2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC): Track 1: Fundamentals and PHY

Fully Distributed Energy-Efficient Synchronization for Half-duplex D2D Communications

Onur Karatalay*, Ioannis Psaromiligkos*, Benoit Champagne* and Benoit Pelletier†
*Department of Electrical and Computer Engineering, McGill University, Montréal, PQ, Canada.
†InterDigital Canada Ltée, Montréal, PQ, Canada.
Email: onur.karatalay@mail.mcgill.ca; ioannis.psaromiligkos@mcgill.ca;
benoit.champagne@mcgill.ca; benoit.pelletier@interdigital.com

***Abstract*—Synchronization is a challenging problem especially for distributed systems, such as out-of-coverage D2D networks, as no common reference time is available. In such cases, devices use distributed synchronization algorithms, however, accurately determining when to stop the synchronization process is as challenging as achieving synchronization since they do not have the synchronization status of other devices in the network. From energy efficiency and performance perspective, the synchronization process should be stopped at all devices at the same time. In addition, to counteract the effect of propagation delays during synchronization, timing-advance (TA) clocks should be employed. This could be achieved in the synchronization process, however, after the devices are synchronized, there is no central mechanism to instruct them on TA clocks for transmitting or receiving data packets. In this paper, we propose a synchronization algorithm which, in an energy-efficient manner, allows devices to (i) acquire the synchronization status of others and terminate the synchronization process as soon as all devices in the network are synchronized, (ii) allow the synchronized devices to properly advance/regress their clocks prior to data communication by tracking their relative timing. We numerically demonstrate that the maximum synchronization error over multipath channels is around 0.6μs and it can be maintained during data communication.**

I. Introduction

In distributed systems, such as out-of-coverage Device-to-Device (D2D) networks operating in the absence of a coordinating base-station (BS), no common reference time is available, hence, devices must use distributed synchronization algorithms [1]-[3]. A commonly used synchronization approach for distributed networks is pulse-coupled synchronization, which offers a scalable physical (PHY) layer solution for initial synchronization prior to data transmission [3].

Pulse-coupled synchronization is often implemented by using distributed phase-locked loops (DPLLs) which is an iterative procedure in which devices receive (and transmit) synchronization pulses that are used to update their clocks. DPLLs enable the devices to reach consensus on a common clock, but their performance is sensitive to signal propagation delays [3], [4]. In the literature, propagation delays and other environmental disturbances such as multipath channels or changes to network size are often not considered [5]-[7]. In practice, to counteract the effect of propagation delays, guard intervals or timing advance (TA) clocks should be employed, i.e., the clocks of the receiver devices are advanced with respect to the clocks of the transmitter devices [8].

The duplexing mode (i.e., full-duplex vs half-duplex) is also a crucial factor in the synchronization process as it directly affects the ability of devices to receive synchronization signals and correct their clocks [8]. Full-duplex clock synchronization is more commonly studied, see [3]-[7] and the references therein. However, full-duplex technology is not practical, especially on the device side. Hence, half-duplex synchronization is a more realistic option for D2D networks [9], [10].

From the energy efficiency and performance perspectives it is beneficial to stop the synchronization process at all devices at the same time and, ideally, as soon as they become synchronized. In out-of-coverage D2D networks, which are the focus of this paper, a device does not know the synchronization status of the rest. This calls for a distributed method to track the synchronization status across the network and then determine when to stop the synchronization process. A second challenge faced by D2D networks, is how a device can determine whether it should advance or regress its clock during data communication in order to counteract the effect of propagation delays. In out-of-coverage D2D networks, there is no BS to coordinate the devices by instructing them on how to time-advance their clocks; the devices have to determine their relative timing themselves during synchronization in a distributed manner.

In this paper, we address both of the above-mentioned challenges. First, we build upon the synchronization algorithm in [9] and propose a communication method that allows devices to track the synchronization status of the network in a distributed manner, and enables them to terminate the synchronization process almost simultaneously. Second, we propose an algorithm which allows the synchronized devices to properly advance/regress their clocks prior to data communication by tracking their relative timing in an energy-efficient manner, i.e., without

This work was supported in part by the Natural Science and Engineering Research Council under the Discovery Grant program, and by InterDigital Canada Ltée and MITACS Canada under the MITACS Accelerate program.

978-1-5386-8110-7/19/$31.00 ©2019 IEEE

needing re-synchronization.

The paper is organized as follows. In Section II we present the system model and we formally describe the problem under consideration. The proposed method is described in detail in Section III, and its performance is discussed in Section IV. Finally, Section V concludes the paper.

II. System Model and Problem Statement

We consider a fully distributed D2D network that consists of $\mathcal{J}$ wireless devices indexed by $j \in \{1, ..., \mathcal{J}\}$; devices may join or leave the network at any time. The physical clock of the jth device is $t_j(t) = \alpha_j t + \theta_j$ [9], where t is the universal time, α_j is the clock skew, $\theta_j \in [0, T_0)$ is the clock phase with T_0 being the clock period. For simplicity, we assume frequency synchronized clocks, that is, $\alpha_j = 1 \ \forall j \in \{1, ..., \mathcal{J}\}$, however, generalizations are possible.

By uniformly sampling the physical clock at $t = \nu T_0$ with $\nu \in \mathbb{Z}$, we obtain the discrete logical clock of the jth device as $t_j[\nu] = t_j(\nu T_0) = \nu T_0 + \theta_j$. We refer to time $t_j[\nu]$ as the νth *clock tick* of the jth device. Moreover, we can partition the universal time axis into a sequence of non-overlapping time slots $[\nu T_0, (\nu+1)T_0)$, each of which contains a clock tick, i.e., $t_j[\nu]$, as shown in Fig. 1. The clock tick difference between the ith and the jth device represents their timing offset (TO), $\Delta t_{ij}[\nu] = t_i[\nu] - t_j[\nu]$.

A. Signal Model in Half-Duplex Communication

We assume half-duplex communication, where each device either operates in transmitter mode (TX) or in receiver mode (RX), as depicted in Fig. 1. We denote the transceiver mode of the jth device at the clock tick ν as $M_\nu^j \in \{\text{TX}, \text{RX}\}$, and the sets of transmitter and receiver devices at the νth clock tick as $\mathcal{T}_\nu$ and $\mathcal{R}_\nu$, respectively.

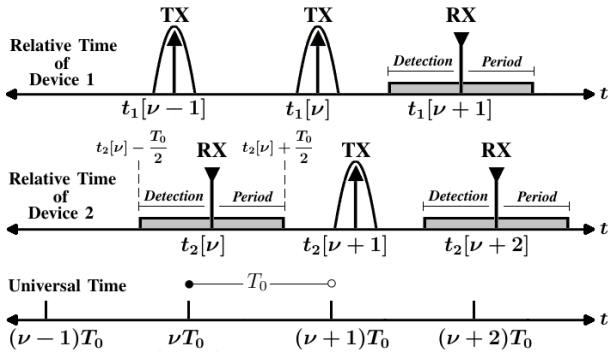


Fig. 1. Sequences of clock ticks of two devices relative to the universal time t. In the νth time slot, device 1 operates as a transmitter denoted by TX (upward arrow) while device 2 denoted by RX (downward arrow) starts as a receiver.

Furthermore, we consider two different *stages* for devices to operate in. During the first stage, which is the *synchronization stage*, the devices try to minimize their synchronization error in a distributed manner. When the devices deem that they have synchronized their clocks with the rest of the network, they switch to the second one, which is called *energy-efficient stage*, where they become idle, i.e., only operate as a receiver to conserve power, unless they have data to transmit.

In the synchronization stage, a transmitter device $i \in \mathcal{T}_\nu$ broadcasts a synchronization signal with length $N_s = 2N$ formed by concatenating two Zadoff-Chu (ZC) sequences of length N with root indices $\hat{u}$ and $-\hat{u}$ [6]:

$$s[n; \hat{u}] = \begin{cases} e^{-\mathrm{j}\frac{\pi}{N}\hat{u}(n-N)^2}, & 0 \le n \le N-1 \\ e^{\mathrm{j}\frac{\pi}{N}\hat{u}n^2}, & N \le n \le 2N-1 \end{cases} \quad (1)$$

where $\mathrm{j} = \sqrt{-1}$.

At each tick of its clock, a device $i \in \mathcal{T}_\nu$ broadcasts a synchronization signal $x(t - t_i[\nu]; \hat{u})$, where

$$x(t; \hat{u}) = \sum_{n=0}^{N_s-1} s[n; \hat{u}] g(t - nT_p). \quad (2)$$

In (2), $g(t) \in \mathbb{R}$ is a unit-energy baseband pulse and T_p denotes the pulse spacing with $N_s T_p \ll T_0$. Note that before broadcasting, (2) is upconverted to passband using a carrier frequency f_i as $\widetilde{x}_i(t, \nu; \hat{u}) = x(t - t_i[\nu]; \hat{u}) e^{\mathrm{j}2\pi f_i t}$.

A receiver device $j \in \mathcal{R}_\nu$, regardless of the stage it is in, listens for the broadcasted synchronization signals over the *detection period* $[t_j[\nu] - \frac{T_0}{2}, t_j[\nu] + \frac{T_0}{2})$, centered at its own clock tick $t_j[\nu]$ (see Fig. 1). The received passband signal equals to:

$$\widetilde{y}_j(t) = \sum_{\eta \in \{\nu, \nu \pm 1\}} \sum_{i \in \mathcal{T}_\eta} \widetilde{x}_i(t, \eta; \hat{u}) * h_{ij}(t) + w_j(t) \quad (3)$$

where $h_{ij}(t)$ is the impulse response of the multipath channel between the ith and jth device, $*$ denotes convolution and $w_j(t)$ is an additive noise term. The multipath channel impulse response is given by $h_{ij}(t) = \sum_{p=1}^{P} \rho_{ijp} \delta(t - \tau_{ijp})$, where p is the path index and P is the number of resolvable paths which is assumed to be the same for all devices. Moreover, $\rho_{ijp} \in \mathbb{C}$ and $\tau_{ijp} \in \mathbb{R}_+$ are the complex gain and propagation delay of the pth path, respectively. As explained in [9], the outer summation in (3) includes the possible signal contributions from the adjacent detection periods of the νth clock tick.

Finally, the jth receiver device downconverts (3) with its carrier frequency f_j and samples it at time instances kT_s with T_s being the sampling period. The baseband signal in discrete-time is obtained by truncating it over the time interval $\left[t_j[\nu] - \frac{T_0}{2}, t_j[\nu] + \frac{T_0}{2}\right)$:

$$\begin{aligned} y_j[k; \nu] &= y_j(kT_s + t_j[\nu]) \\ &= \sum_{\eta \in \{\nu, \nu \pm 1\}} \sum_{i \in \mathcal{T}_\eta} \sum_{p=1}^{P} \rho_{ijp} x(kT_s + t_j[\nu] - t_i[\eta] - \tau_{ijp}; \hat{u}) \\ &\quad \times e^{\mathrm{j}2\pi \Delta f_{ij} k T_s} + w_j[k] \end{aligned} \quad (4)$$

where $\Delta f_{ij} = f_i - f_j$ is the CFO between the ith and jth device and $w_j[k]$ is the discrete-time noise process. Constructing the synchronization sequences as in (1) decouples the effect of CFO in TO estimation [6]. Note that the complex factor $e^{-\mathrm{j}2\pi f_i \tau_{ijp}}$ is absorbed by ρ_{ijp} in (4) with no loss of generality.

Finally, the jth receiver device cross-correlates (4) with two sequences $x_-[k;\hat{u}]$ and $x_+[k;\hat{u}]$ as follows:

$$R_{y_j x_\mp}[l,\nu;\hat{u}] = \sum_{k\in\mathbb{Z}} y_j[k+l;\nu] x^*_\mp[k;\hat{u}] \tag{5}$$

The sequences $x_-[k;\hat{u}]$ and $x_+[k;\hat{u}]$ are the two parts of the discrete-time equivalent of $x(t;\hat{u})$, obtained by sampling $x(t;\hat{u})$ with a sampling period T_s, that correspond to the ZC sequence constructed with $-\hat{u}$ and $+\hat{u}$, respectively. To estimate the TO, the jth receiver device uses a weighted average across the lags l:

$$q_j^\mp[\nu;\hat{u}] \triangleq \frac{\sum_l l|R_{y_j x_\mp}[l,\nu;\hat{u}]|^2}{\sum_l |R_{y_j x_\mp}[l,\nu;\hat{u}]|^2}. \tag{6}$$

The estimates in (6) are obtained in discrete-time, that is $q_j^-[\nu;\hat{u}] \approx f_s d_j$, where $f_s = \frac{1}{T_s}$ is the sampling frequency and d_j is the weighted average TO seen from the jth device in continuous-time. Furthermore, $q_j^+[\nu;\hat{u}]$ is equal to the sum of the weighted average TO and the duration of signal $x_-[k;\hat{u}]$ due to the concatenation in (1), hence, $q_j^+[\nu;\hat{u}] \approx f_s(d_j + NT_p)$. Thus, the weighted average TO estimate in continuous-time between the contributing transmitters and the jth device equals to:

$$\begin{aligned}\Delta t_j[\nu;\hat{u}] &= \frac{\left(q_j^-[\nu;\hat{u}] + q_j^+[\nu;\hat{u}]\right) T_s - NT_p}{2} \\ &\approx \overline{\Delta t_j}[\nu] + \overline{\tau_j}[\nu]\end{aligned} \tag{7}$$

where $\overline{\Delta t_j}[\nu]$ and $\overline{\tau_j}[\nu]$ are the weighted average of relative clock differences and a bias term due to the propagation delays at the jth device, respectively [9].

B. Problem Statement

To achieve time synchronization, the jth receiver device updates its clock by using DPLLs [3], where the next clock tick is corrected with the TO estimate as follows:

$$t_j[\nu+1] = t_j[\nu] + T_0 + \epsilon \Delta t_j[\nu;\hat{u}],\ \forall j \in \mathcal{R}_\nu \tag{8}$$

with $\epsilon \in (0,1]$ being a scaling term. On the other hand, due to half-duplex constraint the transmitter devices only advance their clock:

$$t_i[\nu+1] = t_i[\nu] + T_0,\ \forall i \in \mathcal{T}_\nu. \tag{9}$$

The network is synchronized when the synchronization error reaches a steady-state at all receiver devices, i.e., when $|\Delta t_j[\nu;\hat{u}]|\ \forall j \in \mathcal{R}_\nu$ stays below a given threshold. At this point, the devices can leave the synchronization stage and enter the energy-efficient stage, where they stop broadcasting the synchronization signals and become idle.

However, in a distributed network, the devices do not necessarily reach the steady-state at the same time. In pulse-coupled synchronization, transition from the steady-state to the energy-efficient stage by a single device at the νth clock tick with no coordination with the rest may result in the following. First, a perturbation occurs in the TO estimate at other devices due to the sudden absence of signal contribution from this device, hence, they may no longer be in the steady-state at the $(\nu+1)$th clock tick. Second, since an idle device does not update its clock, as soon as it listens for broadcasted signals, the device will re-join the synchronization process because the other devices are still broadcasting. Thus, in distributed pulse-coupled synchronization, the devices should not stop the synchronization process without coordination.

The first objective of this paper, therefore, is to develop an algorithm that allows the devices to exit the synchronization stage as soon as possible but in a coordinated manner (ideally, at the same time). Our second objective is to allow synchronized devices to initiate data communication in a distributed and in an energy-efficient manner. Since there is no central mechanism to instruct them to properly advance/regress their clocks prior to data communication, the devices should be able to achieve that distributively and without re-initiating the synchronization process. Thus, the devices can exchange data packets by exploiting TA clocks without depleting their resources.

III. Proposed Method

We propose an intermediate stage, the *transition stage*, between the synchronization and energy-efficient stages, during which the devices acquire the synchronization status of the network and then simultaneously terminate the synchronization process. The devices may enter the transition stage at different times but they all exit at the same time by switching to the energy-efficient stage.

During both the synchronization and transition stages the devices either broadcast synchronization signals or update their clocks to reduce/maintain their synchronization errors. What differentiates these two stages is the type of synchronization signal used. During the synchronization stage, a transmitter device broadcasts one type of synchronization signal, i.e., it sets the ZC root index to $\hat{u} = u_1$. When it enters the transition stage, it starts broadcasting a second type of the synchronization signal by changing the ZC root index to $\hat{u} = u_2$, to inform the network that it has achieved synchronization. In other words, the use of the ZC root index u_2 by a device serves as a declaration to the rest of the network that the device has deemed it is synchronized. Consequently, the absence of the synchronization signal of the first type, i.e., $\widetilde{x}_i(t,\nu;u_1)$, signifies that all devices in the network are in the transition stage and can, therefore, terminate the synchronization process in a coordinated manner.

The received signal at a device can be of one of four kinds: It only contains synchronization signals that use the ZC root index u_1 (this occurs when all transmitter devices are in the synchronization stage), or it only contains synchronization signals that use the ZC root index u_2 (this occurs when all the transmitter devices are in the transition stage). In addition, the received signal either contains a mix of synchronization signals (when some devices are in the synchronization stage and others in the transition stage) or no synchronization signals at all (this may occur when there are no transmitter devices). Thus, a receiver device must be able to reliably detect the presence or

absence of synchronization signals that use the ZC root index u_1 or u_2.

Based on [9], we propose to perform the above-mentioned detection task as follows. After cross-correlating the received signal in (3) with both $x^*_{\mp}[k;u_1]$ and $x^*_{\mp}[k;u_2]$ at every clock tick, the receiver device calculates two decision statistics $\psi(\nu,u_1)$ and $\psi(\nu,u_2)$, where

$$\psi(\nu,\hat{u}) \triangleq \max_{l}(|R_{y_j x_{\mp}}[l,\nu;\hat{u}]|) - N/2. \quad (10)$$

Then it makes one of the following four decisions:

$$\begin{aligned}
\mathrm{D}^{00}_{\nu}&: \ \psi(\nu,u_1)<0 \ \wedge \ \psi(\nu,u_2)<0\\
\mathrm{D}^{10}_{\nu}&: \ \psi(\nu,u_1)\geq 0 \ \wedge \ \psi(\nu,u_2)<0\\
\mathrm{D}^{01}_{\nu}&: \ \psi(\nu,u_1)<0 \ \wedge \ \psi(\nu,u_2)\geq 0\\
\mathrm{D}^{11}_{\nu}&: \ \psi(\nu,u_1)\geq 0 \ \wedge \ \psi(\nu,u_2)\geq 0
\end{aligned}$$

where the superscripts of the decisions indicate the presence, i.e., 1, or the absence, i.e., 0, of the ZC root indices u_1 and u_2. Finally, it forms its TO estimate as follows:

$$\Delta t_j[\nu] = \begin{cases} 0 & , \ \mathrm{D}^{00}_{\nu} \\ \Delta t_j[\nu;u_1] & , \ \mathrm{D}^{10}_{\nu} \\ \Delta t_j[\nu;u_2] & , \ \mathrm{D}^{01}_{\nu} \\ \dfrac{\Delta t_j[\nu;u_1]+\Delta t_j[\nu;u_2]}{2} & , \ \mathrm{D}^{11}_{\nu} \end{cases} \quad (11)$$

Furthermore, each receiver device estimates the bias and removes its effect during the DPLL update to reduce its synchronization error. The following DPLLs clock update rule is used at the receiver devices [9]:

$$t_j[\nu+1]=t_j[\nu]+T_0-2\hat{\phi}_j[\nu]+\Delta t_j[\nu], \ \forall j \in \mathcal{R}_{\nu} \quad (12)$$

where $\hat{\phi}_j[\nu]$ is the bias estimate of the jth device at the νth clock tick. The transmitter devices only advance their clocks as previously given in (9).

The operation of the proposed method is described by the state-transition diagram in Fig. 2. In what follows we describe in detail the states as well as the conditions for transitioning from one state to another from the point of view of the j^{th} device.

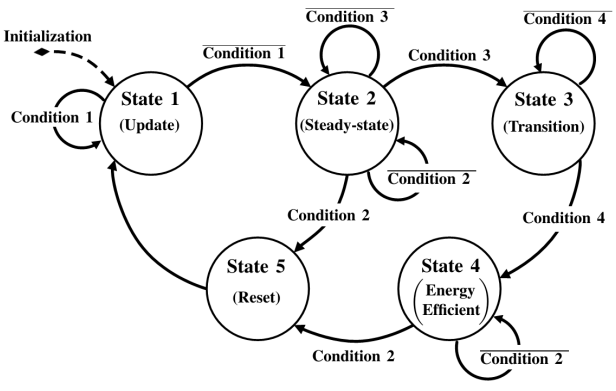


Fig. 2. State-transition diagram of the proposed algorithm. Transitions happen at each clock tick, where $\overline{\textbf{Condition x}}$ denotes the contrary of **Condition x**.

Initialization: Set

$$\Gamma_j = 0 \text{ and } \xi_j = 0 \quad (13)$$

$$\hat{\phi}_j[\nu] = \hat{\phi}^{init}_j \quad (14)$$

$$M_j \leftarrow M^j_{\nu} \quad (15)$$

$$\hat{u} \leftarrow u_1 \text{ if } M^j_{\nu} = \mathrm{TX} \quad (16)$$

where Γ_j and ξ_j are steady-state and energy-efficiency counters, respectively. For selecting the transceiver mode under the half-duplex constraint, we use the approach proposed in [9]. Specifically, when a device joins the network it randomly determines its initial transceiver mode: It either becomes a transmitter with probability p_{tr} or a receiver with probability $1-p_{tr}$. Then, the device alternates its mode from a transmitter to a receiver or vice-versa at each clock tick. On the other hand, when a device cannot detect a synchronization signal of any type, which happens when no transmitters are present in the detection period (see D^{00}_{ν} above), the device re-determines its transceiver mode randomly as described earlier.

Moreover, when a device joins the network at clock tick ν, its initial transceiver mode M^j_{ν} is saved to M_j, which will be used later by the device for the coordinated termination of the synchronization process. In addition, transmitter devices broadcast the first type of the synchronization signal by selecting the corresponding ZC index in (16). We note that by using the alternating transceiver mode algorithm in [9] along with (12), the clocks of the receiver devices become advanced with respect to the clocks of the transmitter devices as they remove their bias estimate twice. This leads to TA clocks which results in reduced synchronization error.

Finally, $\hat{\phi}^{init}_j$ is the initial estimate of the bias and it is set as follows. We assume that the coordinates of the jth receiver device X_j, Y_j and the ith transmitter device X_i, Y_i are independent and identically distributed (i.i.d.), where $X_j, Y_j, X_i, Y_i, \sim \mathcal{U}(0,d)$ and d being the maximum grid distance in meters. Thus, the average propagation delay seen by the receiver device is $\overline{\tau}_{prop} = \frac{1}{c}\mathbf{E}\big(\sqrt{(X_j-X_i)^2+(Y_j-Y_i)^2}\big)$, where c is the speed of light and $\mathbf{E}(\cdot)$ denotes expectation. In addition to propagation delay, multipath channel introduces delay spread, which affects the average TO estimate. Thus, we also include the channel effect and initialize the bias as $\hat{\phi}^{init}_j = \overline{\tau}_{prop} + \overline{\tau}_{spread}$, where $\overline{\tau}_{spread}$ is the average channel delay spread. In other words, $\hat{\phi}^{init}_j$ is a rough approximation of the bias for the first iteration.

In **State 1** and **State 2** which comprise the synchronization stage as well as in **State 3** which comprises the transition stage, a device updates its clock to reduce its synchronization error by performing the following state-specific operations:

State 1: Update the bias estimate

$$\hat{\phi}_j[\nu+1] = \begin{cases} \hat{\phi}_j[\nu]-\delta, & \Delta t_j[\nu]<0 \\ \hat{\phi}_j[\nu]+\delta, & \Delta t_j[\nu]>0 \end{cases} \quad (17)$$

where $\delta \in \mathbb{R}_+$ is the step-size for bias tracking, as long as Condition 1 shown below is satisfied.

Condition 1: $|\Delta t_j[\nu]| \leq |\Delta t_j|_{min} \vee |\Delta t_j|_{min} > \lambda_{th}$

where $|\Delta t_j|_{min}$ is the minimum value of the weighted average TO encountered up to the νth iteration and λ_{th} is the synchronization error threshold, which can be greater than $|\Delta t_j|_{min}$.

State 2: Keep the bias estimate corresponding to $|\Delta t_j|_{min}$

$$\hat{\phi}_j[\nu+1] = \hat{\phi}_j[\nu] \tag{18}$$

$$\Gamma_j = \begin{cases} \Gamma_j + 1 \ , M_\nu^j = \text{RX} \\ \Gamma_j \quad\quad , M_\nu^j = \text{TX} \end{cases} \tag{19}$$

In this state, the jth receiver device fixes its bias estimate to the estimate that yields $|\Delta t_j|_{min}$, then increases Γ_j until **Condition 2** or **Condition 3** is satisfied. These conditions are:

Condition 2: $\big||\Delta t_j[\nu]| - |\Delta t_j|_{min}\big| > \lambda_{th}$

Condition 3: $\Gamma_j \geq \lambda_{ss}$

The former condition signifies a perturbation in the system, e.g., a new device joins the network, whereas the latter indicates that the j^{th} device is in the steady-state for a number of consecutive clock ticks controlled by a predefined threshold λ_{ss}.

State 3: Transition stage between the synchronization stage and the energy-efficient stage:

$$\hat{u} \leftarrow u_2 \text{ if } M_\nu^j = \text{TX} \tag{20}$$

If $M_\nu^j = \text{RX}$:

$$\xi_j = \begin{cases} \xi_j + 1 \ , \text{D}_\nu^{01} \ \vee \ (\text{D}_\nu^{00} \ \wedge \ \xi_j > 0) \\ 0 \quad\quad , \text{D}_\nu^{10} \ \vee \ \text{D}_\nu^{11} \end{cases} \tag{21}$$

If $M_\nu^j = \text{TX}$:

$$\xi_j = \begin{cases} \xi_j + 1 \ , (M_j{=}\text{TX}) \wedge \text{D}_{\nu-1}^{01} \\ 0 \quad\quad , (M_j{=}\text{TX}) \wedge (\text{D}_{\nu-1}^{10} \vee \text{D}_{\nu-1}^{11}) \end{cases} \tag{22}$$

The main purpose of this state is to allow the devices to make a simultaneous transition from the synchronization stage to the energy-efficient stage. First, a device operating as a transmitter informs the network by broadcasting the second type of the synchronization signal as in (20). Then, if a device operates as a receiver at the νth clock tick, i.e., $M_\nu^j = \text{RX}$, and detects the presence of synchronization signals with root index u_2 but not of synchronization signals with root index u_1, it increases its energy-efficiency counter ξ_j. In addition, in this state, a receiver device may detect no synchronization signals, i.e., D_ν^{00}, after it has already increased its counter, i.e., $\xi_j > 0$. The device interprets this as that the previously detected devices have become idle by switching to the *energy-efficient stage*, hence, it continues to increase its counter.

We should note that if a device only took actions when it is a receiver, then the devices would not stop the synchronization process at the same clock tick as the transmitter devices would have to wait for the next clock tick to become a receiver. This is why we distinguish the devices based on their transceiver mode at the first clock tick they join the network (stored in M_j). Thus, when a device operates as a transmitter at the νth clock tick, i.e., $M_\nu^j = \text{TX}$, it also increases its counter if and only if its first transceiver mode is TX, that is $M_\nu^j = M_j = \text{TX}$. However, a transmitter device cannot receive any signals to obtain decision statistics at the νth clock tick (10). Hence, it uses the decisions taken when it was a receiver at the $(\nu-1)$th clock tick given in (22).

On the other hand, whenever a device in this state detects synchronization signals of the first type, it resets its counter as $\xi_j = 0$ when the conditions in (21) or (22) are satisfied. This is interpreted as having at least one device in the synchronization stage, hence, the device waits for the other device(s) to transition out of the synchronization stage.

Condition 4:

If $M_\nu^j = \text{RX}$:

$$(M_j = \text{RX}) \wedge \big(\xi_j > \lambda_{ee} \ \vee \ (\text{D}_\nu^{00} \ \wedge \ \xi_j > 0)\big)$$

If $M_\nu^j = \text{TX}$:

$$(M_j = \text{TX}) \ \wedge \ \xi_j > \lambda_{ee}$$

Each device switches to the energy-efficient stage only when its current transceiver mode at the νth clock tick, i.e., M_ν^j, is the same as its initial transceiver mode, i.e., M_j. In addition, we note that a receiver device might not detect any more synchronization signal of any kind. This can happen when all transmitter devices have become idle by switching to the energy-efficient stage. In this case, the device also switches its state to become idle.

State 4: This state comprises the energy efficiency stage. Each device sets its current transceiver mode to receiver, i.e., $M_j^\nu = \text{RX}$, hence, they become idle. Thus, the devices conserve transmit power but listen for possible perturbations by checking **Condition 2**. If a perturbation is detected, then the device switches to the next state where all its variables are reset.

State 5: Reset the variables

$$\Gamma_j = 0 \text{ and } \xi_j = 0 \tag{23}$$

$$|\Delta t_j|_{min} = \Delta t_j[\nu] \tag{24}$$

$$\hat{\phi}_j[\nu] = \hat{\phi}_j^{init} \tag{25}$$

$$\hat{u} \leftarrow u_1 \tag{26}$$

$$M_j^\nu \leftarrow M_j \tag{27}$$

A perturbation is a result of a new device joining the network and initiating its own syncronization process. In this case, the devices stop being idle by setting their transceiver mode to the one in M_j. Then, they re-initiate the synchronization process by including the new device.

A. Data Communication

By using the proposed algorithm, the devices are separated into two clusters ($\mathcal{T}_\nu$ and $\mathcal{R}_\nu$) based on their transceiver mode. In Fig. 3, we illustrate this separation. Recall that when devices are in **State 3** they alternate their groups at each clock tick; however, we label each cluster as the TX or RX cluster, according to the *initial* transceiver

mode of the devices in it. In addition, due to the decisions taken in **Condition 4**, the devices are aware of their cluster right before switching to the energy-efficient stage, **State 4**, where they become idle. Importantly, when they enter this stage, they know whether their clocks are advanced or lagging with respect to the opposite cluster. Note that the gap between the two clusters corresponds to the average propagation delay between them, which can be used to exploit TA clocks when data communication takes place from a device in the TX cluster towards the devices in the RX cluster.

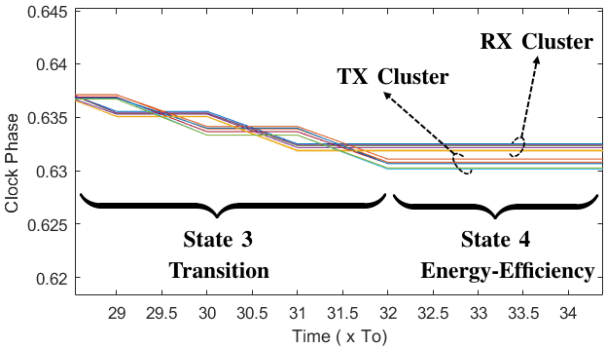


Fig. 3. A snapshot of simultaneous transition from *synchronization stage* to *energy-efficient stage* given by the state-transition diagram of the proposed algorithm.

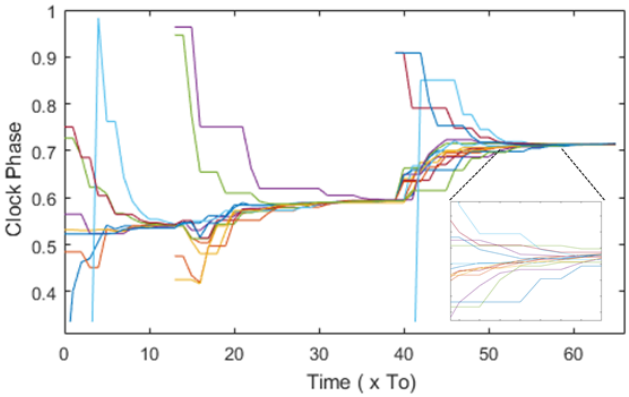
However, a device should be able to exploit TA clocks to initiate data communication regardless of its cluster. If a device is in the TX cluster but anticipates receiving data packets, it can simply advance its clock by using its bias estimate $\hat{\phi}_j[\nu]$ to approach the clocks of the devices in the RX cluster. On the other hand, if a device is in the RX cluster, yet it has data to transmit, it regresses its clock for possible receivers in the RX cluster. These clock alignments for data communication are given as follows:

$$t_j[\nu+1] = \begin{cases} t_j[\nu]+T_0-\hat{\phi}_j[\nu], M_j = \text{RX s.t. } B_j \neq \emptyset \\ t_j[\nu]+T_0+\hat{\phi}_j[\nu], M_j = \text{TX s.t. } B_j = \emptyset \end{cases} \quad (28)$$

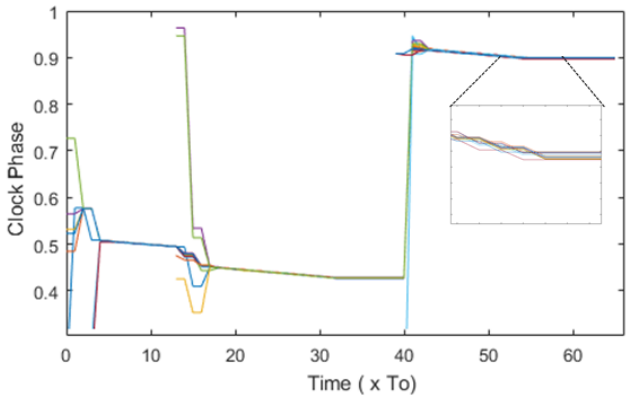
where B_j is the data buffer of the jth device, which indicates whether the device anticipates transmitting or receiving data packets, i.e., $B_j \neq \emptyset$ and $B_j = \emptyset$, respectively. If the device uses (28), then it can also simply revert this after the data communication is done. In this way, any device can initiate data communication without disrupting the achieved synchronization and re-join their original cluster when it is not transmitting/receiving data packets.

IV. Numerical Results

The proposed algorithm is implemented in MATLAB for a stationary, dense urban network where the channel model follows the Manhattan grid scenario [12]. The parameters used in the simulations are given in Table I and other specifications are chosen accordingly from [13]. First, we investigate the convergence of the device clocks in Fig. 4. Then, we focus on the absolute synchronization error achieved in any receiver device shown in Fig. 5, where we compare the proposed algorithm with DPLL clock updates (8) and (9) with different transmitter probabilities p_{tr} as a benchmark. The proposed algorithm not only rapidly mitigates the synchronization error but also achieves smooth clock convergence with less variance. Even though perturbations occur at different states such as during the steady-state at the 14th clock tick and the energy-efficient state at 40th clock tick, the proposed algorithm can successfully detect them and reduce the synchronization error to the previously achieved error levels (cf. Fig. 5).



(a) Random transceiver mode with $p_{tr} = 0.5$.



(b) Proposed algorithm.

Fig. 4. A snapshot of convergence to a common clock with multiple perturbations at 14th and 40th clock ticks.

TABLE I. System parameters for MATLAB simulations.

Parameter Description	Symbol	Value
Number of Devices	$\mathcal{J}$	15
Scaling Term of DPLL	ϵ	0.5
Zadoff-Chu Index	u_1, u_2	7, 13
Zadoff-Chu Sequence Length	N	839
Clock Period	T_0	1 ms
Maximum Network Distance	d	500 m
Operating Frequency	f	2 GHz
Transmit and Reception Powers	$P_{\text{TX}}, P_{\text{RX}}$	23 dBm, 8 dBm
Delay Compensation Step Size	δ	33 ns
Transceiver mode selection probability	p_{tr}	0.5
Synchronization Error Threshold	λ_{th}	1.5 μs
Steady-state Counter	λ_{ss}	2
Energy-efficiency Counter	λ_{ee}	2

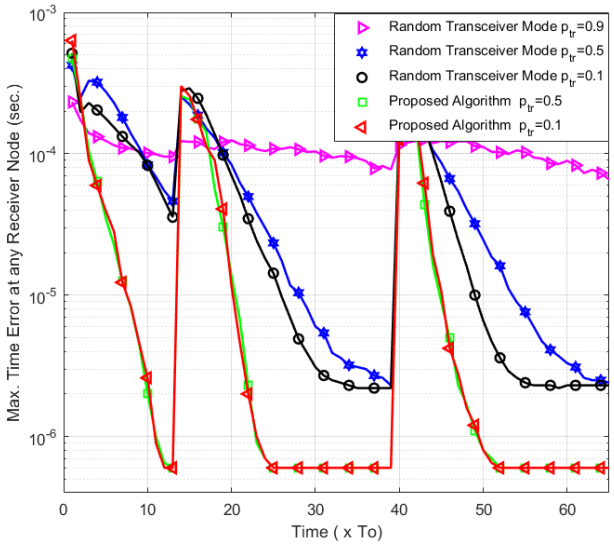


Fig. 5. Maximum synchronization error seen by any receiver device.

Furthermore, we analyze the total energy used during synchronization for the devices present from the beginning to the end of the synchronization process. Note that devices that join the network later in time may consume less energy but achieve the same synchronization error. We consider different power levels for each transceiver mode, namely P_{TX} and P_{RX}, to operate as a transmitter and as a receiver device, respectively. The total energy consumed by the jth device up to the νth clock tick during synchronization is calculated recursively as follows:

$$E_j^{\nu+1} = \begin{cases} E_j^{\nu} + P_{\text{TX}} T_0 \ , & \forall j \in \mathcal{T}_\nu \\ E_j^{\nu} + P_{\text{RX}} T_0 \ , & \forall j \in \mathcal{R}_\nu \end{cases} \tag{29}$$

We only track the devices that are present throughout the synchronization process.

As seen in Fig. 5 and Table II by using the proposed algorithm, the devices can achieve and successfully maintain a synchronization error of around 0.6μs while consuming the average energy of $E_{sync} = \frac{1}{\mathcal{J}} \sum_{j=1}^{\mathcal{J}} \sum_{\nu=0}^{\infty} E_j^{\nu} = 54.47$mJ over the synchronization time $T_{sync} = 130$. On the other hand, random transceiver modes not only fail to achieve the same synchronization error level but also consume more energy except for $p_{tr} = 0.1$. However, we note that in the proposed algorithm, 87.74% of the total energy is used for signal transmission, i.e., E_{sync}^{TX}, while only 12.26% of the total energy is used for signal reception, i.e., E_{sync}^{RX}. This means that if the simulation time T_{sync} increases, then the energy efficiency of the proposed algorithm also increases over time as the devices operate only as a receiver device which consumes less power.

TABLE II. Energy-efficiency comparisons.

	Proposed Algorithm	Random $p_{tr} = 0.1$	Random $p_{tr} = 0.5$	Random $p_{tr} = 0.9$
E_{sync} (mJ)	54.47	34.09	133.02	226.53
$E_{sync}^{\text{TX}}(\%)$	87.74	78.42	96.90	99.65
$E_{sync}^{\text{RX}}(\%)$	12.26	21.58	3.10	0.35

V. Conclusion

In this paper, we proposed an energy-efficient synchronization algorithm for distributed D2D networks over a realistic scenario, in which the devices are arbitrarily joining or leaving the network and communicating over multipath channels. The proposed algorithm orchestrates the switch from the synchronization phase to the data communications phase. Importantly, each device is aware of the global synchronization status and can initiate data communication by properly using TA clocks.

References

[1] L. Cao, T. Peng and J. Liu, "An effective distributed synchronization method subject to timing error for D2D communication," in *Proc. Asia-Pacific conference on communications*, pp. 1–6, Dec. 2017

[2] K. Manolakis and W. Xu, "Time synchronization for multi-link D2D/V2X communication," *IEEE Vehic. Tech. Conf.-Fall*, pp. 1–6, Sept. 2016.

[3] D. Tétreault-La Roche, B. Champagne, I. Psaromiligkos and B. Pelletier, "On the use of distributed synchronization in 5G device-to-device networks," in *Proc. IEEE Int. Conf. on Commun.*, pp. 1938–1883, Jul. 2017.

[4] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. H. Strogatz, "Distributed synchronization in wireless networks," *IEEE Signal Process. Mag.*, vol. 25, no. 5, pp. 81–97, Sept. 2008.

[5] W. Sun, F. Brännström and E. G. Ström, "Network synchronization for mobile Device-to-Device systems," *IEEE Trans. on Commun.*, vol. 65, no. 3, pp. 1193–1206, Mar. 2017.

[6] M. M. U. Gul, X. Ma and S. Lee, "Timing and frequency synchronization for OFDM downlink transmissions using Zadoff-Chu sequences," in *IEEE Trans. on Wireless Commun.*, vol. 14, no. 3, pp. 1716–1729, Mar. 2015.

[7] M. A. Alvarez, B. Azari, U. Spagnolini, "Time and frequency self-synchronization in dense cooperative network," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, pp. 1811–1815, Nov. 2014.

[8] D. Tse and P. Viswanath, Fundamentals of Wireless Communication. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[9] O. Karatalay, I. Psaromiligkos, B. Champagne and B. Pelletier, "Fast converging Distributed pulse-coupled clock synchronization for half-duplex D2D communications over multipath channels," in *Proc IEEE ISSPIT*, pp. 123–128, Dec. 2018.

[10] M. A. Alvarez, U. Spagnolini, "Half-duplex scheduling in distributed synchronization," in *Proc. IEEE Int. Conf. on Commun.*, pp. 6240–6245, June 2015.

[11] D. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 531–532, Jul. 1972.

[12] "Mobile and wireless communications enablers for the twenty-twenty information society (METIS)," Deliverable D1.4 METIS Channel Models, ICT-317669-METIS/D1.4, Feb. 2015. [Online]

[13] "Technical specification group radio access network; study on LTE device-to-device proximity services," 3rd Generation Partnership Project (3GPP), TR 36.843, Mar. 2014, Sections A.2.1.1 - A.2.1.2. [Online].