2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Synchronizing clocks in distributed networks

Weiguo Xia and Ming Cao

Abstract— **While various time synchronization protocols for clocks in wired and/or wireless networks are under development, recently it has been shown by Freris, Graham and Kumar that clocks in distributed networks cannot be synchronized precisely even in idealized situations. In this paper by determining the clock synchronization errors in the similar settings of the impossibility result just mentioned, we are able to show that the clocks can get synchronized within an acceptable level of accuracy. After studying the basic case of synchronizing two clocks with asymmetric time delays in the two-way message passing process, we first analyze the directed ring networks, in which neighboring clocks are likely to experience severe asymmetric time delays. We then discuss connected undirected networks with two-way message passing between each pair of adjacent nodes. In the end, we expand the discussions to networks with directed topologies that are strongly connected.**

I. Introduction

As physical devices, such as computational units, sensors and actuators, are more and more frequently working together over distances, people are more and more concerned with the problem of how to synchronize the clocks that are installed at those physical devices and connected through wired and/or wireless data networks [6]. Clock synchronization has been discussed intensively in the area of theoretical computer science especially in the 1980's [8], [15], and various impossibility results and bounds for synchronization errors have been reported [11], [10]. More recently, with the growing interest in the application of large-scale networks, in particular ad hoc and sensor networks, clock synchronization problems have attracted considerable attention [12], [1], [14], [13].

Very recently, Freris, Graham and Kumar have shown that in an idealized setting the clocks *cannot* be synchronized precisely in distributed networks when asymmetric time delays are present [7]. This result is obtained by using tools from linear system theory and is consistent with the results obtained previously in theoretical computer science. On the other hand, in engineering practice when clocks are adjusted repeatedly to compensate the differences between their time displays, their displays can indeed get synchronized within an acceptable level of accuracy in a distributed fashion. In [16], the Time-Diffusion synchronization Protocol (TDP) has been proposed to enable sensor networks to synchronize their clocks with bounded errors. In [9], both synchronous and asynchronous versions of a rate-based diffusion protocol have been discussed, in which clocks adjust their displays repeatedly by taking the weighted average of the displays of themselves and their adjacent clocks.

In this paper, we determine the clock synchronization errors in the similar settings to that in [7] under which the impossibility result for clock synchronization has been achieved. By updating all clocks repeatedly, we are able to derive explicit expressions of the synchronization errors in steady states, which are within an acceptable range even when the time delays are asymmetric. We first look into directed ring networks, in which neighboring clocks may experience severe asymmetric time delays in the two directed paths that connect them. We then investigate connected undirected networks, in which each pair of adjacent nodes can exchange messages with each other. In the end we discuss clock synchronization in networks with strongly connected directed topologies.

The rest of the paper is organized as follows. In Section II, we review the basic setting in [7] for synchronizing two clocks with asymmetric time delays and then analyze the linear system model to determine the asymptotic clock synchronization errors through state augmentation. We analyze the synchronization errors in directed ring networks, in connected undirected networks and finally in strongly connected directed networks in Sections III, IV, and V, respectively. Concluding remarks are given in Section VI.

II. Synchronizing two clocks

As in [7], we consider affine models for clocks. Let $i > 0$ be the label of a clock in a network, and denote its display by x_i. Then the evolution of x_i can be described by

$$x_i(t) = a_i t + b_i, \tag{1}$$

where t is the time of a standard reference clock, $a_i > 0$ is called the *skew* that is the ratio of the speed of clock i with respect to the reference clock, and b_i is called the *offset* that is the difference between the display of clock i and the reference clock at time $t = 0$. In practice, the values of clocks' skews are very close to one. As shown in [7], when the skews of the clocks are fixed, e.g. not affected by the changes in the environmental temperature, and the communications within the network are noiseless and fault-free, clocks can always estimate precisely their skews through message passing. Hence, clocks can always compensate the differences in their skews or even adjust their skews to a common value [4]. Consequently, to simplify the analysis in this paper, in what follows, we assume that $a_i = 1$ for all i.

This work was supported in part by grants from the Dutch Organization for Scientific Research (NWO) and the Dutch Technology Foundation (STW).

Weiguo Xia and Ming Cao are with the Faculty of Mathematics and Natural Sciences, ITM, University of Groningen, The Netherlands {w.xia,m.cao}@rug.nl

We consider the case when clocks are installed at nodes in a distributed network. We use the label of the clock to denote the node where the clock is installed. It is assumed in [7] that when a message is sent from node i to another node j, the latter can only receive it after a fixed but unknown time delay $d_{ij} > 0$. In addition, the time delays are not necessarily symmetric, and in fact for a pair of distinct nodes i and j, d_{ij} is in general not equal to d_{ji}.

For analysis purposes, we can always describe the message passing process with respect to the standard reference clock, although such a clock is *not* known to any of the two clocks. In the sequel, we use the sequence $\{t_k\}$, $k \geq 0$, to denote the set of time instances embedded in the reference time axis t at which a clock sends or receives messages. Then the message exchange process for two clocks 1 and 2 trying to get synchronized is illustrated in Figure 1. At time t_0, node 1 sends a message of its current value of $x_1(t_0)$ to node 2. We say node 1 has sent a message *time stamped* by its clock just before the transmission. Node 2 records the time $x_2(t_1)$ when it receives the message $x_1(t_0)$ and after a constant time w_1, it sends the message $x_2(t_1)$ at the time t_2 back to node 1 with the time-stamp $x_2(t_2)$. Correspondingly, node 1 receives this message at time t_3 and records the time $x_1(t_3)$. It then sends a message after a constant time w_2. In this manner the messages are sent back and forth. Hence, from the information contained in the exchanged time-stamped messages, the round-trip time delay $d_{12}+d_{21}$ can be obtained accurately by

$$d_{12} + d_{21} = x_1(t_3) - x_2(t_2) + x_2(t_1) - x_1(t_0).$$

However, the individual time delays d_{12} and d_{21} can *never* be determined precisely and this is part of the synchronization impossibility result for a pair of clocks shown in [7], which as argued in the same paper leads to synchronization errors that cannot be eliminated.

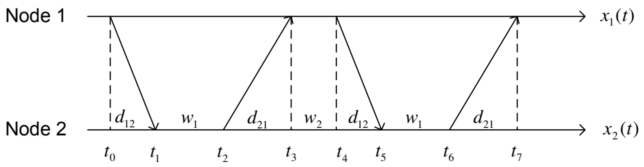

Fig. 1. Message exchanges between two clocks.

Now we try to synchronize the two clocks by repeatedly updating their displays. We use D to denote the round-trip time delay $d_{12} + d_{21}$. When the two clocks update their displays, they use the average delay $\bar{D} = \frac{D}{2}$ as the *nominal* delay to compensate the time-stamped messages they receive about the most recent values of the other clock's display. For example, when clock 1 receives a message of $x_2(t_k)$ from clock 2, it takes $x_2(t_k) + \bar{D}$ as the estimated current value of the display of clock 2. The same estimation strategy is adopted by both of the two clocks. To get synchronized, after a clock receives a new message from the other, it always updates its display to the average of its current display and the latest estimation of the other clock's current display. We assume the updates take place instantaneously and the message exchanges are carried out repeatedly.

The embedding technique to write down a distributed system's dynamics with respect to a common reference time axis for analysis purposes has been used before when studying distributed and parallel computations and asynchronous systems [2], [3]. Following this approach, we use the sequence $\{t_k\}$, $k \geq 0$, embedded in the reference time axis t, to write the system equations. Since the clocks have the same skew and both of them update periodically, we know that for any time $\tau > 0$, there always exists $k \geq 0$ such that $t_k \leq \tau < t_{k+1}$ and $x_1(\tau) - x_2(\tau) = x_1(t_k) - x_2(t_k)$. For the sake of conciseness, in the sequel we use the notation $x_i(k)$ instead of $x_i(t_k)$. Then the system equations of the updating process of the two clocks after embedding can be written as

$$\begin{cases} x_1(4k+1) = x_1(4k) + d_{12} \\ x_2(4k+1) = \frac{1}{2}((x_1(4k) + \bar{D}) + (x_2(4k) + d_{12})) \end{cases}$$
$$\begin{cases} x_1(4k+2) = x_1(4k+1) + l_1 d_{12} \\ x_2(4k+2) = x_2(4k+1) + l_1 d_{12} \end{cases}$$
$$\begin{cases} x_1(4k+3) = \frac{1}{2}((x_2(4k+2) + \bar{D}) + (x_1(4k+2) + d_{21})) \\ x_2(4k+3) = x_2(4k+2) + d_{21} \end{cases}$$
$$\begin{cases} x_1(4(k+1)) = x_1(4k+3) + l_2 d_{12} \\ x_2(4(k+1)) = x_2(4k+3) + l_2 d_{12}, \end{cases} \tag{2}$$

where $k \geq 0$ and $l_i = \frac{w_i}{d_{12}}$, $i = 1, 2$.

The main result in this section is to show that during the above updating process (2), the synchronization error converges to a constant determined by the differences between the delays d_{12} and d_{21}.

Theorem 1: As t goes to infinity, the difference $x_1(t) - x_2(t)$ between the two clocks converges to $\frac{1}{2}(d_{12} - d_{21})$.

Proof: Let $e(k) \triangleq x_1(k) - x_2(k)$ for $k \geq 0$. Then from (2), one has

$$\begin{aligned} e(4k+1) &= \frac{1}{2}e(4k) + \frac{1}{4}(d_{12} - d_{21}) \\ e(4k+2) &= e(4k+1) \\ e(4k+3) &= \frac{1}{2}e(4k+2) + \frac{1}{4}(d_{12} - d_{21}) \\ e(4(k+1)) &= e(4k+3). \end{aligned} \tag{3}$$

Substituting the first three equations of (3) into the last equation of (3), we obtain

$$\begin{aligned} e(4(k+1)) &= (\frac{1}{2})^2 e(4k) + \frac{3}{8}(d_{12} - d_{21}) \\ &= (\frac{1}{2})^{2(k+1)} e(0) + \frac{3}{8}(d_{12} - d_{21}) \sum_{i=0}^{k+1} \frac{1}{4^i}. \end{aligned}$$

Since the geometric series $\sum_{i=0}^{\infty} \frac{1}{4^i}$ converges, we know

$$\lim_{k\to\infty} e(4(k+1)) = \frac{3(d_{12} - d_{21})}{8} \sum_{i=0}^{\infty} \frac{1}{4^i} = \frac{d_{12} - d_{21}}{2}. \tag{4}$$

Combining equation (4) with (3), one can check that

$$\lim_{k\to\infty} e(4k+i) = \frac{1}{2}(d_{12}-d_{21}),\ 1 \le i \le 4. \quad (5)$$

From (5), we know that for any $\epsilon > 0$, there exists a positive integer N, such that for any $n > N$, $|e(4n+i) - \frac{1}{2}(d_{12} - d_{21})| < \epsilon$, $1 \le i \le 4$. Hence, for any $k > 4(N+1)$, it always holds that $|e(k) - \frac{1}{2}(d_{12} - d_{21})| < \epsilon$, which is equivalent to

$$\lim_{k\to\infty} e(k) = \frac{1}{2}(d_{12}-d_{21}). \quad (6)$$

This completes the proof. □

Note that when applying the Network Time Protocol (NTP) [12], it is assumed that most of the time delays are symmetric between a pair of distinct nodes in a network, namely $d_{ij} = d_{ji}$ for $i \neq j$. In fact, in view of Theorem 1, when $d_{12} = d_{21}$, the two clocks can indeed get synchronized precisely.

Corollary 1: When $d_{12} = d_{21}$, the synchronization error $x_1(t) - x_2(t)$ between the two clocks goes to zero asymptotically.

In the next three sections, we will study how the main idea of compensation with nominal delays can be applied to larger networks by utilizing the message passing mechanism just described. It has proven convenient to use graphs to describe topologies of general networks. A graph $\mathbb{G}$ with the node set $\mathcal{N} = \{1, \ldots, n\}$ and the edge set $\mathcal{E} \subset \{(i,j) : i,j \in \mathcal{N}\}$ can be used to describe the topology of a network consisting of n nodes. In $\mathbb{G}$, there is a directed edge from node i to j if i can send messages to j; correspondingly, there is an undirected edge between i and j if both i and j can send messages to each other. A directed path in a directed graph $\mathbb{G}$ is a sequence of distinct nodes $i_1, \ldots, i_k$ such that $(i_s, i_{s+1}) \in \mathcal{E}$ for $s = 1, \ldots, k-1$. $\mathbb{G}$ is said to be strongly connected if there is a directed path from every node to every other node. For an undirected graph, strongly connectedness is equivalent to connectedness.

Since among the networks with the same number of nodes, the network with a directed ring topology can lead to the greatest difference in the delays of d_{ij} and d_{ji} for a given pair of adjacent nodes i and j, we first study synchronizing clocks in networks with directed ring topologies.

III. Synchronizing clocks in directed ring networks

A. Synchronizing three clocks in a directed ring network

We fist consider a ring network of three nodes 1, 2 and 3 and three directed edges $(1,2)$, $(2,3)$ and $(3,1)$. Similar to the message passing process for the 2-clock case discussed in the previous section, we illustrate the message passing process among the three clocks in Fig. 2, where d_{12}, d_{23}, d_{31} and w_i, $i = 1,2,3$, are the time delays and idling times, respectively.

Although the delays d_{12}, d_{23} and d_{31} cannot be determined from the time-stamped messages, the round-trip delay $D = d_{12} + d_{23} + d_{31}$ can be determined precisely by

$$D = x_1(5) - x_3(4) + x_3(3) - x_2(2) + x_2(1) - x_1(0).$$

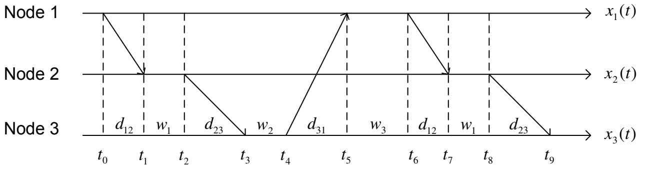

Fig. 2. Message passing among three clocks with directed links.

We take $\bar{D} = \frac{D}{3}$ as the nominal delay for the three clocks when they update their displays. To be more specific, we take time t_1, when node 2 receives a message from node 1, as an example. At t_1 clock 2 updates its display to the average of its current display and the current estimate of clock 1's display $x_1(0) + \bar{D}$. And w_1 time units later, clock 2 sends the message $x_2(2)$ to clock 3, which in turn updates its display following the same averaging rule. This procedure repeats periodically. As one can see from Fig. 2, every link is used exactly once in each period from t_{6k} to $t_{6(k+1)}$ for $k \ge 0$.

Now we write down the system equations. Define

$$x(k) = [x_1(k), x_2(k), x_3(k)]^T, \quad v = [d_{12}, d_{23}, d_{31}]^T.$$

Then for $k \ge 0$,

$$\begin{aligned}\begin{bmatrix} x_1(6k+1) \\ x_2(6k+1) \\ x_3(6k+1) \end{bmatrix} &= \begin{bmatrix} x_1(6k) + d_{12} \\ \frac{1}{2}\Big((x_1(6k) + \bar{D}) + (x_2(6k) + d_{12})\Big) \\ x_3(6k) + d_{12} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(6k) \\ x_2(6k) \\ x_3(6k) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} d_{12} \\ d_{23} \\ d_{31} \end{bmatrix}.\end{aligned}$$

Through a similar procedure, one can obtain

$$x(6k+i) = A_i x(6k+i-1) + B_i v, \quad 1 \le i \le 6, \quad (7)$$

where

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, A_5 = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 1 & 0 & 0 \\ \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ 1 & 0 & 0 \end{bmatrix}, B_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{bmatrix}, B_5 = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix},$$

$$A_2 = A_4 = A_6 = I_3,\ B_{2j} = l_j \begin{bmatrix} \mathbf{1}_3 & O_{3\times 2} \end{bmatrix},\ j = 1,2,3.$$

Here, I_3 is the 3-dimensional identity matrix, $\mathbf{1}_3$ is the 3-dimensional all-one column vector, $O_{3\times 2}$ is the 3×2 zero matrix, and $l_j = \frac{w_j}{d_{12}}$. We can further obtain the following system equation in an iterative form

$$x(6(k+1)) = A_6 A_5 \cdots A_1 x(6k) + \sum_{i=1}^{6} A_6 \cdots A_{i+1} B_i v.$$

Define $A \triangleq A_6 A_5 \cdots A_1 = A_5 A_3 A_1$ and $B \triangleq \sum_{i=1}^{6} A_6 \cdots A_{i+1} B_i$, then we have

$$x(6(k+1)) = A^{k+1} x(0) + \sum_{i=0}^{k} A^i B v, \quad k \ge 0. \quad (8)$$

We first show the following convergence result.

Proposition 1: As k goes to infinity, $x_i(6(k+1)) - x_j(6(k+1))$ converge to some constants for $i,j = 1,2,3,\ i \neq j$.

The proof of Proposition 1 will be present in the full-length version of the paper.

If we take t_2 or t_4 in Fig. 2 as the starting time of the system evolution, one can get that $x_i(6k+2) - x_j(6k+2)$ and $x_i(6k+4) - x_j(6k+4)$ both converge to some constants for $1 \leq i,j \leq 3,\ i \neq j$, as $k \to \infty$. Since

$$x_i(6k+r) - x_j(6k+r) = x_i(6k+r-1) - x_j(6k+r-1),$$

hold for $r = 2,4,6$, one can get the following conclusion.

Proposition 2: As k goes to infinity, $x_i(6k+r) - x_j(6k+r)$ converge to some constants for all $r = 1,\dots,6$, and $i,j = 1,2,3,\ i \neq j$.

From Proposition 2, we know that we can define

$$e_{ij}(6k+r) \triangleq x_i(6k+r) - x_j(6k+r),$$
$$e(6k+r) \triangleq [e_{12}(6k+r), e_{23}(6k+r)]^T,$$

and the constants

$$e_{ij}^r \triangleq \lim_{k\to\infty} e_{ij}(6k+r),$$

where $i,j = 1,2,3,\ i \neq j$, and $r = 1,\dots,6$. From the system equations (7), one can get a set of equations

$$e(6k+i) = \tilde{A}_i e(6k+i-1) + \tilde{B}_i v,\ 1 \leq i \leq 6, \tag{9}$$

where

$$\tilde{A}_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{bmatrix},\ \tilde{A}_3 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix},\ \tilde{A}_5 = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ 0 & 1 \end{bmatrix},$$
$$\tilde{B}_1 = \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{3} & \frac{1}{6} & \frac{1}{6} \end{bmatrix},\ \tilde{B}_3 = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{6} & \frac{1}{3} & -\frac{1}{6} \end{bmatrix},$$
$$\tilde{B}_5 = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & -\frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix},$$
$$\tilde{A}_2 = \tilde{A}_4 = \tilde{A}_6 = I_2,\ \tilde{B}_2 = \tilde{B}_4 = \tilde{B}_6 = O_{2\times 3}.$$

By iteration, one has

$$\begin{aligned} e(6(k+1)) &= \tilde{A}_6\tilde{A}_5\cdots\tilde{A}_1 e(6k) + \sum_{i=1}^{6} \tilde{A}_6\cdots\tilde{A}_{i+1}\tilde{B}_i v \\ &= \tilde{A}^{k+1}e(0) + \sum_{i=0}^{k} \tilde{A}^i \tilde{B} v, \quad k \geq 0, \end{aligned}$$

where $\tilde{A} \triangleq \tilde{A}_6\tilde{A}_5\cdots\tilde{A}_1 = \tilde{A}_5\tilde{A}_3\tilde{A}_1$ and $\tilde{B} = \sum_{i=1}^{6} \tilde{A}_6\cdots\tilde{A}_{i+1}\tilde{B}_i$. Taking k to infinity, one has

$$\lim_{k\to\infty} e(6(k+1)) = \lim_{k\to\infty} \tilde{A}^{k+1}e(0) + \lim_{k\to\infty} \sum_{i=0}^{k} \tilde{A}^i \tilde{B} v,\ k \geq 0.$$

Since the limit $\lim_{k\to\infty} e(6(k+1))$ exists for any initial condition and any time delays from Proposition 2, it must be true that both $\lim_{k\to\infty} \tilde{A}^{k+1}$ and $\lim_{k\to\infty}\sum_{i=0}^{k} \tilde{A}^i$ converge, from which we conclude that $\rho(\tilde{A}) < 1$, namely, the spectral radius of $\tilde{A}$ is strictly less than 1.

In view of the fact that $e_{i,i+1}^{r+1} = e_{i,i+1}^r,\ i = 1,2,\ r = 1,3,5$, we define

$$e^r \triangleq [e_{12}^r, e_{23}^r]^T,\ e \triangleq [(e^1)^T, (e^3)^T, (e^5)^T]^T.$$

Then we get the equation of the asymptotic synchronization errors between clocks by taking k on both sides of (9) to infinity:

$$e = \bar{A}e + \bar{B}v,$$

where

$$\bar{A} = \begin{bmatrix} O & O & \tilde{A}_1 \\ \tilde{A}_3 & O & O \\ O & \tilde{A}_5 & O \end{bmatrix},\ \bar{B} = \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_3 \\ \tilde{B}_5 \end{bmatrix},$$

with O being the zero matrix of compatible dimension. If the matrix $I - \bar{A}$ is invertible, where I is the identity matrix of compatible dimension, the error e can be calculated as $e = (I - \bar{A})^{-1}\bar{B}v$.

Lemma 1: The matrix $I - \bar{A}$ is invertible.

The proof of Lemma 1 makes use of the fact that $\rho(\tilde{A}) < 1$ and will be present in the full-length version of the paper.

Thus, by calculating $e = (I - \bar{A})^{-1}\bar{B}v$, one has

$$e_{12}^r = d_{12} - \bar{D},\ e_{23}^r = d_{23} - \bar{D},\ r = 1,\dots,6.$$

Hence, we have proved the following.

Theorem 2: As time goes to infinity, the synchronization errors between clocks in the three-clock ring network converge and

$$\lim_{t\to\infty} (x_i(t) - x_{[i]}(t)) = d_{i,[i]} - \bar{D}, \quad i = 1,2,3,$$

where $[i] = i+1$ if $i = 1,2$ and $[i] = 1$ if $i = 3$.

The following result is a direct consequence of Theorem 2.

Corollary 2: For the three clocks in the ring network, if the delays are all equal, namely $d_{12} = d_{23} = d_{31}$, the clocks can get synchronized asymptotically.

In the next subsection, we extend the results that we have obtained for the three-clock ring network to general ring networks with $n \geq 3$ nodes.

B. Synchronizing more clocks in a directed ring network

Now we consider a directed ring network of $n \geq 3$ nodes. The message passing procedure in the network with unidirectional communications is illustrated in Fig. 3, where $d_{i,[i]}$ and w_i, $i = 1,\dots,n$, are time delays and idling times respectively. Here, $[i]$ is defined to be $i+1$ when $i = 1,\dots,n-1$ and 1 when $i = n$.

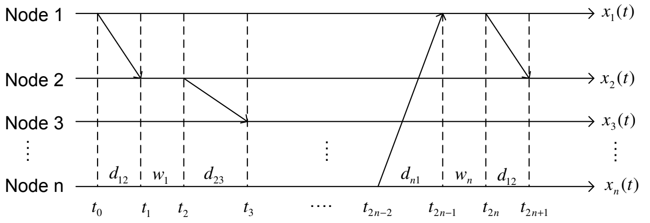

Fig. 3. Message passing among $n \geq 3$ clocks with directed links.

Although the time delays $d_{i,[i]},\ i = 1, \ldots, n$, between clocks cannot be determined precisely no matter how many time-stamped messages are exchanged, the round-trip delay $D = \sum_{i=1}^{n} d_{i,[i]}$ can be calculated after sufficiently many messages are delivered

$$D = \sum_{i=0}^{n-1} \left(x_{[i+1]}(2i+1) - x_{i+1}(2i) \right).$$

Similar to the three-clock case in Subsection III-A, we use $\bar{D} = \frac{D}{n}$ as the nominal delay for all the clocks which they use when updating their displays. Using similar arguments to that in Subsection III-A, one can prove the following result.

Theorem 3: As time goes to infinity, the synchronization errors between clocks in the n-clock ring network, $n \geq 3$, converge and

$$\lim_{t \to \infty} (x_i(t) - x_{[i]}(t)) = d_{i,[i]} - \bar{D}, \quad i = 1, \ldots, n.$$

In the next section, we discuss how to synchronize clocks in connected undirected networks.

IV. Synchronizing clocks in connected undirected networks

A. Synchronizing three clocks in a connected undirected network

We fist consider a network of three nodes with undirected edges $(1, 2)$, $(2, 3)$ and $(1, 3)$. Similar to the message passing process for the 2-clock case discussed before, we illustrate the message passing process among the three clocks in Fig. 4.

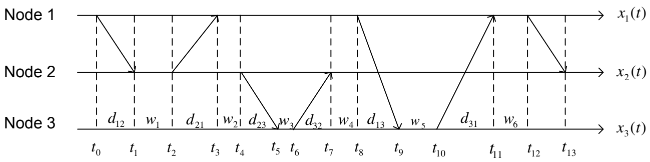

Fig. 4. Message passing among three clocks with undirected links.

Although the delays $d_{ij},\ 1 \leq i, j \leq 3$, cannot be determined from the time-stamped messages, the round-trip delay between each pair of connected clocks can be calculated precisely. For example, the round-trip delay D_{12} between clocks 1 and 2 is

$$D_{12} = d_{12} + d_{21} = x_1(3) - x_2(2) + x_2(1) - x_1(0).$$

We take $\bar{D}_{ij} = \frac{D_{ij}}{2}$ for a pair of adjacent clocks i and j when they update their displays, where $D_{ij} = d_{ij} + d_{ji}$ is the round-trip delay between clocks i and j. As before the clocks update following the same average rule and this procedure repeats periodically. It can be seen from Fig. 4 that, in each update period from t_{12k} to $t_{12(k+1)}$ for $k \geq 0$, a pair of adjacent nodes exchange messages exactly once.

Define $x(k) \triangleq [x_1(k), x_2(k), x_3(k)]^T$ and $v = [d_{12}, d_{21}, d_{23}, d_{32}, d_{13}, d_{31}]^T$. Then we obtain the system equations

$$x(12k + i) = A_i x(12k + i - 1) + B_i v, \qquad (10)$$

where A_i and B_i can be derived similarly as in Subsection III-A, $1 \leq i \leq 12$ and $k \geq 0$. By iteration, we have

$$x(12(k+1)) = A^{k+1} x(0) + \sum_{i=0}^{k} A^i B v, \quad k \geq 0,$$

where $A = A_{12} A_{11} \cdots A_1$ and $B = \sum_{i=1}^{12} A_{12} \cdots A_{i+1} B_i$. Following similar arguments to that in Subsection III-A, one can prove the following result.

Proposition 3: As k goes to infinity, $x_i(12k + r) - x_j(12k + r)$ converge to some constants for all $r = 1, \ldots, 12$, and $i, j = 1, 2, 3,\ i \neq j$.

Define $e_{ij}^r \triangleq \lim_{k \to \infty} (x_i(12k + r) - x_j(12k + r))$, where $i, j = 1, 2, 3,\ i \neq j$, and $r = 1, \ldots, 12$, $e^r \triangleq [e_{12}^r, e_{23}^r]^T$ and $e \triangleq [(e^1)^T, (e^3)^T, \ldots, (e^{11})^T]^T$. We can get the equation for the synchronization errors between clocks

$$e = \bar{A} e + \bar{B} v, \qquad (11)$$

where

$$\bar{A} = \begin{bmatrix} O & O & \cdots & O & \tilde{A}_1 \\ \tilde{A}_3 & O & \cdots & O & O \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ O & O & \ddots & O & O \\ O & O & \cdots & \tilde{A}_{11} & O \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_3 \\ \vdots \\ \tilde{B}_9 \\ \tilde{B}_{11} \end{bmatrix},$$

and $\tilde{A}_i$ and $\tilde{B}_i$ can be derived similarly as in Subsection III-A. Since $(I - \bar{A})$ is invertible, which can be proved using similar arguments as in Lemma 1, the error e can be calculated by $e = (I - \bar{A})^{-1} \bar{B} v$. Thus we have proved the following result.

Theorem 4: As time goes to infinity, the synchronization errors between each pair of distinct clocks in the three-clock connected undirected network will approach permanent oscillations among at most 6 values, which are determined by

$$e = (I - \bar{A})^{-1} \bar{B} v.$$

Although the synchronization errors between a pair of distinct clocks in general will oscillate, in some cases, the errors will converge.

Corollary 3: If $d_{12} + d_{23} + d_{31} = d_{13} + d_{32} + d_{21}$, then as time goes to infinity, the synchronization errors between clocks in the three-clock undirected network converge and

$$\lim_{t \to \infty} (x_i(t) - x_j(t)) = d_{ij} - \bar{D}_{ij},\ i \neq j.$$

Specifically, if the time delays are symmetric, namely $d_{ij} = d_{ji}\ i \neq j$, then the three clocks can get synchronized asymptotically.

In the next subsection, we extend the results that we have obtained for the three-clock connected network to general connected networks with bidirectional links.

B. Synchronizing more clocks in a connected undirected network

We consider a connected network consisting of n nodes and m undirected edges. For the ease of describing the message passing process, we assume that the edges have been labeled and in each update period, a pair of connected nodes exchange messages exactly once. The indices of the edges determine the ordering of the pair of nodes that are activated to exchange messages. For the two nodes associated with an edge, the one with the smaller index starts the message exchange process. For the sth edge of the graph, let $s_1 < s_2$ denote the indices of the associated two nodes. Then s_1 always sends a message to s_2 first, and then s_2 replies. Taking the three clocks in Subsection IV-A as an example, we label the edges $(1, 2)$, $(2, 3)$ and $(1, 3)$ by ①, ② and ③, respectively. For the 2nd edge $(2, 3)$, node $\textcircled{2}_1 = 2$ always sends a message to node $\textcircled{2}_2 = 3$ first, and after waiting for some idling time, node 3 sends back a message to node 2. Thus the message passing process can be illustrated more in detail in Fig. 4.

Using similar arguments to that in Subsection IV-A, one can obtain the following result.

Theorem 5: As time goes to infinity, the synchronization errors between each pair of distinct clocks in the n-clock connected undirected network will approach permanent oscillations among at most $2m$ values.

Networks with tree topologies are preferred when applying network clock synchronization protocols [5], the following corollary suggests the reason behind it.

Corollary 4: If the communication graph $\mathbb{G}$ is an undirected tree, the synchronization errors between clocks in the network converge and

$$\lim_{t\to\infty}(x_i(t) - x_j(t)) = d_{ij} - \bar{D}_{ij},\ i \neq j,$$

where $(i, j) \in \mathcal{E}$ and $\bar{D}_{ij} = \frac{1}{2}(d_{ij} + d_{ji})$.

In the next section, we discuss how to synchronize clocks in networks with strongly connected directed topologies.

V. Extension to strongly connected networks

In order to synchronize n clocks in a network with strongly connected topology, we may use only some of the edges in the network. To better explain this idea, we need to introduce some more notions.

For a graph $\mathbb{G} = (\mathcal{N}, \mathcal{E})$, a subgraph $\mathbb{G}' = (\mathcal{N}', \mathcal{E}')$ of $\mathbb{G}$ is a graph such that $\mathcal{N}' \subseteq \mathcal{N}$ and $\mathcal{E}' \subseteq \mathcal{E}$. Since $\mathbb{G}$ is strongly connected, we can find subgraphs $\mathbb{G}_i = (\mathcal{N}_i, \mathcal{E}_i),\ i = 1, \ldots, p$, of $\mathbb{G}$ such that $\cup_{i=1}^{p} \mathcal{N}_i = \mathcal{N}$ and each $\mathbb{G}_i$ is a directed ring graph. Those edges in $\cup_{i=1}^{p} \mathcal{E}_i$ are to be utilized in the message passing process. We divide each update period of the overall network into p stages. Each stage corresponds to a directed ring subgraph $\mathbb{G}_i$, in which the message passing process is the same as that in Subsection III-B. Note that $\mathbb{G}_i$ might share common edges and the nodes associated with these edges will carry out message passing more than once in each period. We take the message passing process in Subsection IV-A as an example since connected undirected graphs can always be viewed as strongly connected directed graphs. The graph corresponds to Fig. 4 is $\mathbb{G} = (\mathcal{N}, \mathcal{E})$ with $\mathcal{N} = \{1, 2, 3\}$ and $\mathcal{E} = \{(1, 2), (2, 1), (2, 3), (3, 2), (1, 3), (3, 1)\}$. Define $\mathbb{G}_i = (\mathcal{N}_i, \mathcal{E}_i),\ i = 1, 2, 3$, with $\mathcal{N}_1 = \{1, 2\}$, $\mathcal{E}_1 = \{(1, 2), (2, 1)\}$, $\mathcal{N}_2 = \{2, 3\}$, $\mathcal{E}_2 = \{(2, 3), (3, 2)\}$, and $\mathcal{N}_3 = \{1, 3\}$, $\mathcal{E}_3 = \{(1, 3), (3, 1)\}$. It is easy to check that $\cup_{i=1}^{3} \mathcal{N}_i = \mathcal{N}$ and $\mathbb{G}_i$ are directed ring graphs for $i = 1, 2, 3$. Thus each update period can be divided into 3 stages, and each stage corresponds to a subgraph $\mathbb{G}_i$. The message passing process in each stage is the same as that in Subsection III-B. Let $|\mathcal{E}|$ be the cardinality of the set $\mathcal{E}$. One can obtain the following result which is similar to that in the previous section.

Theorem 6: As time goes to infinity, the synchronization errors between each pair of distinct clocks in the n-clock strongly connected network will approach permanent oscillations among at most $\sum_{i=1}^{p} |\mathcal{E}_i|$ values.

VI. Concluding remarks

We have presented explicit expressions for the asymptotic synchronization errors between two interconnected clocks, and expanded the results to larger networks with directed ring topologies, connected undirected topologies, and general strongly connected directed topologies respectively. The obtained synchronization errors complements the impossibility results for clock synchronization in the literature. Our future research will focus on the determination of clock synchronization errors when the time delays are random, which is closer to reality in distributed data networks.

References

[1] K. Arvind. Probabilistic clock synchronization in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 5:474–487, 1994.

[2] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Athena Scientific, Nashua, 1997.

[3] M. Cao, A. S. Morse, and B. D. O. Anderson. Agreeing asynchronously. *IEEE Transactions on Automatic Control*, 53:1826–1838, 2008.

[4] R. Carli and S. Zampieri. Networked clock synchronization based on second order linear consensus algorithms. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 7259–7264, 2010.

[5] J. C. Eidson. *Measurement, Control, and Communication Using IEEE 1588*. Springer-Verlag, New York, 2006.

[6] N. Freris, H. Kowshik, and P. R. Kumar. Fundamentals of large sensor networks: Connectivity, capacity, clock and computation. *Proceedings of the IEEE*, 98:1828–1846, 2010.

[7] N. M. Freris, S. R. Graham, and P. R. Kumar. Fundamental limits on synchronizing clocks over networks. *IEEE Transactions on Automatic Control*, 55:1352–1364, 2011.

[8] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21:558–565, 1978.

[9] Q. Li and D. Rus. Global clock synchronization in sensor networks. *IEEE Transactions on Computers*, 55:214–226, 2006.

[10] J. Lundelius and N. Lynch. A new fault-tolerant algorithm for clock synchronization. In *Proceedings of the third ACM Symposium on Principles of Distributed Computing*, pages 75–88, 1984.

[11] J. Lundelius and N. Lynch. An upper and lower bound for clock synchronization. *Information and Control*, 62:190–204, 1984.

[12] D. L. Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 39:1482–1493, 1991.

[13] S. Palchaudhuri, A. K. Saha, and D. B. Johnson. Adaptive clock synchronization in sensor networks. In *Proceedings of the third ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2004.

[14] K. Romer. Time synchronization in Ad Hoc networks. In *Proceedings of the second ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2001.

[15] T. K. Srikanth and S. Toueg. Optimal clock synchronization. *Journal of the ACM*, 34:626–645, 1987.

[16] W. Su and F. Akyildiz. Time-diffusion synchronization protocol for wireless sensor networks. *IEEE Transactions on Networking*, 13:384–397, 2005.