Distributed Time Synchronization in Ultra Dense Networks

Debjani Goswami[1], Nicola Marchetti[2], and Suvra Sekhar Das[1]
G. S. Sanyal School of Telecommunications[1], Indian Institute of Technology Kharagpur, India.
Trinity College[2], Dublin
Email: debjani.ami.89@gmail.com, nicola.marchetti@tcd.ie, and suvra@gssst.iitkgp.ernet.in

***Abstract*—To fulfill a higher data rate requirement in fifth-generation (5G) systems, mobile network operators are steering the access architecture towards ultra-dense network (UDN) deployments. However, the coexisting small cells and macrocells in UDN require highly accurate timing synchronization among the base stations (BSs). Traditional master-slave-based time synchronization techniques are not well-suited in several new deployment scenarios of 5G like dense urban, urban canyon, etc., which do not easily access the global positioning system (GPS). Moreover, a good back-haul requirement and high signaling overhead limit the applicability of existing techniques in small cell deployment scenarios. Despite a large amount of work done in the area, an *efficient* timing synchronization technique for dense networks is not well addressed, which this work aims to investigate. By applying an efficient approach for collecting information from an enlarged neighborhood at a minimal signaling cost in our modified timing update process, we provide a *low-complexity* and *faster* GPS-independent timing synchronization solution for a dense network. Numerical investigations evaluate the impact of the graph topology and nodes number on the network synchronization speed. Finally, we prove that the proposed algorithm achieves asymptotic convergence with probability one.**

***Index Terms*—Ultra dense networks (UDN), small cell base-station (SBS), connectivity factor (CF), synchronization accuracy, small cell network (SCN).**

I. Introduction

The 5G cellular network needs to support a 1000-fold increase in traffic volume compared to the 4G systems while ensuring a higher user-experienced data rate. However, the scarcity of available spectrum resources drives the search for efficient solutions to meet the increasing user needs with the maximum utilization of resources. Apart from using multiple antennas, the users' experienced data rate can be enhanced by densifying the network using small-range low-power BSs, also known as small cell base stations (SBSs). This positive aspect of small cells incentivizes the mobile network operators to steer the access architecture towards the UDNs. The Long Term Evolution-Advanced-enhanced Inter-Cell Interference Coordination (LTE-A-eICIC) allows cooperative small cells operating at frequencies of 1.9 GHz-2.6 GHz to coexist with a macro cell, i.e., 3GPP small cell scenario 1 [1]. However, the mismatches in these heterogeneous BSs timings increase the inter-BS interference resulting in low spectral efficiency in the network. Maintaining a standard time scale among all the network BSs can minimize the upraised inter-cell interference (ICI) in a network [2]. In this context, Fig. 1 depicts the various application scenarios in 5G that demand a highly accurate timing synchronization.

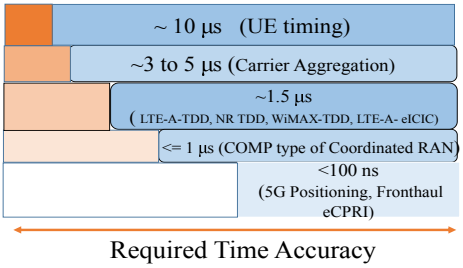

Fig. 1: Radio Access Network Synchronization Requirements [3].

A. Traditional Time Synchronization Schemes with Limitations

In this section, we start the discussion of the conventional time synchronization techniques and their implementation limitations. Network time protocol (NTP) is one of the most established protocols currently in use, which synchronizes computer clocks over a packet network [4]. However, the sensitivity to delay attacks, unicast mode of application, and the lack of end-to-end protection direct the search for an alternative solution in LTE-Advanced systems. The primary techniques for time synchronization used by the LTE-Advanced and the LTE-time division duplex (TDD) systems, i.e., "global navigation satellite system (GNSS) everywhere", "precision time protocol (PTP) with full path support (G.8275.1)" and "PTP with partial on-path support (G.8275.2)", follow the server-client configuration and use the universal time coordinated (UTC) information received from the GNSS [2]. The requirement of a reliable back-haul condition (e.g., fiber or ethernet) and length constraint of the coaxial cable that connects the PTP reference clock to a GPS antenna limits the deployment location options [5]. In addition, the variable queuing delays, the packet delay variations, and the asymmetry of sources (like switch delay and network path variation) degrade the timing accuracy achieved by PTP [2] significantly, requiring proprietary changes and on-site calibration in the network to overcome these issues. Moreover, the high signaling overhead in PTPv2.1 (IEEE 1588v2) limits its application in the small-cell scenario [6]. Sync-Ethernet technology of delivering synchronization to BSs demands Sync-E support for each device in between master and slave clocks, which makes it a costlier deployment solution [7]. Apart from these shortcomings mentioned above, the traditional master-slave based timing synchronization solutions experience some additional

drawbacks [8]: 1) Due to the accumulated time errors in every element, a highly accurate timing solution is difficult to ensure in the network; 2) Although the master-slave configuration ensures absolute accuracy, a highly accurate relative time error between neighboring BSs required for various 5G applications is not guaranteed. The above shortcomings drive the search for an alternative to the master-slave-based timing synchronization solution. Moreover, the events of solar activity, intentional or unintentional jamming, interference, spoofing, signal blockage, and the constellation failure make the GNSS signal unavailable to the receiver [9]. In addition, the limited satellite visibility, multi-path interference, and the non-line of sight sky view pose a threat of GNSS unavailability [9]-[10], which necessitates a GPS-independent backup solution for timing synchronization in a dense urban network [2]. The next section mentions some known facts on distributed time synchronization and discusses the suitability of existing GPS-independent timing synchronization solutions in dense networks.

B. Literature Survey in Distributed Time Synchronization

Among a massive volume of works on consensus-based distributed timing synchronization, such as [11–14], the authors of [13] exploit algebraic graph theory to achieve an average consensus both for fixed and time-varying networks. The average consensus in the context of that work signifies that the post-synchronized clocks maintain the mean of the initial clock timings for any directed graph only when it is *balanced* in nature, i.e., for each node in the network, the sum of incoming nodes is equal to the sum of outgoing nodes. Moreover, prior works [13]-[15] confirm that strong connectivity can ensure network synchronization. The requirement for a complete adjacency matrix's knowledge limits the application of semidefinite program (SDP)-based optimal solution for fastest average consensus [16] in large networks as it introduces significant communication overhead. In this context, an excessive signaling overhead caused by the periodic information transmission motivates the authors of [12] to implement decentralized event-triggered control strategies, which consider the presence of continuous virtual clocks in the network.

C. Time Synchronization Without Satellite Time Reference

The issues with GPS availability in multiple deployment scenarios (e.g., dense urban, urban canyon) encouraged the International Telecommunication Union-Telecommunication (ITU-T) to introduce the coherent network primary reference time clock (cnPRTC) [17], whose performance analysis is still in progress. In this context, authors in [18] propose a distributed time synchronization (DTS) approach, where the cooperative grand-master (GM) clocks achieve the synchronization in a self-organizing way. However, the BSs use the PTP protocol to collect the timing information from their associated GM clock, thereby experiencing the PTP protocol's limitations. Note that self-synchronization is very important for multi-link device-to-device (D2D)/vehicle-to-anything (V2X) communications, inter-vehicle communication, and smart grid applications in 5G systems [14], [19]. Apart from cellular networks, synchronization is equally crucial to WiFi-based systems as a highly desynchronized wireless device-access point pair introduces a rapid energy-draining in the system [20]. Motivated by the above fact, [20] reduces the energy consumption of wireless devices by ensuring a tighter synchronization between these two entities. More specifically, to minimize the energy expenditure by a wireless device, the difference between the actual local beacon reception time and the beacon reception wake-up time (calculated from the previous iteration) is minimized by modifying the beacon reception wake-up time iteratively. The above discussion confirms a high similarity of the work in [20] with the basic approach used in traditional distributed time synchronization schemes. The authors in [14] address the time synchronization solution for D2D communication, prioritizing the oldest timing references while updating its local timing. The convergence proof in [14] is beyond the scope of this work. [21] compares the superimposed signal-based synchronization technique in dense internet of things networks with the one that avoids the superimposition of signaling in the network. The authors confirm that the signature ensemble synchronization scheme is superior to the master-slave-based configuration, which motivates further investigations in collision-based synchronization in dense networks. In this context, [22, 23] compensate the negative impact of the propagation delay and multipath propagation on the widely used distributed phase-locked loop (DPLL) based self-synchronization scheme suggested in [24]. Motivated by the half-duplex nature of the transmission scheme, [23] allows each SBS to self-determine its suitable mode of operation. Unlike master-slave-based time synchronization, distributed decision-making allows all nodes to play an essential role in achieving the network time synchronization. To mitigate the negative impact of newly joined unsynchronized nodes or malicious nodes, [25] develops a neighbor-aware time synchronization protocol (NTSP), where each node can choose its suitable neighborhood that participates in the consensus process. Recent work in [26] adopts a hybrid Bayesian approach for clock offset and the skew estimation to achieve synchronization in 5G networks. A large number of required message-passing actions make the potentially strong believe-propagation technique unsuitable for periodic synchronization in large networks. From now on, this discussion is directed towards the gap analysis and the contributions of our work.

D. Gap Analysis and Contributions of Our Work

The readers should note that [15] achieves the time synchronization without any additional signaling exchange in the network. In particular, to achieve the synchronized network, each node extracts the timing information from its neighboring nodes and broadcasts the same within the network. This low signaling solution in [15] motivates us to investigate the framework's efficiency for densely deployed networks. Our numerical investigation confirms that applying the existing method in the presence of a large network size results in a rapid degradation in the synchronization speed and is hence best-fitted only for small network sizes. The above discussion confirms the requirement of a cost-effective, low-complexity, and low signaling overhead self-synchronization solution for dense networks that motivates our work. In this context, [19]

proposes a time synchronization solution for multi-link D2D / V2X communications, targeting deployment scenarios where the users are out-of-coverage from any network apparatus like evolved Node B (eNB), BS, etc. The highly similar application scenarios motivate a comparative study of [19] with the approach we propose, see Fig. 12. The numerical investigations confirm that our proposed work achieves faster synchronization than [15] and [19] even for large networks.
The contributions of this paper are summarized as follows:

- Different from [15], we come up with an efficient approach for collecting information from an enlarged neighborhood at a minimal signaling cost (see section III). The numerical investigation confirms that the application of the above idea in (9) can resolve the issue with the rate of speed degradation for large networks. We present the convergence proof of the proposed algorithm in section VI.
- The impact of the network topology and network size on the synchronization speed, standard deviation among the node timings, expected relative time offset, and deviation from the average consensus are evaluated and compared with the prior known work in [15] in section IV.
- Besides, we propose a slightly modified version of the timing update in (14) to encounter added hardware complexity.

II. System Model

This work considers a homogeneous cellular network for outdoor deployment scenarios, like public space, open stadium, etc., where the macro cell coverage is not present [27]. Note that this work follows the system model described in [15]. As shown in Fig. 2, we have considered a small cell network (SCN) consisting of K number of picocells, i.e., $\kappa = \{1, 2, 3, ..., K\}$. In the latter part of the discussion, picocells and small cells are used interchangeably. The S1 interfaces connect the mobile core network with their respective picocells, while the direct connection between small base stations (SBSs) is ensured by the X2 interfaces [28]. From now on, the terms SBSs and nodes are used interchangeably throughout the paper. As mentioned earlier, the unavailability of GPS disables the operation of the traditional master/slave synchronization techniques in many real-life scenarios, even in the presence of macro BS also. Hence, this work aims to provide an alternative solution for timing synchronization, which does not rely on the satellite time reference. In particular, in our proposed scheme, all the SBSs can cooperatively achieve a similar timing value, therefore not experiencing limitations like those of the PTP protocol. To achieve the goal of time synchronization, let us assume that each SBS k follows the local timing of $t^k(n)$ at the n^{th} discrete instant of time, where $k \in \kappa$ and $n \in \{0, 1, 2, 3, ...\}$. We consider a homogeneous network scenario, where all SBSs use the same fixed transmit power, P_t. More specifically, each SBS radiates a periodic train of impulses by following its local timing. As an example, the i^{th} SBS radiates a periodic train of $\sum_n g(t - t^i(n));\ i \in \kappa$ [29], where $t^i(n)$ is the local timing of i^{th} SBS at the n^{th} instant of time. The timing information of individual SBS can be decoded only by the neighbor SBSs. Unlike the existing works in this domain, the neighborhood of this work is distinguished into two subclasses. The incoming

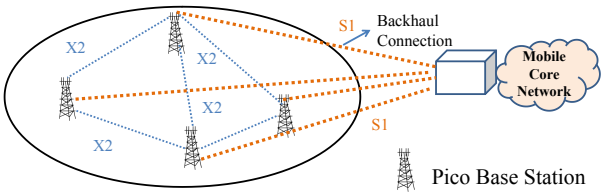

Fig. 2: System Model.

neighborhood of i^{th} SBS at the n^{th} instant of time is defined as:

$$N_i^I(n) = \{j : P_{ij}(n) \geq P_0\};\ i, j \in \{1, ., K\} \text{ and } i \neq j \quad (1)$$

where $P_{ij}(n)$ denotes the power received by i^{th} SBS from j^{th} SBS at the n^{th} instant of time and P_0 is the power threshold that determines the connectivity of the interference graph. Eq. (1) reveals that j^{th} node is considered to be the incoming neighbor of i^{th} node if and only if the interference power coming from j^{th} node to i^{th} node is greater than the power threshold value, P_0. Furthermore, the outgoing neighborhood of i^{th} SBS at the n^{th} instant of time is defined as:

$$N_i^O(n) = \{k : P_{ki}(n) \geq P_0\};\ i, k \in \{1, ., K\} \text{ and } i \neq k \quad (2)$$

This work assumes a perfect estimation of the timings of neighboring nodes by an individual receiver. Moreover, we assume that each node knows its distances from all the neighboring nodes in the network. We consider a Rayleigh fading channel, which ensures the presence of a non-line-of-sight communication in the outdoor deployment scenario. Thus, the channel gain G_{ij}, is equal to the square of channel coefficient from the j^{th} to the i^{th} SBS (i.e., $|h_{ij}|^2$), is represented as an exponential random variable. Moreover, the reciprocity in the channel gain confirms that $G_{ij} = G_{ji}$. The received strength of the interfering signal, which is coming from the j^{th} SBS to the i^{th} SBS, can be calculated by:

$$P_{ij}(n) = P_t |h_{ij}(n)|^2 d_{ij}^{-\alpha} = P_t G_{ij}(n) d_{ij}^{-\alpha} \quad (3)$$

where α represents the path loss exponent and d_{ij} is the distance between i^{th} SBS and the j^{th} SBS. Eq. (3) shows that in the presence of the reciprocal fading channel, the received power is symmetric, i.e., $P_{ij}(n) = P_{ji}(n)$. Let us assume that at a particular instant of time, n, $G(n) = (\kappa, E(n), \mathbf{A}(n))$ is the directed interference graph with fixed connectivity, which consists of K number of vertices. In particular, the SBSs are acting as the vertices of the interference graph $G(n)$ at the n^{th} instant of time. $E(n)$ and $\mathbf{A}(n)$ describe the edge connectivity and the adjacency matrix, respectively, at the n^{th} instant of time. We assume that the number of vertices remains the same across all time instants, which excludes the possibility of adding or removing any nodes in/from the network. However, we consider the variation in the connectivity graph, which captures the varying channel coefficient at different instants of time. Eq. (4) reveals that the weight between two SBSs is proportional to the measured interfering signal strength from one SBS to another, where the weight value $(a_{ij}(n))$ of the edge from j^{th} SBS to i^{th} SBS is assigned by following [15]:

$$a_{ij}(n) = \frac{P_{ij}(n)}{\sum_{j \in N_i^I} P_{ij}(n)} \tag{4}$$

Note that even in the presence of symmetric power, the difference between the incoming aggregated interferences received in i^{th} and j^{th} SBS makes the adjacency matrix **A** asymmetric in nature (i.e., $a_{ij}(n) \neq a_{ji}(n)$), which further makes the weighted interference graph unbalanced. Moreover, the assumption of symmetric power and homogeneity makes the network bidirectional (i.e., $a_{ij}(n) > 0$ iff $a_{ji}(n) > 0$).

III. Proposed Algorithm for Distributed Time Synchronization

A. Existing Works and Prerequisites of Time Synchronization

In this section, we have proposed a faster distributed timing synchronization technique for the SCN. Before the detailed discussion of our proposed work, we start with the established timing update process from [15]:

$$t^k(n+1) = t^k(n) + \epsilon \sum_{j=1,\ j\neq k}^{K} a_{kj}(n)[t^j(n) - t^k(n)] \tag{5}$$

where $t^k(n+1)$ denotes the local time value of the k^{th} SBS at the time instant $(n+1)$ and ϵ is the step size $(0 < \epsilon < 1)$ [15]. Eq. (5) reveals that each node uses the timing information of its incoming neighboring nodes to update its local timing. Moreover, the dependency increases for a higher interfering SBS, which is captured in the weight value in matrix **A** [15]. Please note that $a_{ii}(n) = 0$ and $\sum_{j=1,j\neq i}^{K} a_{ij}(n) = 1\ ; \forall i,\ \forall n$. As mentioned earlier, the work proposed in [15] experiences a rapid speed degradation problem with increasing network size, hence necessitates a modification to address suitable timing synchronization solution in densely deployed scenarios. Hereinafter, this section provides some prerequisites of the timing synchronization requirements in 5G systems and subsequently provides a detailed discussion of the proposed algorithm. In practice, to ensure cost-effectiveness, small cells use low-cost temperature-controlled crystal oscillators (TCXO) or voltage-controlled crystal oscillators (VCXO), which experience the maximum frequency error of ± 1 part per million. The presence of jitters and receiver imperfections introduces drifts in the node timings, leading to the necessity for periodic synchronization in the networks. In this regard, a complete guideline on the resynchronization interval for various frequency errors is presented in [30], which specifies that the required timing accuracy for small cells is less than 3 μsec. A set of nodes is said to be δ_{error}-synchronized at a particular instant of time n, iff the timing deviation between any two nodes is less than equal to δ_{error}, i.e., $|t^1(n) - t^2(n)| \leq \delta_{error}$. In other words, the loss of the δ_{error}-synchronization between any node pair necessitates the resynchronization in the network. Note that the maximum relative drift between any two node timings is 2ρ, where ρ represents the maximum drift of each node timing. Hence, the maximum relative drift in the timings between any pair of nodes for the observation time of τ_{max} is expressed as $2\tau_{max}\rho$. Above discussion reveals that the resynchronization will be performed when $2\tau_{max}\rho > \delta_{error}$, which confirms that the resynchronization interval, $T_{SyncIn} \leq \frac{\delta_{error}}{2\rho}$ [31]

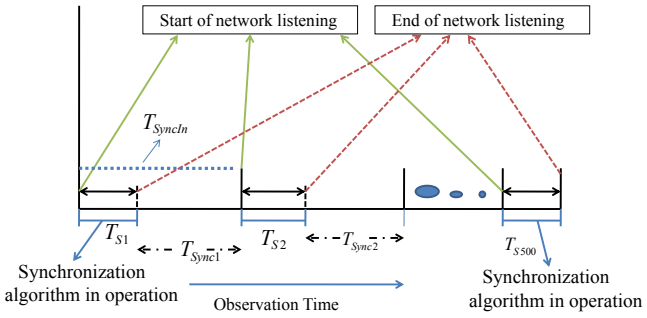

Fig. 3: Timing Diagram of the proposed algorithm consisting of the algorithmic operational time slot (T_{Sj}), data transfer phase (T_{Syncj}), and the resynchronization interval (T_{SyncIn}).

B. Proposed Algorithm and Computational Complexity

This work proposes a periodic synchronization algorithm, as shown in Fig. 3, which uses the resynchronization interval (T_{SyncIn}) of 1.5 sec (following δ_{error} = 3 μsec and $\rho = \pm 1$ part per million). Fig. 3 shows that the number of resynchronizations performed within the observed time duration is 500. Moreover, each resynchronization period, j contains an algorithmic operational time slot (T_{Sj}) and the data transfer phase (T_{Syncj}). In particular, T_{Sj} during which the proposed synchronization executes is expressed as

$$T_{Sj} = \sum_{n=1}^{n_{max}} \Delta_{nj}; \qquad j \in \{1, 2, ..., 500\} \tag{6}$$

where,

$$\Delta_{nj} = \begin{cases} 1; \text{if } \sqrt{\frac{\sum_{i=1}^{K}\left(t^i(n,t^i(n-1),\mathbf{A}(n),\bar{\mathbf{A}}) - \bar{t}(n,t(n))\right)^2}{K-1}} > \delta \\ 0; \text{else} \end{cases}$$

where n_{max} and δ represent the maximum allowed number of iterations during each algorithmic operation and the maximum standard deviation (SD) allowed among the node timings, respectively. Hereinafter, we define each T_{Sj} slot as a time duration during which all nodes listen to their neighborhood. Note that in (6), T_{Sj} is expressed as the summation of the Δ_{nj}'s, where n varies from 1 to n_{max} and the Δ_{nj} is an indicator function (i.e., $\Delta_{nj} \in \{0, 1\}\ ;\ n \in \{1, 2, ..., n_{max}\}$), whose value is '1' for all the iterations until the standard deviation among the timings meets the threshold value, δ. Hence, T_{Sj} calculates the number of iterations required to achieve the network synchronization at the j^{th} snapshot of observation, where $j \in \{1, 2, ..., 500\}$. One can easily follow that the thermal instability and the aging introduce a continuous drift in the node timings during both the periods of T_{Sj} and T_{Syncj}. The effect of thermal instability have considered in our numerical investigation. However, the negligible aging effect in TCXO (10-20 ppm/10 years) is ignored in the performed investigations. We calculate the time drift from the expression of the fractional change of frequency due to the temperature variation [32]:

$$F_{Error}(\text{ppm}) = \beta(T_O - T)^2 \tag{7}$$

where T_O is the turnover temperature of 25° C, T and β

represent the node temperature and the temperature coefficient, respectively. In particular, our work overcomes the rapid speed degradation issue of [15] by exploiting both the information of incoming and outgoing interferences in the timing update process stated in (9). Although less signaling overhead is always preferred in communication networks, the authors in [33] suggest a physical signal-to-interference and noise ratio (SINR) model-based collision aware low complexity broadcast algorithm that completes the broadcasting of local information within the time slot of complexity $\mathcal{O}(\log K)$. Motivated by this work, we propose a faster distributed synchronization algorithm for the dense network scenario, which allows a minimal message exchange within the network. According to (1), the nodes that interfere with a particular SBS are defined as the incoming neighbors of this SBS. By utilizing the received signal strength indicator (RSSI), each SBS can measure the received signal strengths of its incoming neighbor SBSs. However, the lack of synchronization prevents the exchange of information from one SBS to another, which makes the information gathering from the outgoing neighborhood difficult for an asynchronous network. Our algorithm, whose pseudocode is shown in Algorithm 1, suggests an idea to exploit information from both the subclasses of neighborhood while ensuring a minimal signaling overhead in the networks. The initial phase of the proposed algorithm considers a fully synchronized network that allows the exchange of information within it. If such a synchronized network is not available, then the initial stage of the algorithm achieves the synchronized state by following the update process in (5). Note that in a synchronized network, each SBS can easily collect the aggregated incoming interference information from its outgoing neighbors ($\bar{N}_i^O;\ \forall i$). However, the collection of information from *all of the* outgoing neighbors is possible only for bidirectional networks, as bidirectionality implies the presence of all outgoing neighbors in the incoming proximity as well $\left(\text{i.e., } \bar{N}_i^O = \bar{N}_i^I;\ \forall i\right)$. In the next snapshot, when the network becomes asynchronous, each SBS updates its local timing by exploiting the previously gathered information of its outgoing neighbors and the currently measured interference information from the incoming neighborhood. More specifically, each node collects information from its respective outgoing neighbors only once in the data transfer phase of the immediately previous synchronized network, thereby requiring K number of additional information exchanges in each synchronization period. To better describe the proposed idea, by following (4), the adjacency matrix $\mathbf{A}$ for a network consisting of $K = 3$ SBSs is defined as:

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{P_{12}(n)}{\sum_{j\in N_1^I} P_{1j}(n)} & \frac{P_{13}(n)}{\sum_{j\in N_1^I} P_{1j}(n)} \\ \frac{P_{21}(n)}{\sum_{j\in N_2^I} P_{2j}(n)} & 0 & \frac{P_{23}(n)}{\sum_{j\in N_2^I} P_{2j}(n)} \\ \frac{P_{31}(n)}{\sum_{j\in N_3^I} P_{3j}(n)} & \frac{P_{32}(n)}{\sum_{j\in N_3^I} P_{3j}(n)} & 0 \end{bmatrix} \tag{8}$$

where $\sum_{j\in N_i^I} P_{ij}(n)$ denotes the aggregated incoming interference toward i^{th} SBS. From the adjacency matrix in (8), one can easily follow that $a_{12} = \frac{P_{12}(n)}{\sum_{j\in N_1^I} P_{1j}(n)}$ and $a_{21} = \frac{P_{21}(n)}{\sum_{j\in N_2^I} P_{2j}(n)}$. As discussed earlier, each SBS, say SBS_1, can measure the interference power from the SBS_2 (P_{12}), as $SBS_2 \in N_1^I$. Moreover, the assumption of symmetric power implies that $P_{12} = P_{21}$. Hence, the knowledge of the aggregated incoming interference in SBS_2, i.e., $\left(\sum_{j\in N_2^I} P_{2j}(n)\right)$ enables SBS_1 to calculate the weight coefficient, a_{21}. Similarly, SBS_1 can calculate all the a_{j1}; where $\forall j \in N_1^O \cap N_1^I$, if it can collect the aggregated incoming interference information from all these outgoing neighbors. Hence, at the initial stage of the algorithm, each SBS broadcasts the information of its aggregated incoming interference along with collecting the same from its outgoing neighbors. Initially, our proposed algorithm takes the input of the gathered information from the first snapshot (i.e., $T_{index} = 1$) and executes the synchronization process from $T_{index} = 2$. The collection of information from the immediately previous synchronized state continues for all the resynchronizing operation instants. Note that the packet losses and the asynchrony in the timings of the SBSs may introduce a significant communication delay for exchanging the timing information within the neighborhood. The work in [34] confirms that the constrained consensus is possible to achieve in the presence of the random but bounded communication delay in an unbalanced time-varying graph if the union of the time switching graphs holds the property of strong connectivity. Our proposed algorithm ensures that the union of the time switching graphs is a complete graph (see section VI), which confirms the convergence of this algorithm even in the presence of random but time-bounded communication delay. During the asynchronous state of the network, each SBS updates its local time value, which uses the current entries of the $\mathbf{A}$ matrix corresponding to the incoming neighbor SBSs and the past gathered information of the outgoing neighbors ($\bar{\mathbf{A}}$ matrix). In particular, our proposed scheme follows the update procedure of:

$$\begin{aligned} t^k(n+1) =& t^k(n) + \epsilon \sum_{j\in\kappa,\ j\neq k} \frac{a_{kj}(n) + \bar{a}_{jk}}{2} \left[t^j(n) - t^k(n)\right] \\ =& t^k(n) + \underbrace{\epsilon \sum_{j\in\kappa,\ j\neq k} \frac{a_{kj}(n)}{2} \left[t^j(n) - t^k(n)\right]}_{X} + \\ & \underbrace{\epsilon \sum_{j\in\kappa,\ j\neq k} \frac{\bar{a}_{jk}}{2} \left[t^j(n) - t^k(n)\right]}_{Y} \end{aligned} \tag{9}$$

where, $a_{kj}(n) = 0$ if $j \notin N_k^I$, and $\bar{a}_{jk} = 0$ if $j \notin \bar{N}_k^O \cap N_k^I$. $\bar{a}_{jk}$ signifies the entries of $\bar{\mathbf{A}}$ matrix corresponding to the outgoing neighbors of k^{th} SBS in the previous snapshot. Note that the k^{th} SBS can gather the information only from the nodes j, where $j \in N_k^I$ in $\bar{\mathbf{A}}$ matrix, which signifies $j \in \bar{N}_k^O \cap \bar{N}_k^I$ and $j \subseteq N_k^I$, where equality in $j \subseteq \bar{N}_k^I$ holds for the bidirectional network. a_{kj}'s are instead gathered from the current adjacency matrix ($\mathbf{A}$). The above-discussion reveals that this work assumes that all the SBSs are capable of multicasting their aggregated incoming interference information within their neighborhood. In this context, 3GPP Rel-17 newly introduces the multicast / broadcast support at 5G core network [35], which confirms that the proposed work can

be considered as a candidate time synchronization solution in future networks. In this work, we define the connectivity factor

Algorithm 1: Algorithm for a Faster Distributed Time Synchronization.

Input: Initial power matrix $(\mathbf{P}^1)_{K\times K}$, Maximum number of snapshots (T_{max}), Maximum number of iterations per snapshot $(n_{max}-1)$, Initial adjacency matrix $(\mathbf{A}^1)_{K\times K}$, Distance matrix $(\mathbf{D})_{K\times K}$, Maximum SD allowed (δ), Objective function $C^{(0)}(\mathbf{A}_n^{T_{Index}}, \bar{\mathbf{A}})$, Initial time vector $(\mathbf{T}^{(0)})_{1\times K}$.
Initialize: $\mathbf{P}^1$, $n = 0$, $T_{Index} = 2$, $\bar{\mathbf{A}} = \mathbf{A}^1$, $C^{(0)}(\mathbf{A}_n^{T_{Index}}, \bar{\mathbf{A}}) = 0$,
for $T_{Index} \leq T_{max}$ **do**
  Generation of power matrix $\left(\boldsymbol{P}_n^{T_{Index}}\right)$ for all SBSs by following (3).
  Generate the adjacency matrix $\mathbf{A}_n^{T_{Index}}$ using (4) and the time vector $(\mathbf{T}^{(n+1)})$ by following (9)
  Calculate, $C^{(n+1)}(\mathbf{A}_n^{T_{Index}}, \bar{\mathbf{A}})$ by using (11).
  if $(C^{(n+1)}(\boldsymbol{A}_n^{T_{Index}}, \bar{\boldsymbol{A}})) > \delta$ *and* $n < n_{max}$ **then**
    $n = n + 1$;
    Generation of power matrix $\left(\boldsymbol{P}_n^{T_{Index}}\right)$ for all SBSs by following (3) and corresponding adjacency matrix $\mathbf{A}_n^{T_{Index}}$.
    Update the time values of all SBSs $(\mathbf{T}^{(n+1)})$ using $(\mathbf{T}^{(n)})$ by following (9) and calculate the SD among time values $(C^{(n+1)}(\mathbf{A}_n^{T_{Index}}, \bar{\mathbf{A}}))$ by using (11);
    go to the line 6
  else
  end
  Record $C^{(n+1)}(\mathbf{A}_n^{T_{Index}}, \bar{\mathbf{A}})$, n, and $\mathbf{T}^{(n+1)}$ for this snapshot.
  $\bar{\mathbf{A}} = \mathbf{A}_n^{T_{Index}}$
  $T_{Index} = T_{Index} + 1$
end
Average the $C^{(n+1)}(\mathbf{A}_n^{T_{Index}}, \bar{\mathbf{A}})$ and n over all snapshots.
Return: SD between time values $(C_{\text{avg}}^{(n+1)}(\mathbf{A}^{T_{Index}}, \bar{\mathbf{A}}))$, Number of required iterations to converge n_{avg}.

(CF) as

$$C_F = \frac{\sum_{i=1}^{K}(|N_i^I| + |N_i^O|)}{2\binom{K}{2}} \tag{10}$$

which captures the average degree of connectivity between the SBSs in the networks. As the numerator of (10) counts each edge in both directions, the connectivity of the directed graph is normalized by a factor of 2, and the SD of the n^{th} instant of time is defined as

$$C^{(n)}(\mathbf{A}, \bar{\mathbf{A}}) = S_D(n) = \sqrt{\frac{\sum_{i=1}^{K}(t^i(n) - \bar{t}(n))^2}{K-1}} \tag{11}$$

where, $\bar{t}(n)$ denotes the mean value of timing vector at the n^{th} instant of time. Although the numerical investigations in section IV confirm the effectiveness of our proposed work in terms of the achievable synchronization speed, here below we investigate the communication complexity of the proposed framework, as having a low signaling overhead is one of the prime service requirements for ultra-dense networks [36]. Table I provides a comparative study of the communication complexity experienced by our work with some prior known results in this domain, which confirms that our proposed algorithm exploits comparatively minimal signaling overhead than all the references mentioned in this table.

TABLE I: Signaling overhead comparison among the schemes of time synchronization.

Time Synchronization Algorithms	**Communication Complexity**
Faster Synchronization [16]	$K\kappa_{avg}$
Precision Time Protocol (PTP) [37]	$2(k_0+k_N)+4(k_1+k_2+\ldots+k_{N-1})$
Wireless Precision Time Protocol (WPTP) [37]	$k_0+2(k_1+k_2+\ldots+k_N)$
Proposed solution for distributed time synchronization	K

PTP and WPTP in Table I use multiple levels of hierarchy, where the level n contains k_n number of nodes, such that $\sum_{n=0}^{N} k_n = K$, where K is the maximum number of nodes in the network. n_{avg} denotes the average number of iterations required to achieve the synchronization in work [16].

IV. Numerical Analysis

This section evaluates the impact of the graph topology and the network size on the speed of synchronization. In particular, we have quantified the speed of the convergence in terms of the number of required iterations expressed by (6) to achieve the synchronization state in the network. In Table II, we list the values of the system parameters that are used in the investigation. In this section, we provide a brief discussion on our performed numerical investigation. The analysis considers 500 synchronization periods, each having a maximum of 20,000 iterations. The synchronization algorithm exploits these iterations to diminish the drifts in node timings until it converges or reaches the maximum number of iterations. All nodes experience different realizations of received powers at distinct instants of time due to the temporal variation in a communication network. We consider the fixed location of picocells, where the SBSs are uniformly placed in a two-dimensional square grid having an area of 100 meter^2. Moreover, we assume that the initial drifts of SBSs timings are uniformly distributed in the range of [0,40 μs], as the maximum drift possible in one resynchronization interval is 40 μs by following (7). In addition, this investigation considers the drift introduced due to the thermal instability during both the periods of T_{Sj} and T_{Syncj}, which are calculated by using (7). Furthermore, we assume that the channel remains constant for each iteration of the resynchronization period, varying over different iterations. The study about the impact of step size on the network synchronization speed in [15] confirms a slower convergence in the presence of small step size. Motivated by the fact that step size lies in the range of (0,1) and by following the value used in [15], we set the step size value to 0.9 to investigate the best achievable synchronization speed in the network.

Initially, the numerical investigation in Fig. 4 compares the speed of convergence of our proposed algorithm with [15] and

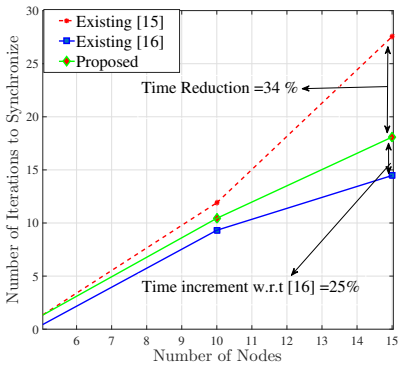

Fig. 4: Number of iterations required to achieve synchronization as a function of the number of network nodes, for the maximum number of nodes = 15.

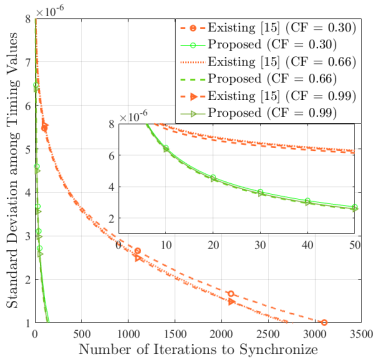

Fig. 5: Standard deviation among the timing values as a function of the number of iterations required to achieve synchronization for different CF values and for a network size of 250.

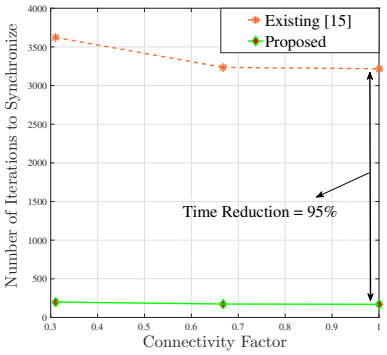

Fig. 6: Number of iterations required to achieve synchronization as a function of the CF in the networks for the number of nodes = 250.

TABLE II: Simulation Parameters.

Parameter	Value	Parameter	Value
Number of SBSs	250	Path loss exponent	4
Power Thresold	-110 dBm [38]	ϵ	0.9
Base station transmit power	24 dBm [36]	Node Temperature	0 – 50°
Temperature Coefficient (β)	-.042 ppm/$^\circ C^2$[32]	Maximum SD allowed (δ)	1e-06

that of the optimal solution described in [16]. Although the optimal solution in Fig. 4 confirms an earlier synchronization in the network, the time complexity restricts the application of the optimal solution in a large network (Table III). Due to the high computational complexity of the optimal solution, in Fig. 4, we restrict the numerical comparison with the optimal solution for small network size. Hereinafter, the numerical investigations consider a fixed large network size and subsequently capture the impact of network size on the synchronization speed. Fig. 5 computes the SD among the timings of all SBSs for the different values of CF. This figure confirms that an increment in the CF reduces the SD among the time values for a fixed network size. Eq. (11) shows that the SD captures the degree of closeness among the timing values of all SBSs, where more mismatches in the timing values are captured by a higher value of $S_D(n)$. Furthermore, CF measures the average connectivity among the SBSs in the networks. Hence, Fig. 5 can be interpreted as follows: an increment in CF implies the timing information exchange within a higher number of SBSs in the networks. Intuitively, with an increased number of communicating SBSs in the system, the possibility that the timing update will be more precise increases, thus reducing the SD value in the network. Fig. 6 exhibits the number of iterations required to achieve the synchronization as a function of the CF for fixed network size. This figure reveals that an increase in the CF makes the synchronization speed faster, which echoes the earlier finding of [15]. Fig. 5 justifies the observed parametric trend in Fig. 6, which shows that an increased CF leads to faster synchronization. Hence, Fig. 5 and Fig. 6 reveal that an increase in the CF value causes a rapid decrease of SD, which confirms that the network synchronizes earlier. In the case the graph is strongly connected, [13] shows that convergence is guaranteed even if the graph is unbalanced. However, the average consensus is not guaranteed in such a case. Although the consensus point of any directed graph (balanced/unbalanced) is known for a fixed network topology [13], to the best of our knowledge the consensus point of an unbalanced directed graph with a switching topology is still not well-investigated in the existing literature. Therefore, we will then investigate how close our proposed solution is to that of the average consensus. Fig. 7 addresses this query. In particular, this figure measures the degree of closeness of the final timing value to that of the average consensus. More specifically, Fig. 7 captures the variation in the accuracy value for a fixed size of the network, where the accuracy is measured in terms of the deviation of the final timing (i.e., $t^i(F) \;\; ; \forall \;\; i$) values from the mean value of the initial time vector, $\bar{t}(0)$, which is expressed as:

$$A_c \;=\; \sum_{i=1}^{K} \left[|(t^i(F) - \bar{t}(0))| \right] \tag{12}$$

This figure confirms that our proposed scheme returns a final timing solution comparable to the average consensus.

The discussion so far is limited to a given size of the network. Hereinafter, we evaluate the effect of the network size on the speed of synchronization. Fig. 8 depicts the variation in the SD for different sizes of the network, showing a larger SD in the presence of a higher number of SBSs. The observed trend can be justified as follows: an increment in the network size leads to a slower reduction of the timing mismatches within the networks. Fig. 9 shows the impact of different network sizes on the speed of synchronization, which reveals that an increment in the number of nodes reduces the synchronization speed in the network. The observed trend is explained as follows: as mentioned earlier, growth in the

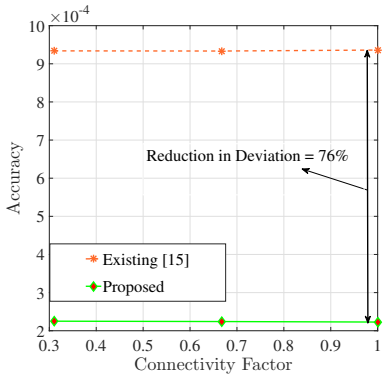

Fig. 7: Deviation from the average consensus as a function of the CF in the networks for the number of nodes = 250.

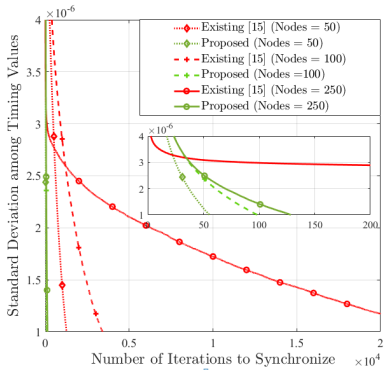

Fig. 8: Standard deviation among the timing values as a function of the number of iterations required to achieve synchronization for different network sizes.

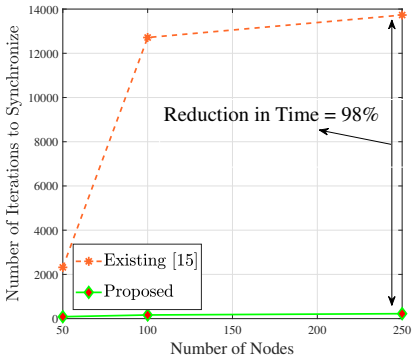

Fig. 9: Number of iterations required to achieve synchronization as a function of the network size.

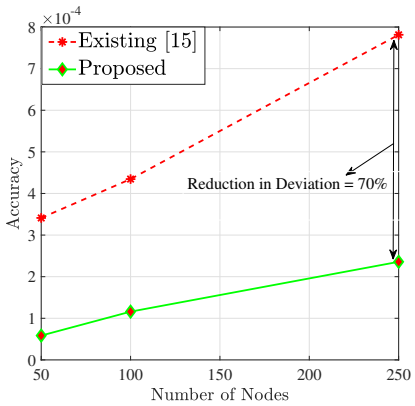

Fig. 10: Deviation from the average consensus as a function of the number of nodes in the network.

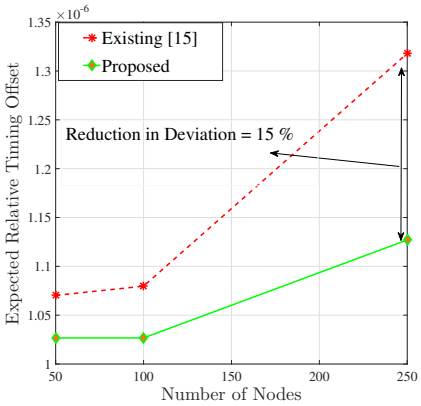

Fig. 11: Expected Relative Deviation in timing values as a function of number of nodes in the networks.

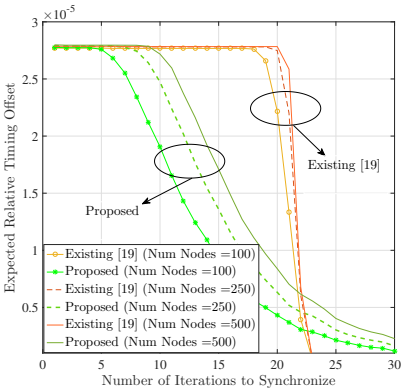

Fig. 12: Relative deviation among timing values as a function of number of iterations to synchronize for different network sizes.

network size requires a longer time to diminish the mismatches among the timing vector, slowing down the synchronization speed. Fig. 8 explains the observed trend of this figure as it confirms a rapidly decreasing SD value for the small number of nodes in the networks. Fig. 9 confirms a 98% faster synchronization than that of work in [15] for large networks, which makes it best-fitted in the densely deployed scenarios. Fig. 10 depicts the impact of the network size on the accuracy in the network. A prominent reduction in the deviation from the average consensus is observed in comparison to [15] for a large network. The numerical investigation in Fig. 11 exhibits the mean relative time offset as a function of the network size. In particular, the expected relative timing offset is expressed as:

$$R_D = \mathbb{E}[|t_i(n) - t_j(n)|] \tag{13}$$

where the expectation is taken over both the all possible pairs of nodes and all the time instants. In this context, the numerical investigation in Fig. 12 compares our proposed work with the work in [19] by measuring the variation of the expected relative offset as a function of the number of iterations to achieve the same. This figure reveals that our proposed framework can achieve a faster synchronization in terms of the relative timing error in comparison to [19].

V. Implementation of Proposed Algorithm

This section comprises a brief discussion on the practical implementation aspects of our proposed framework. In more detail, our work follows the discrete-time phase locked loop (DPLL)-based time synchronization approach that is described in [15], as this algorithm scales well in large networks among all the alternatives ways of achieving the synchronization

[22], like bio-inspired firefly algorithm [39], MAC timestamp exchanges [40], etc. To highlight the required modification of the time difference detector setting mentioned in [15], we start our discussion with the DPLL-based time synchronization technique proposed in [15]. Note that the implementation of X portion of (9) is possible with the help of a single match filter-based time difference detector setting as described in [15]. In their proposed implementation technique, each node, k transmits a bandlimited waveform (like square-root raised cosine pulse) centered at the local time instant with symbol period $1/F_s$. Then an individual receiver extracts multiple peaks from the received combination of waveforms by performing base-band filtering matched to the transmitted waveform followed by sampling the signal at L/T_s, where L is the oversampling factor (i.e., $L \geq 1$) and $T_s = 1/F_s$. However, unlike the X portion which does not require the separation of a single time difference from the incoming pulses of multiple transmitters, the usage of $\bar{a}_{jk}$'s in the Y part of (9) requires the additional information of the node identifiers corresponding to all the extracted timings by each receiver, hence requiring modification of the above mentioned single matched filter-based time difference detector setting. Hereinafter, we discuss the modified time difference detector setting which can separate the time differences from an individual node. In this setup, each transmitter sends a pulse shape that is both timelimited between $[-\frac{T}{2}, \frac{T}{2}]$ and bandlimited between $[-\frac{B_i}{2}, \frac{B_i}{2}]$ corresponding to the transmitter index i. Furthermore, all of the transmitted pulses are orthogonal in the frequency domain. Each receiving node uses a bank of bandpass filters (BPFs) individually tuned to different B_i's followed by the matched filter with threshold detection for peak search method. This set-up can be further simplified with the help of a prolate spheroidal wave function (PSWF) pulse transmission and detection [41]. This would remove the need for BPF at the receiver end as PSWF pulses can be tuned to provide user defined bandlimited and timelimited pulses. At the receiver end, the impulse response of the matched filter can be tuned to constrain the particular bandwidth corresponding to each of the transmitted pulses. Interested readers are referred to [42] that demonstrates the experimental set up to transmit and receive data sequence as PSWF. The above-discussion can be summarized as follows: To separate all time differences, each transmitter should send its pulses using a distinct frequency band. Moreover, instead of the single matched filter-based time difference detector [15] setting, each receiving node uses a bank of matched filters that are matched with different transmitted waveforms. Moreover, the afore-mentioned implementation setting requires global knowledge of the pulses used by different nodes and their allocated bandwidth in the networks, which can be exchanged during the data transfer phase of initially synchronized networks. The afore-mentioned added hardware complexity is realizable for the infrastructure-supported pico BSs. On the other hand, to encounter the increased hardware complexity, we propose herewith an easily implementable timing update method by slightly modifying the coefficient of the Y portion of the local time update process. More specifically, instead of using

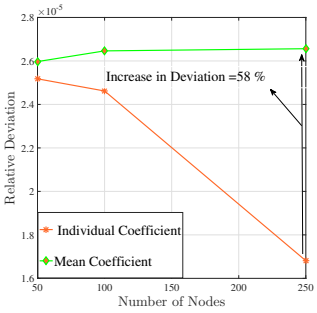

Fig. 13: Comparative study of the relative time offsets as a function of network size.

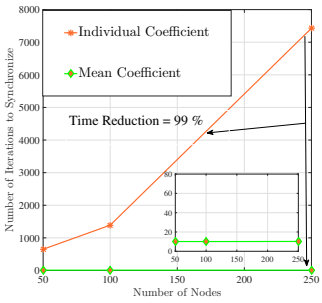

Fig. 14: Comparative study of number of iterations required to achieve a specific standard deviation as a function of network size.

specific values of $\bar{a}_{jk}$'s, each node k uses the mean coefficient value $\bar{a}_{k_{\text{mean}}}$ for all the outgoing neighboring nodes in (9), where $\bar{a}_{k_{\text{mean}}} = \frac{\sum_{j \in \bar{N}_k^0} \bar{a}_{jk}}{|\bar{N}_k^0|}$. Hence, the modified time update process in our proposed work is defined as:

$$t^k(n+1) = t^k(n) + \epsilon \sum_{j \in \kappa,\ j \neq k} \frac{a_{kj}(n) + \bar{a}_{k_{\text{mean}}}}{2} \left[t^j(n) - t^k(n)\right] \tag{14}$$

where, $a_{kj}(n) = 0$ if $j \notin N_k^I$, and $\bar{a}_{jk}$ in $\bar{a}_{k_{\text{mean}}}$ is equal to 0 if $j \notin N_k^O \cap N_k^I$. As the modified update process does not require any separation of information from multiple transmitters, it can then be implemented using the single matched filter-based time difference detector setting. Although the usage of an individual coefficient in the update process can provide a comparatively lower relative time error (Fig. 13), the mean-coefficient based update process can achieve an earlier synchronization in the SD among timing values (Fig. 14). Moreover, the application of minimal hardware in the mean coefficient-based update process makes it a cost-effective mutual time synchronization solution, especially for the low-infrastructure distributed applications.

VI. Convergence Proof of Proposed Algorithm

Theorem 1. *Consider $\mathbb{G}$ be the time sequence of the bidirectional directed graphs $\mathcal{G}(n) = \{\kappa, E(n), \boldsymbol{A}(n)\}$ with a common node set, $\kappa = \{1, ..., K\}$ and the set of the observation time instants $\mathcal{N}$, where, $n \in \mathcal{N}$. $E(n)$ and $\boldsymbol{A}(n)$ represent the edge connectivity and the adjacency matrix, respectively, at the n^{th} instant of time. All K agents of the linear update protocol in (9) will converge to a common time value, t (i.e., $t^1(F) = t^2(F) = ... = t^K(F) = t$) as $N \to \infty$ with probability one.*

Proof. The convergence of the distributed timing synchronization algorithm ensures a common time scale in all the SBSs in the network. Notice that the convergence of the algorithm highly depends on the graph topology. Moreover, the graph connectivity is directly influenced by the threshold value of power (P_0). Eq. (1) reveals that the formation of an edge from

i^{th} to j^{th} SBS, $E_{ji}(n)$, depends on the value of $P_{ji}(n)$, where

$$E_{ji}(n) = \begin{cases} 1 \ ; & \text{if } P_{ji}(n) \geq P_0 \\ 0 \ ; & \text{else} \end{cases} ; \quad (15)$$

Eq. (15) shows that at a particular instant of time, no edge will be formed from i^{th} SBS to the j^{th} SBS (i.e., $E_{ji}(n) = 0$) iff $P_{ji}(n) < P_0 \Longrightarrow G_{ji}(n) < \frac{P_0 d_{ji}^{\alpha}}{P_t}$, which follows from (3). In this work, we assume that G_{ji} is an exponentially distributed random variable with mean '1', which captures the effect of fading from the i^{th} SBS to the j^{th} SBS. Hence, at the n^{th} instant of time, the no-edge formation probability from i^{th} SBS to j^{th} SBS (i.e., $\mathbb{P}(E_{ji}(n) = 0)$) is defined as:

$$\mathbb{P}\left(G_{ji}(n) < \frac{P_0 d_{ji}^{\alpha}}{P_t}\right) = \quad 1 \ - \ \exp\left(-\frac{P_0 d_{ji}^{\alpha}}{P_t}\right) \quad (16)$$

Eq. (16) confirms that the no-edge formation probability from the i^{th} SBS to the j^{th} SBS ($\mathbb{P}(E_{ji}(n) = 0)$) is strictly less than one that equivalently confirms the positive edge formation probability. Now, to show that the proposed algorithm converges with probability *one*, we will use relevant results from [15] and [43] that are stated in *Fact 1* and *Fact 2*.

Fact 1: Let $\mathbb{G}$ be the sequence of the weighted directed graph $\mathcal{G}(t) = \{\kappa, E(t), \mathbf{A}(t)\}$ with a common node set, $\kappa = \{1, ..., K\}$ and $t \in \mathcal{N}$, where $\mathcal{N}$ is the set of the observation time instants. $E(t)$ and $\mathbf{A}(t)$ describe the edge connectivity and the adjacency matrix, respectively, at the time instant, t. If the graph is bidirectional $\forall t \in \mathcal{N}$, then the K components of the linear update protocol in (5) will converge to a common value as t $\rightarrow \infty$, only when for all $t_0 \in \mathcal{N}$, there is a node connected to all other nodes across $[t_0, \infty)$. In particular, a node $k \in \kappa$ is considered to be connected to a node $l \in \kappa \backslash \{k\}$ across an interval of $\mathcal{I}$ if k is connected to l for the directed graph of $(\kappa, \cup_{t\in\mathcal{I}} E(t), \cup_{t\in\mathcal{I}} \mathbf{A}(t))$ [43], where $\mathcal{I}$ is the observations time instant sequence and $\mathcal{I} \subseteq \mathcal{N}$.

Fact 2: The immediate interpretation of *Fact 1* is applied in [15] to propose the distributed synchronization of the multi-agent wireless network. The authors in [15] confirm that the distributed synchronization asymptotically converges to a common value of phases for all the agents in the networks, i.e., $\tau_1(n) = \tau_2(n) = ... = \tau_K(n) = \tau$; for $n \rightarrow \infty$ iff the associated graph $\mathcal{G}(n)$ is strongly connected across the observed interval of $\mathcal{I} \subseteq [n_0, \infty)$ for any $n_0 = \{0, 1, 2, 3..\}$, i.e., $\mathcal{G}(n) = (\kappa, \cup_{n\in\mathcal{I}} E(n) \cup_{n\in\mathcal{I}} \mathbf{A}(n))$ is strongly connected.

It may be noted that the strong connectivity ensures a directed path between any pair of nodes, where an individual node can be connected to another node via some intermediate node. Hence, a complete graph, which has a direct connection between any pair of nodes, always fulfils property of the strong connectivity. According to *Fact 1* and *Fact 2*, the timing synchronization is guaranteed for any complete graph. Hence, to prove the convergence, it is sufficient to show that $\lim_{N\rightarrow\infty} (\kappa, \cup_{n\in\mathcal{I}} E(n), \cup_{n\in\mathcal{I}} \mathbf{A}(n))$ is a complete graph with probability *one*. Note that the symmetric power assumption confirms that $E_{ij}(n) = 1$ iff $E_{ji}(n) = 1$. Suppose the edge between SBS_1 and SBS_2 is labeled as E_1. Let us assume that at a particular instant of time, the no-edge formation probability (i.e., $\mathbb{P}(E_1 = 0)$) between SBS_1 and SBS_2 is p. The edge formation probability in each instant of time is independent of each other. Hence, the probability that no edge will be formed between the SBS_1 and SBS_2 within the N observation instants is p^N. In other words, the probability of edge formation between the SBS_1 and SBS_2 for at least one instant of time can be expressed as $(1 - p^N)$, where N be the maximum instant of time for observation. This argument is true for all edges in the network. Now, suppose M denotes the maximum number of possible edges in the network consisting of K number of nodes, i.e., $M = \binom{K}{2}$ and all edges are marked as E_m, where $m \in \{1, 2, .., \frac{K\times(K-1)}{2}\}$. Moreover, the edges form independently of each other. Hence, the probability of formation of all M edges for at least one time instant is:

$$\prod_{m=1}^{M} \mathbb{P}(E_m = 1) = (1 - p^N)^M \quad (17)$$

For a sufficiently large instant of observation time, $N \rightarrow \infty$, (17) can be restated as:

$$\lim_{N\rightarrow\infty} \prod_{m=1}^{M} \mathbb{P}(E_m = 1) = \lim_{N\rightarrow\infty} (1 - p^N)^M \quad (18)$$

As (16) reveals that the no edge formation probability p is strictly less than *one*. As a result, the term p^N tends to zero for $N \rightarrow \infty$, and (18) then tends to one, ensuring the presence of the complete graph with probability one, which completes our proof. □

VII. Complexity Analysis

This section (Table III) provides a comparative study of the time complexities of different distributed timing synchronization algorithms. The proposed algorithm starts with 'Block A' in Fig. 15, and the 'Block B' exhibits an iterative execution of the timing update for the achievement of the timing synchronization.

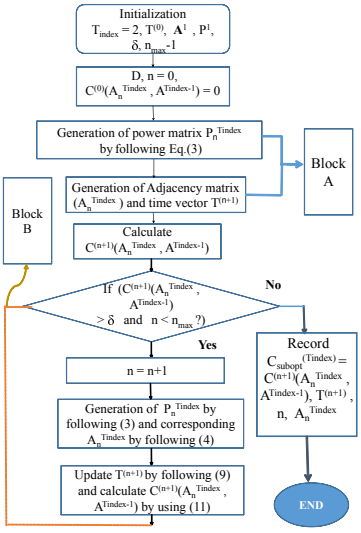

Fig. 15: Flow chart of proposed algorithm for $T_{Index} = 2$.

The worst-case time complexities of 'Block A' and 'Block B' are $\mathcal{O}(K^2)$ and $n_{max}(\delta)\mathcal{O}(K^2)$, respectively, where K is the maximum number of nodes in the network and n_{max} is the maximum number of iterations required for the algorithm to converge, which is a function of the maximum SD, δ. Hence, the sequential execution of the 'Block A', and 'Block B' have the time complexity of $n_{max}(\delta)\mathcal{O}(K^2)$, which is the

TABLE III: Complexity comparison among the schemes for distributed time synchronization.

Distributed Time Synchronization Algorithms	**Complexity**
Optimal solution for Faster Synchronization [16]	$\left(\frac{10}{3}K^3 + \frac{m^3}{3}\right) n_s(\delta)$
Distributed time synchronization [15]	$n_{max}(\delta)\mathcal{O}(K^2)$
Proposed solution for time synchronization	$n_{max}(\delta)\mathcal{O}(K^2)$

overall complexity of the proposed algorithm. In Table III, m is the number of edges, and $n_s(\delta)$ is the maximum number of iterations required to find the optimal weights by using the SDP algorithm.

VIII. CONCLUSION

In this work, we have proposed a *faster* and *low complexity* distributed timing synchronization solution for dense networks. In particular, we suggest a modified time update that exploits an efficient approach for the information exchange from an enlarged neighborhood at a minimal signaling cost. Our proposed work achieves a 98% faster convergence for large networks than the prior work. The numerical investigations exhibit that the higher connectivity and fewer nodes in the network ensure a faster network synchronization. Moreover, the high accuracy obtained in our work ensures that the achieved post-synchronized time is comparable to the mean value of the initial time vector. Unlike the traditional master/slave configuration-based synchronization techniques that use GPS as an external time reference source, our proposed algorithm does not rely on any external timing information. Therefore, it can be used as the backup solution in future networks, especially for scenarios where the synchronizing devices are beyond the coverage of any macro BS or the timing information from other external sources is unavailable. Furthermore, the self-organizing nature and a tighter relative accuracy among the node timings make the solution suitable for device-to-device, unmanned aerial vehicles, and vehicle-to-anything communications.

IX. ACKNOWLEDGEMENT

This work was supported in part by a Research Grant from Science Foundation Ireland and cofunded under the European Regional Development Fund under Grant 13/RC/2077 and Grant 13/RC/2077_P2, in part by CHIST-ERA (call 2017) via the FIREMAN consortium, which is funded by the following national foundations: Academy of Finland under Grant 326270 and Grant 326301, and in part by Irish Research Council, Spanish and Catalan Government under Grant TEC2017-87456-P and Grant 2017-SGR-891, respectively.

REFERENCES

[1] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects, TR 36.872, V12.1.0, 2013/12," Tech. Rep.

[2] R. Behrends, L. K. Dillon, S. D. Fleming, and R. E. K. Stirewalt, "White paper: Timing and Synchronization for LTE-TDD and LTE-Advanced Mobile Networks," Microsemi, USA, Tech. Rep., January 2014.

[3] ITU-T, "Time and phase synchronization aspects of packet networks, G.8271/Y.1366, 2012/02," Tech. Rep.

[4] I. Parvez, A. I. Sarwat, J. Pinto, Z. Parvez, and M. A. Khandaker, "A gossip algorithm based clock synchronization scheme for smart grid applications," in *2017 North American Power Symposium (NAPS)*, Morgantown, July. 2017, pp. 1–6.

[5] K. J. Zou, K. W. Yang, Mao Wang, Bingyin Ren, Jingsong Hu, Jingjing Zhang, Min Hua, and Xaohu You, "Network synchronization for dense small cell networks," *IEEE Wireless Communications*, vol. 22, no. 2, pp. 108–117, 2015.

[6] E. Shereen, F. Bitard, G. Dán, T. Sel, and S. Fries, "Next Steps in Security for Time Synchronization: Experiences from implementing IEEE 1588 v2.1," in *2019 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, Portland, OR, USA, Sept. 2019, pp. 1–6.

[7] Z. Chaloupka, "Technology and Standardization Gaps for High Accuracy Positioning in 5G," *IEEE Communications Standards Magazine*, vol. 1, no. 1, pp. 59–65, 2017.

[8] L. Han, X. Hu, H. Li, L. Wang, and N. Hua, "First demonstration of distributed time synchronization system over transport network towards 5G requirements," in *Optical Fiber Communications Conference and Exhibition (OFC)*, Los Angeles, CA, USA, Mar. 2017, pp. 1–3.

[9] L. Wang, P. D. Groves, and M. K. Ziebart, "Multi-Constellation GNSS Performance Evaluation for Urban Canyons using Large Virtual Reality City Models," *Journal of Navigation*, vol. 65, no. 3, p. 459–476, 2012.

[10] N. Zhu, J. Marais, D. Bétaille, and M. Berbineau, "GNSS Position Integrity in Urban Environments: A Review of Literature," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2762–2778, 2018.

[11] H. Wang, L. Chen, M. Li, and P. Gong, "Consensus-based clock synchronization in wireless sensor networks with truncated exponential delays," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1425–1438, 2020.

[12] E. Garcia, S. Mou, Y. Cao, and D. W. Casbeer, "An event-triggered consensus approach for distributed clock synchronization," in *American Control Conference (ACC)*, Seattle, WA, USA, May 2017, pp. 279–284.

[13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[14] N. Abedini, S. Tavildar, J. Li, and T. Richardson, "Distributed synchronization for device-to-device communications in an LTE network," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1547–1561, 2016.

[15] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, no. 1, p. 057054, June 2007.

[16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[17] K. Arai and M. Murakami, "Trends in Standardization

of High-precision Time- and Frequency-synchronization Technology for Creating a 5G Mobile Network," *NTT Technical Review*, vol. 17, no. 1, 2019.

[18] L. Han and N. Hua, "A distributed time synchronization solution without satellite time reference for mobile communication," *IEEE Communications Letters*, vol. 17, no. 7, pp. 1447–1450, July 2013.

[19] K. Manolakis and W. Xu, "Time synchronization for multi-link D2D and cellular communication," U.S. Patent 10,827,546 B2, Nov. 3, 2020.

[20] H. Zhu, V. Lakamraju, and A. Finn, "Beacon Synchronization in Wi-Fi based Systems," U.S. Patent 9,125,152 B2, Sep. 1, 2015.

[21] M. A. Alvarez and U. Spagnolini, "Collision vs non-collision distributed time synchronization for dense IoT deployments," in *IEEE International Conference on Communications (ICC)*, France, May 2017, pp. 1–6.

[22] D. T. Roche, B. Champagne, I. Psaromiligkos, and B. Pelletier, "On the use of distributed synchronization in 5G device-to-device networks," in *IEEE International Conference on Communications (ICC)*, Paris, France, May 2017, pp. 1–7.

[23] O. Karatalay, I. Psaromiligkos, B. Champagne, and B. Pelletier, "A Distributed Pulse-Based Synchronization Protocol for Half-Duplex D2D Communications," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 245–261, 2021.

[24] M. A. Alvarez, B. Azari, and U. Spagnolini, "Time and frequency self-synchronization in dense cooperative network," in *48th Asilomar Conference on Signals, Systems and Computers*, USA, Nov. 2014, pp. 1811–1815.

[25] L.-A. Phan, T. Kim, and T. Kim, "Robust neighbor-aware time synchronization protocol for wireless sensor network in dynamic and hostile environments," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1934–1945, 2021.

[26] M. Goodarzi, D. Cvetkovski, N. Maletic, J. Teran, and E. Grass, "Synchronization in 5G Networks: a Hybrid Bayesian Approach towards Clock Offset/skew Estimation and Its Impact on Localization," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, 04 2021.

[27] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; LTE; scenarios and requirements for small cell enhancements for E-UTRA and E-UTRAN , TR 136 932 v 13.0.0 (2016-01)," Tech. Rep.

[28] A. De Domenico, E. Strinati, and M. Di Benedetto, "Cognitive Strategies for Green Two-tier Cellular Networks: A critical Overview," *Handbook of Green Information and Communication Systems*, pp. 1–33, 2015.

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

[30] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); TDD Home eNode B (HeNB) Radio Frequency (RF) requirements analysis scenarios and requirements, TR 36.922 v10.0.0 (2011-05)," Tech. Rep.

[31] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

[32] F. Tirado-Andrés and A. Araujo, "Performance of clock sources and their influence on time synchronization in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 15, p. 155014771987937, 09 2019.

[33] D. Yu, Q. Hua, Y. Wang, and F. C. M. Lau, "An o(log n) distributed approximation algorithm for local broadcasting in unstructured wireless networks," in *2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems*, China, May 2012, pp. 132–139.

[34] F. Xiao and L. Wang, "Asynchronous Consensus in Continuous-Time Multi-Agent Systems With Switching Topology and Time-varying Delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804–1816, 2008.

[35] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on architectural enhancements for 5G multicast-broadcast services,TR 23.757 V 15.0.0,2018/07," Tech. Rep.

[36] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Base station (BS) radio transmission and reception, TS 36.104 V15.3.0, 2018/07," Tech. Rep.

[37] A. Garg, A. Yadav, A. Sikora, and A. S. Sairam, "Wireless Precision Time Protocol," *IEEE Communications Letters*, vol. 22, no. 4, pp. 812–815, 2018.

[38] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management, TS 36.133 version 13.3.0 Release 13," Tech. Rep., May 2016.

[39] S.-L. Chao, H.-Y. Lee, C.-C. Chou, and H.-Y. Wei, "Bio-inspired proximity discovery and synchronization for D2D communications," *IEEE Communications Letters*, vol. 17, no. 12, pp. 2300–2303, 2013.

[40] W. Sun, M. R. Gholami, E. G. Ström, and F. Brännström, "Distributed clock synchronization with application of D2D communication without infrastructure," in *2013 IEEE Globecom Workshops (GC Wkshps)*, 2013, pp. 561–566.

[41] C. Chen and P. P. Vaidyanathan, "MIMO radar space–time adaptive processing using prolate spheroidal wave functions," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 623–635, 2008.

[42] T. Wang and I. B. Djordjevic, "Experimentation with prolate spheroidal wave function pulses for physical-layer security," in *14th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, Nis, Serbia, Oct. 2019, pp. 174–177.

[43] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, Feb 2005.