \ldots$. *Then the sequence converges. Denote the limiting point as* $(\boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*, \boldsymbol{x}^*)$. *Moreover,* $\boldsymbol{x}^*$ *is the optimum of the primal model* (4.1) *and* $\boldsymbol{u}^*, \boldsymbol{v}^*$ *and* $\boldsymbol{w}^*$ *are the optimums of the dual model.*

## 5. Numerical Experiments

In this section, we present a series of experiments to demonstrate the validity of our model and the advantage of the SGS-ADMM algorithm. Given the unknown ground truth of delays in the real network, our experiments rely on highly realistic simulation. We observe the performance of LERW with various $\alpha$ and compare them with LE and PTP. In addition, the computation speed and memory consumption are evaluated on SGS-ADMM and the interior point method. All the experiments are run on Python 3.7, deployed on a PC with 16G RAM and Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz.

Here we need first to establish the communication of the host network, mainly how to set routers and generate actual OWDs. In our simulations, we adopt mesh-distributed routers with size $20 \times 20$ for the underlying communication[‡] and set $N$ hosts randomly within the plane, shown in Fig. 2. Each host is directly connected to the unique nearest router. Routing paths in both directions between them are selected with equal probability from ones with minimum hop count to simulate different transmission paths frequently occurring in WANs [1, 25]. The OWD between any two hosts equals the sum of the link delays in their router path.

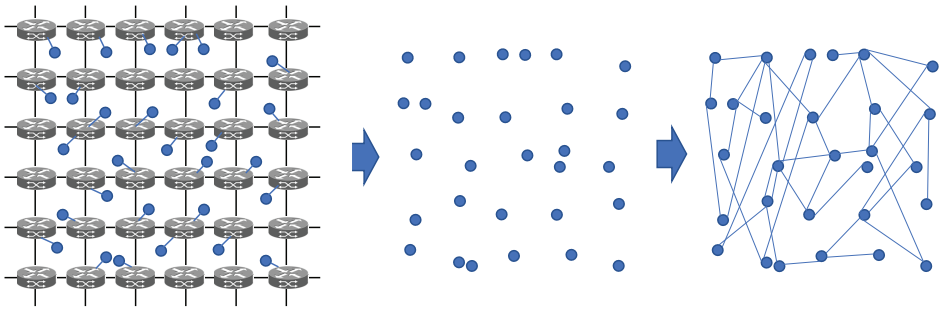


Figure 2: Hosts communicate via grid-distributed routers, then a logically complete connected host network is formed (only part of the edges are drawn). Here the gray disks are routers, and the blue nodes are hosts.

Let $d_{ij} = a_{ij} + c_{ij}$ be the link delay from router $i$ to router $j$. The basic component $a_{ij}$ characterizes the diversity of link lengths, while the small increment $c_{ij}$ creates the asymmetry between forward delay and reverse delay. We ensure $a_{ij} = a_{ji}$ and then sample $a_{ij}$ from a uniform distribution $U(1ms, 1.5ms)$. In this way, the delays between the most distant routers are about $40 \sim 50$ms, close to China's Internet state. There are two ways to generate $c_{ij}$ viz.

(1) **Unidirectional communication.** Terms $c_{ij}$ are sampled from $U(0ms, 0.01ms)$. Some networks are equipped with unidirectional fiber pairs to connect routers for a low cost. When laying fibers, it is hard to ensure the same lengths in both directions, giving rise to random delay asymmetry.

[‡]It is realistic because routers in real-world the routers are evenly distributed over an area and tend to be directly connected to the nearest ones.

(2) **Bidirectional communication.** Set $c_{ij} = \theta \cdot a_{ij},\ c_{ji} = 0$ or $c_{ij} = 0,\ c_{ji} = \theta \cdot a_{ji}$, where $0 < \theta < 1$. For networks using bidirectional single fiber, different wavelengths of light used in opposite directions cause distinct propagation distances. The asymmetry $|c_{ij} - c_{ji}|$ is proportional to the distance, and the coefficient $\theta$ depends on the specific fiber material [10].

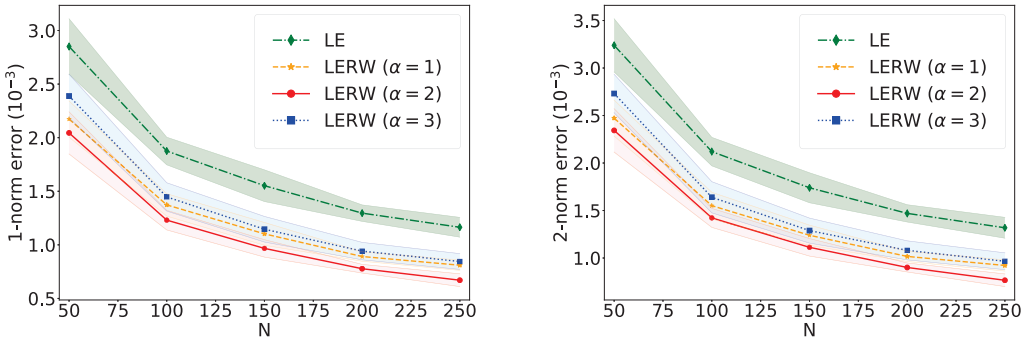
## 5.1. Unidirectional communication

In this part, we investigate the performance of LERW in unidirectional communication. We set the number of hosts $N$ to $50, 100, 150, 200$ in turn and repeat the experiments 10 times with various random seeds for each $N$. Feasible regions $\Omega$ of models are derived by finding fundamental loops of the graphs and accumulating the real OWDs along them [24]. The SGS-ADMM algorithm proposed is used to solve all models, with $\sigma$ selected from $\{10, 100, 1000, 1000\}$, $\gamma = 1.618$.

Table 1 compares the average relative errors with respect to 1-norm and 2-norm. Fig. 3 visualizes the comparison and additionally shows the standard deviations. As expected, in all cases, LERW and LE are overwhelmingly superior to PTP, and LERW outperforms LE significantly. Beyond the inherent superiority of the loop estimation, this empirically demonstrates the helpfulness of the RTT weighting. Interestingly, we find that LERW works best at $\alpha = 2$. It seems natural, consistent with the order of $|x_{ij} - x_{ji}|$. On the other hand, the error of LERW is observed to decrease as $N$ increases regardless of the value of $\alpha$, which means that large-scale networks facilitate the effect of the model. It brings important enlightenment that we can artificially introduce additional hosts to form a larger network for the purpose of improving the accuracy of delay estimation. Our fast algorithm SGS-ADMM is just designed for this.

Table 1: Relative error comparison in unidirectional communication. 1-norm: $\|\boldsymbol{x} - \boldsymbol{x}^*\|_1 / \|\boldsymbol{x}^*\|_1$, 2-norm: $\|\boldsymbol{x} - \boldsymbol{x}^*\|_2 / \|\boldsymbol{x}^*\|_2$. The best results are in bold.

| N | LERW | | | LE | PTP | Criterion |
|---|---|---|---|---|---|---|
| | $\alpha = 1$ | $\boldsymbol{\alpha = 2}$ | $\alpha = 3$ | | | |
| 50 | 2.17e-3 | **2.04e-3** | 2.39e-3 | 2.85e-3 | 13.0e-3 | 1-norm |
| 100 | 1.37e-3 | **1.23e-3** | 1.45e-3 | 1.88e-3 | 12.9e-3 | |
| 150 | 1.10e-3 | **9.67e-4** | 1.15e-3 | 1.55e-3 | 12.9e-3 | |
| 200 | 8.91e-4 | **7.77e-4** | 9.42e-4 | 1.30e-3 | 13.0e-3 | |
| 250 | 8.10e-4 | **6.70e-4** | 8.45e-4 | 1.16e-3 | 12.9e-3 | |
| 50 | 2.47e-3 | **2.34e-3** | 2.73e-3 | 3.24e-3 | 16.3e-3 | 2-norm |
| 100 | 1.55e-3 | **1.42e-3** | 1.64e-3 | 2.12e-3 | 16.3e-3 | |
| 150 | 1.24e-3 | **1.11e-3** | 1.29e-3 | 1.74e-3 | 16.2e-3 | |
| 200 | 1.02e-3 | **9.00e-4** | 1.08e-3 | 1.47e-3 | 16.3e-3 | |
| 250 | 9.22e-4 | **7.66e-4** | 9.66e-4 | 1.32e-3 | 16.3e-3 | |




a) 1-norm errors b) 2-norm errors

Figure 3: Visualization of relative error comparison in unidirectional communication. The thicker lines in the middle show the means on 10 repetitions. The widths of the light-colored regions are equal to twice the corresponding standard deviations.

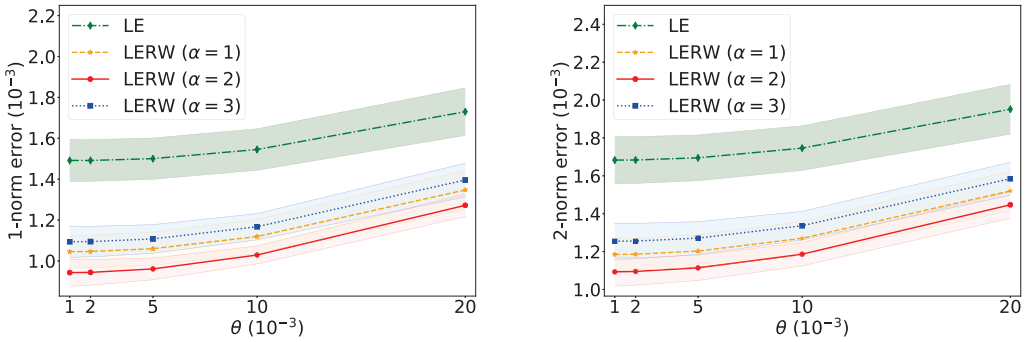
## 5.2. Bidirectional communication

In bidirectional communication, we obtain similar results for a given $\theta$ to those in the previous subsection. For brevity, we here will not report them. In order to study the robustness of LERW under different optical fiber materials, we observe its performance on $\theta = 0.001, 0.002, 0.005, 0.01, 0.02$ and compare it with LE and PTP, with the number of hosts $N = 150$. Each experiment is also repeated 10 times.

Table 2 and Fig. 4 show the errors with two evaluation criteria as did in Section 5.1. LERW beats the others again and still works best at $\alpha = 2$, with errors about two-thirds of those of LE. As $\theta$ increases, the accuracies of all methods decrease. Even so, the performance of LERW at $\theta = 0.02$ keeps an advantage over the others at $\theta = 0.001$. It shows that LERW is in a position to deal with bidirectional communication well.

Table 2: Relative error comparison in bidirectional communication. The best results are in bold.

| $\theta$ | LERW | | | LE | PTP | Criterion |
|---|---|---|---|---|---|---|
| | $\alpha = 1$ | $\boldsymbol{\alpha = 2}$ | $\alpha = 3$ | | | |
| 0.001 | 1.05e-3 | **9.43e-4** | 1.09e-3 | 1.49e-3 | 13.0e-3 | |
| 0.002 | 1.05e-3 | **9.44e-4** | 1.09e-3 | 1.49e-3 | 13.0e-3 | |
| 0.005 | 1.06e-3 | **9.61e-4** | 1.11e-3 | 1.50e-3 | 13.1e-3 | 1-norm |
| 0.01 | 1.12e-3 | **1.03e-3** | 1.17e-3 | 1.55e-3 | 13.1e-3 | |
| 0.02 | 1.35e-3 | **1.27e-3** | 1.40e-3 | 1.73e-3 | 13.3e-3 | |
| 0.001 | 1.18e-3 | **1.09e-3** | 1.26e-3 | 1.68e-3 | 16.4e-3 | |
| 0.002 | 1.19e-3 | **1.09e-3** | 1.26e-3 | 1.68e-3 | 16.4e-3 | |
| 0.005 | 1.20e-3 | **1.11e-3** | 1.27e-3 | 1.70e-3 | 16.4e-3 | 2-norm |
| 0.01 | 1.27e-3 | **1.19e-3** | 1.34e-3 | 1.75e-3 | 16.4e-3 | |
| 0.02 | 1.52e-3 | **1.45e-3** | 1.59e-3 | 1.95e-3 | 16.5e-3 | |




a) 1-norm errors b) 2-norm errors

Figure 4: Visualization of relative error comparison in bidirectional communication. The thicker lines in the middle show the means on 10 repetitions. The widths of the light-colored regions are equal to twice the corresponding standard deviations.

## 5.3. Comparison of SGS-ADMM and IPM

Our SGS-ADMM algorithm and the traditional interior point method (IPM) can both be utilized for solving LERW models. In this part, we compare their efficiency on LERW with $\alpha = 2$. The quadratic programming solver in CVXOPT[§] is applied to implement IPM. Table 3 shows the CPU time of the algorithms on two types of communication and the difference in their computation results. We can see that the two algorithms bring about close results, while SGS-ADMM has an outstanding speed advantage. SGS-ADMM is about two orders of magnitude faster than IPM at $N = 200$ and this gap in performance widens further with larger values of $N$.

Table 3: Comparison of SGS-ADMM and IPM in running time and results on LERW with $\alpha = 2$. A dash means it cannot be calculated because of memory error.

| Type | $N$ | $m$ | $l$ | CPU time (sec) | | speed-up ratio | $\frac{\|x_{\text{ADMM}} - x_{\text{IPM}}\|_2}{\|x^*\|_2}$ |
|---|---|---|---|---|---|---|---|
| | | | | ADMM | IPM | | |
| Unidirectional | 50 | 2450 | 2401 | 0.52 | 0.87 | 1.67 | 1.22e-4 |
| | 100 | 9900 | 9801 | 6.75 | 22.5 | 3.33 | 6.95e-5 |
| | 150 | 22350 | 22201 | 9.86 | 300 | 30.4 | 6.04e-5 |
| | 200 | 39800 | 39601 | 16.48 | 2438 | 148 | 6.67e-5 |
| | 250 | 62250 | 62001 | 50.2 | - | - | - |
| Bidirectional ($\theta = 0.001$) | 50 | 2450 | 2401 | 0.34 | 0.78 | 2.29 | 1.20e-4 |
| | 100 | 9900 | 9801 | 6.46 | 14.5 | 2.24 | 6.24e-5 |
| | 150 | 22350 | 22201 | 9.28 | 338 | 26.4 | 6.43e-5 |
| | 200 | 39800 | 39601 | 14.1 | 2448 | 173 | 7.79e-5 |
| | 250 | 62250 | 62001 | 69.7 | - | - | - |

[§]https://cvxopt.org/

Fig. 5 presents the memory usage of the two algorithms. Communication type has no impact on the size of the LERW model and thus on the memory consumption, meaning the two subgraphs are nearly identical. It is evident from Fig. 5 that SGS-ADMM also performs remarkably well in terms of memory consumption. Specifically, it takes less than 300MB at $N = 250$, while the usage of IPM has reached about 6GB at a smaller scale $N = 200$. The usage of IPM at $N = 250$ is not given because it is beyond the memory limit of approximately 10.5GB (system processes and some necessary programs occupy about 35% of the machine's memory).

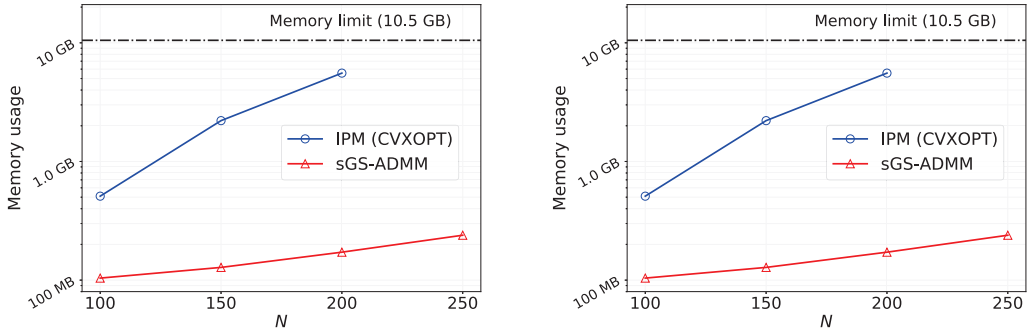



Figure 5: The memory usage of SGS-ADMM and IPM. Left: Unidirectional communication. Right: Bidirectional communication with $\theta = 0.001$.

## 6. Conclusion

Considering the positive correlation between RTT and asymmetry in WANs, we proposed the LERW model. RTT is used to add a weight to each term in the summation so that pairs of hosts with large RTTs can dominate greater asymmetries. We proved the rationality of the model in theory and verified its excellent effect in experiments. Moreover, we found the model performs best at $\alpha = 2$. Hence, we recommend using $\alpha = 2$ in the engineering. In order to adapt to the large scale, we also designed a solving algorithm SGS-ADMM, which is significantly superior to the general interior point method in both memory consumption and computation speed.

## Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (Grant No. 12171271).

## References

[1] M. Allman and V. Paxson, *On estimating end-to-end network path properties*, ACM SIGCOMM Comput. Commun. Rev. **29**, 263–274 (1999).

[2] G. Almes, S. Kalidindi and M. Zekauskas, *A one-way delay metric for IPPM*, IETF RFC 2679 (1999).
[3] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM (1994).
[4] S.E. Brak, M. Bouhorma, M.E. Brak and A.A. Boudhir, *VoIP applications over manet: Codec performance enhancement by tuning routing protocol parameters*, J. Theore. Appl. Inf. Technol. **50**, 68–75 (2013).
[5] X. Chang, J. Bai, D. Song and S. Liu, *Linearized symmetric multi-block ADMM with indefinite proximal regularization and optimal proximal parameter*, Calcolo **57**, 38 (2020).
[6] C. Chen, M. Li, X. Liu and Y. Ye, *Extended ADMM and BCD for nonseparable convex minimization models with quadratic coupling terms: Convergence analysis and insights*, Math. Program. **173**, 37–77 (2019).
[7] L. Chen and D. Sun, *An efficient inexact symmetric Gauss-Seidel based majorized ADMM for high-dimensional convex composite conic programming*, Math. Program. **161**, 237–270 (2017).
[8] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak and M. Bowman, *PlanetLab: An overlay testbed for broad-coverage services*, ACM SIGCOMM Comput. Commun. Rev. **33**, 3–12 (2003).
[9] B.N. Datta, *Numerical Linear Algebra and Applications*, SIAM (2010).
[10] E.F. Dierikx, A.E. Wallin, T. Fordell, J. Myyry, P. Koponen, M. Merimaa, T.J. Pinkert, J.C. Koelemeij, H.Z. Peek and R. Smets, *White rabbit precision time protocol on long-distance fiber links*, IEEE Trans. Ultrason. Ferroelectr. Freq. Control **63**, 945–952 (2016).
[11] J.C. Eidson, M. Fischer and J. White, *IEEE-1588™ standard for a precision clock synchronization protocol for networked measurement and control systems*, in: *Proceedings of the 34th Annual Precise Time and Time Interval Systems and Applications Meeting*, pp. 243–254, ION (2002).
[12] N.M. Freris, S.R. Graham and P.R. Kumar, *Fundamental limits on synchronizing clocks over networks*, IEEE Trans. Automat. Control **56**, 1352–1364 (2010).
[13] Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum and A. Vahdat, *Exploiting a natural network effect for scalable, fine-grained clock* in: *15th USENIX Symposium on Networked Systems Design and Implementation*, pp. 81–94, USENIX (2018).
[14] E.M. Gertz and S.J. Wright, *Object-oriented software for quadratic programming*, ACM Trans. Math. Software **29**, 58–81 (2003).
[15] O. Gurewitz, I. Cidon and M. Sidi, *Network time synchronization using clock offset optimization*, in: *11th IEEE International Conference on Network Protocols*, pp. 212–221, IEEE (2003).
[16] O. Gurewitz, I. Cidon and M. Sidi, *One-way delay estimation using network-wide measurements*, IEEE Trans. Inf. Theory **52**, 2710–2724 (2006).
[17] O. Gurewitz and M. Sidi, *Estimating one-way delays from cyclic-path delay measurements*, in: *20th Annual Joint Conference of the IEEE-Computer-Society*, pp. 1038–1044, IEEE (2001).
[18] D. Han, L. Zhang and D. Sun, *Linear rate convergence of the alternating direction method of multipliers for convex composite programming*, Math. Oper. Res. **43**, 622–637 (2017).
[19] B. He and X. Yuan, *A class of ADMM-based algorithms for three-block separable convex programming*, Comput. Optim. Appl. **70**, 791–826 (2018).
[20] X. Li, D. Sun and K.C. Toh, *A block symmetric Gauss-Seidel decomposition theorem for convex composite quadratic programming and its applications*, Math. Program. **175**, 395–418 (2019).
[21] D.L. Mills, *Internet time synchronization: The network time protocol*, IEEE Trans. common. **39**, 1482–1493 (1991).
[22] C.D. Murta, P.R. Torres Jr and P. Mohapatra, *Qrpp1-4: Characterizing quality of time and topology in a time synchronization network*, in: *IEEE Globecom 2006*, pp. 1–5, IEEE (2006).

[23] A. Pathak, H. Pucha, Y. Zhang, Y. Hu and Z. Mao, *A measurement study of internet delay asymmetry*, Lect. Notes. Artif. Intell. **4979**, 182–191 (2008).
[24] K. Paton, *An algorithm for finding a fundamental set of cycles of a graph*, Commun. ACM **12**, 514–518 (1969).
[25] V. Paxson, *End-to-end routing behavior in the internet*, IEEE-ACM Trans. Netw. **5**, 601–615 (1997).
[26] J. Postel and J. Reynolds, *File Transfer Protocol*, IETF RFC 959 (1985).
[27] L.D. Vito, S. Rapuano and L. Tomaciello, *One-way delay measurement: State of the art*, IEEE Trans. Instrum. Meas. **57**, 2742–2750 (2008).