Communication-efficient Clock Synchronization

Peng Fei, Zhen Chen, Zhiying Wang, Syed A. Jafar

Center for Pervasive Communications and Computing (CPCC)
University of California, Irvine, USA
{pfei1, zhenc4, zhiying, syed}@uci.edu

***Abstract*—The problem of clock synchronization is studied in an arbitrary network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}|$ server nodes and $|\mathcal{E}|$ edges. Every pair of adjacent servers has a time discrepancy (edge information) that is only known approximately to one or both of the two adjacent servers. A master node aims to coordinate the otherwise independent clocks of the servers by eliminating the loop-wise offset surplus in the network. The goal is to minimize the communication cost between server nodes and the master node. Optimal schemes are found for the two cases where each time discrepancy is known by 1) both adjacent servers, and 2) only one of the adjacent servers. Notably, the scheme for the first case is robust to a straggler (slow or failed server). An algorithm that outperforms the natural (uncoded) baseline is proposed for the general setting that is a mix of the two cases. Classes of such mixed setting are identified where the algorithm represents the optimal solution.**

I. INTRODUCTION

Clock synchronization is broadly used in distributed systems for many network and database applications [1], [2]. Performance requirements of such applications these days involve synchronization accuracy to within microsecond or even nanosecond level [3]–[6]. As a result, there are increasing demands on the speed and reliability of implementation of synchronization techniques. In particular, the communication overhead required for clock synchronization is one of the limiting factors that can significantly affect the accuracy of synchronization [3]. Motivated by this concern, in this work we explore ways to reduce the communication cost of clock synchronization.

The body of research on clock synchronization in distributed systems largely focuses on two aspects: two-clock synchronization, and multiple-clock synchronization for a whole graph consensus. Two-clock synchronization studies ways to reduce the one-way delay between two servers that may arise due to various factors such as path length, temperature, and fluctuations of switch times. Multiple-clock synchronization uses the information from the whole network to generate approximate global time [7] of the system [3], [8]–[11]. Our focus in this work is on the multiple-clock synchronization problem.

Among the latest multiple-clock synchronization algorithms, HUYGENS [3] enables delay-sensitive applications in datacenters by achieving an accuracy of tens to hundreds of nanoseconds. To obtain one-way delay, or clock time discrepancy, between a pair of connected servers, HUYGENS proposes a coded probe filter to purify the training data and uses support vector machines (SVM) [12] to estimate the synchronized time between two clocks. Then, it develops a loop-wise algorithm to eliminate the asymmetric mistakes between multiple loops.

To implement the HUYGENS algorithm, each server node exchanges time information with its neighbouring nodes, and then all server nodes send information to a master node who runs the synchronization algorithm. As the clocks drift constantly, it is essential to ensure timely communication between the servers and the master to guarantee the synchronization accuracy. In this work, we focus on optimizing, from a coding perspective, the information transfer that is required by the HUYGENS algorithm between the servers and the master node.

For our purpose, the HUYGENS algorithm may be abstracted as follows [3]: The network is described by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents server nodes, and $\mathcal{E}$ is the set of edges. Each edge is associated with a weight representing the approximate time discrepancy, also called the edge information. A node may know the edge information of its incident edges or a subset thereof. A master wants to collect all the edge information that is necessary to compute a particular linear function of such information.

In this context, we identify an opportunity to improve communication efficiency by coding, which leads to the following question: assuming each server can linearly code its edge information, what is the fundamental limit on the communication cost from the servers to the master, and how can this optimal cost be achieved?

The question falls under the broad umbrella of network computing problems where source nodes generate independent messages and a master node computes a target function of the messages [13]–[18]. In general, network computing considers arbitrary graphs and arbitrary target functions. The clock synchronization problem can be regarded as a special case of network computing that considers linear coding for the computation of a particular linear function over a particular graph. As such, the clock synchronization problem is comprised of $|\mathcal{E}|$ messages (corresponding to the $|\mathcal{E}|$ time discrepancies) that originate at distinct source nodes and are made available to $|\mathcal{V}|$ server nodes depending on the adjacency and knowledge structure, and a master node that wishes to compute a linear function (involving a pseudo-inverse) by downloading coded

Peng Fei and Zhen Chen made equal contributions.

information from the servers. The goal is to minimize the total amount of information downloaded by the master node.

Our contributions consist of three parts. First, for the case where each time discrepancy is known by both incident servers, we propose a coding scheme that achieves the optimal rate. Remarkably, the scheme can also tolerate a straggler (slow or failed server) at no extra cost. Second, for the case where each time discrepancy is known by only one of the incident servers, we show that the optimal solution is the trivial solution that entails sending all the time discrepancy information to the master. Third, we design a general algorithm for mixed scenarios where some edge discrepancies are known by both incident servers while others are known only by one of the incident servers. The algorithm outperforms the natural baseline of uncoded transmission and is shown to be optimal for some cases.

Notation: We use calligraphic characters to denote sets and bold characters to denote matrices. For a positive integer N, $[N]$ stands for the set $\{1, 2, \ldots, N\}$. For a set $\mathcal{S}$, $|\mathcal{S}|$ represents its cardinality.

II. Problem Statement

We first review the HUYGENS algorithm [3] for clock synchronization. Then, we introduce our communication model for the algorithm.

A. Clock Synchronization

HUYGENS [3] first implements the pair-wise synchronization algorithm between two neighboring server clocks. It then builds a synchronization algorithm that considers the whole network effect.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the server network with server nodes $\mathcal{V}$ and communication links (edges) $\mathcal{E}$. For convenience, every edge has a predefined direction. The edge from Node U to Node V is denoted as UV, for $U, V \in \mathcal{V}$.

In the first step of HUYGENS, the edge information, or the time discrepancy, is determined between every pair of adjacent nodes. We assume the edge information is only known by one or both of the two incident nodes. The edge information for Edge UV is denoted as l_{UV}. In Figure 1.a, for example, Servers A and B both believe that B's time is 20 time units earlier than A and the time discrepancy $l_{AB} = 20$ is only known by Node A and/or Node B. In Figure 1.a, the *original time discrepancy vector*

$$\mathbf{\Delta}^P = [l_{AB}, l_{BC}, l_{CA}, l_{BD}, l_{DA}]^T = [20, -15, 5, 25, -15]^T \tag{1}$$

corresponds to the time discrepancy on directed edges AB, BC, CA, BD, and DA.

However, notice that in the loop $A \to B \to C \to A$, Server C is 5 time units later than A and 15 time units later than B. Then B should be only 10 time units earlier than A. This means that there are 10 time units loop-wise offset surplus, due to the inaccuracy of the clock measurement.

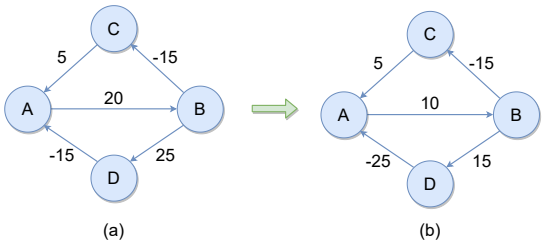

Fig. 1. HUYGENS algorithm. The figure is redrawn from [3].

The second step of HUYGENS is to eliminate the loop-wise offset surplus in the clock network. It calibrates the original discrepancy vector $\mathbf{\Delta}^P$ to the *final time discrepancy vector*

$$\mathbf{\Delta}^F = [10, -15, 5, 15, -25]^T \tag{2}$$

as shown in Figure 1.b. The transformation from $\mathbf{\Delta}^P$ to $\mathbf{\Delta}^F$ is explained below.

Remark. From the example above, we see that if the graph $\mathcal{G}$ has several connected components, then the loop-wise surplus can be eliminated for each component independently. Moreover, if some nodes are not included in any loops, they can be ignored during clock synchronization. Therefore, we assume $\mathcal{G}$ is connected and all nodes are in loops without loss of generality.

Given a server network graph $\mathcal{G}$, denote $\mathbf{A}$ as the *loop-composition matrix*. The columns of $\mathbf{A}$ are indexed by the edges. The rows are indexed by the largest set of linearly independent loops in $\mathcal{G}$. If an edge occurs in a loop, directly or reversed, the corresponding entry in $\mathbf{A}$ is 1 or -1; otherwise, the entry equals 0. Since the loops are all linearly independent, matrix $\mathbf{A}$ has full row rank. For a connected graph,

$$rank(A) = |\mathcal{E}| + |\mathcal{V}| - 1. \tag{3}$$

For example, in Figure 1, let the 5 columns of $\mathbf{A}$ be indexed by edges AB, BC, CA, BD, DA. The three loops $A \to B \to C \to A$, $A \to B \to D \to A$, and $A \to C \to B \to D \to A$ can be denoted respectively by three rows. The last row (loop) is dependent on the first two, and hence the loop-composition matrix is

$$\mathbf{A} = \begin{array}{c} \begin{matrix} AB & BC & CA & BD & DA \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}. \tag{4}$$

The vector $\mathbf{Y} = \mathbf{A}\mathbf{\Delta}^P$ gives the original loop-wise surplus in each independent loop of $\mathcal{G}$. In order to apply the loop-wise correction, we look for a vector $\mathbf{N}$ which also solves $\mathbf{Y} = \mathbf{A}\mathbf{N}$ and posit the correction to be $\mathbf{\Delta}^F = \mathbf{\Delta}^P - \mathbf{N}$. As a result, the final loop-wise surplus vector is $\mathbf{A}\mathbf{\Delta}^F = \mathbf{0}$. Now, $\mathbf{A}$ has full row rank. Further, since the number of linearly independent loops in $\mathcal{G}$ equals $|\mathcal{E}| - |\mathcal{V}| + 1$ which is less than $|\mathcal{E}|$, the equation $\mathbf{Y} = \mathbf{A}\mathbf{N}$ is under-determined and has multiple solutions. We look for the minimum-norm solution since this is most likely the best explanation of the errors in the loop-wise surpluses: $\min_{\mathbf{N}:\mathbf{Y}=\mathbf{A}\mathbf{N}} ||\mathbf{N}||_2$. The pseudo-inverse is well-known to give the minimum-norm solution [19]: $\mathbf{N} =$

$\mathbf{A}_{right}^{-1}\mathbf{Y} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\boldsymbol{\Delta}^P$, where the $\mathbf{A}_{right}^{-1}$ is the right pseudo-inverse of $\mathbf{A}$ [20]. HUYGENS can be concluded in the following formula:

$$\boldsymbol{\Delta}^F = \boldsymbol{\Delta}^P - \mathbf{N} = (\mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A})\boldsymbol{\Delta}^P = \mathbf{B}\boldsymbol{\Delta}^P, \quad (5)$$

where $\mathbf{I}$ is an $|\mathcal{E}| \times |\mathcal{E}|$ identity matrix and

$$\mathbf{B} = \mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}. \quad (6)$$

Note that $\mathbf{A}$, $\mathbf{I}$, and hence $\mathbf{B}$ are determined by the known graph structure. It can be easily checked that the final time discrepancy in Figure 1.b satisfies (5).

B. Communication Model

As proposed by HUYGENS, the calculation of Equation (5) is run on a master [3]. Assume each edge information is known by one or both incident server nodes, and each server node can linearly code its known edge information. We aim to minimize the *communication cost* (CC), defined as the number of symbols communicated from all the server nodes to the master node, such that (5) can be calculated.

Trivial scheme. We can see that HUYGENS relies on $\boldsymbol{\Delta}^P$ which is gathered from every pair of neighboring nodes. To obtain $\boldsymbol{\Delta}^F$, a trivial solution is for each edge, one of the incident nodes should send the information to the master. In this scenario, the communication cost is $CC = |\mathcal{E}|$ symbols.

To motivate the general communication scheme, let us consider the matrix $\mathbf{B}$ in Figure 1's example,

$$\mathbf{B} = \begin{bmatrix} 0.5 & -0.25 & -0.25 & -0.25 & -0.25 \\ -0.25 & 0.625 & -0.375 & 0.125 & 0.125 \\ -0.25 & -0.375 & 0.625 & 0.125 & 0.125 \\ -0.25 & 0.125 & 0.125 & 0.625 & -0.375 \\ -0.25 & 0.125 & 0.125 & -0.375 & 0.625 \end{bmatrix}. \quad (7)$$

In this case, $rank(\mathbf{B}) = 3$. It means that the optimal communication cost is not less than 3 symbols. If every clock knows all the time discrepancies, it is obvious that the communication cost $CC = 3$ because one node can calculate 3 (row) bases of $\mathbf{B}$, denoted by $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$, and send to the master $\mathbf{B}_1\boldsymbol{\Delta}^P, \mathbf{B}_2\boldsymbol{\Delta}^P, \mathbf{B}_3\boldsymbol{\Delta}^P$. The master can obtain $\boldsymbol{\Delta}^F = \mathbf{B}\boldsymbol{\Delta}^P$ from $\mathbf{B}_1\boldsymbol{\Delta}^P, \mathbf{B}_2\boldsymbol{\Delta}^P, \mathbf{B}_3\boldsymbol{\Delta}^P$.

However, each node only has the information of edges that are connected with it. Then, is it still possible to achieve the communication cost of $rank(\mathbf{B})$? We will answer this question *affirmatively* if each edge information is known by both of the incident nodes.

Next, we describe an equivalent matrix representation of any linear communication scheme, and define the corresponding communication cost. For any graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider a matrix $\mathbf{X}' \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, whose rows are indexed by the nodes and columns are indexed by the edges. The entry in Row $U \in \mathcal{V}$ and Column $e \in \mathcal{E}$ equals 0 if Node U does not have the time discrepancy information of Edge e. Otherwise, the entry can be set as any value. Denote by $\mathbf{x}_U$ the row corresponding to Node U. If Node U transmits a symbol to the master, it must be in the form of $\mathbf{x}_U\boldsymbol{\Delta}^P$. The overall transmitted symbols must be in the form of $\mathbf{X}\boldsymbol{\Delta}^P$, where rows of $\mathbf{X}$ are chosen from rows of $\mathbf{X}'$ (a row may be chosen multiple times but the non-zero entries can be set differently). Finally, the master obtains $\mathbf{B}\boldsymbol{\Delta}^P = \mathbf{M}\mathbf{X}\boldsymbol{\Delta}^P$ for some transformation matrix $\mathbf{M}$. The *communication scheme* consists of the matrices $\mathbf{X}$ and $\mathbf{M}$, such that $\mathbf{B} = \mathbf{M}\mathbf{X}$. The *communication cost* equals the number of rows in $\mathbf{X}$.

III. Communication-efficient Clock Synchronization

In this section, we show our main results for the communication cost under three cases: the edge information is known by 1) both incident nodes, 2) one incident node, and 3) either one or two incident nodes.

For the first case, we claim that the communication cost of $rank(\mathbf{B})$ is achievable, and since we already established that the communication cost cannot be less than $rank(\mathbf{B})$, the solution is optimal.

Theorem 1. For any connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if every node has all its edge information, there exists an optimal solution where all but one node send one symbol, and the desired $\mathbf{B}\boldsymbol{\Delta}^P$ is recovered at the master. The total communication cost $CC = rank(\mathbf{B}) = |\mathcal{V}| - 1$.

For our purpose, in the matrix $\mathbf{X}' \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$, Row

$$\mathbf{x}_U, U \in \mathcal{V}, \quad (8)$$

is defined such that each column (edge) starting from U has entry 1, each column going to U has entry -1, and the other columns are 0.

We first demonstrate Theorem 1 using the example in Figure 1. It can be checked that the following communication scheme satisfies $\mathbf{B} = \mathbf{M}\mathbf{X}$.

$$\mathbf{M} = \begin{bmatrix} 0.25 & -0.25 & 0 \\ -0.125 & 0.125 & -0.5 \\ -0.125 & 0.125 & 0.5 \\ 0.375 & 0.625 & 0.5 \\ -0.625 & -0.375 & -0.5 \end{bmatrix}, \quad (9)$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_A \\ \mathbf{x}_B \\ \mathbf{x}_C \end{bmatrix} = \begin{array}{c} \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccccc} AB & BC & CA & BD & DA \end{array} \\ \begin{pmatrix} 1 & 0 & -1 & 0 & -1 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \end{pmatrix} \end{array}. \quad (10)$$

The transmitted symbols are $\mathbf{X}\boldsymbol{\Delta}^P$. This means that we let Nodes A, B, C send a linear combination of the time discrepancies of all their neighbouring edges. In total, the communication cost is $3 = |\mathcal{V}| - 1$ symbols which equals $rank(\mathbf{B})$. After the master receives these symbols, it can get $\boldsymbol{\Delta}^F = \mathbf{M}\mathbf{X}\boldsymbol{\Delta}^P = [10, -15, 5, 15, -25]^T$.

In fact, for $[\mathbf{x}_A{}^T, \mathbf{x}_B{}^T, \mathbf{x}_D{}^T]^T$, $[\mathbf{x}_A{}^T, \mathbf{x}_C{}^T, \mathbf{x}_D{}^T]^T$, $[\mathbf{x}_B{}^T, \mathbf{x}_C{}^T, \mathbf{x}_D{}^T]^T$, we can also find the corresponding $\mathbf{M}$. In general, it is sufficient to have any $|\mathcal{V}| - 1$ nodes send coded information to the master. Therefore, this scheme tolerates 1 straggler.

The proof of Theorem 1 is broken into several steps. We first show a rank condition in Lemma 1. Then we show the achievability in Lemma 2 and the converse in Lemma 3.

Lemma 1. Let $\mathbf{E} \in \mathbb{C}^{m\times n}, \mathbf{F} \in \mathbb{C}^{n\times p}$ be two matrices such that $\mathbf{E}\mathbf{F} = \mathbf{0}$, and the null space of $\mathbf{E}$ lies in the column span of $\mathbf{F}$. Then $rank(\mathbf{E}) + rank(\mathbf{F}) = n$.

Proof. Since the null space of $\mathbf{E}$ is in the column span of $\mathbf{F}$,

$$n - rank(\mathbf{E}) \leq rank(\mathbf{F}). \tag{11}$$

On the other hand, the rank of the product of $\mathbf{E}$, $\mathbf{F}$ satisfies Sylvester inequality:

$$0 = rank(\mathbf{EF}) \geq rank(\mathbf{E}) + rank(\mathbf{F}) - n. \tag{12}$$

Combining (11) and (12) we see the lemma is proved. □

Define a *loop set* to be a set of loops. We allow disjoint loops as well as loops with common edges. Define the corresponding *loop vector* $\mathbf{y}$ of length $|\mathcal{E}|$ to be a column vector that lists the number of times (with signs) each edge appears in the loop set, which are called the *weights* of the edges. A negative sign means that the edge is in the reverse direction. For example, in Figure 1, $\{A \to B \to C \to A, A \to B \to D \to A\}$ is a loop set. The loop vector is

$$\begin{matrix} AB & BC & CA & BD & DA \\ (2 & 1 & 1 & 1 & 1)^T. \end{matrix} \tag{13}$$

The weight of AB is 2, and the weight of BC is 1, etc. For each loop vector, there can be multiple associated loop sets. By the definition of the loop-composition matrix $\mathbf{A}$, a loop vector is a vector in the column span of $\mathbf{A}^T$.

Lemma 2. Let $\mathbf{X}$ be the matrix of size $(|\mathcal{V}| - 1) \times |\mathcal{E}|$, and its rows be any $|\mathcal{V}| - 1$ rows from (8). Then $\mathbf{B} = \mathbf{MX}$, for $\mathbf{M} = \mathbf{B}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$.

Proof. We first show $\mathbf{B}^T$ is in the column span of $\mathbf{X}^T$, and hence $\mathbf{B} = \mathbf{MX}$ for

$$\mathbf{M} = \mathbf{B}\mathbf{X}_{right}^{-1}, \tag{14}$$

where $\mathbf{X}_{right}^{-1}$ is the right preudo-inverse of $\mathbf{X}$. Then we find the formula for $\mathbf{M}$.

Assume $\mathbf{a}$ is any column of $\mathbf{A}^T$, which corresponds to a loop. Let $\mathbf{x}_U$ be the row vector as in (8) for Node U, whose ± 1 entries correspond to all incident edges of Node U. If the loop does not pass Node U, then there is no overlap between the edges in $\mathbf{x}_U$ and $\mathbf{a}$. In this case, $\mathbf{x}_U\mathbf{a} = 0$. Otherwise, every time the loop passes Node U, exactly one edge goes into node U, and exactly one edge goes out from node U. In this case, we also have $\mathbf{x}_U\mathbf{a} = 0$. Therefore, $\mathbf{X}\mathbf{A}^T = \mathbf{0}$.

Now let us prove that the rows of $\mathbf{X}$ are linearly independent. Consider a vector $\mathbf{y}$ in the null space of $\mathbf{X}$, i.e., $\mathbf{X}\mathbf{y} = 0$. We show that $\mathbf{y}$ is a loop vector. Since $\mathbf{x}_U\mathbf{y} = 0$, for any Node U, the sum weight in $\mathbf{y}$ for Node U's incoming edges equals the sum weight of its outgoing edges. By Veblen's theorem [21], a directed graph admits a decomposition into directed cycles if and only if the sum weight of the incoming edges equals the sum weight of the outgoing edges for every node. Therefore, $\mathbf{y}$ must be a loop vector, which is in the column span of $\mathbf{A}^T$. Combining the facts that $\mathbf{X}\mathbf{A}^T = \mathbf{0}$ and $rank(A) = |\mathcal{E}| + |\mathcal{V}| - 1$, we use Lemma 1 to conclude that

$$rank(\mathbf{X}) = |\mathcal{E}| - rank(\mathbf{A}) = |\mathcal{V}| - 1, \tag{15}$$

which is equal to the number of rows in $\mathbf{X}$.

Since matrix $\mathbf{A}$ of size $(|\mathcal{E}| - |\mathcal{V}| + 1) \times |\mathcal{E}|$ has full row rank, its null space has dimension $|\mathcal{V}| - 1$. Due to $\mathbf{A}\mathbf{X}^T = \mathbf{0}$ and (15), we known the null space of $\mathbf{A}$ equals the column span of $\mathbf{X}^T$. Moreover,

$$\mathbf{B}\mathbf{A}^T = \mathbf{A}^T - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{A}^T = \mathbf{A}^T - \mathbf{A}^T = \mathbf{0}. \tag{16}$$

Therefore, $\mathbf{B}^T$ is in the null space of $\mathbf{A}$, and hence is in the column span of $\mathbf{X}^T$. Thus, Equation (14) holds.

Finally, since $\mathbf{X}$ is full rank and $\mathbf{A}\mathbf{X}^T = \mathbf{0}$, we apply the pseudo-inverse formula to get

$$\mathbf{M} = (\mathbf{I} - \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A})\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}. \tag{17}$$

The proof is completed. □

Lemma 3. $rank(\mathbf{B}) = |\mathcal{V}| - 1$.

Proof. We will show that the null space of $\mathbf{B}^T$ is in the column span of $\mathbf{A}^\mathbf{T}$, and $\mathbf{B}^T\mathbf{A}^T = \mathbf{0}$ so as to use Lemma 1. In that case,

$$rank(\mathbf{B}) = |\mathcal{E}| - rank(\mathbf{A}) = |\mathcal{V}| - 1. \tag{18}$$

First,

$$\mathbf{A}\mathbf{B} = \mathbf{A} - \mathbf{A}\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A} = \mathbf{A} - \mathbf{A} = \mathbf{0}, \tag{19}$$

and thus $\mathbf{B}^T\mathbf{A}^T = \mathbf{0}$. Second, let $\mathbf{y}$ be any vector in the null space of $\mathbf{B}^T$, namely, $\mathbf{y}^T\mathbf{B} = \mathbf{0}$. Then

$$\begin{aligned} \mathbf{y}^T &= \mathbf{y}^T\mathbf{I} = \mathbf{y}^T\left(\mathbf{B} + \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{A}\right) \\ &= \left(\mathbf{y}^T\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\right)\cdot\mathbf{A}, \end{aligned} \tag{20}$$

which belongs to the row span of $\mathbf{A}$. Namely, $\mathbf{y}$ is in the column span of $\mathbf{A}^T$. □

Proof of Theorem 1. The scheme in Lemma 2 has a communication cost of $CC = rank(\mathbf{X}) = |\mathcal{V}| - 1$, which is equal to $rank(\mathbf{B})$ according to Lemma 3. Theorem 1 is proved. □

We note that in our optimal scheme, only $|\mathcal{V}| - 1$ nodes transmit, one symbol each, and we can tolerate 1 straggler.

Lemma 4. Let $\mathbf{X}'$ contain all $|\mathcal{V}|$ rows as in (8). If the master obtains $\mathbf{X}'\mathbf{\Delta}^P$, then it can perform row operations to get $\mathbf{B}\mathbf{\Delta}^P$. Moreover, $rank(\mathbf{X}') = rank(\mathbf{B}) = |\mathcal{V}| - 1$.

Proof. From Lemma 2, it is obvious that $\mathbf{X}'$, which includes all rows of $\mathbf{X}$, can be transformed into $\mathbf{B}$ by row operations.

By the same argument as Equation (15), one can show that $\mathbf{X}'\mathbf{A}^T = \mathbf{0}$, and the null space of $\mathbf{X}'$ is in the column span of $\mathbf{A}^T$. By Lemma 1 and Lemma 3,

$$rank(\mathbf{X}') = |\mathcal{E}| - rank(\mathbf{A}) = |\mathcal{V}| - 1 = rank(\mathbf{B}). \tag{21}$$

The proof is completed. □

While (21) holds for a connected graph, for $\mathcal{G}$ with n connected components, since $\mathbf{X}'$ is the incidence matrix, its rank is [22, Prop. 4.3]

$$rank(\mathbf{X}') = |\mathcal{V}| - n. \tag{22}$$

The above lemma indicates that the master's desired information $\mathbf{B}\mathbf{\Delta}^P$ can be equivalently represented by $\mathbf{X}'\mathbf{\Delta}^P$. The *desired dimensions* refers to either $\mathbf{B}$ or $\mathbf{X}'$ interchangeably.

Next, we consider the cases where the edge information is known by one or both of the incident nodes. We state in Lemma 5 a converse of the communication cost, which directly follows from the min-cut bound of linear network computing [16].

To that end, define $\mathcal{E}(U)$ as the set of the edges which are in $\mathcal{E}$ and known by Node U, for $U \in \mathcal{V}$. For a set of nodes $\mathcal{U} \subseteq \mathcal{V}$, define $\mathcal{E}(\mathcal{U})$ as the set of the edges known by Nodes $\mathcal{U}$. Denote matrix $\mathbf{X}'_{\mathcal{E}(U)}$ as the sub-matrix of $\mathbf{X}'$ obtained by choosing the columns corresponding to the edges in $\mathcal{E}(U)$. It represents the master's desired dimensions restricted to information known by U. The matrix $\mathbf{X}'_{\mathcal{E}(\mathcal{U})}$ is defined similarly. It can be easily seen that for any Node $U \in \mathcal{V}$, $\mathbf{X}'_{\mathcal{E}(U)}$ always contains a diagonal matrix (after row and column permutations) and is full rank $rank(\mathbf{X}'_{\mathcal{E}(U)}) = |\mathcal{E}(U)|$.

Lemma 5 (Min-cut bound)**.** Let CC_U denote the communication cost from Server U. The total communication cost satisfies

$$CC \geq \min \sum_{U\in\mathcal{V}} CC_U, \tag{23}$$

$$\text{s.t.} \sum_{U\in\mathcal{U}} CC_U \geq rank(\mathbf{X}'_{\mathcal{E}(\mathcal{U})}), \text{ for all } \mathcal{U} \subseteq \mathcal{V}. \tag{24}$$

Algorithm 1 Algorithm for graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where some edge information are singletons.

1: Send the singletons directly from the corresponding servers. Let $\mathcal{E}_s$ be the corresponding edges.
2: Let $\mathcal{H}$ be the graph $(\mathcal{V}, \mathcal{E} \backslash \mathcal{E}_s)$.
3: Let n be the number of connected components of $\mathcal{H}$.
4: Let $\mathbf{X}'$ be as in (8) for graph $\mathcal{H}$. Let $\mathbf{X}$ be $|\mathcal{V}| - n$ rows of $\mathbf{X}'$ such that one row is excluded from $\mathbf{X}'$ for each connected component.
5: Let $\mathbf{M} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$.
6: Use the communication scheme $\mathbf{X}, \mathbf{M}$.
7: Combine the singletons and $\mathbf{M}\mathbf{X}\mathbf{\Delta}^P$ to obtain $\mathbf{B}\mathbf{\Delta}^P$.

For example, if we set $\mathcal{U} = \mathcal{V}$ in (24), we get the aforementioned communication cost bound $CC \geq \min \sum_{U \in \mathcal{V}} CC_U \geq rank(\mathbf{X}') = rank(\mathbf{B})$.

The following theorem states that when each edge information is known by only one node, the trivial scheme is optimal.

Theorem 2. For graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, if each edge information is only known by one of its incident nodes, the optimal solution for the master to obtain the desired $\mathbf{B}\mathbf{\Delta}^P$ is to send all the edge information individually. The total communication cost is $CC = |\mathcal{E}|$.

Proof. The achievability is obvious. We only need to show the converse. The communication cost of any Node U must satisfy $CC_U \geq rank\left(\mathbf{X}'_{\mathcal{E}(U)}\right) = |\mathcal{E}(U)|$ based on the min-cut bound with $\mathcal{U} = \{U\}$. The total communication cost

$$CC \geq \min \sum_{U \in \mathcal{V}} CC_U \geq \sum_{U \in \mathcal{V}} rank\left(\mathbf{X}'_{\mathcal{E}(U)}\right) \tag{25}$$

$$= \sum_{U \in \mathcal{V}} |\mathcal{E}(U)| = |\mathcal{E}|, \tag{26}$$

where the last equality holds since each edge is only known by one node. □

Finally, let us consider the case where each edge information is known by either one or both incident nodes. We say the time discrepancy information on an edge is *singleton* if it is known by just one node. We provide an achievable scheme in Algorithm 1. It trivially sends singletons to the master, and removes the corresponding edges. Call the remaining graph $\mathcal{H}$. Then it solves the remaining desired dimensions on graph $\mathcal{H}$ as in Lemma 2. The following lemma is straightforward from the algorithm.

Lemma 6. Let m be the number of singletons. Let n be the number of connected components of $\mathcal{H}$. Algorithm 1 achieves the communication cost of $CC = |\mathcal{V}| - n + m$.

Algorithm 1 is optimal for certain cases. For example, in Figure 1, let $\mathcal{E}(A) = \{AB, CA\}$, $\mathcal{E}(B) = \{BD, BC\}$, $\mathcal{E}(C) = \{CA\}$, $\mathcal{E}(D) = \{DA, BD\}$. There are $m = 3$ singletons, i.e., l_{AB}, l_{DA}, l_{BC}. After removing the singleton edges, the graph $\mathcal{H}$ consists of all the 4 nodes but only 2 edges CA and BD. Then $\mathcal{H}$ contains $n = 2$ connected components, such that the first component contains Nodes A, C, and the second component contains Nodes B, D. The communication cost of Algorithm 1 is $CC = |\mathcal{V}| - n + m = 5$. On the other hand, set $\mathcal{U} = \{A, C\}$ and $\mathcal{U} = \{B, D\}$ in (24),

$$\mathbf{X}'_{\mathcal{E}(\{A,C\})} = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cc} AB & CA \end{array} \\ \begin{pmatrix} 1 & -1 \\ -1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \end{array}, \tag{27}$$

$$\mathbf{X}'_{\mathcal{E}(\{B,D\})} = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{ccc} BC & BD & DA \end{array} \\ \begin{pmatrix} 0 & 0 & -1 \\ 1 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 1 \end{pmatrix} \end{array}. \tag{28}$$

We obtain the lower bound

$$CC_A + CC_C \geq rank\left(\mathbf{X}'_{\mathcal{E}(\{A,C\})}\right) = 2, \tag{29}$$

$$CC_B + CC_D \geq rank\left(\mathbf{X}'_{\mathcal{E}(\{B,D\})}\right) = 3, \tag{30}$$

$$CC \geq \min \sum_{U \in \mathcal{V}} CC_U \geq 5. \tag{31}$$

Therefore, the algorithm gives the optimal communication cost in the example. In general, we have the following sufficient condition for Algorithm 1 to be optimal.

Theorem 3. Algorithm 1 is optimal under the following condition: $\mathcal{H}$ contains $n \geq 2$ connected components, each component has more than 1 nodes, every singleton edge is between different components, and singleton edges known by distinct nodes in one component are not connected.

Proof. We show that the cut-set bound matches the communication cost in Lemma 6. Let $\mathcal{V}_i, \mathcal{E}_i$ be the nodes and edges in the i-th connected component of $\mathcal{H}$, and $\mathcal{E}'_i$ the singleton edges known by $\mathcal{V}_i$, $1 \leq i \leq n$. Then, the set of edges known by the i-th component is $\mathcal{E}(\mathcal{V}_i) = \mathcal{E}_i \cup \mathcal{E}'_i$. Consider $\mathbf{X}'_{\mathcal{E}(\mathcal{V}_i)}$,

$$\mathbf{X}'_{\mathcal{E}(\mathcal{V}_i)} = \begin{array}{c} \\ \mathcal{V}_i \\ \mathcal{V} \backslash \mathcal{V}_i \end{array} \begin{array}{c} \begin{array}{cc} \mathcal{E}_i & \mathcal{E}'_i \end{array} \\ \begin{pmatrix} \mathbf{X}_1 & * \\ \mathbf{0} & \mathbf{X}_2 \end{pmatrix} \end{array}, \tag{32}$$

where we list rows (nodes) in the i-th component on the top. The matrix $*$ is not of interest. Matrix $\mathbf{X}_1$ is the matrix $\mathbf{X}'$ as in (8) for the graph $(\mathcal{V}_i, \mathcal{E}_i)$, whose rank is $rank(\mathbf{X}_1) = |\mathcal{V}_i| - 1$. Matrix $\mathbf{X}_2$ corresponds to the singletons known by the i-th component. Due to the given condition on the singletons in the theorem, the subgraph induced by edges $\mathcal{E}'_i$ is simply several disconnected star graphs (one center node connected to the other nodes), where nodes in the i-th component are the centers. Hence, it can be seen that $\mathbf{X}_2$ is a diagonal block matrix, and every block corresponds to one star. Since $\mathbf{X}_2$ does not include the rows corresponding to the center nodes in $\mathcal{V}_i$, by (22) each block is full rank, which is equal to the number of vertices in the star minus 1, or the number of edges. Overall, the diagonal block matrix satisfies $rank(\mathbf{X}_2) = |\mathcal{E}'_i|$.

Therefore, the cut-set bound gives

$$\sum_{U\in\mathcal{V}_i} CC_U \geq rank(\mathbf{X}'_{\mathcal{E}(\mathcal{V}_i)}) \tag{33}$$

$$=rank(\mathbf{X}_1)+rank(\mathbf{X}_2)=|\mathcal{V}_i|-1+|\mathcal{E}'_i|, \tag{34}$$

$$CC \geq \min \sum_{U\in\mathcal{V}} CC_U = \min \sum_{i=1}^{n} \sum_{U\in\mathcal{V}_i} CC_U \tag{35}$$

$$\geq \sum_{i=1}^{n} (|\mathcal{V}_i|-1+|\mathcal{E}'_i|) = |\mathcal{V}|-n+m. \tag{36}$$

The proof is completed. □

IV. Discussion

Besides the clock synchronization, the idea of this work may have other applications. For example, given pairwise evaluations by customers for how much more one item is worth than another, our method can give a communication-efficient global evaluation of all values, subject to minimum norm perturbation.

References

[1] N. Van Tu, J. Hyun, G. Y. Kim, J.-H. Yoo, and J. W.-K. Hong, "Intcollector: A high-performance collector for in-band network telemetry," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 10–18.

[2] A. Kemper and T. Neumann, "Hyper: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots," in *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 2011, pp. 195–206.

[3] Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosunblum, and A. Vahdat, "Exploiting a natural network effect for scalable, fine-grained clock synchronization," in *Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'18. USENIX Association, 2018, p. 81–94.

[4] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: algorithms and analysis," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 4915–4920.

[5] Y. S. Patel, A. Page, M. Nagdev, A. Choubey, R. Misra, and S. K. Das, "On demand clock synchronization for live VM migration in distributed cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 138, pp. 15–31, 2020.

[6] Y. Li, G. Kumar, H. Hariharan, H. Wassel, P. Hochschild, D. Platt, S. Sabato, M. Yu, N. Dukkipati, P. Chandra *et al.*, "Sundial: fault-tolerant clock synchronization for datacenters," in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020, pp. 1171–1186.

[7] H. Kopetz and W. Ochsenreiter, "1987." *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 933–940, 1987.

[8] M. Leng and Y.-C. Wu, "Distributed clock synchronization for wireless sensor networks using belief propagation," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5404–5414, 2011.

[9] M. K. Maggs, S. G. O'keefe, and D. V. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE sensors Journal*, vol. 12, no. 6, pp. 2269–2277, 2012.

[10] R. Solis, V. S. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 2734–2739.

[11] E. Mallada, X. Meng, M. Hack, L. Zhang, and A. Tang, "Skewless network clock synchronization without discontinuity: convergence and performance," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1619–1633, 2014.

[12] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[13] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, "Network coding for computing: cut-set bounds," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 1015–1030, 2011.

[14] ——, "Linear codes, target function classes, and network computing capacity," *IEEE Transactions on Information Theory*, vol. 59, no. 9, pp. 5741–5753, 2013.

[15] J. Zhan, S. Y. Park, M. Gastpar, and A. Sahai, "Linear function computation in networks: duality and constant gap results," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 620–638, 2013.

[16] R. Appuswamy and M. Franceschetti, "Computing linear functions by linear coding over networks," *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 422–431, 2014.

[17] C. Huang, Z. Tan, S. Yang, and X. Guang, "Comments on cut-set bounds on network function computation," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6454–6459, 2018.

[18] X. Guang, R. W. Yeung, S. Yang, and C. Li, "Improved upper bound on the network function computing capacity," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3790–3811, 2019.

[19] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, p. 406–413, 1955.

[20] M. OpenCourseWare, "Left and right inverses; pseudoinverse," *Online. http://ocw. mit. edu/courses/mathematics/18-06sc-linear-algebra-fall-2011/positive-definite-matrices-and-applications/left-and-right-inverses-pseudoinverse/MIT18_06SCF11_Ses3. 8sum. pdf. Accessed March*, 2015.

[21] J. Bondy and U. Murty, *Graph theory*. Springer, 2008.

[22] N. Biggs, N. L. Biggs, and B. Norman, *Algebraic graph theory*. Cambridge university press, 1993, no. 67.