DETERMINATION OF CLOCK SYNCHRONIZATION ERRORS IN DISTRIBUTED NETWORKS
WEIGUO XIA AND MING CAO

Abstract. It has been shown by Freris, Graham and Kumar that clocks in distributed networks cannot be synchronized precisely in the presence of asymmetric time delays even in idealized situations. Motivated by that impossibility result, we test under similar settings the performance of some existing clock synchronization protocols and show that the synchronization errors between neighboring nodes can be bounded within an acceptable level of accuracy that is determined by the degree of asymmetry in time delays. After studying the basic case of synchronizing two clocks 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.

Key words. Clock synchronization, time delay, distributed networks, stochastic matrices.

1. Introduction. As a class of collective behavior of groups of interacting units, synchronization has been discovered widely in natural, social and engineered networks and systems. The last decades have witnessed major advances in the understanding of synchronization phenomena of coupled dynamical systems [19, 26, 9, 18]. While 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 [5]. Clock synchronization has been discussed intensively in the area of theoretical computer science especially in the 1980’s [11, 23], and various impossibility results and bounds for synchronization errors have been reported [15, 14]. 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 [16, 1, 20, 17, 13, 25].

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 [6]. This result is obtained by using tools from linear system theory and consistent with the results obtained previously in theoretical computer science. Such impossibility results point out insightfully the fundamental limit of distributed clock synchronization strategies and underscore the urgent need to carry out in-depth theoretical analysis for various clock synchronization protocols. 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 [24], the Time-Diffusion synchronization Protocol (TDP) has been proposed to enable sensor networks to synchronize their clocks with bounded errors. In [12], 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. IEEE 1588 protocol [8] has been applied widely to networked measurement and control systems.

In this paper, we determine the clock synchronization errors based on the same models for...
clocks as in [6] 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 for neighboring nodes 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 packets 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 2, we review the models for clocks in [6]. In Section 3, we determine the asymptotic clock synchronization errors between two clocks through state augmentation. In Section 4, we analyze the synchronization errors in directed ring networks, in connected undirected networks and finally in strongly connected directed networks, respectively. Illustrative examples are given in Section 5.

2. Models for clocks in networks. As in [6], 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

$$x_i(t) = a_i t + b_i,$$

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 $t = 0$. Here we consider the idealized case 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.

Assume that the 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 [6] that when a packet 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 $d_{ji}$.

It has proven convenient to use graphs to describe topologies of general networks. A graph $G$ 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 $G$, there is a directed edge from node $i$ to $j$ if $i$ can send packets to $j$; correspondingly, there is an undirected edge between $i$ and $j$ if both $i$ and $j$ can send packets to each other. A directed path in a directed graph $G$ is a sequence of distinct nodes $i_1, \ldots, i_k$ such that $(i_s, i_{s+1}) \in \mathcal{E}$ for $s = 1, \ldots, k-1$. $G$ 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.

3. Synchronizing two clocks. In this section, we consider two clock synchronization. For analysis purposes, we can always describe the packet passing process with respect to the standard reference clock. In the sequel, we use the sequence $\{t_k\}$, $k \geq 0$, to denote the set of time instants embedded in the reference time axis $t$, at which a clock sends or receives packets. Then the packet 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 packet to node 2 and its current time-stamp $x_1(t_0)$ measured by its clock just before the transmission is included in the packet. Node 2 records the time $x_2(t_1)$ according to its local clock just after it receives the packet $x_1(t_0)$ and after a constant time $w_1$, it sends a packet including the time-stamp $x_2(t_2)$ at the time $t_2$ back to node 1. Correspondingly, node 1 receives this packet at time $t_3$ and records the time $x_1(t_3)$ according to its local clock. It then sends a packet after a constant time $w_2$. In this manner the packets are sent back and
forth. Each packet is allowed to contain information about all the past receiving times of all
prior packets recorded at that node. Hence, each node has the full knowledge of the sending and
receiving times of packets between them.

\[
\begin{align*}
D &= \frac{x_2(t_5) - x_2(t_1)}{x_1(t_4) - x_1(t_0)}, \\
d_{12} + d_{21} &= x_1(t_3) - x_1(t_0) - \frac{1}{a_2}(x_2(t_2) - x_2(t_1)).
\end{align*}
\]

However, the individual time delays \(d_{12}\) and \(d_{21}\) can never be determined precisely when they
are asymmetric and this is part of the synchronization impossibility result for a pair of clocks
shown in [6], which as argued in the same paper leads to synchronization errors that cannot be
eliminated. We also refer the interested reader to [10] for more information about phase and
skew estimators.

Now we try to synchronize the two clocks by repeatedly updating their displays. Consider
first the simple case when \(a_2 = 1\) as well; in other words, the skews of the two clocks are the same.
We use \(D\) to denote the round-trip time delay \(D \triangleq d_{12} + d_{21} = x_1(t_3) - x_1(t_0) - (x_2(t_2) - x_2(t_1))\).
Upon receiving a packet, the receiving node uses the average delay \(\bar{D} = \frac{D}{2}\) as the nominal delay
to compensate the time-stamped packet it receives about the most recent value of the other
clock’s display and it immediately updates its display to the average of its current display and
the estimated value of the other clock’s display. For example, when clock 1 receives a time-
stamped packet from clock 2 at time \(t_{k+1}\), it takes \(x_2(t_k) + \bar{D}\) as the estimated current value of
the display of clock 2. To get synchronized, it immediately updates its display to the average
of its current display \(x_1(t_{k+1})\) and \(x_2(t_k) + \bar{D}\). The same estimation and update strategies are
adopted by both of the two clocks. We assume the updates take place instantaneously and the
packet 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.
Although the two clocks update periodically according to their own clocks, since the clocks
have the same skew, 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 of equations of the updating process
of the two clocks after embedding can be written as

\[
\begin{align*}
&\begin{cases}
x_1(4k + 1) = x_1(4k) + d_{12} \\
x_2(4k + 1) = \frac{1}{2}((x_1(4k) + \bar{D}) + (x_2(4k) + d_{12}))
\end{cases} \\
&\begin{cases}
x_1(4k + 2) = x_1(4k + 1) + l_1 d_{12} \\
x_2(4k + 2) = x_2(4k + 1) + l_1 d_{12}
\end{cases} \\
&\begin{cases}
x_1(4k + 3) = \frac{1}{2}((x_2(4k + 2) + \bar{D}) + (x_1(4k + 2) + d_{21})) \\
x_2(4k + 3) = x_2(4k + 2) + d_{21}
\end{cases} \\
&\begin{cases}
x_1(4(k + 1)) = x_1(4k + 3) + l_2 d_{12} \\
x_2(4(k + 1)) = x_2(4k + 3) + l_2 d_{12},
\end{cases}
\end{align*}
\]

where \( k \geq 0 \) and \( l_i = \frac{a_i}{d_{12}} \), \( i = 1, 2 \).

We first show that during the above updating process (3.3), the synchronization error converges to a constant determined by the difference between the delays \( d_{12} \) and \( d_{21} \).

**Theorem 3.1.** 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 \( e(k) = x_1(k) - x_2(k) \) for \( k \geq 0 \). Then from (3.3), one has

\[
\begin{align*}
e(4k + 1) &= \frac{1}{2} e(4k) + \frac{1}{4} (d_{12} - d_{21}) \\
e(4k + 2) &= e(4k + 1) \\
e(4k + 3) &= \frac{1}{2} e(4k + 2) + \frac{1}{4} (d_{12} - d_{21}) \\
e(4(k + 1)) &= e(4k + 3).
\end{align*}
\]

Substituting the first three equations of (3.4) into the last equation of (3.4), we obtain

\[
e(4(k + 1)) = \left( \frac{1}{2} \right)^2 e(4k) + \frac{3}{8} (d_{12} - d_{21}) \\
= \left( \frac{1}{2} \right)^2 e(0) + \frac{3}{8} (d_{12} - d_{21}) + \sum_{i=0}^{k+1} \frac{1}{4^i}.
\]

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

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

Combining equation (3.5) with (3.4), one can check that

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

From (3.6), we know that for any \( \epsilon > 0 \), there exists a positive integer \( N \), such that for any \( n > N \), \(|e(4n + i) - \frac{1}{2} (d_{12} - d_{21})| < \epsilon, \ 1 \leq i \leq 4\). 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

\[
\lim_{k \to \infty} e(k) = \frac{1}{2} (d_{12} - d_{21}).
\]
This completes the proof. \( \square \)

Note that when applying the Network Time Protocol (NTP) [16], 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 3.1, when \( d_{12} = d_{21} \), the two clocks can indeed get synchronized precisely.

**Corollary 3.2.** When \( d_{12} = d_{21} \), the synchronization error \( x_1(t) - x_2(t) \) between the two clocks goes to zero asymptotically.

Now consider the general case when \( a_2 \) is different from 1. We first interpret Theorem 3.1 in a different way motivated by the approach proposed in [21]. Note that the models of the two clocks with the same skew are

\[
x_1(t) = t + b_1, \quad x_2(t) = t + b_2.
\]

Since the two clocks are with the same skew, to get them synchronized can be regarded as to synchronize the two clocks with respect to a virtual clock

\[
x(t) = t + b
\]

with \( b \) undetermined. Suppose that each clock has an estimate of the virtual clock

\[
\hat{x}_1(t) = t + b_1 + o_1(t), \quad \hat{x}_2(t) = t + b_2 + o_2(t).
\]

Thus the update of the displays of the two clocks in equation (3.3) is equivalent to the update of \( o_i(t) \) as follows

\[
\begin{align*}
o_1(4k + 1) &= o_1(4k) \\
o_2(4k + 1) &= o_2(4k) + \frac{1}{2}((\hat{x}_1(4k) + D) - (\hat{x}_2(4k) + d_{12})) \\
o_1(4k + 2) &= o_1(4k + 1) \\
o_2(4k + 2) &= o_2(4k + 1) \\
o_1(4k + 3) &= o_1(4k + 2) + \frac{1}{2}((\hat{x}_2(4k + 2) + D) - (\hat{x}_1(4k + 2) + d_{21})) \\
o_2(4k + 3) &= o_2(4k + 2) \\
o_1(4k + 4) &= o_1(4k + 3) \\
o_2(4k + 4) &= o_2(4k + 3),
\end{align*}
\]

where we use the notation \( o_i(k) \) instead of \( o_i(t_k) \), \( o_1(0) = o_2(0) = 0 \), and \( o_i(t) = o_i(k) \) for \( t \in [t_k, t_{k+1}) \). During the packet transmission process, the estimates of the virtual clock at that node are also included in the packet. Then Theorem 3.1 says that the difference between the estimates \( \hat{x}_1(t) - \hat{x}_2(t) = b_1 + o_1(t) - (b_2 + o_2(t)) \) converges to \( \frac{1}{2}(d_{12} - d_{21}) \) as \( t \) goes to infinity.

When the skews of the two clocks are different, consider the models

\[
x_1(t) = t + b_1, \quad x_2(t) = a_2 t + b_2,
\]

where \( a_2 \) is close to 1. Since the skew of clock 2 can be estimated through packet passing as shown in (3.1), a transformation of the model of clock 2 leads to the same-skew case

\[
\hat{x}_2(t) = \frac{1}{a_2} x_2(t) = t + \frac{1}{a_2} b_2.
\]

Let the estimates of a virtual clock be

\[
\hat{x}_1(t) = t + b_1 + o_1(t), \quad \hat{x}_2(t) = t + \frac{b_2}{a_2} + o_2(t).
\]
From Theorem 3.1, one has that 
\[
\hat{x}_i(t) - \hat{x}_2(t) = b_1 + o_1(t) - (t + o_2(t)) \text{ converges to } \frac{1}{2}(d_{12} - d_{21})
\]
as \( t \) goes to infinity. In other words, the result stated in Theorem 3.1 applies also to the general case when \( a_2 \neq 1 \).

In the next section, we will study how the main idea of compensation with nominal delays can be applied to larger networks by utilizing the packet passing mechanism just described.

4. Synchronizing clocks in networks. Now we consider a network of \( n \) clocks that are described by (2.1) with \( i = 1, \ldots, n, a_1 = 1, \) and \( a_i \) close to 1 for \( i = 2, \ldots, n. \) Since the skews \( a_i, i = 2, \ldots, n, \) of the clocks can be estimated through packet passing, similar to the discussion at the end of Section 3, a transformation will lead to the same-skew case
\[
\hat{x}_i(t) = \frac{1}{a_i} \bar{x}_i(t) = t + \frac{1}{a_i} b_i, \quad i = 2, \ldots, n.
\]

Hence, in what follows, we will only consider the case when the skews of the clocks are the same, namely \( a_i = 1 \) for all \( i. \)

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 the clocks, we take time delay \( d_{i,j} \), namely \( d_{1,2} = 1. \) Hence, in what follows, we will only consider the case when the skews of the clocks are the same, namely \( a_i = 1 \) for all \( i. \)

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 the clocks, we take time delay \( d_{i,j} \), namely \( d_{1,2} = 1. \) Hence, in what follows, we will only consider the case when the skews of the clocks are the same, namely \( a_i = 1 \) for all \( i. \)

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 the clocks, we take time delay \( d_{i,j} \), namely \( d_{1,2} = 1. \) Hence, in what follows, we will only consider the case when the skews of the clocks are the same, namely \( a_i = 1 \) for all \( i. \)

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 the clocks, we take time delay \( d_{i,j} \), namely \( d_{1,2} = 1. \) Hence, in what follows, we will only consider the case when the skews of the clocks are the same, namely \( a_i = 1 \) for all \( i. \)


4.1.1. 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 packet passing process for the 2-clock case discussed in the previous section, we illustrate the packet 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.

![Fig. 2. Message exchanges among three clocks with directed connections.](image)

Although the delays \( d_{12}, d_{23} \) and \( d_{31} \) cannot be determined from the time-stamped packets, the round-trip delay \( D = d_{12} + d_{23} + d_{31} \) can be determined precisely by
\[
D = x_1(5) - x_3(3) - x_2(2) + x_2(1) - x_1(0).
\]

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 packet from node 1, as an example. Upon receiving a packet from node 1 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 a packet with time-stamp \( x_2(2) \) 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 \( k \geq 0. \)

Now we write down the system of 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 $k \geq 0$,
\[
\begin{bmatrix}
  x_1(6k+1) \\
  x_2(6k+1) \\
  x_3(6k+1)
\end{bmatrix} = \begin{bmatrix}
  \frac{1}{2} (x_1(6k) + d_{12}) \\
  \frac{1}{2} (x_2(6k) + x_3(6k) + d_{12}) \\
  x_3(6k) + d_{12}
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 \\
  \frac{1}{2} & \frac{1}{2} & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x_1(6k) \\
  x_2(6k) \\
  x_3(6k)
\end{bmatrix} + \begin{bmatrix}
  1 & 0 & 0 \\
  \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\
  1 & 0 & 0
\end{bmatrix} \begin{bmatrix}
  d_{12} \\
  d_{23} \\
  d_{31}
\end{bmatrix}.
\]

Through a similar procedure, one can obtain
\[(4.1) \quad x(6k+i) = A_ix(6k+i-1) + B_iv, \quad 1 \leq i \leq 6,
\]

where
\[
A_1 = \begin{bmatrix}
  1 & 0 & 0 \\
  \frac{1}{2} & \frac{1}{2} & 0 \\
  0 & 0 & 1
\end{bmatrix}, \quad A_3 = \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 1 & 0 \\
  0 & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}, \quad A_5 = \begin{bmatrix}
  \frac{1}{2} & 0 & \frac{1}{2} \\
  0 & 1 & 0 \\
  0 & 0 & 1
\end{bmatrix},
\]
\[
B_1 = \begin{bmatrix}
  1 & 0 & 0 \\
  \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\
  1 & 0 & 0
\end{bmatrix}, \quad B_3 = \begin{bmatrix}
  0 & 1 & 0 \\
  0 & 1 & 0 \\
  \frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}, \quad B_5 = \begin{bmatrix}
  \frac{1}{2} & \frac{1}{6} & \frac{2}{3} \\
  0 & 0 & 1 \\
  0 & 0 & 1
\end{bmatrix},
\]
\[
A_2 = A_4 = A_6 = I_3, \quad B_{2j} = l_j \begin{bmatrix}
  1 & 0 & 0 \\
  0 & 0 & 0 \\
  0 & 0 & 0
\end{bmatrix}, \quad j = 1, 2, 3.
\]

Here, $I_3$ is the 3-dimensional identity matrix, $1_3$ is the 3-dimensional all-one column vector, $O_{3 \times 2}$ is the $3 \times 2$ zero matrix, and $l_j = \frac{w_j}{d_{ij}}$. We can further obtain the following system of equations in an iterative form
\[
x(6k+1) = A_6A_5 \cdots A_1x(6k) + \sum_{i=1}^{6} A_6 \cdots A_{i+1}B_iv.
\]

Define $A \triangleq A_6A_5 \cdots A_1 = A_5A_3A_1$ and $B \triangleq \sum_{i=1}^{6} A_6 \cdots A_{i+1}B_i$, then we have
\[(4.2) \quad x(6(k+1)) = A^{k+1}x(0) + \sum_{i=0}^{k} A^iBv, \quad k \geq 0.
\]

We first prove the following convergence result.

**Proposition 4.1.** As $k$ goes to infinity, $x_i(6(k+1)) - x_j(6(k+1))$ converge to some constants for $i, j = 1, 2, 3, \ i \neq j$.

To prove this proposition, we need to use some results about stochastic matrices and scrambling matrices. A matrix $P \in \mathbb{R}^{m \times m}$ is a stochastic matrix if its entries are all nonnegative and its row sums are all equal to one. A stochastic matrix $P$ is called a scrambling stochastic matrix if for any pair of distinct row indices $i$ and $j$, there always exists a column index $k$ such that both $p_{ik}$ and $p_{jk}$ are positive.

For a stochastic matrix $P$, define
\[
\gamma(P) \triangleq 1 - \min_{1 \leq i,j \leq m} \sum_{k=1}^{m} \min\{p_{ik}, p_{jk}\}.
\]
Then $0 \leq \gamma(P) \leq 1$ and $\gamma(P)$ is strictly less than one if $P$ is a scrambling matrix.

**Lemma 4.2.** Let $P = (p_{ij})_{m \times m}$ be a square stochastic matrix. Then for all $y \in \mathbb{R}^m$

$$
\varphi(Py) \leq \gamma(P)\varphi(y),
$$

where $\varphi(\cdot) : \mathbb{R}^m \to \mathbb{R}^+$ is defined by

$$
\varphi(y) \triangleq \max_{1 \leq i \leq m} y_i - \min_{1 \leq i \leq m} y_i.
$$

For any matrix $Q = [\alpha_1 \alpha_2 \cdots \alpha_m]^T \in \mathbb{R}^{m \times m}$, where $\alpha_i \in \mathbb{R}^m$, $i = 1, \ldots, m$, are column vectors, we use

$$
\sigma_{ij}(Q) \triangleq \alpha_i^T - \alpha_j^T
$$

to denote the difference between the $i$th and $j$th rows of $Q$, where $1 \leq i, j \leq m$, $i \neq j$. The following result will be useful in the proof for Proposition 4.1.

**Lemma 4.3.** Let $P \in \mathbb{R}^{m \times m}$ be a scrambling matrix, then $\sigma_{ij}(\sum_{i=0}^k P^i)$ converge to some constant row vectors as $k$ goes to infinity for $1 \leq i, j \leq m$, $i \neq j$.

**Proof:** Since the convergence to be proved is meant in the element-wise sense, without loss of generality, we only need to prove the convergence of the first element of $\sigma_{12}(\sum_{i=0}^k P^i)$.

Let $(\sigma_{12}(P^i))_1$ be the first element of $\sigma_{12}(P^i)$ for $i \geq 0$ and $(\sigma_{12}(\sum_{i=0}^k P^i))_1$ be the first element of $\sigma_{12}(\sum_{i=0}^k P^i)$. It is easy to check that the operator $\sigma_{12}(\cdot)$ has the property that

$$
(\sigma_{12}(\sum_{i=0}^k P^i))_1 = \sum_{i=0}^k (\sigma_{12}(P^i))_1.
$$

Thus it suffices to prove that $\sum_{i=0}^\infty (\sigma_{12}(P^i))_1$ converges. Towards this end, let $u_i = (\sigma_{12}(P^i))_1$ and

$$
P^i = \begin{bmatrix} \eta_1^{(i)} & \eta_2^{(i)} & \cdots & \eta_m^{(i)} \end{bmatrix},
$$

where $\eta_1^{(i)}, \ldots, \eta_m^{(i)}$ are $m$ column vectors of the matrix $P^i$. Denote $\varphi(\eta_1^{(1)}) = M \geq 0$, then one can prove by induction using Lemma 4.2 that

$$
\varphi(\eta_1^{(i)}) \leq \gamma(P)^{i-1}M, \quad i \geq 1.
$$

Let $s_0 = 1$ and $s_i = \gamma(P)^{i-1}M$, $i \geq 1$, then in view of the definitions of $\sigma_{12}(\cdot)$ and $\varphi(\cdot)$, one has

$$
|u_i| = |(\sigma_{12}(P^i))_1| \leq \varphi(\eta_1^{(i)}) \leq s_i
$$

for $i \geq 1$. It is obvious that $u_0 = 1$ because $P^0 = I_m$. Then we know $|u_i| \leq s_i$ for all $i \geq 0$. Since $P$ is a scrambling matrix, $0 \leq \gamma(P) < 1$, which implies the convergence of the series

$$
\sum_{i=0}^\infty s_i = 1 + \sum_{i=1}^\infty \gamma(P)^{i-1}M = 1 + \frac{M}{1 - \gamma(P)}.
$$

Hence, $\sum_{i=0}^\infty |u_i|$ converges, and so does $\sum_{i=0}^\infty u_i$. This completes the proof. \qed

Now we are ready to prove Proposition 4.1.

8
By iteration, one has

\[
\frac{\sigma_{ij}(A^k)}{k} = x_k(6(k + 1)) - x_j(6(k + 1)) = \sigma_{ij}(A^{k+1})x(0) + \sigma_{ij}(\sum_{i=0}^{k} A^i)Bv.
\]

As \( k \to \infty \), it follows from Lemma 4.3 that \( x_i(6(k+1)) - x_j(6(k+1)) \) converge to some constants.

\[
\lim_{k \to \infty} \sigma_{ij}(A^k) = 0, \quad 1 \leq i, j \leq 3, \quad i \neq j.
\]

In view of (4.2), one has

\[
x_i(6(k + 1)) - x_j(6(k + 1)) = \sigma_{ij}(A^{k+1})x(0) + \sigma_{ij}(\sum_{i=0}^{k} A^i)Bv.
\]

As \( k \to \infty \), it follows from Lemma 4.3 that \( x_i(6(k+1)) - x_j(6(k+1)) \) converge to some constants.

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

\[
x_i(6(k + r)) - x_j(6(k + r)) = x_i(6(k + r - 1)) - x_j(6(k + r - 1)),
\]

hold for \( r = 2, 4, 6 \), one can get the following conclusion.

**Proposition 4.4.** As \( k \to \infty \), \( x_i(6(k+1)) - x_j(6(k+1)) \) converge to some constants for all \( r = 1, \ldots, 6 \), and \( i, j = 1, 2, 3, \quad i \neq j \).

From Proposition 4.4, we know that we can define

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

and the constants

\[
e_{ij}^o = \lim_{k \to \infty} e_{ij}(6k + r), \quad e^o = [e_{12}^o, e_{23}^o]^T,
\]

where \( i, j = 1, 2, 3, \quad i \neq j \), and \( r = 1, \ldots, 6 \). From the system of equations (4.1), one can get a set of equations

\[
e(6k + i) = \tilde{A}_i e(6k + i - 1) + \tilde{B}_i v, \quad 1 \leq i \leq 6,
\]

where

\[
\tilde{A}_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{bmatrix}, \quad \tilde{A}_3 = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}, \quad \tilde{A}_5 = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ 0 & 1 \end{bmatrix},
\]

\[
\tilde{B}_1 = \begin{bmatrix} \frac{1}{3} & -\frac{1}{6} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{1}{2} & -\frac{1}{6} \end{bmatrix}, \quad \tilde{B}_3 = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{6} & \frac{1}{2} & -\frac{1}{6} \end{bmatrix}, \quad \tilde{B}_5 = \begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \end{bmatrix},
\]

\[
\tilde{A}_2 = \tilde{A}_4 = \tilde{A}_6 = I_2, \quad \tilde{B}_2 = \tilde{B}_4 = \tilde{B}_6 = O_{2 \times 3}.
\]

By iteration, one has

\[
e(6(k + 1)) = \tilde{A}_6 \tilde{A}_5 \cdots \tilde{A}_1 e(6k) + \sum_{i=1}^{6} \tilde{A}_6 \cdots \tilde{A}_{i+1} \tilde{B}_i v
\]

\[
= \tilde{A}^{k+1} e(0) + \sum_{i=0}^{k} \tilde{A}^i \tilde{B} v, \quad k \geq 0,
\]
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 4.4, 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 $e^{r+1} = e^r$, $r = 1, 3, 5$, we define

$$e \triangleq [(e^1)^T, (e^3)^T, (e^5)^T]^T.$$  

Then we get the equation of the asymptotic synchronization errors between clocks by taking $k$ on both sides of (4.3) to infinity:

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

where

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

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

**Lemma 4.5.** The matrix $I - \tilde{A}$ is invertible.

**Proof:** We prove this Lemma by showing that $(I - \tilde{A}) y = 0$ has a unique solution $y = 0$. From

$$(I - \tilde{A}) y = \begin{bmatrix} I & O & -\tilde{A}_1 \\ -\tilde{A}_3 & I & O \\ O & -\tilde{A}_5 & I \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = 0,$$

one has

(4.4)

$$y_1 = \tilde{A}_1 y_3, \quad y_2 = \tilde{A}_3 y_1, \quad y_3 = \tilde{A}_5 y_2.$$  

Substituting the first two equations of (4.4) into the last one of (4.4), we obtain

$$y_3 = \tilde{A}_5 \tilde{A}_3 \tilde{A}_1 y_3 = \tilde{A}_5 y_3.$$  

Since $\rho(\tilde{A}) < 1$, $y_3 = \tilde{A}_5 y_3$ has a unique solution $y_3 = 0$. Substituting $y_3 = 0$ into the equations (4.4), one has $y = 0$. $\square$

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

$$e_r = d_{i+1} - \tilde{D}, \quad d_r = d_{i+1} - \tilde{D}, \quad r = 1, 2, 3.$$  

Hence, we have proved the following.

**Theorem 4.6.** As time goes to infinity, the synchronization errors between clocks in the three-clock directed ring network converge and

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

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

The following result is a direct consequence of Theorem 4.6.

**Corollary 4.7.** For the three clocks in the directed 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 directed ring network to general directed ring networks with $n \geq 3$ nodes.
4.1.2. Synchronizing more clocks in a directed ring network. Now we consider a
directed ring network of $n \geq 3$ nodes. The packet passing procedure in the network with
unidirectional communications is illustrated in Fig. 3, where $d_{i,[i]}$ and $w_i$, $i = 1, \ldots, n$, are time
delays and idling times respectively. Here, $[i]$ is defined to be $i+1$ when $i = 1, \ldots, n-1$ and 1
when $i = n$.

![Figure 3](image)

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

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

Similar to the three-clock case in Subsection 4.1.1, we use $\bar{D} = \frac{D}{n}$ as the nominal delay for all
the clocks when they update their displays.

Define

$$x(k) = [x_1(k), x_2(k), \ldots, x_n(k)]^T, \quad v = [d_{i,12}, d_{i,23}, \ldots, d_{i,n}]^T.$$

Then we have the system of equations in state space

$$x(2nk + i) = A_i x(2nk + i - 1) + B_i v,$$

for $1 \leq i \leq 2n$ and $k \geq 0$, where

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad A_{2n-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & 0 & \cdots & \frac{1}{2} \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} \frac{1}{n+1} & 0 & 0 & \cdots & 0 \\ \frac{1}{2n} & 0 & 0 & \cdots & 0 \\ \frac{1}{2n} & \frac{1}{2n} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2n} & \frac{1}{2n} & \frac{1}{2n} & \cdots & \frac{1}{2n} \end{bmatrix}, \quad B_{2n-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \frac{1}{2n} & 0 & 0 & \cdots & \frac{1}{2n} \\ 0 & 0 & \cdots & \frac{1}{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

$$A_2 = A_4 = \cdots = A_{2n} = I_n, \quad B_{2j} = l_j \begin{bmatrix} 1_n & O_{n \times (n-1)} \end{bmatrix}, \quad j = 1, \ldots, n.$$
and $l_j = \frac{w_j}{x_{l_{12}}}$. We can write down the iterative equations

$$x(2n(k + 1)) = A_{2n}A_{2n-1} \cdots A_1x(2nk) + \left(\sum_{i=1}^{2n} A_{2n} \cdots A_{i+1}B_i\right)v$$

$$= A^{k+1}x(0) + \sum_{i=0}^{k} A^iBv.$$
We take $D_{ij} = \frac{d_{i,j}}{2}$ as the nominal delay for a pair of adjacent clocks $i$ and $j$ when they update their displays, where $D_{ij} = d_{i,j} + d_{j,i}$ 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 packets exactly once.

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

$$x(12(k + 1)) = A_ix(12k + i - 1) + B_iv,$$

for $1 \leq i \leq 12$ and $k \geq 0$, where

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

$$A_7 = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_9 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}, \quad A_{11} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} \frac{3}{4} & 0 & 0 \\ \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_3 = \begin{bmatrix} \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_5 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{4} & \frac{1}{4} & 0 & 0 \end{bmatrix},$$

$$B_7 = \begin{bmatrix} \frac{3}{4} & 0 & 0 \\ \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_9 = \begin{bmatrix} \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_{11} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3}{4} & \frac{1}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$A_2 = \cdots = A_{12} = I_3, \quad B_{12} = l_j \begin{bmatrix} 1_3 & O_{3 \times 5} \end{bmatrix}, \quad 1 \leq j \leq 6.$$

Here $l_j = \frac{w_j}{\Delta r_{ij}}$. By iteration, we have

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

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 4.1.1, one can prove the following result.

**Proposition 4.9.** 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

$$e_{ij}(12k+r) \triangleq x_i(12k+r) - x_j(12k+r),$$

$$e(12k+r) \triangleq [e_{12}(12k+r), e_{23}(12k+r)]^T,$$

and the constants

$$e_{ij}^r \triangleq \lim_{k \to \infty} e_{ij}(12k+r), \quad e^r \triangleq [e_{12}^r, e_{23}^r]^T,$$

where $i, j = 1, 2, 3$, $i \neq j$, and $r = 1, \ldots, 12$. From the system of equations (4.6), one gets a set of equations

$$e(12k+i) = \tilde{A}_i e(12k+i-1) + \tilde{B}_i v, \quad 1 \leq i \leq 12,$$
that as the error $e$ approaches the equation for the synchronization errors between clocks

takes the exact average of its current display and the latest estimation of the other clock’s
current displays. For example, if every clock updates its display using a stochastic approximation
algorithm with diminishing stepsizes when it receives a packet, the synchronization errors behave

1. In the three-clock directed ring network, the synchronization errors between clocks converge to some constants; for example, $\lim_{t \to \infty} (x_1(t) - x_2(t)) = d_{12} = \bar{D} = e_{12}$ for all $r = 1, \ldots, 6$. However, in the three-clock connected undirected network, the synchronization errors between a pair of distinct clocks may not converge, which in general will oscillate; for example, $\lim_{t \to \infty} (x_1(t) - x_2(t))$ may not exist because $e_{12}^r$ may not be equal to $e_{12}^r$ for some $r_1, r_2, 1 \leq r_1, r_2 \leq 12$.

2. In the present setting, the synchronization errors between pairs of clocks in general oscillate among several values. Different update strategies may be adopted by a clock instead of taking the exact average of its current display and the latest estimation of the other clock’s current displays. For example, if every clock updates its display using a stochastic approximation algorithm with diminishing stepsizes when it receives a packet, the synchronization errors behave
differently, although not guaranteed to converge; however, the corresponding analytical analysis will be more complicated. An algorithm that always leads to converging synchronization errors between clocks is an interesting open research topic.

Although the synchronization errors between a pair of distinct clocks in general will oscillate, it is easy to see that if

\[(4.9)\]

\[e^{r_1} = e^{r_2}, \forall r_1, r_2 = 1, 3, \ldots, 11,\]

then the errors converge to some constant values. Substituting (4.9) into (4.8), one has

\[e^{r_1}_{12} = d_{12} - D_{12}, \quad e^{r_1}_{23} = d_{23} - D_{23}, \quad e^{r_1}_{12} + e^{r_1}_{23} = d_{13} - D_{13},\]

where \(r = 1, 3, \ldots, 11\). Since (4.8) has a unique solution \(e = (I - A)^{-1}Bv\), we can conclude that if \(d_{12} - D_{12} + d_{23} - D_{23} = d_{13} - D_{13}\), namely, \(d_{12} + d_{23} + d_{31} = d_{13} + d_{32} + d_{21}\), then \(e^{r_1}_{12} = d_{12} - D_{12}, \quad e^{r_1}_{23} = d_{23} - D_{23}, \quad r = 1, 3, \ldots, 11\), is indeed the solution to (4.8). We summarize.

**Corollary 4.11.** 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

\[\lim_{t \to \infty} (x_i(t) - x_j(t)) = d_{ij} - D_{ij}, \quad i \neq j.\]

Specifically, if the time delays are symmetric, namely \(d_{ij} = d_{ji}, \quad i \neq j\), 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.

**4.2.2. 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 packet passing process, we assume that the edges have been labeled and in each update period, a pair of connected nodes exchange packets exactly once. The indices of the edges determine the ordering of the pair of nodes that are activated to exchange packets. For the two nodes associated with an edge, the one with the smaller index starts the packet exchange process. For the 3th edge of the graph, let \(s_1 < s_2\) denote the indices of the associated two nodes. Then \(s_1\) always sends a packet to \(s_2\) first, and then \(s_2\) replies. Taking the three clocks in Subsection 4.2.1 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 \(2_1\) = 2 always sends a packet to node \(2_2\) = 3 first, and after waiting for some idling time, node 3 sends back a packet to node 2. Thus the packet passing process can be illustrated more in detail in Fig. 4.

Define \(x(k) = [x_1(k), \ldots, x_n(k)]^T\), and \(v = [d_{i_1,1}, d_{i_2,1}, \ldots, d_{m_1,m_2}, d_{m_2,m_1}]^T\). Then we can derive the system of equations through a similar procedure as that in Subsection 4.1.1

\[(4.10)\]

\[x(4mk + i) = A_x x(4mk + i - 1) + B_x v, \quad 1 \leq i \leq 4m, \quad k \geq 0,\]

where

\[A_x = \cdots = A_{4m} = I_n, \quad B_{2j} = l_j \begin{bmatrix} I_n & O_{n \times (2m-1)} \end{bmatrix},\]

and \(l_j = \frac{w_j}{d_{i_1,1}}, \quad w_j\) are idling times for \(1 \leq j \leq 2m\), and when \(i = 4(s-1) + 1\), for \(1 \leq s \leq m\),

\[A_i = \text{diag}\{I_{s_1-1}, A_i', I_{n-s_2}\},\]

\[B_i = \begin{bmatrix} O_{n \times (2s-2)} & B_i' \end{bmatrix},\]

\[C_i = \begin{bmatrix} O_{n \times (m-2s)} & C_i' \end{bmatrix},\]

\[D_i = \begin{bmatrix} O_{n \times (m-2s)} & D_i' \end{bmatrix}.\]
with

\[ A'_i = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2} & 0 & \cdots & \frac{1}{2} \end{bmatrix}, \quad B'_i = \begin{bmatrix} 1_{s_1-1} & 0_{s_1-1} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}, \]

where \( 0_{s_1-1} \) is the \((s_1 - 1)\)-dimensional all-zero column vector. When \( i = 4(s - 1) + 3 \), for \( 1 \leq s \leq m \),

\[ A_i = diag\{I_{s_1-1}, A'_i, I_{n-s_2}\}, \]
\[ B_i = [O_{n \times (2s-2)} \quad B'_i \quad O_{n \times (2m-2s)}], \]

with

\[ A'_i = \begin{bmatrix} \frac{1}{2} & 0 & \cdots & \frac{1}{2} \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \quad B'_i = \begin{bmatrix} 0_{s_1-1} & 1_{s_1-1} \\ \frac{1}{4} & \frac{3}{4} \end{bmatrix}. \]

We can further obtain

\[ x(4m(k+1)) = A^{k+1}x(0) + \sum_{j=0}^{k} A'^j B v, \quad k \geq 0, \]

where \( A = A_{4m} A_{4m-1} \cdots A_1 \) and \( B = \sum_{i=1}^{4m} A_{4m} \cdots A_{i+1} B_i \). Following similar arguments to that in Subsection 4.1.1, we can conclude that as \( k \) goes to infinity, \( x_i(4mk + r) \to x_i(4mk + r) \) converge to some constants for all \( r = 1, \ldots, 4m \), and \( i, j = 1, \ldots, n, i \neq j \).

Then define \( e_{ij}(4mk + r) \triangleq x_i(4mk + r) - x_j(4mk + r) \), \( e(4mk + r) \triangleq [e_{12}(4mk + r), e_{23}(4mk + r), \ldots, e_{n-1,n}(4mk + r)]^T \), and the constants \( e_{ij}^r \triangleq \lim_{k \to \infty} e_{ij}(4mk + r) \), \( e^r \triangleq [e_{12}^r, e_{23}^r, \ldots, e_{n-1,n}^r]^T \), where \( i, j = 1, \ldots, n, i \neq j \), and \( r = 1, \ldots, 4m \). From the system of equations (4.10), one can get a set of equations

\[ (4.11) \quad e(4mk + i) = \tilde{A}_i e(4mk + i - 1) + \tilde{B}_i v, \quad 1 \leq i \leq 4m, \]

where

\[ \tilde{A}_2 = \cdots = \tilde{A}_{4m} = I_{n-1}, \quad \tilde{B}_2 = \cdots = \tilde{B}_{4m} = O_{(n-1) \times 2m}, \]

and when \( i = 4(s - 1) + 1 \), for \( 1 \leq s \leq m \),

\[ \tilde{A}_i = diag\{I_{s_1-1}, \tilde{A}'_i, I_{n-s_2}\}, \]
\[ \tilde{B}_i = [O_{(n-1) \times (2s-2)} \quad \tilde{B}'_i \quad O_{(n-1) \times (2m-2s)}], \]

with

\[ \tilde{A}'_i = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 & 0 \\ -\frac{1}{2} & \cdots & -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}, \quad \tilde{B}'_i = \begin{bmatrix} 0_{s_2-2} & 0_{s_2-2} \\ \frac{1}{4} & -\frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix}. \]
when $i = 4(s - 1) + 3$, for $1 \leq s \leq m$,

\[ \tilde{A}_i = \text{diag}\{I_{s-2}, \tilde{A}'_i, I_{n-s} \}, \]
\[ \tilde{B}_i = [O_{(n-1)\times(2s-2)} \ O_{(n-1)\times(2m-2s)}], \]

with

\[
\tilde{A}'_i = \begin{bmatrix}
1 & \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} \\
0 & 1 & -\frac{1}{2} & \cdots & -\frac{1}{2} \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1
\end{bmatrix}, \quad \tilde{B}'_i = \begin{bmatrix}
0_{s-2} & 0_{s-2} \\
\frac{1}{2} & \frac{1}{2} \\
0_n & 0_{n-s-1} \\
0_{n-s-1} & 0_{n-s-1}
\end{bmatrix}.
\]

Let

\[ e \triangleq [(e^1)^T, (e^3)^T, \ldots, (e^{4m-1})^T]^T. \]

Then the equation of the synchronization errors can be written as

\[ e = \tilde{A}e + \tilde{B}v, \]

where

\[
\tilde{A} = \begin{bmatrix}
O & O & \cdots & O & \tilde{A}_1 \\
\tilde{A}_3 & O & \cdots & O & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
O & O & \cdots & O & 0 \\
O & O & \cdots & \tilde{A}_{4m-1} & O
\end{bmatrix}, \quad \tilde{B} = \begin{bmatrix}
\tilde{B}_1 \\
\tilde{B}_3 \\
\vdots \\
\tilde{B}_{4m-3} \\
\tilde{B}_{4m-1}
\end{bmatrix}.
\]

Since the matrix $I - \tilde{A}$ is invertible, the errors can be calculated by $e = (I - \tilde{A})^{-1} \tilde{B}v$.

**Theorem 4.12.** As time goes to infinity, the synchronization errors between each pair of distinct clocks in the $n$-clock undirected connected network will approach permanent oscillations among at most $2m$ values, which are determined by

\[ e = (I - \tilde{A})^{-1} \tilde{B}v. \]

Networks with tree topologies are preferred when applying network clock synchronization protocols [4], the following corollary suggests the reason behind it.

**Corollary 4.13.** If the communication graph $\mathcal{G}$ is an undirected tree, the synchronization errors between clocks in the network converge and

\[ \lim_{t \to \infty} (x_i(t) - x_j(t)) = d_{ij} - \tilde{D}_{ij}, \quad i \neq j, \]

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

For the synchronization errors between two non-neighboring nodes, we have the following discussion.

**Remark 3.** Let $\max_{(i,j) \in \mathcal{E}} |d_{ij} - \tilde{D}_{ij}| = \delta$. Since $\mathcal{G}$ is connected, for two non-neighboring nodes $i, j$, there exists a shortest path $i_0, i_1, \ldots, i_s$ of length $s$ connecting $i$ and $j$, where $i_0 = i$ and $i_s = j$. Hence, $\lim_{t \to \infty} (x_i(t) - x_j(t)) = \sum_{k=0}^{s-1} (d_{i_k,i_{k+1}} - \tilde{D}_{i_k,i_{k+1}}) \leq s\delta$. The synchronization errors between two non-neighboring nodes could increase linearly as the length of the shortest path between them increases.

In the next subsection, we discuss how to synchronize clocks in networks with strongly connected directed topologies.
4.3. Expansion to strongly connected directed networks. In order to synchronize \( n \) clocks in a network with strongly connected directed 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 \( G = (\mathcal{N}, \mathcal{E}) \), a subgraph \( G' = (\mathcal{N}', \mathcal{E}') \) of \( G \) is a graph such that \( \mathcal{N}' \subseteq \mathcal{N} \) and \( \mathcal{E}' \subseteq \mathcal{E} \). Since \( G \) is strongly connected, we can find subgraphs \( G_i = (\mathcal{N}_i, \mathcal{E}_i) \), \( i = 1, \ldots, p \), of \( G \) such that \( \bigcup_{i=1}^p \mathcal{N}_i = \mathcal{N} \) and each \( \mathcal{G}_i \) is a directed ring graph. Those edges in \( \bigcup_{i=1}^p \mathcal{E}_i \) are to be utilized in the packet passing process. We divide each update period of the overall network into \( p \) stages. Each stage corresponds to a directed ring subgraph \( G_i \), in which the packet passing process is the same as that in Subsection 4.1.2. Note that \( G_i \) might share common edges and the nodes associated with these edges will carry out packet passing more than once in each period.

We take the packet passing process in Subsection 4.2.1 as an example since connected undirected graphs can always be viewed as strongly connected directed graphs. The graph corresponds to the case when the delays take random values in the intervals \( d \) where receiving a packet before sending its own packet, namely \( D \) may be idling times to be \( (1 - d_1), (2 - d_1), (2, 3), (3, 2), (1, 3), (3, 1) \). Define \( \mathcal{G}_1 = (\mathcal{N}_1, \mathcal{E}_1) \), \( i = 1, 2, 3 \), 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 \( \bigcup_{i=1}^3 \mathcal{N}_i = \mathcal{N} \) and \( \mathcal{G}_i \) 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 \( \mathcal{G}_i \). The packet passing process in each stage is the same as that in Subsection 4.1.2.

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 4.14.** 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.

The synchronization errors between clocks, which in general will oscillate, are determined by the choices of the subgraphs and the time delays. A proper choice of the subgraphs can lead to the convergence of synchronization errors. One example is that if we only choose the subgraphs \( \mathcal{G}_1, \mathcal{G}_2 \) of \( G \) defined in the previous example for packet passing, the synchronization errors will converge to some constants in view of Corollary 4.13.

5. Illustrative examples. **Example 1.** We first consider three clocks with the same skew in a directed ring network, whose packet passing procedure is shown in Fig. 2. The three time delays, not known by the clocks, are \( d_{12} = 0.2 \), \( d_{23} = 0.4 \), and \( d_{31} = 0.9 \). Then the round-trip delay \( D = 1.5 \) and the nominal delay \( \bar{D} = 0.5 \). Every clock waits for one time unit after receiving a packet before sending its own packet, namely \( w_i = 1 \), \( i = 1, 2, 3 \). We set the initial time displays of the three clocks to be \( |x(0)| = [10, 40, 20]^T \). The simulation results of the evolution of the displays of the three clocks are shown in Fig. 5. One can see that the three clocks do not synchronize, but the asymptotic synchronization error between clocks 1 and 2 is \( -0.3 \) and that between clocks 2 and 3 is \( -0.1 \), which agrees with our theoretical analysis. If we set all the three time delays to be equal, namely \( d_{12} = d_{23} = d_{31} = 0.5 \), then from Theorem 4.6 it follows that the three clocks are synchronized asymptotically as shown in Fig. 6.

Since the time delays are random variables in real distributed networks, we re-run the simulation for the case when the delays take random values in the intervals \( d_{12} \in [0.15, 0.25] \), \( d_{23} \in [0.3, 0.5] \), and \( d_{31} \in [0.8, 1.0] \). The expected round-trip delay is still \( D = 1.5 \), and thus \( \bar{D} = 0.5 \). The simulation results are shown in Fig. 7, from which one can tell the clock synchronization errors are bounded in a small range.

**Example 2.** We consider three clocks with the same skew in an undirected network, whose packet passing procedure is shown in Fig. 4. The time delays are \( d_{12} = 0.2 \), \( d_{21} = 0.3 \), \( d_{23} = 0.3 \), \( d_{32} = 0.4 \), \( d_{13} = 0.4 \), and \( d_{31} = 0.3 \). Thus the nominal delays for the three pair of clocks \( (1, 2), (2, 3) \), and \( (1, 3) \) are \( \bar{D}_{12} = 0.25 \), \( \bar{D}_{23} = 0.35 \), and \( \bar{D}_{13} = 0.35 \) respectively. We set the idling times to be \( w_i = 0.5 \), \( i = 1, \ldots, 6 \), and the initial time displays of the three clocks to be
find that the clock synchronization errors are still bounded in a small range. 

The three clocks are shown in Fig. 5.

The simulation results of the evolution of the displays of the three clocks are shown in Fig. 8, from which one can find that the synchronization errors are bounded in a small range without converging to some constants.

When we rerun the simulations for the case when the delays take random values in the intervals $d_{12} \in [0.15, 0.25]$, $d_{21} \in [0.2, 0.4]$, $d_{23} \in [0.2, 0.4]$, $d_{24} \in [0.3, 0.5]$, $d_{41} \in [0.3, 0.5]$, and $d_{31} \in [0.2, 0.4]$. The expected nominal delays are still the same as above. From Fig. 9 one can find that the clock synchronization errors are still bounded in a small range.

**Example 3.** We consider four clocks with the same skew in a strongly connected graph $G = (\mathcal{N}, \mathcal{E})$ with $\mathcal{N} = \{1, 2, 3, 4\}$ and $\mathcal{E} = \{(1, 2), (2, 3), (3, 1), (2, 4), (4, 1)\}$ as shown in Fig. 10. Let $G_1 = (\mathcal{N}_1, \mathcal{E}_1) = \{(1, 2), (2, 3), (3, 1)\}$ and $G_2 = (\mathcal{N}_2, \mathcal{E}_2) = \{(1, 2), (2, 4), (4, 1)\}$. Then $G_1$ and $G_2$ are both directed ring graphs and $\mathcal{N}_1 \cup \mathcal{N}_2 = \mathcal{N}$. The time delays are $d_{12} = d_{24} = d_{41} = 0.2$, $d_{23} = 0.3$, and $d_{31} = 0.4$. The nominal delay for the directed ring graph $G_1$ is $D_1 = \frac{1}{3}(d_{12} + d_{23} + d_{31}) = 0.3$ and the nominal delay for $G_2$ is $D_2 = \frac{1}{3}(d_{12} + d_{24} + d_{41}) = 0.2$. We set the idling times to be $w_i = 0.5$ and the initial time
Fig. 9. Time displays of three clocks with time-varying delays in an undirected network.

Fig. 10. The graph topology of a general strongly connected graph.

displays of the four clocks to be $[x_1(0), x_2(0), x_3(0), x_4(0)]^T = [10, 40, 20, 30]^T$. The simulation results of the evolution of the displays of the four clocks are shown in Fig. 11, from which one can find that the synchronization errors oscillate among several values.

6. 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 the reality in distributed data networks.

REFERENCES

Fig. 11. Time displays of four clocks in a strongly connected network.