Synchronization in 5G networks: a hybrid Bayesian approach toward clock offset/skew estimation and its impact on localization

Meysam Goodarzi[1,2]*, Darko Cvetkovski[1,2], Nebojsa Maletic[1], Jesús Gutiérrez[1] and Eckhard Grass[1,2]

*Correspondence:
goodarzm@hu-berlin.de
[2] Humboldt University
of Berlin, Berlin, Germany
Full list of author information
is available at the end of the
article

Abstract

Clock synchronization has always been a major challenge when designing wireless networks. This work focuses on tackling the time synchronization problem in 5G networks by adopting a hybrid Bayesian approach for clock offset and skew estimation. Furthermore, we provide an in-depth analysis of the impact of the proposed approach on a synchronization-sensitive service, i.e., localization. Specifically, we expose the substantial benefit of belief propagation (BP) running on factor graphs (FGs) in achieving precise network-wide synchronization. Moreover, we take advantage of Bayesian recursive filtering (BRF) to mitigate the time-stamping error in pairwise synchronization. Finally, we reveal the merit of hybrid synchronization by dividing a large-scale network into local synchronization domains and applying the most suitable synchronization algorithm (BP- or BRF-based) on each domain. The performance of the hybrid approach is then evaluated in terms of the root mean square errors (RMSEs) of the clock offset, clock skew, and the position estimation. According to the simulations, in spite of the simplifications in the hybrid approach, RMSEs of clock offset, clock skew, and position estimation remain below 10 ns, 1 ppm, and 1.5 m, respectively.

Keywords: 5G, Hybrid synchronization, Belief propagation, Bayesian recursive filtering, Joint synchronization and localization

1 Introduction

The Fifth Generation (5G) of mobile networks is expected to deliver a wide range of location-based services [1]. To pave the way for those services, a myriad of precise positioning techniques have been introduced in the literature, the majority of which rely on the cooperation between the Access Points (APs) serving the Mobile Users (MUs) [2]. In particular, to estimate the location, these techniques capitalize on the time measurements carried out between the agents, i.e., MUs and APs, requiring them to have a common time base [3]. Therefore, for the cooperative approaches to function, the APs need to be accurately synchronized among each other as well as with MUs [4, 5].

Considerable effort has been made to design fast, continuous, and precise synchronization algorithms across different networks, from Wireless Sensor Networks (WSNs) to wireless communication networks [6]. Generally, state-of-the-art synchronization

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 2 of 22

algorithms adopt two main macroscopic approaches: (a) designing a *network-wide synchronization* algorithm from scratch [7–10], and (b) employing the existing *pairwise synchronization* protocols in a structural manner, e.g., layer-by-layer pairwise synchronization [11–13].

Network-wide synchronization in WSNs has been addressed in [7, 9, 10] by employing the Belief Propagation (BP) algorithm. Typically, BP runs on a Factor Graph (FG) corresponding to the network and calculates the marginals at each node by iteratively exchanging beliefs between neighboring nodes [14]. The algorithm is advantageous in the sense that it is fully distributed and estimates the clock offset and skew with high accuracy. However, the time required to compute the pairwise conditional *probability distribution functions* (pdfs) needed for FG, and then conducting the iterative message passing, can be considered as a potential drawback rendering its practical applicability limited.

Pairwise synchronization is mostly conducted by exchanging time-stamps between the nodes using the Precision Time Protocol (PTP) [15]. To perform network synchronization in a layer-by-layer manner, PTP is then combined with the Best Master Clock Algorithm (BMCA), whose role is to determine the Master Node (MN) in the network. While this combination operates sufficiently robust in tree-structured networks with medium time-sensitivity (sub-μs range), BMCA's poor performance in networks with mesh topology on one hand, and uncertainty in time-stamping on the other hand, render the algorithm futile in highly time-sensitive (sub-hundred ns range) loopy networks.

Despite the attempts in [11, 16] to address the time-stamping uncertainty (or error) by the virtue of Kalman filtering, this approach is not optimal in the Bayesian sense since all the information available from time-stamps is not utilized. Instead, the Bayesian Recursive Filtering (BRF) utilized in [17] can be employed to capture all the available information in time-stamps, thereby optimally rectifying the time-stamping error. We have already revealed the outstanding merit of BRF in the mitigation of time-stamping error in [18].

Although all aforementioned techniques have made invaluable contribution, none of them alone can be expected to meet the global and local time precision aimed by 5G for accurate localization [19]. Instead, a combination of these algorithms is more likely to deliver a superior performance owing to diverse network typologies (mesh, tree, or a combination thereof) [16]. In particular, to successfully achieve precise network synchronization, it is suggested by [20] that the architecture of a large-scale network should consist of common synchronization areas and multiple synchronization domains. Therefore, equipping networks with different synchronization algorithms (or a combination thereof) appears to be a balanced approach, whereby each domain can, based on its topology and capabilities, leverage the most suitable algorithm. In this manner, it is easier to satisfy the requirement of the relative time error in the synchronization domains while keeping the absolute time error low. This is particularly of interest in applications where ultra-high time accuracy is required in a specific synchronization domain, e.g., positioning services.

In [16], we have introduced and thoroughly described the idea of hybrid synchronization, whereby clock offset can be precisely estimated and correspondingly corrected. The extension to incorporate the clock skew estimation was proposed in [18]. In this paper,

we expand on [16, 18] and design a hybrid synchronization algorithm based on *asymmetric time-stamp exchange* to allow for accurate localization [21]. The merit of asymmetric time-stamp for localization has been revealed in [3, 22, 23]. Furthermore, the *fine time measurement* standard introduced in [24] allows for implementation of such time-stamp exchange mechanism. Given that, in addition to analysis of clock offset and skew estimation, we examine the impact of the proposed hybrid approach on a localization algorithm based on the technique presented in [22].

The contribution of this paper can then be briefly summarized as follows:

- We present the principles of BP-based network-wide and BRF-based pairwise synchronization based on asymmetric time-stamp exchange.
- We develop a hybrid statistical synchronization algorithm by combining the two above-mentioned Bayesian approaches.
- We analyze the performance of the hybrid approach when estimating the clock offset and skew as well as its impact on a localization algorithm.

The rest of this paper is structured as follows: In Sect. 2, the system model is introduced. Section 3 deals with the estimation methods for network-wide, pairwise, and hybrid synchronization. We present and discuss the simulation results in Sect. 4. Section 5 is devoted to the impact of hybrid synchronization on MU localization. Finally, Sect. 6 concludes this work and points to the future work.

1.1 Notation

The boldface capital A and lower case a letters denote matrices and vectors, respectively. $\mathbf{a}(n)$ indicates the nth element of vector $\mathbf{a}$. A^T represents the transposed of matrix A. I_N denotes a N dimensional identity matrix. $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates a random vector $\mathbf{x}$ distributed as Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $\mathrm{diag}(x_1, \ldots, x_K)$

denotes a diagonal matrix with the diagonal elements given by $(x_1, \ldots, x_K)$. The symbol $\sim$ stands for "is distributed as," and the symbol $\propto$ represents the linear scalar relationship between two functions.

2 System model

In this section, we firstly present the clock model for each node in the communication network. Then, we explain the components constructing the clock offset in details. Finally, the time-stamp exchange mechanism is described comprehensively.

2.1 Clock model

The clock behavior for each node i is modeled as [25]

$$c_i(t) = \gamma_i t + \theta_i, \tag{1}$$

where $c_i(t)$ shows the local time at each node, t represents the reference time, γ_i denotes the clock skew, and θ_i is the clock offset. We consider the parameter γ_i as random and varying over time. However, it is common to assume that it remains unchanged in the course of one synchronization period [9, 11]. Moreover, θ_i consists of several components, all thoroughly discussed in the following subsection. In light of above-mentioned
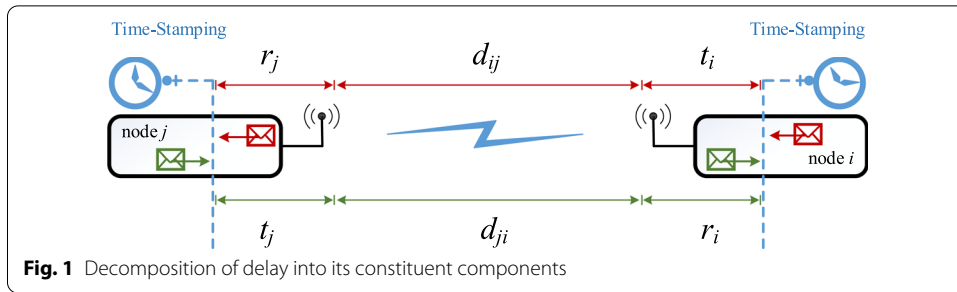
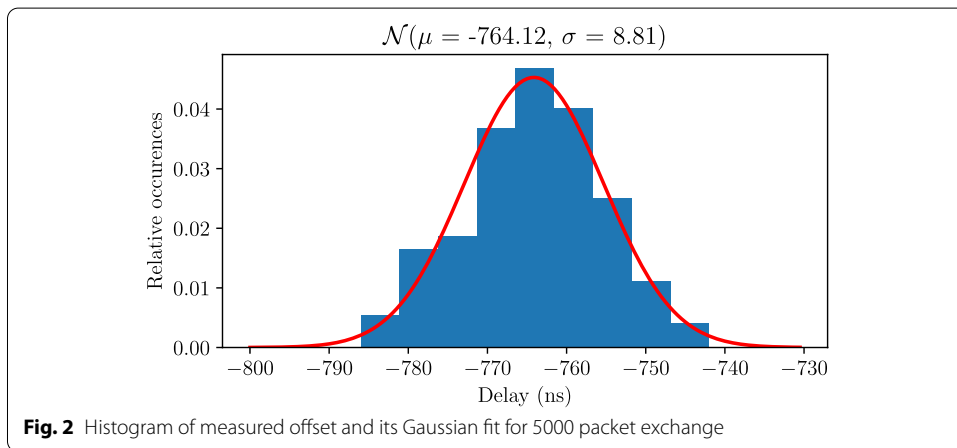**Fig. 1** Decomposition of delay into its constituent components

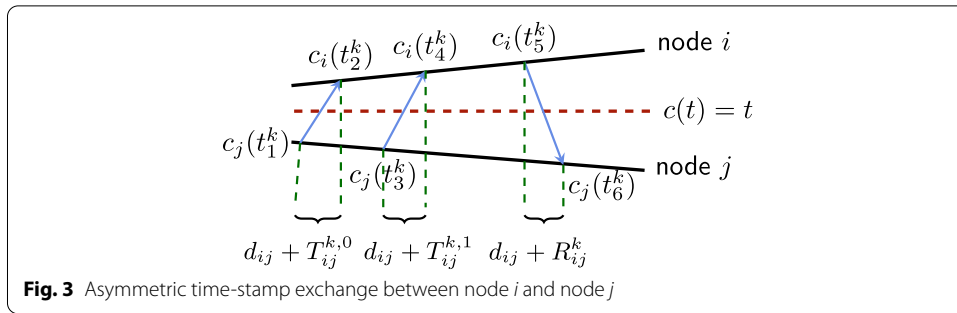**Fig. 2** Histogram of measured offset and its Gaussian fit for 5000 packet exchange

points, time synchronization can be deemed equivalent to estimation of γ_i and θ_i (or transformations thereof) for each node. Corrections are then applied such that, ideally, all the clocks show the same time as the reference time t.

2.2 Clock offset decomposition

We decompose the clock offset θ_i as shown in Fig. 1, thereby elaborating on its constituent components. The parameter t_i/t_j is the time it takes for a packet to leave the transmitter after being time-stamped (the term "time-stamp" is refered to hardware time-stamping hereafter), d_{ij}/d_{ji} denote the propagation delay, and r_i/r_j represents the time that a packet needs to reach the time-stamping point upon arrival at the receiver. Generally,

$$t_i + d_{ij} + r_j \neq t_j + d_{ji} + r_i,$$

meaning that the packets sent from node i to node j do not experience the same delay as the packets sent from node j to node i. In particular $T_{ij} = t_i + r_j$, and $R_{ij} = t_j + r_i$ are random variables due to multiple hardware-related random independent processes and can, therefore, be assumed i.i.d. Gaussian random variables distributed as $\mathcal{N}(\mu_T, \sigma_T^2)$ and $\mathcal{N}(\mu_R, \sigma_R^2)$,, respectively [7, 9, 10]. Conversely, d_{ij} and d_{ji} are usually assumed to be deterministic and symmetric ($d_{ij} = d_{ji}$) [7]. Figure 2 depicts the histogram of the clock

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 5 of 22

Fig. 3 Asymmetric time-stamp exchange between node i and node j

offset and its Gaussian fit for 5000 packet exchange between two Commercial Off-The-Shelf (COTS) street nodes.[1] In particular, the variance of offset turns out to be around 9 ns, what is crucial to know if we are to reduce the error in the clock offset/skew estimation.

2.3 Time-stamp exchange mechanism

We employ the asymmetric time-stamping mechanism introduced in [21] and shown in Fig. 3. It functions as follows: node j transmits a *sync* message wherein the local time $c_j(t_1^k)$ is incorporated. Node i receives the packet and records the local reception time $c_i(t_2^k)$. After a certain time, the process repeats again with $c_j(t_3^k)$ and $c_i(t_4^k)$. Subsequently, at local time $c_i(t_5^k)$, node i sends back a *sync* message to node j with $c_i(t_2^k)$, $c_i(t_4^k)$ and $c_i(t_5^k)$ incorporated. Upon reception, node j records the local time $c_j(t_6^k)$. Given that, the relation between local clocks can be written as:

$$\frac{1}{\gamma_i}(c_i(t_2^k) - \theta_i) = \frac{1}{\gamma_j}(c_j(t_1^k) - \theta_j) + d_{ij} + T_{ij}^{k,0}, \tag{2}$$

$$\frac{1}{\gamma_i}(c_i(t_4^k) - \theta_i) = \frac{1}{\gamma_j}(c_j(t_3^k) - \theta_j) + d_{ij} + T_{ij}^{k,1}, \tag{3}$$

$$\frac{1}{\gamma_i}(c_i(t_5^k) - \theta_i) = \frac{1}{\gamma_j}(c_j(t_6^k) - \theta_j) - d_{ij} - R_{ij}^k, \tag{4}$$

where $(t_1^k, t_3^k)/t_6^k$ and $t_5^k/(t_2^k, t_4^k)$ are the time points where neighboring nodes j and i send/receive the *sync* messages, respectively. Stacking the weighted sum of (2), (3) and (4) for K rounds of time-stamp exchange gives

$$\mathbf{W}_{ji}\boldsymbol{\xi}_i + \mathbf{W}_{ij}\boldsymbol{\xi}_j = \mathbf{z}_{ij}, \tag{5}$$

where $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$ are $K \times 2$ matrices with the kth row being

---

[1] The uncertainty in time-stamping is due to the precision of the devices as well as the manner of hardware time-stamping implementation. For the nodes in this experiment, the precision of time-stamping was 8 ns meaning that the time-stamps were always an integer of 8 ns.

$$\left[\frac{1}{2}\left(c_i(t_2^k) + c_i(t_4^k)\right) + c_i(t_3^k), -2\right],$$

and

$$-\left[\frac{1}{2}\left(c_i(t_3^k) + c_i(t_4^k)\right) + c_j(t_4^k), -2\right],$$

, respectively. Moreover, we introduce the vector variables $\boldsymbol{\xi}_i \triangleq \left[\frac{1}{\gamma_i}, \frac{\theta_i}{\gamma_i}\right]^T$, and $\boldsymbol{\xi}_j \triangleq \left[\frac{1}{\gamma_j}, \frac{\theta_j}{\gamma_j}\right]^T$ with $\frac{1}{\gamma_i}$, $\frac{\theta_i}{\gamma_i}$, $\frac{1}{\gamma_j}$, and $\frac{\theta_j}{\gamma_j}$ being Gaussian distributed [3, 10]. Finally, $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \sigma_{ij}^2 \mathbf{I}_K)$, where $\sigma_{ij}^2 = \frac{\sigma_{T_{ij}}^2}{2} + \sigma_{R_{ij}}^2$. In concrete terms, what (5) implicitly states is that for given $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$, the probability that we measure $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$ is equal to $\mathcal{N}(\mathbf{z} = \mathbf{W}_{ji}\boldsymbol{\xi}_i + \mathbf{W}_{ij}\boldsymbol{\xi}_j|\mathbf{0}, \sigma_{ij}^2 \mathbf{I}_N)$. This can be expressed as

$$p(\mathbf{W}_{ji}, \mathbf{W}_{ij}|\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \sim \mathcal{N}(\mathbf{z} = \mathbf{W}_{ji}\boldsymbol{\xi}_i + \mathbf{W}_{ij}\boldsymbol{\xi}_j|\mathbf{0}, \sigma_{ij}^2 \mathbf{I}_N). \tag{6}$$

3 Methods of clock offset and skew estimation

In this section, first the principles of BP-based network-wide synchronization are described. Subsequently, we introduce the BRF-based pairwise synchronization. Lastly, we present an approach, where both techniques are employed in a hybrid manner.

3.1 Network-wide offset and skew estimation

In network-wide synchronization, we aim to synchronize each node with a global MN. Alternatively, we can restate the problem as estimation of parameters γ_i and θ_i (or vector parameter $\boldsymbol{\xi}_i$), based on the observation matrices $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$. Mathematically, this is translated to the following marginal calculation:

$$\begin{aligned}
&p(\boldsymbol{\xi}_i|\{\mathbf{W}_{ji}, \mathbf{W}_{ij}\}_{i=1:M, j \in \mathcal{I}_i}) \\
&= \int \cdots \int p(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M|\{\mathbf{W}_{ji}, \mathbf{W}_{ij}\}_{i=1:M, j \in \mathcal{I}_i}) \\
&\quad d\boldsymbol{\xi}_1 \cdots d\boldsymbol{\xi}_{i-1} d\boldsymbol{\xi}_{i+1} \cdots d\boldsymbol{\xi}_M,
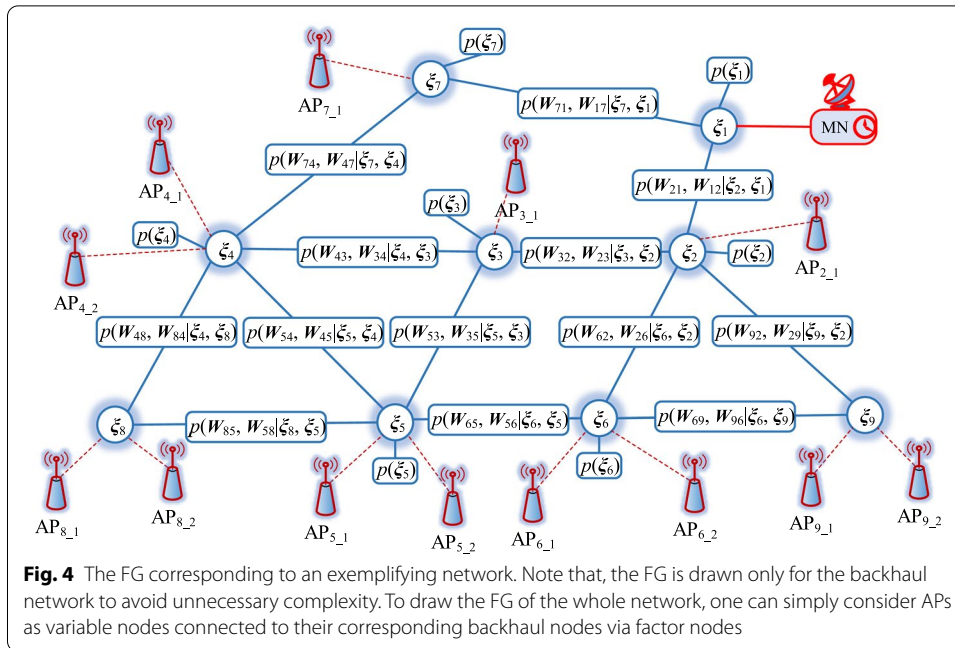\end{aligned} \tag{7}$$

where $\mathcal{I}_i$ denotes the set of neighboring nodes of node i and M is total number of the nodes in the network. Consequently, $\boldsymbol{\xi}_i$ can be estimated as

$$\hat{\boldsymbol{\xi}}_i = \arg\max_{\boldsymbol{\xi}_i} p(\boldsymbol{\xi}_i|\{\mathbf{W}_{ji}, \mathbf{W}_{ij}\}_{i=1:M, j \in \mathcal{I}_i}). \tag{8}$$

Unfortunately, the computation cost and complexity of the marginal pdf in (7) are extremely high. Instead, as a compromise, one can resort to approximating the integrand of (7). This is carried out in the sequel with the aid of *variational methods.*

3.1.1 Variational methods

The basic idea underpinning variational methods is to approximate an intractable complex distribution $p(\mathbf{x})$ by a straightforward tractable distribution $q(\mathbf{x})$. To this end, one can minimize the discrepancy measure Kullback–Leibler (KL) divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$, given by [14]

Fig. 4 The FG corresponding to an exemplifying network. Note that, the FG is drawn only for the backhaul network to avoid unnecessary complexity. To draw the FG of the whole network, one can simply consider APs as variable nodes connected to their corresponding backhaul nodes via factor nodes

$$D_{KL}(p\|q) = \int_{-\infty}^{+\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x}. \tag{9}$$

The minimization is then achieved by drawing on the *Bethe method*, which imposes the following structure on $q(\mathbf{x})$ [26]:

$$q(\mathbf{x}) \propto \prod_i q(x_i) \prod_{i,j} q(x_i, x_j), \tag{10}$$

where x_j and x_i are neighboring nodes. The structure in (10) can be appropriately represented by FG. Furthermore, to efficiently infer the marginal beliefs, BP is typically run on the FG [14]. Therefore, in the sequel, we briefly describe FG and BP.

3.1.2 Factor graph

An FG is a bipartite graph that depicts a pdf with the factorized form, e.g., that of (10). In particular, an FG comprises several nodes known as variable nodes, and a number of factor nodes, each being a function of its neighboring variable nodes (Fig. 4).

We construct the graphical model in Fig. 4, where a number of APs are backhauled by a mesh network, each represented by $\boldsymbol{\xi}_i$. The main objective is then to compute the marginal illustrated in (7). Adopting the approximation outlined in Sect. 3.1.1, the conditional probability under the integral of (7) turns into

$$p(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M | \{\mathbf{W}_{ji}, \mathbf{W}_{ij}\}_{i=1:M, j \in ne(i)}) \propto$$
$$\prod p(\boldsymbol{\xi}_i) \prod p(\mathbf{W}_{ij}, \mathbf{W}_{ji} | \boldsymbol{\xi}_i, \boldsymbol{\xi}_j), \tag{11}$$

where $p(\boldsymbol{\xi}_i)$ indicates the Gaussian distributed prior knowledge on $\boldsymbol{\xi}_i$ and $p(\mathbf{W}_{ji}, \mathbf{W}_{ij} | \boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$ is the pairwise conditional probability computed from (6). In the

Fig. 5 Message passing principles in BP
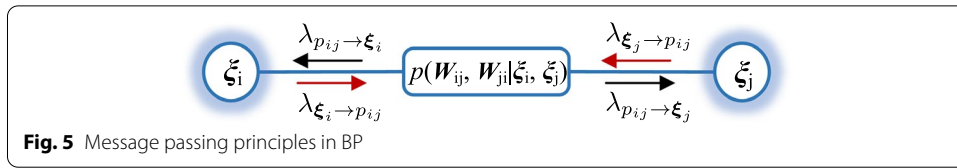
sequel, we briefly describe the principles of BP as an efficient algorithm to obtain the estimation in (8).

3.1.3 Belief propagation

BP is a technique which relies primarily on the exchange of beliefs between neighboring nodes to infer the marginals. This inference is proved to be exact when the graphs are singly connected and approximate if they contain loops [14]. While generally there is no guarantee that the algorithm converges in the loopy graphs, [9, 10] have indicated that, if there exist at least one MN in the network, the convergence of BP is certain. Figure 5 depicts the principles of the message passing in BP for the nodes $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$. For the sake of simplicity, we denote the factor $p(\mathbf{W}_{ji}, \mathbf{W}_{ij}|\xi_i, \xi_j)$ with p_{ij}. The message from a factor vertex p_{ij} to a variable vertex $\boldsymbol{\xi}_i$ in iteration l is then given by [14]

$$
\begin{aligned}
&\lambda^{(l)}_{p_{ij}\to\boldsymbol{\xi}_i}(\boldsymbol{\xi}_i) \\
&= \int p(\mathbf{W}_{ji}, \mathbf{W}_{ij}|\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)\lambda^{(l)}_{\boldsymbol{\xi}_j\to p_{ij}}(\boldsymbol{\xi}_j)\mathrm{d}\boldsymbol{\xi}_j,
\end{aligned}
\tag{12}
$$

where $\lambda^{(l)}_{\boldsymbol{\xi}_j\to p_{ij}}(\boldsymbol{\xi}_j)$ denotes the message from a variable node $\boldsymbol{\xi}_j$ to the variable vertex p_{ij} and is given by

$$
\lambda^{(l)}_{\boldsymbol{\xi}_j\to p_{ij}}(\boldsymbol{\xi}_j) = p(\boldsymbol{\xi}_j) \prod_{k\in\{\mathcal{I}(j)\setminus i\}} \lambda^{(l-1)}_{p_{kj}\to\boldsymbol{\xi}_j}(\boldsymbol{\xi}_j).
\tag{13}
$$

Finally,

$$
b^{(l)}(\boldsymbol{\xi}_i) \propto p(\boldsymbol{\xi}_i) \prod_{k\in\mathcal{I}i} \lambda^{(l)}_{p_{ik}\to\boldsymbol{\xi}_i}(\boldsymbol{\xi}_i),
\tag{14}
$$

where $b^{(l)}(\boldsymbol{\xi}_i)$ denotes the marginal belief of variable node $\boldsymbol{\xi}_i$ in the lth iteration. It is expected that the result of the integral in (12) is Gaussian distributed as its arguments are also Gaussian distributed. We note that, in practice, both (12) and (13) are locally computed at each node and only $\lambda^{(l)}_{p_{ij}\to\boldsymbol{\xi}_i}(\boldsymbol{\xi}_i)$ is transmitted from node j to node i as shown in Fig. 6.

Let $\lambda^{(l)}_{j\to i}(\boldsymbol{\xi}_i) \sim \mathcal{N}(\boldsymbol{\xi}_i|\boldsymbol{\mu}^{(l)}_{j\to i}, \boldsymbol{Q}^{(l)}_{j\to i})$ denote the message sent from j to i. Considering (12) and (13), the covariance matrix $\boldsymbol{Q}^{(l)}_{j\to i}$ can be calculated by [10, 18, 27]

$$
\boldsymbol{Q}^{(l)}_{j\to i} = \left[\mathbf{W}^T_{ji}\left(\boldsymbol{\Omega}^{(l-1)}_{j\to i}\right)^{-1}\mathbf{W}_{ji} \right]^{-1},
\tag{15}
$$

where

Goodarzi *et al. J Wireless Com Network* (2021) 2021:91

Page 9 of 22

Fig. 6 $\lambda_{i \to j} = \lambda_{p_{ij} \to \xi_j}$ and $\lambda_{j \to i} = \lambda_{p_{ij} \to \xi_i}$ are the BP messages exchanged between physical nodes in practice
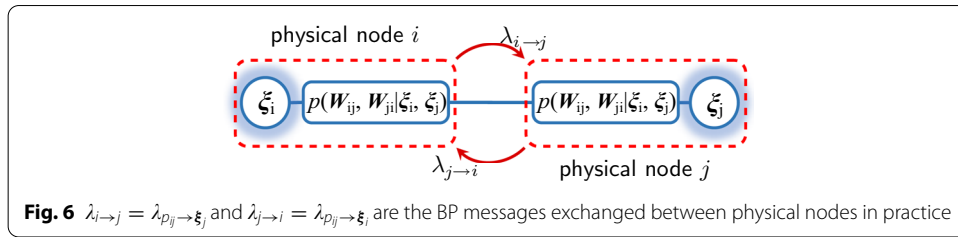
$$\begin{aligned}
\boldsymbol{\Omega}_{j \to i}^{(l-1)} \\
= \sigma_{ij}^2 \mathbf{I}_N + \mathbf{W}_{ij} \left[\boldsymbol{Q}_j^{-1} + \sum_{k \in ne(j) \setminus i} \left(\boldsymbol{Q}_{k \to j}^{(l-1)} \right)^{-1} \right]^{-1} \mathbf{W}_{ij}^T,
\end{aligned} \tag{16}$$

and $\boldsymbol{Q}_j$ is the covariance matrix of $p(\boldsymbol{\xi}_j)$. Furthermore,

$$\begin{aligned}
\boldsymbol{\mu}_{j \to i}^{(l)} = -\boldsymbol{Q}_{j \to i}^{(l)} \mathbf{W}_{ji}^T \boldsymbol{\Omega}_{j \to i}^{(l-1)} \mathbf{W}_{ij} \\
\times \left[\boldsymbol{Q}_j^{-1} + \sum_{k \in ne(j) \setminus i} \left(\boldsymbol{Q}_{k \to j}^{(l-1)} \right)^{-1} \right]^{-1} \\
\times \left[\boldsymbol{Q}_j^{-1} \boldsymbol{\mu}_j + \sum_{k \in ne(j) \setminus i} \left(\boldsymbol{Q}_{k \to j}^{(l-1)} \right)^{-1} \boldsymbol{\mu}_{k \to j}^{(l-1)} \right],
\end{aligned} \tag{17}$$

where $\boldsymbol{\mu}_j$ represents the mean vector of $p(\boldsymbol{\xi}_j)$. It should be noted that $\boldsymbol{Q}_j$ and $\boldsymbol{\mu}_j$ remain unchanged during the message updating process.

The BP algorithm initializes the message from node j to node i as $\lambda_{j \to i}^{(0)}(\boldsymbol{\xi}_i) \sim \mathcal{N}(\boldsymbol{\xi}_i | \mathbf{0}, +\infty \mathbf{I}_2)$. Node j computes its outgoing message to node i according to (15) and (17) in iteration l with its available $\boldsymbol{Q}_{k \to j}^{(l-1)}$ and $\boldsymbol{\mu}_{k \to j}^{(l-1)}$ ($k \in ne(j) \setminus i$). The belief of node i is then computed as

$$b^{(l)}(\boldsymbol{\xi}_i) \sim \mathcal{N}(\boldsymbol{\xi}_i | \boldsymbol{\eta}_i^{(l)}, \boldsymbol{\Gamma}_i^{(l)}), \tag{18}$$

where

$$\boldsymbol{\Gamma}_i^{(l)} = \left[\boldsymbol{Q}_i^{-1} + \sum_{j \in ne(i)} \left(\boldsymbol{Q}_{j \to i}^{(l-1)} \right)^{-1} \right]^{-1}, \tag{19}$$

and

$$\boldsymbol{\eta}_i^{(l)} = \boldsymbol{\Gamma}_i^{(l)} \left[\boldsymbol{Q}_i^{-1} \boldsymbol{\mu}_i + \sum_{j \in ne(i)} \left(\boldsymbol{Q}_{j \to i}^{(l-1)} \right)^{-1} \boldsymbol{\mu}_{j \to i}^{(l-1)} \right]. \tag{20}$$

Finally, the clock skew and offset estimation can be computed by

Fig. 7 Bayesian representation of offset and skew estimation
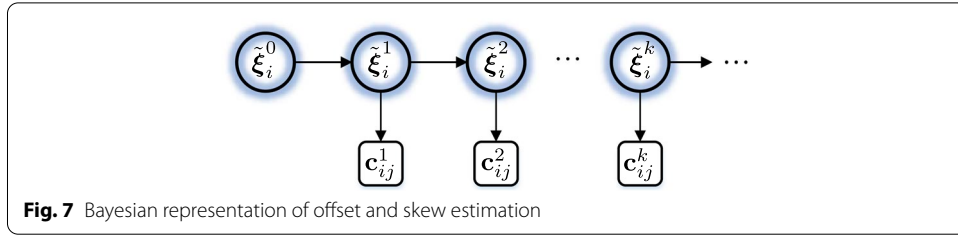
$$\hat{\gamma}_i^{(l)} = \frac{1}{\boldsymbol{\eta}_i^{(l)}(1)}, \hat{\theta}_i^{(l)} = \frac{\boldsymbol{\eta}_i^{(l)}(2)}{\boldsymbol{\eta}_i^{(l)}(1)}. \tag{21}$$

3.2 Pairwise offset and skew estimation

In pairwise synchronization, one node is assumed to be the MN. In particular, in Fig. 3, instead of a global reference $c(t) = t$, we take node j as MN. We can then introduce the transformations

$$\frac{1}{\tilde{\gamma}_i} = \frac{\gamma_j}{\gamma_i}, \tag{22}$$

$$\tilde{\theta}_i = \theta_i - \tilde{\gamma}_i \theta_j, \tag{23}$$

$$\tilde{d}_{ij} + \tilde{T}_{ij}^k = \gamma_j (d_{ij} + T_{ij}^k), \tag{24}$$

$$\tilde{d}_{ij} - \tilde{R}_{ij}^k = \gamma_j (d_{ij} - R_{ij}^k). \tag{25}$$

For the sake of simplicity, as done in [28], we assume $\tilde{d}_{ij} = d_{ij}$, $\tilde{R}_{ij}^k = R_{ij}^k$, and $\tilde{T}_{ij}^k = T_{ij}^k$. This is valid owing to $\gamma_j \approx 1$ and the value of $d_{ij} + T_{ij}^k$ and $d_{ij} - R_{ij}^k$ being low. Finally, (2), (3) and (4) turn into

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_2^k) - \tilde{\theta}_i) = c_j(t_1^k) + d_{ij} + T_{ij}^{k,0}, \tag{26}$$

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k) - \tilde{\theta}_i) = c_j(t_3^k) + d_{ij} + T_{ij}^{k,1}, \tag{27}$$

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_5^k) - \tilde{\theta}_i) = c_j(t_6^k) - d_{ij} - R_{ij}^k. \tag{28}$$

By the end of the kth round of time-stamp exchange, each node is expected to have collected the time-stamps $\mathbf{C}_{ij} = \left[\mathbf{c}_{ij}^1, \ldots, \mathbf{c}_{ij}^k\right]^T$, where

$$\mathbf{c}_{ij}^k = \left[c_j(t_1^k), c_i(t_2^k), c_j(t_3^k), c_i(t_4^k), c_i(t_5^k), c_j(t_6^k)\right].$$

Let $\tilde{\boldsymbol{\xi}}_i^k$ be the state of the vector variable $\tilde{\boldsymbol{\xi}}_i \triangleq \left[\frac{1}{\tilde{\gamma}_i}, \frac{\tilde{\theta}_i}{\tilde{\gamma}_i}\right]^T$ after kth round of time-stamp exchange (visualized in Fig. 7). Similar to (7) the pdf corresponding to the kth state can be written as

$$p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{C}_{ij}) = \int p(\tilde{\boldsymbol{\xi}}_i^0, \dots, \tilde{\boldsymbol{\xi}}_i^k | \mathbf{C}_{ij}) \, d\Theta^{k-1}, \tag{29}$$

where $\Theta^{k-1} = \left[\tilde{\boldsymbol{\xi}}_i^0, \dots, \tilde{\boldsymbol{\xi}}_i^{k-1}\right]$. Following the steps explained in "Appendix", (29) can be simplified to

$$p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{C}_{ij}) \propto p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{c}_{ij}^{1:k-1}) p(\mathbf{c}_{ij}^k | \tilde{\boldsymbol{\xi}}_i^k) \sim \mathcal{N}(\boldsymbol{\mu}_i^k, \mathbf{Q}_i^k). \tag{30}$$

The term $p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{c}_{ij}^{1:k-1})$ is known as *prediction* step, while the term $p(\mathbf{c}_{ij}^k | \tilde{\boldsymbol{\xi}}_i^k)$ is referred to as *measurement update* or *correction* step [29]. Considering the clock properties discussed in Sect. 2.1, it is typical in wireless networks to assume that $\tilde{\boldsymbol{\xi}}_i^k$ is Gaussian distributed [3, 9, 28]. Given this assumption, in the sequel, we show that the relation between the states is linear, implying that the marginal in (30) is also Gaussian distributed.

3.2.1 Prediction

Assuming constant skew in one synchronization period ($= K$ rounds of time-stamp exchange), a reasonable prediction for $\tilde{\boldsymbol{\xi}}_i^k$ is given by [11]

$$\tilde{\boldsymbol{\xi}}_i^k = \mathbf{A}\tilde{\boldsymbol{\xi}}_i^{k-1} + \mathbf{n}_i^{k-1}, \tag{31}$$

where $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ c_j(t_1^k) - c_j(t_1^{k-1}) & 1 \end{bmatrix}$, and $\mathbf{n}_i^{k-1}$ denotes the Gaussian noise vector. Given (31), the prediction term can be written as

$$p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{c}_{ij}^{1:k-1}) \sim \mathcal{N}(\tilde{\boldsymbol{\xi}}_i^k | \boldsymbol{\mu}_{\mathrm{p}}, \boldsymbol{\Sigma}_{\mathrm{p}}), \tag{32}$$

where $\boldsymbol{\mu}_{\mathrm{p}} = \mathbf{A}\boldsymbol{\mu}_i^{k-1}$ and $\boldsymbol{\Sigma}_{\mathrm{p}} = \mathbf{A}\boldsymbol{\Sigma}_i^{k-1}\mathbf{A}^T + \mathbf{Q}_n$ where $\mathbf{Q}_n$ denotes the noise covariance matrix.

3.2.2 Correction

To obtain the correction term in (30), we conduct the following mathematical manipulations. Subtracting (26) from (27) leads to

$$\begin{aligned} \frac{1}{\tilde{\gamma}_i}(c_i(t_4^k) - c_i(t_2^k)) \\ = c_j(t_3^k) - c_j(t_1^k) + T_{ij}^{k,1} - T_{ij}^{k,0}, \end{aligned} \tag{33}$$

while weighted sum of (26)–(28) gives

$$\begin{aligned} \frac{1}{\tilde{\gamma}_i}\left(\frac{c_i(t_2^k) + c_i(t_4^k)}{2} + c_i(t_5^k) - 2\tilde{\theta}_i\right) \\ = \frac{c_j(t_1^k) + c_j(t_3^k)}{2} + c_j(t_6^k) + \frac{T_{ij}^{k,0} + T_{ij}^{k,1}}{2} - R_{ij}^k, \end{aligned} \tag{34}$$
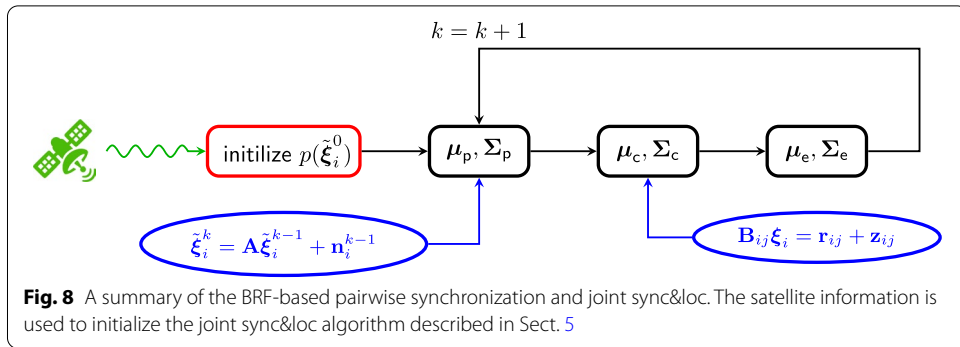
Goodarzi *et al. J Wireless Com Network* (2021) 2021:91

Page 12 of 22

Fig. 8 A summary of the BRF-based pairwise synchronization and joint sync&loc. The satellite information is used to initialize the joint sync&loc algorithm described in Sect. 5

where, given the assumptions in Sect. 2.2, $\frac{T_{ij}^{k,0}+T_{ij}^{k,1}}{2} - R_{ij}^k$ and $T_{ij}^k - T_{ij}^{k-1}$ are zero mean and have the variances $\frac{\sigma_{T_{ij}}^2}{2} + \sigma_{R_{ij}}^2$ and $2\sigma_{T_{ij}}^2$, respectively. This is straightforward to observe since they are linear subtraction of independent random processes. Alternatively, we can write (33) and (34) in matrix form as

$$\mathbf{B}_{ij}\boldsymbol{\xi}_i = \mathbf{r}_{ij} + \mathbf{z}_{ij}, \tag{35}$$

where $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{R}_{ij})$ with

$$\mathbf{R}_{ij} = \text{diag}\left([2\sigma_{T_{ij}}^2, \frac{\sigma_{T_{ij}}^2}{2} + \sigma_{R_{ij}}^2] \right),$$

$$\mathbf{B}_{ij} = \begin{bmatrix} c_i(t_4^k) - c_i(t_2^k) & 0 \\ \frac{c_i(t_2^k)+c_i(t_4^k)}{2} + c_i(t_5^k) & -2 \end{bmatrix},$$

and $\mathbf{r}_{ij} = \left[c_j(t_3^k) - c_j(t_1^k), \frac{c_j(t_1^k)+c_j(t_3^k)}{2} + c_j(t_6^k) \right]^T$.

Consequently,

$$p(\mathbf{c}_{ij}^k|\boldsymbol{\xi}_i^k) \sim \mathcal{N}(\tilde{\boldsymbol{\xi}}_i^k|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), \tag{36}$$

where $\boldsymbol{\mu}_c = \mathbf{B}_{ij}^{-1}\mathbf{r}_{ij}$ and $\boldsymbol{\Sigma}_c = \mathbf{B}_{ij}^{-1}\mathbf{R}_{ij}\mathbf{B}_{ij}^{-T}$.

3.2.3 Estimation

Considering (32) and (36), the estimated distribution in (30) is given by

$$p(\tilde{\boldsymbol{\xi}}_i^k|\mathbf{C}_{ij}) \sim \mathcal{N}(\tilde{\boldsymbol{\xi}}_i^k|\boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e), \tag{37}$$

where

$$\boldsymbol{\mu}_e = \left[\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_c\right]^{-1}\left(\boldsymbol{\Sigma}_c\boldsymbol{\mu}_p + \boldsymbol{\Sigma}_p\boldsymbol{\mu}_c\right), \tag{38}$$

$$\boldsymbol{\Sigma}_e = \left[\boldsymbol{\Sigma}_p^{-1} + \boldsymbol{\Sigma}_c^{-1}\right]^{-1}. \tag{39}$$

The parameters in (32), (36), and (37) are calculated recursively and, in each iteration *k*, the estimation of the clock skew and offset can be obtained by

$$\tilde{\gamma}_i^k = \frac{1}{\boldsymbol{\mu}_{\mathrm{e}}(1)} \text{ and } \tilde{\theta}_i^k = \frac{\boldsymbol{\mu}_{\mathrm{e}}(2)}{\boldsymbol{\mu}_{\mathrm{e}}(1)}. \tag{40}$$

Figure 8 summarizes this recursive process.

3.3 Hybrid synchronization

Given Sects. 3.1 and 3.2, to ensure a low end-to-end synchronization error at the global level, BP can be run over the backhaul network. At the same time, we can employ the BRF algorithm to perform synchronization between the backhaul nodes and the APs at the edge of the network where fast and frequent synchronization is required to keep the relative time error small. This is, in particular, crucial to a number of applications such as localization as will be discussed in Sect. 5.

Algorithm 1 Hybrid synchronization algorithm

1: Determine the suitable algorithm for each network section (BP-nodes and BRF-nodes).
2: Commence time-stamp exchange and the BRF algorithm at BRF-nodes.
3: Start the time-stamp exchange between adjacent BP-nodes and, subsequently, construct $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$ for each pair.
4: **for** $l = 1, 2, \cdots, L$ **do**
5:     Calculate the messages using (15) and (17) for each BP-node and send them to its neighboring nodes.
6:     Update the beliefs at each BP-node by means of (18)
7:     **if** $\hat{\boldsymbol{\xi}}_i^{(l)} - \hat{\boldsymbol{\xi}}_i^{(l-1)} \leq \epsilon \; \forall i$ **then**
8:         Compute the offset and skew estimation using (21).
9:         Go to step 3.
10:     **end if**
11: **end for**

The steps of the hybrid synchronization are described in algorithm 1. Firstly, step 1 determines the network sections suitable for BP and BRF (they are labeled as BP-nodes and BRF-nodes, respectively). Then, step 2 initiates the time-stamp exchange mechanism (Fig. 3) and, correspondingly, the BRF algorithm at BRF-nodes. Step 3 triggers the time-stamp exchange among the BP-nodes, thereby collecting the required time-stamps to construct the matrices $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$. Step 4 is where the BP iterations commence and continue until it converges, or the maximum number of iterations L is achieved. In step 5, the outgoing messages are computed by each BP-node using (15) and (17). They are then sent to their corresponding nodes. Step 6 updates each node's belief. Lastly, in steps 7-10, we check for the convergence by comparing the difference between clock offset and skew estimation in iterations (l) and $(l-1)$ with a predefined small value ϵ. If the algorithm is converged, the clock offset and skew estimation are calculated by means of (18) and (21), respectively. Note that step 2 and steps 3-11 can run simultaneously.

3.3.1 Convergence analysis

Convergence of hybrid synchronization algorithm depends on the behavior of BRF, and BP. In particular, at the edge of the network where we aim to locally synchronize the APs using BRF the convergence is of no meaning. Nevertheless, as a measure to evaluate the estimator's performance, given the set of linear equations presented in Sect. 3.2, we can refer to BRF with Gaussian parameters as minimum variance unbiased estimator [30].

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 14 of 22

Table 1 Simulation parameters

Parameters	Values in synchronization (Sect. 4)	Values in joint sync&loc (Sect. 5)
Number of independent simulations	10000	1000
Master node	$\boldsymbol{\xi}_1$	$\boldsymbol{\xi}_1$
Initial random delays interval	[− 1000, 1000] ns	[− 1000, 1000] ns
Number of time-stamp exchange (*K*)	10	—
Standard deviation of T_{ij}^k and R_{ij}^k	9 ns	[2, 4, 6, 8, 10] ns
Random propagation delay between each pair of nodes	[200, 300] ns	[200, 300] ns
Initial pdf of the offset/skew for each node	$\mathcal{N}(0, +\infty)/\mathcal{N}(1, 10^{-4})$	—
Initial pdf of the offset/skew of MN	$\mathcal{N}(0, 0)/\mathcal{N}(1, 0)$	—
STD of acceleration noise $(\sigma_{a_x}, \sigma_{a_y})$	—	2.5 m/s^2
Period of joint sync&loc (Δ)	—	200 ms
Process noise covariance matrix ($\mathbf{Q}_n$)	diag($10^{-12}, 10^{-2}$)	diag($10^{-12}, 10^{-2}, (0.5\sigma_{a_x}\Delta)^2,$ $(0.5\sigma_{a_y}\Delta)^2, \sigma_{a_x}^2, \sigma_{a_y}^2$)
Constant velocity	—	2 (m/s)
AN density	—	50 m

Given that, convergence of Algorithm 1 depends solely on the convergence of BP which is of crucial importance for global synchronization.

While it is known that BP converges to the exact marginal on loop-free FGs, its convergence on loopy FGs is highly conditional. In the context of clock synchronization, detailed convergence analysis of loopy BP has been conducted in [7, 9, 10, 31]. For the set of message passing formulas presented in this paper, we can leverage on [10, Lemma 1] and [10, Lemma 2] to prove that the mean vector $\boldsymbol{\eta}_i^{(l)}$ in (20) and the covariance matrix $\boldsymbol{\Gamma}_i^{(l)}$ in (19) of the belief $b^{(l)}(\boldsymbol{\xi}_i)$ in (18) converge to constant vector/matrix regardless of the network topology [10, Theorem 1], [10, Theorem 2]. The crucial point of this proof is that, regardless of the network topology, the belief parameters (mean vector and covariance matrix) converge as long as there is an informative prior, i.e., there exist at least one MN in the network.

4 Simulation results and discussion

In this section, we evaluate the performance of the hybrid synchronization algorithm proposed in this work. Detailed analysis of its impact on the achievable performance of the joint sync&loc algorithm at the edge of the network is left to the next section.

4.1 Network synchronization

Figure 4 exemplifies a wireless network where the algorithm proposed in this work can be applied. It comprises a number of APs, all backhauled by a wireless mesh network and delivering services to MUs. The following scenarios are simulated: a) synchronizing the whole network using only BP (the APs in Fig. 4 are assumed to be variable nodes connected to the mesh network via factor nodes), b) performing hybrid synchronization as described in Algorithm 1, where we synchronize the mesh backhaul network by means of BP and the APs at the edge of the network using BRF, and c) carrying out synchronization across the rounds of time-stamp exchange *K*. Scenario (a) is considered as the

Fig. 9 BP applied on the whole network

Fig. 10 BP and BRF applied to the network in a hybrid manner
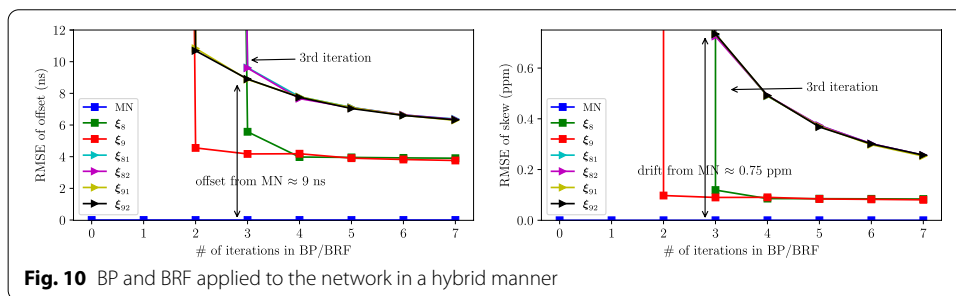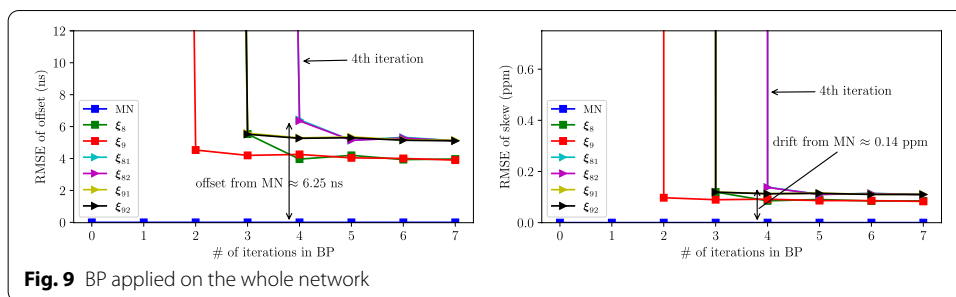
baseline for comparison with the hybrid approach. Furthermore, we compute the Root Mean Square Error (RMSE) of clock offset and skew estimation as a measure to evaluate the performance. For the sake of simplicity and without loss of generality, in (a), (b), and (c) we consider only the nodes ξ_8 and ξ_9 and their corresponding APs. Further simulation parameters can be found in Table 1.

Figure 9 shows the RMSEs of the clock offset and skew estimation versus the number of message passing iterations for scenario (a). The RMSEs of offset and skew are indicated in *nanosecond* (ns) and *parts per million* (ppm), respectively. As can be observed, BP converges after four iterations and achieves an offset and skew RMSE below 7 ns and 0.2 ppm, respectively. As shown in [7, 14], when there exist at least one MN in the network, the convergence is guaranteed. However, the value to which BP converges in loopy networks is deemed to be approximate. Note that, although this simulation setup reveals the potential performance of BP, the nodes, and particularly the APs, must wait at least four message passing iterations in addition to *K* rounds of time-stamp exchange (required for obtaining the conditional probabilities) to be fully synchronized. This is particularly unfavorable in certain synchronization-based services such as localization, where continuous time alignment is essential for accurate estimation of the MUs' positions. Therefore, it is necessary that the APs synchronize themselves to the backhaul network more frequently to be able to deliver those services at an increased performance as required in 5G networks.

Figure 10 depicts the RMSEs of the clock offset and skew estimation versus the number of message passing and BRF iterations for scenario (b). We can observe a slight deterioration in performance (RMSE increases by 2–3 ns for the offset and 0.5–0.6 ppm for the skew) compared to scenario (a). In fact, this is the cost of economizing on the number of BP iterations as well as rounds of time-stamp exchange. To clarify, BP commences

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 16 of 22

only when the nodes have already conducted K rounds of time-stamp exchange (to construct $\mathbf{W}_{ij}$ and $\mathbf{W}_{ji}$). Even then, it takes four iterations, or n if there are n nodes between an AP and the MN, to estimate the clock parameters and correspondingly perform synchronization. Conversely, BRF is faster, as it is directly applied after each round of time-stamp exchange and runs independently (does not need any information from the other network sections as BP does). Therefore, it can conduct more iterations, thereby continuously fulfilling the requirement of very low relative time error on a local level. Given the above-mentioned properties for BP and BRF, the hybrid approach sacrifices a fraction of global accuracy to rapidly achieve synchronization at a local level, which is crucial to a number of applications such as MU localization.

Figure 11 presents the RMSEs of the clock offset and skew estimation versus the rounds of time-stamp exchange K. As can be observed, the RMSEs of both offset and skew estimation decrease as K grows, indicating that the higher number of time-stamp exchanges leads to a more accurate estimation. The gradient is, however, slightly smaller for the APs owing to the fact that their RMSEs comprises two components, i.e., the synchronization error of the backhaul mesh network and the error arising when synchronizing APs with their corresponding backhaul nodes. Although the former decreases as K grows, the latter remains constant resulting in a slower decline of RMSEs of clock offset and skew estimation at the APs.

We note that the network in Fig. 9 is only a random example picked to lucidly convey the fundamental concepts of hybrid synchronization introduced in this work. The intuitions obtained from above simulations are still valid even if we replace the network by any other network with arbitrary size. Nevertheless, while the size of the network, in particular the backhaul network, does not play a role when locally synchronizing adjacent APs, it can prolong the time of convergence for BP depending on the number of nodes between node i and the MN.

5 Impact of hybrid synchronization on localization

To evaluate the impact of hybrid synchronization on the localization accuracy, we draw on the idea of joint synchronization and localization (sync&loc) introduced in [22]. In particular, in this section we focus on the edge of the communication network, as shown in Fig. 12, where the APs, on one hand, synchronize themselves with the backhaul nodes, i.e., the serving Base Stations (BSs). On the other hand, they perform joint sync&loc by exchanging time-stamps with MUs to which they have Line-of-Sight (LoS) connection (Fig. 12). Each MU i is assumed to exchange time-stamps with two APs, i.e., j and l.[2] In the sequel, we briefly describe the principles of joint sync&loc.

5.1 Joint MU synchronization and localization

The principles of Bayesian joint sync&loc are akin to those described in (30), (31), and (35). To incorporate the location estimation into the algorithm, we need to redefine $\tilde{\boldsymbol{\xi}}_i$ as

$$\tilde{\boldsymbol{\xi}}_i \triangleq \begin{bmatrix} \frac{1}{\tilde{\gamma}_i} & \frac{\tilde{\theta}_i}{\tilde{\gamma}_i} & x_i & y_i & v_{x_i} & v_{y_i} \end{bmatrix}^T,$$

---

[2] It is worth mentioning that in [22] each MU exchanges time-stamps with only one AP and the second AP passively listens to their exchange, which might not be implementable in practice. Therefore, in this work we have modified the algorithm as explained above.

where x_i/v_{x_i} and y_i/v_{y_i} denote the position/velocity of the MU i on the x and y axes, respectively. In particular, location-related parameters appear when expanding the propagation delay d_{ij} (or d_{ji}) as

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{41}$$

where x_j and y_j represent the known position of the jth AP on the x and y axes, respectively. Furthermore, each AP is assumed to be equipped with an N-element Uniform Linear Array (ULA) antenna and is able to perform Angle of Arrival (AoA) estimation in each round of time-stamp exchange. This estimation is given by

$$\arctan \frac{y_i - y_j}{x_i - x_j} = \varphi_{ij} + n_\varphi, \tag{42}$$

where φ_{ij} denotes the true AoA and $n_\varphi \sim \mathcal{N}(0, \sigma_\varphi^2)$ is the zero mean Gaussian noise stemming from the AoA estimation algorithm. For the sake of simplicity, in our simulations we rely on the Cramer–Rao Bound (CRB) of AoA estimation, derived in [32], to calculate σ_φ while φ_{ij} is computed knowing the exact trajectory of MU i and the location of AP j. Moreover, the nonlinearity in (41) and (42) is dealt with by resorting to *Taylor expansion*. The details of the approximation can be found in (43)–(46), where v_c represents the speed of light and (x_i^k, y_i^k) denotes the position of MU i predicted by the prediction step in the kth round of joint sync&loc. We note that similar set of equations, i.e., (41)–(45), and (46), can be written for the second AP serving MU i, i.e., AP l.

Given the new prediction and measurement equations, it is clear that $\mathbf{A}$, $\mathbf{Q}_n$, in (31) and $\mathbf{B}_{ij}$, $\mathbf{r}_{ij}$, $\mathbf{R}_{ij}$ in (35) require adjustment to account for the location parameters added to $\tilde{\boldsymbol{\xi}}_i$. The former can readily be updated using motion equations [22, 33], while the latter is adapted with the aid of time-stamp exchange and AoA measurements. The adjusted matrices and vectors are given in (47)–(49), where Δ represents the time between two consecutive rounds of joint sync&loc and $c_i(t)^j/c_i(t)^l$ denotes the time-stamp of MU i when exchanged with AP j/l (this is to distinguish the time-stamps sent by MU i to its two serving APs).

5.2 Performance analysis

We perform our analysis for the pedestrian scenario shown in Fig. 12. In particular, in this scenario, a MU (the pedestrian) moves with a constant velocity of 2 m/s (≈ 7 km/h) and takes the turns randomly until it exits the map. During its journey, we assume that the MU exchanges time-stamps with two APs to which it has LoS connection. Each AP performs AoA estimation as well. Both time-stamps and the AoA estimations are then combined by means of the joint sync&loc algorithm, described in Sect. 5 and depicted in Fig. 8, to estimate the vector variable $\tilde{\boldsymbol{\xi}}_i$. The simulation is conducted for two cases: (1) the APs synchronize themselves with the backhaul network by means of BP, corresponding to scenario (a) in Sect. 4.1, and (2) using the hybrid approach proposed in this work, corresponding to scenario (b) in Sect. 4.1.

Figure 13 depicts the RMSEs of the clock offset and position estimation of the MU versus the uncertainty in time-stamping. As can be noticed, both RMSEs increase with growth of σ_T. Although generally the RMSEs for (b) are larger, the difference in the rate of growth appears to be small, i.e., 0.05 m/ns for the RMSE of position and 0.3 for that of clock offset. This is a negligible cost at which we reduce the complexity of the algorithm by one BP iteration and K rounds of time-stamp exchange. Furthermore, the APs at the edge are able to perform localization immediately without waiting for the BP iterations. Consequently, as shown in Fig. 13, with only 3 more iterations of BRF, the gap between RMSEs can be halved (dotted curves lie in the middle of the solid ones).

$$\frac{d_{ij}}{v_c} \approx a_{j,0}^k + a_{j,x}^k(x_i - x_i^k) + a_{j,y}^k(y_i - y_i^k), \tag{43}$$

$$\arctan(\frac{y_i - y_j}{x_i - x_j}) \approx b_{j,0}^k + b_{j,x}^k(x_i - x_i^k) \\ + b_{j,y}^k(y_i - y_i^k), \tag{44}$$

$$a_{j,0}^k = \frac{1}{v_c}\left(\sqrt{(x_i^k - x_j)^2 + (y_i^k - y_j)^2}\right), \qquad a_{j,x}^k \\ = \frac{x_i^k - x_j}{v_c^2 a_{j,0}^k}, \qquad a_{j,y}^k = \frac{y_i^k - y_j}{v_c^2 a_{j,0}^k}, \tag{45}$$
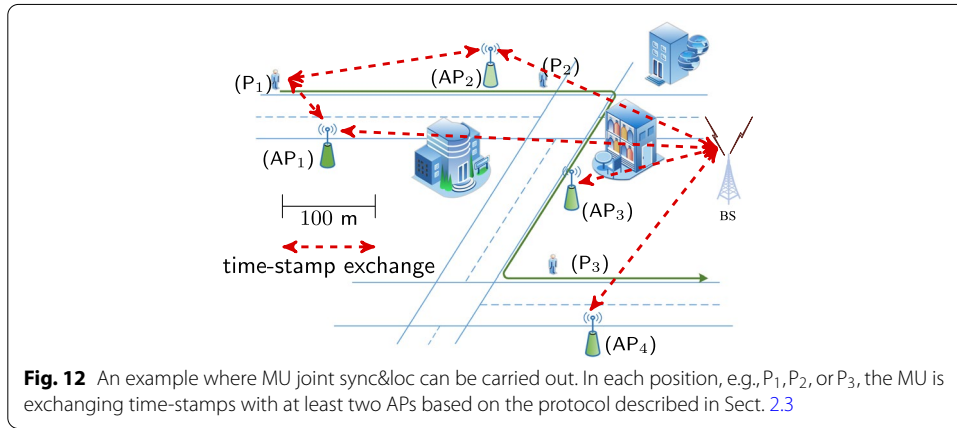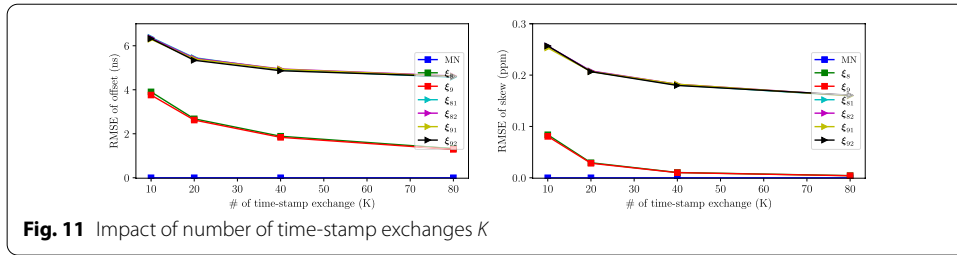
$$b_{j,0}^k = \arctan(\frac{y_i^k - y_j}{x_i^k - x_j}), \qquad b_{j,x}^k \\ = -\frac{y_i^k - y_j}{v_c^2 (a_{j,0}^k)^2}, \qquad b_{j,y}^k = \frac{x_i^k - x_j}{v_c^2 (a_{j,0}^k)^2}. \tag{46}$$

$$\mathbf{B}_{i,jl} = \begin{bmatrix} B_{11} & 0 & & \mathbf{0}_2 & & \mathbf{0}_2 \\ B_{21} & -2 & & & & \\ B_{31} & 0 & a_{j,x}^k - a_{l,x}^k & a_{j,y}^k - a_{l,y}^k & 0 & 0 \\ & & b_{j,x}^k & b_{j,y}^k & & \\ \mathbf{0}_2 & & b_{l,x}^k & b_{l,y}^k & & \mathbf{0}_2 \\ \mathbf{0}_2 & & -\frac{1}{\Delta}\mathbf{I}_2 & & & \mathbf{I}_2 \end{bmatrix},$$

$$B_{11} = \frac{1}{2}\left(c_i(t_4^k)^j - c_i(t_2^k)^j + c_i(t_4^k)^l - c_i(t_2^k)^l\right)$$

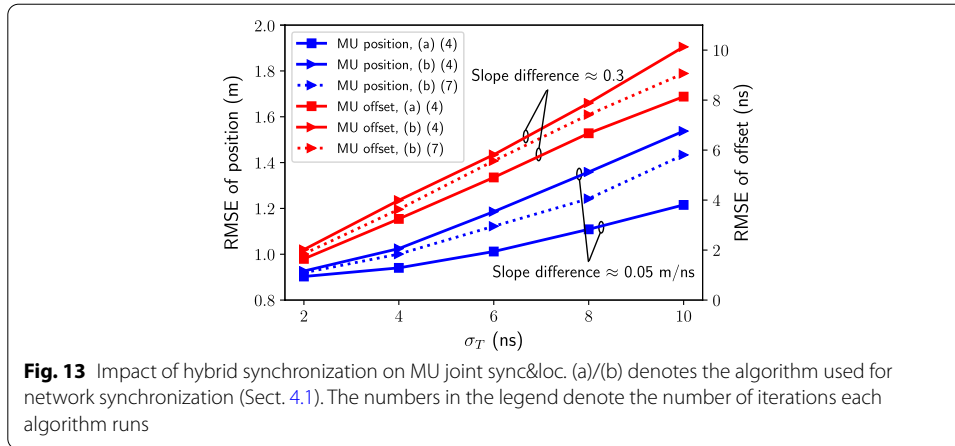$$B_{21} = \frac{1}{4}\left(c_i(t_2^k)^j + c_i(t_4^k)^j + 2c_i(t_5^k)^j \\ + c_i(t_2^k)^l + c_i(t_4^k)^l + 2c_l(t_5^k)^l\right)$$

$$B_{31} = c_i(t_5^k)^j - c_i(t_5^k)^l \tag{47}$$

Goodarzi *et al. J Wireless Com Network* (2021) 2021:91

Page 19 of 22

Fig. 11 Impact of number of time-stamp exchanges K

Fig. 12 An example where MU joint sync&loc can be carried out. In each position, e.g., P_1, P_2, or P_3, the MU is exchanging time-stamps with at least two APs based on the protocol described in Sect. 2.3

$$\mathbf{r}_{i,jl} = \left[r_1, r_2, r_3, r_4, r_5, -\frac{x_i^{k-1}}{\Delta}, -\frac{y_i^{k-1}}{\Delta} \right]^T.$$

$$r_1 = \frac{1}{2}\left(c_j(t_3^k) - c_j(t_1^k) + c_l(t_3^k) - c_l(t_1^k) \right)$$

$$r_2 = \frac{1}{4}\left(c_j(t_1^k) + c_j(t_3^k) + 2c_j(t_6^k) \right.$$

$$r_3 = c_j(t_6^k) - a_{j,0}^k + a_{j,x}^k x_i^k + a_{j,y}^k y_i^k$$
$$\qquad + c_l(t_1^k) + c_l(t_3^k) + 2c_l(t_6^k) \Big) \qquad - \left(c_l(t_6^k) - a_{l,0}^k + a_{l,x}^k x_l^k + a_{l,y}^k y_i^k \right)$$

$$r_4 = \varphi_j^k - b_{j,0}^k + b_{j,x}^k x_i^k + b_{j,y}^k y_i^k$$

$$r_5 = \varphi_l^k - b_{l,0}^k + b_{l,x}^k x_i^k + b_{l,y}^k y_i^k$$

$$(48)$$

$$\mathbf{R}_{i,jl} = \mathrm{diag}\left(\left[\frac{1}{2}\left(\sigma_{T_{ij}}^2 + \sigma_{T_{il}}^2 \right), \frac{1}{4} \right. \right.$$
$$\left(\frac{\sigma_{T_{ij}}^2 + \sigma_{T_{il}}^2}{2} + \sigma_{R_{ij}}^2 + \sigma_{R_{il}}^2 \right),$$
$$\left. \left. \sigma_{R_{ij}}^2 + \sigma_{R_{il}}^2, \sigma_{\varphi_j}^2, \sigma_{\varphi_l}^2, \left(\frac{\sigma_{x_i}^{k-1}}{\Delta} \right)^2, \left(\frac{\sigma_{y_i}^{k-1}}{\Delta} \right)^2 \right] \right).$$

$$(49)$$

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 20 of 22

Fig. 13 Impact of hybrid synchronization on MU joint sync&loc. (a)/(b) denotes the algorithm used for network synchronization (Sect. 4.1). The numbers in the legend denote the number of iterations each algorithm runs

6 Conclusions and future work

We presented two Bayesian approaches toward clock offset and skew estimation in communication networks. In particular, Belief Propagation (BP) was employed to perform high-precision network-wide synchronization, albeit at the cost of a high number of time-stamp exchanges and message passing iterations. Additionally, Bayesian Recursive Filtering (BRF) was leveraged to carry out pairwise synchronization, delivering a superb performance at the edge of the network. Based on these two algorithms, a hybrid Bayesian approach was proposed to not only fulfill a low relative time error at a local level but also to maintain a high synchronization accuracy at a global level. Lastly, we analyzed the impact of the proposed hybrid approach on a joint synchronization and localization (sync&loc) algorithm. Simulation results show that the proposed hybrid approach achieves faster and more frequent synchronization at the cost of only a slight deterioration in performance, i.e., around 3 ns, 0.5 ppm, and 0.1 m in the RMSEs of the clock offset, clock skew, and position, respectively.

Given the promising results, our future work targets the implementation of the hybrid synchronization algorithm presented in this work using Commercial-Off-The-Shelf (COTS) millimeter wave hardware. This would then allow the implementation of the joint synchronization and localization at the edge of the network as well.

Appendix

Employing Bayes rule:

$$p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{C}_{ij}) \propto \int p(\mathbf{C}_{ij} | \tilde{\boldsymbol{\xi}}_i^0, \ldots, \tilde{\boldsymbol{\xi}}_i^k) p(\tilde{\boldsymbol{\xi}}_i^0, \ldots, \tilde{\boldsymbol{\xi}}_i^k)\, d\Theta^{k-1}. \tag{50}$$

Assuming the independent measurements and Markov property [29], the integrands in (50) can be rewritten as

$$\begin{aligned}
p(\mathbf{C}_{ij} | \tilde{\boldsymbol{\xi}}_i^0, \ldots, \tilde{\boldsymbol{\xi}}_i^k) &= p(\mathbf{c}_{ij}^k | \tilde{\boldsymbol{\xi}}_i^k) \cdots p(\mathbf{c}_{ij}^1 | \tilde{\boldsymbol{\xi}}_i^1), \\
p(\tilde{\boldsymbol{\xi}}_i^0, \ldots, \tilde{\boldsymbol{\xi}}_i^k) &= p(\tilde{\boldsymbol{\xi}}_i^k | \tilde{\boldsymbol{\xi}}_i^{k-1}) \cdots p(\tilde{\boldsymbol{\xi}}_i^1 | \tilde{\boldsymbol{\xi}}_i^0) p(\tilde{\boldsymbol{\xi}}_i^0),
\end{aligned} \tag{51}$$

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 21 of 22

where $p(\tilde{\boldsymbol{\xi}}_i^0)$ denotes the prior knowledge on $\tilde{\boldsymbol{\xi}}_i$. Plugging (51) into (50) leads to (30) where

$$
\begin{aligned}
&p(\tilde{\boldsymbol{\xi}}_i^k | \mathbf{c}_{ij}^{1:k-1}) \\
&= \int p(\tilde{\boldsymbol{\xi}}_i^0) \left[\prod_{r=1}^{k-1} p(\tilde{\boldsymbol{\xi}}_i^r | \tilde{\boldsymbol{\xi}}_i^{r-1}) p(\mathbf{c}_{ij}^r | \tilde{\boldsymbol{\xi}}_i^r) \right] p(\tilde{\boldsymbol{\xi}}_i^k | \tilde{\boldsymbol{\xi}}_i^{k-1}) d\Theta^{k-1},
\end{aligned}
\tag{52}
$$

Abbreviations
5G: Fifth generation; AP: Access point; MU: Mobile user; WSN: Wireless sensor network; BP: Belief propagation; FG: Factor graph; pdf: Probability density function; PTP: Precision Time Protocol; BMCA: Best Master Clock Algorithm; MN: Master node; BRF: Bayesian recursive filtering; COTS: Commercial off-the-shelf; KL: Kullback–Leibler; BS: Base Station; LoS: Line-of-sight; sync&loc: Synchronization and localization; CRB: Cramer–Rao Bound; RMSE: Root mean square error; STD: Standard deviation.

Open Access

Author Contributions
Meysam Goodarzi conducted the study presented in this work. All authors took part in the discussions, reviewed and edited this manuscript. All authors read and approved the final manuscript.

Availability of data and materials
The python code used for the evaluations and analysis in this study is available from the corresponding author on reasonable requests.

Declarations

Competing interests
The authors declare that they have no competing interests.

Author details
[1]IHP - Leibniz-Institute für innovative Mikroelektronik, Frankfurt (Oder), Germany. [2]Humboldt University of Berlin, Berlin, Germany.

References
1. A. Kaloxylos, A. Gavras, R. De Peppe, Empowering vertical industries through 5G networks—current status and future trends. Zenodo (2020). https://doi.org/10.5281/zenodo.3698113
2. J. Werner, M. Costa, A. Hakkarainen, K. Leppanen, M. Valkama, Joint user node positioning and clock offset estimation in 5G ultra-dense networks. In *2015 IEEE Global Communications Conference (GLOBECOM)* (IEEE, 2015), pp. 1–7
3. B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer, H. Wymeersch, Cooperative simultaneous localization and synchronization in mobile agent networks. IEEE Trans. Signal Process. **65**(14), 3587–3602 (2017)
4. M. Koivisto, M. Costa, J. Werner, K. Heiska, J. Talvitie, K. Leppänen, V. Koivunen, M. Valkama, Joint device positioning and clock synchronization in 5g ultra-dense networks. IEEE Trans. Wirel. Commun. **16**(5), 2866–2881 (2017)
5. N. Maletic, V. Sark, M. Ehrig, J. Gutiérrez, E. Grass, Experimental evaluation of round-trip ToF-based localization in the 60 GHz band. In *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, pp. 1–6
6. M. Lévesque, D. Tipper, A survey of clock synchronization over packet-switched networks. IEEE Commun. Surv. Tutor. **18**(4), 2926–2947 (2016)
7. M. Leng, Y.-C. Wu, Distributed clock synchronization for wireless sensor networks using belief propagation. IEEE Trans. Signal Process. **59**(11), 5404–5414 (2011)

Goodarzi *et al. J Wireless Com Network*   (2021) 2021:91

Page 22 of 22

8.   K.J. Zou, K.W. Yang, M. Wang, B. Ren, J. Hu, J. Zhang, M. Hua, X. You, Network synchronization for dense small cell networks. IEEE Wirel. Commun. **22**(2), 108–117 (2015)

9.   B. Etzlinger, H. Wymeersch, A. Springer, Cooperative synchronization in wireless networks. IEEE Trans. Signal Process. **62**(11), 2837–2849 (2014)

10. J. Du, Y.-C. Wu, Distributed clock skew and offset estimation in wireless sensor networks: asynchronous algorithm and convergence analysis. IEEE Trans. Wireless Commun. **12**(11), 5908–5917 (2013)

11. G. Giorgi, C. Narduzzi, Performance analysis of Kalman-filter-based clock synchronization in IEEE 1588 networks. IEEE Trans. Instrum. Meas. **60**(8), 2902–2909 (2011)

12. M. Leng, Y.-C. Wu, Low-complexity maximum-likelihood estimator for clock synchronization of wireless sensor nodes under exponential delays. IEEE Trans. Signal Process. **59**(10), 4860–4870 (2011)

13. B. Lv, Y. Huang, T. Li, X. Dai, M. He, W. Zhang, Y. Yang, Simulation and performance analysis of the IEEE 1588 PTP with Kalman filtering in multi-hop wireless sensor networks. J. Netw. **9**(12), 3445 (2014)

14. D. Barber, *Bayesian Reasoning and Machine Learning* (Cambridge University Press, Cambridge, 2012).

15. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008) (2020), pp. 1–499

16. M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, E. Grass, Synchronization in 5G: a Bayesian approach. In *European Conference on Networks and Communications (EuCNC)* (IEEE, 2020), pp. 194–199

17. I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, Y.-C. Wu, Clock synchronization in wireless sensor networks: an overview. Sensors **9**(1), 56–85 (2009)

18. M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, E. Grass, A hybrid bayesian approach towards clock offset and skew estimation in 5G networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (IEEE, 2020).

19. S. Ruffini, P. Iovanna, M. Forsman, T. Thyni, A novel SDN-based architecture to provide synchronization as a service in 5G scenarios. IEEE Commun. Mag. **55**(3), 210–216 (2017)

20. H. Li, L. Han, R. Duan, G.M. Garner, Analysis of the synchronization requirements of 5G and corresponding solutions. IEEE Commun. Stand. Mag. **1**(1), 52–58 (2017)

21. S.P. Chepuri, R.T. Rajan, G. Leus, A.-J. van der Veen, Joint clock synchronization and ranging: asymmetrical time-stamping and passive listening. IEEE Signal Process. Lett. **20**(1), 51–54 (2012)

22. M. Goodarzi, N. Maletic, J. Gutiérrez, E. Grass, Bayesian joint synchronization and localization based on asymmetric time-stamp exchange. In *International Symposium on Networks, Computers and Communications (ISNCC)* (IEEE, 2020). arXiv:2008.08481.pdf

23. F. Meyer, B. Etzlinger, Z. Liu, F. Hlawatsch, M.Z. Win, A scalable algorithm for network localization and synchronization. IEEE Internet Things J. **5**(6), 4714–4727 (2018)

24. IEEE standard for information technology -telecommunications and information exchange between systems local and metropolitan area networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012) (2016), pp. 1–3534

25. B. Sundararaman, U. Buy, A.D. Kshemkalyani, Clock synchronization for wireless sensor networks: a survey. Ad Hoc Netw. **3**(3), 281–323 (2005)

26. L. Zdeborová, F. Krzakala, Statistical physics of inference: thresholds and algorithms. Adv. Phys. **65**(5), 453–552 (2016)

27. O. Shental, P.H. Siegel, J.K. Wolf, D. Bickson, D. Dolev, Gaussian belief propagation solver for systems of linear equations. In *2008 IEEE International Symposium on Information Theory* (IEEE, 2008), pp. 1863–1867

28. Y.-C. Wu, Q. Chaudhari, E. Serpedin, Clock synchronization of wireless sensor networks. IEEE Signal Process. Mag. **28**(1), 124–138 (2010)

29. A.L. Barker, D.E. Brown, W.N. Martin, Bayesian estimation and the Kalman filter. Comput. Math. Appl. **30**(10), 55–77 (1995)

30. Y. Pei, S. Biswas, D.S. Fussell, K. Pingali, An elementary introduction to Kalman filtering. Commun. ACM **62**(11), 122–133 (2019)

31. B. Li, N. Wu, Y.-C. Wu, Distributed verification of belief precisions convergence in gaussian belief propagation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2020), pp. 9115–9119

32. D.A. Fittipaldi, M. Luise, Cramér-rao bound for DOA estimation with antenna arrays and UWB-OFDM signals for PAN applications. In *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications* (IEEE, 2008), pp. 1–5

33. R. Khan, S.U. Khan, S. Khan, M.U.A. Khan, Localization performance evaluation of extended Kalman filter in wireless sensors network. Procedia Comput. Sci. **32**, 117–124 (2014)

Publisher's Note