Integrated Cooperative Synchronization for Wireless Sensor Networks

G. S. S. Chalapathi, Bernhard Etzlinger, S. Gurunarayanan, and Andreas Springer

***Abstract*—A precise decentralized clock synchronization method, referred to as integrated cooperative synchronization (ICS) is proposed in this letter. ICS is a delay compensating method that relies on mean-field message passing, where the measurement phase is integrated into the message passing phase. By selecting an extended factorization of the underlying joint a-posteriori distribution of the clock parameters and an appropriate message scheduling for link initialization, ICS is conceptually much simpler than conventional message-passing methods while achieving similar accuracy and reduced computational complexity. Moreover, it has only slightly higher implementation requirements compared to less accurate consensus methods.**

***Index Terms*—Clock synchronization, mean field, message passing, wireless sensor networks.**

I. Introduction

CLOCK synchronization is one of the basic building blocks for many applications in wireless sensor networks (WSN) or the Internet of Things (IoT) [1]. Thereby, increasing the node density advocates the application of decentralized synchronization methods [2]–[9]. From those methods, delay compensated approaches in [2]–[4] outperform the approaches in [5]–[9] as inaccuracies that result from the communication delay between nodes are mitigated.

From an implementation perspective, an outstanding work is the non-delay compensated method in [6]. In this method, each node records the time-of-arrival of packets broadcast by its neighbors. With the content of the received packet, i.e., the transmission timestamp and the current state estimate, it can update its own clock estimate with simple computations, and then broadcast it to the neighbors. This conceptional simplicity sets a benchmark for decentralized synchronization schemes.

In message passing [2], [3], the Mean-Field (MF) principle is of particular interest, since it inherently supports a broadcasting mechanism of synchronization information, has only moderate computational complexity and has guaranteed convergence. However, its biggest shortcoming compared to [6] is the requirement of two operational phases that are necessary for practical implementation. First, each node has to collect time measurements with all its neighbors and store them for the second phase (requiring additional memory resources). Second, the nodes have to iteratively update and broadcast their local messages. In contrast, the heuristic method in [4] has similar complexity as [6]. However, it uses a flooding principle that extracts a tree topology from the network, which is prone to error propagation.

In this letter, we propose an MF message passing scheme that integrates the measurement phase into the message passing phase. The main novelty is that we rely on an extended factorization of the a-posteriori distribution of the clock parameters and a specific message passing schedule such that with each broadcast packet a node can collect a new measurement and do message passing computations. This mitigates the two major drawbacks of [2] and [3] with only a small loss in achievable accuracy. The proposed method has a similar implementation complexity as the consensus method in [6].

II. System Model and Network Description

A. Network, Clock and Timestamping Model

We consider a network containing $P = M + A$ stationary nodes, consisting of $M \geq 1$ synchronous master nodes and A asynchronous agent nodes. The node indices are collected in the corresponding disjoint sets $\mathcal{M} \triangleq \{1, 2, \dots M\}$ $\mathcal{A} \triangleq \{M+1, M+2, \dots P\}$ for master and agent nodes, respectively, that form the set of all nodes $\mathcal{P} = \mathcal{M} \cup \mathcal{A}$. The network topology is represented by the set $\mathcal{S}$ that contains all node pairs (p, q) that can directly communicate to each other. Moreover, we require that all links are symmetric, i.e., if $(p, q) \in \mathcal{S} \implies (q, p) \in \mathcal{S}$. Further, we define the neighbor set of node $p \in \mathcal{P}$ as $\mathcal{T}_p = \{q | (p, q) \in \mathcal{S}\}$, i.e., it contains all the nodes which are directly connected to the node p.

A stationary skew-offset clock model is considered, where the local time at a node p, $c_p(t)$, is related to the reference time t with clock skew α_p and offset β_p by

$$c_p(t) = \alpha_p t + \beta_p \, . \tag{1}$$

The clock parameters are collected in the vector $\boldsymbol{\vartheta}_p \triangleq [1/\alpha_p \;\; \beta_p/\alpha_p]^{\mathrm{T}}$. Agent nodes have unknown $\boldsymbol{\vartheta}_p$, whereas master nodes are assumed to have $\boldsymbol{\vartheta}_p = [1 \; 0]^{\mathrm{T}}$, i.e., absolute synchronization [10] is considered.

Each node $p \in \mathcal{P}$ broadcasts N_p packets, each indexed by $n_p = 1, \dots, N_p$. All neighbors $q \in \mathcal{T}_p$ will receive the packets. Per link $(p, q) \in \mathcal{S}$, this yields N_p packets $p \to q$ and N_q packets $q \to p$. The n_pth packet $p \to q$ is transmitted at reference time $t_{p,0}^{(n_p)}$, and received at node q at $t_{pq,1}^{(n_p)} = t_{p,0}^{(n_p)} + \Delta_{pq} + \omega_{pq}^{(n_p)}$, where Δ_{pq} is the unknown communication delay that is assumed to remain constant and $\omega_{pq}^{(n_p)}$ is the measurement noise, modeled as normal Gaussian distribution [2] with $\omega_{pq}^{(n_p)} \sim \mathcal{N}(0, \sigma_\omega^2)$. Timestamps are recorded

Manuscript received October 23, 2018; accepted December 2, 2018. Date of publication December 6, 2018; date of current version June 19, 2019. This work was supported by the "LCM—K2 Center for Symbiotic Mechatronics" through the framework of the Austrian COMET-K2 Program. The work of G. S. S. Chalapathi was supported by BITS, Pilani. The associate editor coordinating the review of this paper and approving it for publication was C. T. Chou. *(Corresponding author: G. S. S. Chalapathi.)*

G. S. S. Chalapathi is with the Department of EEE, Birla Institute of Technology and Science, Pilani 333031, India (e-mail: gssc@pilani.bits-pilani.ac.in).

B. Etzlinger is with INRAS GmbH, 4040 Linz, Austria (e-mail: bernhard.etzlinger@inras.at).

S. Gurunarayanan is with the Department of EEE, Birla Institute of Technology and Science, Pilani 333031, India (e-mail: sguru@pilani.bits-pilani.ac.in).

A. Springer is with the Institute for Communications Engineering and RF-Systems, Johannes Kepler University, 4040 Linz, Austria (e-mail: andreas.springer@jku.at).

Digital Object Identifier 10.1109/LWC.2018.2885471

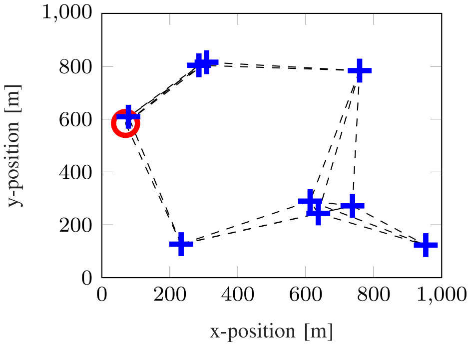

Fig. 1. Topology of a wireless network with 10 randomly placed nodes. Red circles indicate master nodes, blue crosses agent nodes, and the dashed lines the communication links.

at node p at transmission as $c_p(t_{p,0}^{(n_p)})$, and measured at node q at reception as $c_q(t_{pq,1}^{(n_p)})$. Using (1), the receive timestamp is

$$c_q\big(t_{pq,1}^{(n_p)}\big) = \alpha_q \frac{c_p\big(t_{p,0}^{(n_p)}\big) - \beta_p}{\alpha_p} + \alpha_q \omega_{pq}^{(n_p)} + \alpha_q \Delta_{pq} + \beta_q. \quad (2)$$

In the following we will use as short notation for the transmit timestamp $\tilde{c}_p^{(n_p)} \triangleq c_p(t_{p,0}^{(n_p)})$ and for the receive timestamp $c_{qp}^{(n_p)} \triangleq c_q(t_{pq,1}^{(n_p)})$.

B. Stochastic Model

To each node we assign a Gaussian prior distribution $p(\boldsymbol{\vartheta}_p)$ on the clock parameters, and on each link a Gaussian prior distribution $p(\Delta_{pq})$ on the communication delay.

The approximate local likelihood function [2] of $c_{qp}^{(n_p)}$ given the clock parameters $\boldsymbol{\vartheta}_p$ and $\boldsymbol{\vartheta}_q$ is

$$\begin{aligned} &p\big(c_{qp}^{(n_p)} | \boldsymbol{\vartheta_q}, \boldsymbol{\vartheta_p}, \Delta_{pq}\big) \\ &\quad \propto \exp\Big(-\frac{1}{2\sigma_\omega^2} \parallel \mathbf{a}_{pq}\boldsymbol{\vartheta}_q + \mathbf{b}_{pq}\boldsymbol{\vartheta_p} - \Delta_{pq} \parallel^2 \Big), \end{aligned} \quad (3)$$

where $\mathbf{a}_{pq}^{(n_p)} \triangleq [c_{qp}^{(n_p)} \; -1]$ and $\mathbf{b}_{pq}^{(n_p)} = [-\tilde{c}_p^{(n_p)} \; 1]$. Similarly, the likelihood of the n_qth $q \rightarrow p$ packet, $n_q \in \{1, \ldots, N_q\}$ can be derived by interchanging p and q in (3). The pairwise likelihood function of all measurements $\mathbf{c}_{pq} \triangleq [\{c_{qp}^{(n_p)}\}_{n_p=1}^{N_p}, \{c_{pq}^{(n_q)}\}_{n_q=1}^{N_q}]^{\mathrm{T}}$ between p and q is

$$\begin{aligned} p(\mathbf{c}_{pq}|\boldsymbol{\vartheta}_q, \boldsymbol{\vartheta}_p, \Delta_{pq}) &\propto \prod_{n_p=1}^{N_p} p\big(c_{qp}^{(n_p)}|\boldsymbol{\vartheta}_q, \boldsymbol{\vartheta}_p, \Delta_{pq}\big) \\ &\quad \times \prod_{n_q=1}^{N_q} p\big(c_{pq}^{(n_q)}|\boldsymbol{\vartheta}_q, \boldsymbol{\vartheta}_p, \Delta_{pq}\big). \end{aligned} \quad (4)$$

Collecting the pairwise likelihood function over all links and joining the prior distributions yields the global posterior function

$$\begin{aligned} p(\boldsymbol{\vartheta}, \boldsymbol{\Delta}|\mathbf{c}) &\propto \prod_{q' \in \mathcal{P}} p(\boldsymbol{\vartheta_{q'}}) \prod_{(p,q)\in\mathcal{S}} p(\Delta_{pq}) \\ &\quad \times \Big(\prod_{n_p=1}^{N_p} p\big(c_{qp}^{(n_p)}|\boldsymbol{\vartheta}_q, \boldsymbol{\vartheta}_p, \Delta_{pq}\big) \prod_{n_q=1}^{N_q} p\big(c_{pq}^{(n_q)}|\boldsymbol{\vartheta}_q, \boldsymbol{\vartheta}_p, \Delta_{pq}\big) \Big) \quad (5) \\ &= \prod_{q' \in \mathcal{P}} p(\boldsymbol{\vartheta_{q'}}) \prod_{(p,q)\in\mathcal{S}} p(\Delta_{pq}) p(\mathbf{c}_{pq}|\boldsymbol{\vartheta}_q, \boldsymbol{\vartheta}_p, \Delta_{pq}), \quad (6) \end{aligned}$$

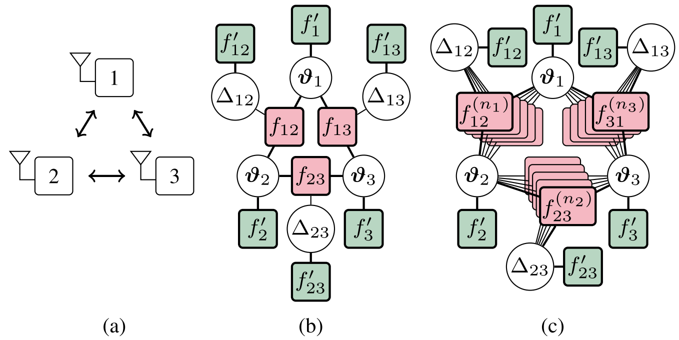

Fig. 2. (a) Wireless network, (b) corresponding FG for conventional MF based cooperative synchronization and delay estimation, (c) FG with extended factorization for MF based integrated cooperative synchronization. The following short notation is used: $f'_p = p(\boldsymbol{\vartheta}_p)$, $f'_{pq} = p(\Delta_{pq})$, $f_{pq}^{(n_q)} = p(c_{pq}^{(n_q)})|\boldsymbol{\vartheta}_p, \boldsymbol{\vartheta}_q, \Delta_{pq})$ and $f_{pq} = p(\mathbf{c}_{pq}|\boldsymbol{\vartheta}_p, \boldsymbol{\vartheta}_q, \Delta_{pq})$.

where the vectors $\mathbf{c}$ and $\boldsymbol{\Delta}$ collect all $\mathbf{c}_{pq}$ and Δ_{pq}, respectively, for $(p,q) \in \mathcal{S}$, and $\boldsymbol{\vartheta}$ all $\boldsymbol{\vartheta}_p$, $p \in \mathcal{P}$. From (6), an estimate of the clock parameters of node p is obtained by

$$\hat{\boldsymbol{\vartheta}}_p = \arg\max \int p(\boldsymbol{\vartheta}, \boldsymbol{\Delta}|\mathbf{c}) \, \mathrm{d} \sim \boldsymbol{\vartheta}_p, \quad (7)$$

where $\sim \boldsymbol{\vartheta}_p$ denotes all entries of $\boldsymbol{\vartheta}$ except $\boldsymbol{\vartheta}_p$.

III. Conventional Mean Field Message Passing

Conventional MF message passing finds an approximate solution $b(\boldsymbol{\vartheta_p}) \approx \int p(\boldsymbol{\vartheta}, \boldsymbol{\Delta}|\mathbf{c}) \, \mathrm{d} \sim \boldsymbol{\vartheta}_p$ to solve (7) in an iterative way [2], [3]. MF synchronization has high accuracy and moderate computational complexity, but requires a measurement phase and message passing phase in practical implementation. Coordinating these phases becomes highly challenging in asynchronous networks and thus prevents a broad application.

In the following, we will shortly introduce the basic MF equations to highlight the shortcomings in its practical implementation. Thereafter, modifications needed for ICS are introduced. Motivated from practical observations, we consider the following simplifications: (i) all links have the same measurement covariance σ_ω^2 and (ii) no prior on the distances is available, i.e., $p(\Delta_{pq})$ has infinite covariance.

Conventional message passing uses the factorization (6) to construct a factor graph (FG). For synchronization, this factor graph matches the topology of the network (see Fig. 2(a) and (b)). Thus, messages that are passed over the edges of the FG directly translate to packet transmissions between nodes. For example, parameters of messages between variable vertex $\boldsymbol{\vartheta}_1$ and $\boldsymbol{\vartheta}_2$ can be transmitted in packets between nodes 1 and 2.

On such a factor graph, MF operates iteratively after all measurements are obtained. The measurements are collected in the matrices $\mathbf{A}_{pq} \triangleq [\mathbf{a}_{qp}^{(1)\,\mathrm{T}}, \ldots, \mathbf{a}_{qp}^{(N_q)\,\mathrm{T}}, \mathbf{b}_{pq}^{(1)\,\mathrm{T}}, \ldots, \mathbf{b}_{pq}^{(N_p)\,\mathrm{T}}]^{\mathrm{T}}$ and $\mathbf{B}_{pq} \triangleq [\mathbf{b}_{qp}^{(1)\,\mathrm{T}}, \ldots, \mathbf{b}_{qp}^{(N_q)\,\mathrm{T}}, \mathbf{a}_{pq}^{(1)\,\mathrm{T}}, \ldots, \mathbf{a}_{pq}^{(N_p)\,\mathrm{T}}]^{\mathrm{T}}$. Starting from an initial guess $\hat{\Delta}_{pq}^{(0)}$ and $\hat{\boldsymbol{\vartheta}}_p^{(0)}$, a node computes[1] at iteration i

$$\hat{\Delta}_{pq}^{(i+1)} = \frac{\mathbf{1}^{\mathrm{T}}\big(\mathbf{B}_{pq}\hat{\boldsymbol{\vartheta}}_q^{(i)} + \mathbf{A}_{pq}\hat{\boldsymbol{\vartheta}}_p^{(i)}\big)}{(N_p + N_q)} \quad (8)$$

$$\hat{\boldsymbol{\vartheta}}_p^{(i+1)} = \mathbf{P}_p^{-1}\Big(\mathbf{m}_{0,p} + \sum_{q \in \mathcal{T}_p} \mathbf{A}_{pq}^{\mathrm{T}}\big(\hat{\Delta}_{pq}^{(i)} - \mathbf{B}_{pq}\hat{\boldsymbol{\vartheta}}_q^{(i)}\big)\Big), \quad (9)$$

[1] Note that (8) and (9) are a simplified version of the message passing equations in [3].

where

$$\mathbf{P}_p = \mathbf{P}_{0,p} + \sum_{q \in \mathcal{T}_p} \mathbf{A}_{pq}^{\mathrm{T}} \mathbf{A}_{pq}, \tag{10}$$

and $\mathbf{P}_{0,p}$ and $\mathbf{m}_{0,p}$ are the precision matrix and weighted mean of the prior $p(\boldsymbol{\vartheta}_p)$. After the computations are completed, the node broadcasts its estimate $\hat{\boldsymbol{\vartheta}}_p^{(i+1)}$ to its neighbors, and vice-versa receives their estimates. The method has two shortcomings. First, the iterations have to be coordinated among all nodes, and second, the measurements have to be collected before starting the message passing, requiring additional coordination of the network and memory for $(2\,N_p + 2\,N_q)$ timestamps that have to be stored at node p per neighbor q.

IV. Integrated Cooperative Synchronization

In contrast to conventional MF, ICS uses the factorization (5), in which each likelihood function obtained with a single packet exchange is considered separately, i.e., it performs message passing on an extended FG (See Fig. 2(c)). In every transmitted packet, the transmission time stamp is appended, and by measuring the receive time, the receiver can construct a single packet likelihood directly after reception. If the transmitter additionally includes its current message parameters, this likelihood function can be directly used for message passing. Thereby, the message passing does not operate iteratively as a message is passed only once over every single edge.

Although the proposed modification seems trivial, the message scheduling requires careful consideration when no prior delay estimate is available. Therefore, we will first introduce the computation rules for standard operation when previous accurate delay estimates are available, and thereafter discuss the special case when a new link is initialized, i.e., when no prior delay estimate is available. The transition between those modes is per node and depends on a simple local decision.

A. Mode I: Standard Operation

Nodes perform message computation for synchronization in the standard operation mode if an accurate estimate of the delay to the neighbor is available. This mode is motivated by observation that $\mathbf{A}_{pq}^{\mathrm{T}} \mathbf{A}_{pq} = \sum \mathbf{a}_{qp}^{(n_q)\,\mathrm{T}} \mathbf{a}_{qp}^{(n_q)} + \sum \mathbf{b}_{pq}^{(n_p)\,\mathrm{T}} \mathbf{b}_{pq}^{(n_p)}$ in (10) and $\mathbf{A}_{pq}^{\mathrm{T}} \mathbf{B}_{pq} = \sum \mathbf{a}_{qp}^{(n_q)\,\mathrm{T}} \mathbf{b}_{pq}^{(n_q)} + \sum \mathbf{b}_{qp}^{(n_p)\,\mathrm{T}} \mathbf{a}_{pq}^{(n_p)}$ in (9). Selecting message passing only in the direction of packet transmission will reproduce the first parts of those sums.

Each node q broadcasts packets to its neighbors, where the n_qth packet includes the transmission timestamp $\tilde{c}_q^{n_q}$, its clock estimate $\hat{\boldsymbol{\vartheta}}_q^{(k_q)}$, and the delay estimates to all its neighbors $\hat{\Delta}_{pq}^{(k_{pq})}$, $p \in \mathcal{T}_q$. Here, k_q and k_{pq} are the recursion indices for the clock parameter estimate and for the delay estimate, respectively. If a node p receives the n_qth packet from node q, it can directly perform the following message passing computations on the branch of the FG that includes $f_{qp}^{(n_q)}$:

$$\mathbf{P}_{qp}^{(n_q)} = \mathbf{a}_{qp}^{(n_q)\mathrm{T}} \mathbf{a}_{qp}^{(n_q)},\; \mathbf{m}_{qp}^{(n_q)} = \mathbf{a}_{qp}^{(n_q)\mathrm{T}} \big(\hat{\Delta}_{pq}^{(k_{pq})} - \mathbf{b}_{qp}^{(n_q)} \hat{\boldsymbol{\vartheta}}_q^{(k_q)} \big), \tag{11}$$

to then update its local variables

$$\mathbf{P}_p^{(k_p+1)} = \mathbf{P}_p^{(k_p)} + \mathbf{P}_{qp}^{(n_q)}, \mathbf{m}_p^{(k_p+1)} = \mathbf{m}_p^{(k_p)} + \mathbf{m}_{qp}^{(n_q)}, \tag{12}$$

and estimates

$$\hat{\Delta}_{pq}^{(k_{pq}+1)} = \frac{k_{pq} \hat{\Delta}_{pq} + \big(\mathbf{a}_{pq}^{(n_q)} \hat{\boldsymbol{\vartheta}}_p^{(k_p)} + \mathbf{b}_{pq}^{(n_q)} \hat{\boldsymbol{\vartheta}}_q^{(k_q)} \big)}{k_{pq} + 1}, \tag{13}$$

$$\hat{\boldsymbol{\vartheta}}_p^{(k_p+1)} = \big(\mathbf{P}_p^{(k_p+1)} \big)^{-1} \mathbf{m}_p^{(k_p+1)}. \tag{14}$$

The updated parameters $\hat{\boldsymbol{\vartheta}}_p^{(k_p+1)}$ and $\hat{\Delta}_p^{(k_p+1)}$ are broadcast with the next packet. If a node is a master node, i.e., $p \in \mathcal{A}$, then the clock parameters are not updated as in (14), but remain constant. Note that no timestamps from previous packets have to be stored, i.e., the message passing happens with the same packet as the measurement recording.

B. Mode II: Link Initialization

We consider the initialization $\mathbf{P}_p^{(0)} = \mathbf{0}$, $\mathbf{m}_p^{(0)} = \mathbf{0}$ for all non-master nodes $p \in \mathcal{A}$ and $\hat{\Delta}_{pq}^{(0)} = 0$ for all links $(p, q) \in \mathcal{S}$. Thus, we assume not having any prior knowledge on the network, except the clock parameters of the master nodes.

When a node $p \in \mathcal{A}$ receives the first packet from a neighbor with accurate clock parameter estimates, i.e., from a master node (or a node q with a sufficiently large clock estimation index k_q, referred to as pseudo-master node), it would not be able to compute the standard mode equations. Firstly, $\mathbf{P}_p^{(1)}$ cannot be inverted in (14) after only one reception, i.e., $\mathbf{P}_p^{(1)} = \mathbf{P}_{qp}^{(1)}$, as it has only rank one (See (11)). This would not be a limitation if node p has already updated $\mathbf{P}_p^{(k_p)}$, $k_p > 1$ times with received packets from other (pseudo-)master nodes. However, secondly, as the delay is not known before, $\mathbf{m}_{qp}^{(1)}$ in (11) will be inaccurate and severely perturb the network synchronization process.

Therefore, we propose that every node has one set of link initialization variables $\mathbf{P}_{p,\mathrm{init}}^{(k_p')}$ and $\mathbf{m}_{p,\mathrm{init}}^{(k_p')}$, and that the following procedure is executed when a node p receives a packet from a neighboring (pseudo-)master q.

1) Node p initializes $\mathbf{P}_{p,\mathrm{init}}^{(0)} = \mathbf{0}$, $\mathbf{m}_{p,\mathrm{init}}^{(0)} = \mathbf{0}$ and $k_p' = 0$
2) Node p updates the local initialization variables for the for the first two packets received from a (pseudo-)master q as in (11), with $\hat{\Delta}_{pq}^{(1)} = \hat{\Delta}_{pq}^{(2)} = 0$.
3) Node p then computes $\hat{\boldsymbol{\vartheta}}_{p,\mathrm{init}} = \mathbf{P}_{p,\mathrm{init}}^{(2)\,-1} \mathbf{m}_{p,\mathrm{init}}^{(2)}$ and adds it to its next broadcast, indexed by n_p'. At this point, a deterministic error occurs on the phase estimate (second entry of $\hat{\boldsymbol{\vartheta}}_{p,\mathrm{init}}$) of the magnitude of the unknown Δ_{pq}.
4) Node q computes the delay estimate with $\hat{\Delta}_{pq}^{(3)} = (\mathbf{a}_{qp}^{(n_p')} \hat{\boldsymbol{\vartheta}}_{p,\mathrm{init}} + \mathbf{b}_{pq}^{(n_p')} \hat{\boldsymbol{\vartheta}}_q^{(k_q)})/2$, and broadcasts the result with its next packet.
5) Node p corrects its previous estimate to $\hat{\boldsymbol{\vartheta}}'_{p,\mathrm{init}} = \hat{\boldsymbol{\vartheta}}_{p,\mathrm{init}} - [0, \hat{\Delta}_{pq}^{(3)}]^{\mathrm{T}}$ and its variable $\mathbf{m}_{p,\mathrm{init}}^{(2)} = \mathbf{P}_{p,\mathrm{init}}^{(2)} \hat{\boldsymbol{\vartheta}}'_{p,\mathrm{init}}$. It then adds the initialization variables to its standard operation variables with $\mathbf{P}_p^{(k_p+1)} = \mathbf{P}_p^{(k_p)} + \mathbf{P}_{p,\mathrm{init}}^{(2)}$ and $\mathbf{m}_p^{(k_p+1)} = \mathbf{m}_p^{(k_p)} + \mathbf{m}_{p,\mathrm{init}}^{(2)}$.
6) Node p joins the standard operation mode, and executes (11)–(14) with the packet received in step 5.

Those steps, graphically depicted in Fig. 3, require the cooperation of node p and q during the transmission of 4 packets.

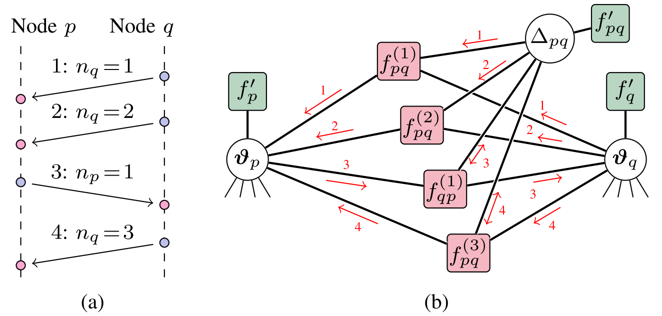

Fig. 3. Link initialization mode, with four packets exchanged between nodes p and q in (a) and their corresponding messages on the FG in (b).

TABLE I
PER NODE COMPUTATIONAL COMPLEXITY IN NUMBER OF OPERATIONS: FOR ICS AND ATS PER RECEIVED PACKET, AND FOR CMF PER ITERATION OF THE MESSAGE PASSING PHASE

Protocol	Additions	Multiplications	Div.
ICS	15	19	2
ATS	10	7	1
CMF	$(\lvert\mathcal{T}_p\rvert(8K-6))+4K+8$	$(\lvert\mathcal{T}_p\rvert 8K)+4K+6$	2

V. EVALUATIONS

For evaluating ICS we used a network of 10 nodes randomly placed in a $1000 \times 1000\,\text{m}^2$ field as shown in Fig. 1, where the connectivity is defined by the node range of 600 m. The noise standard deviation σ_ω was set to 93 ns [2]. The communication delays Δ_{pq} are set to the propagation time, i.e., distance between the nodes divided by the speed of light. The clock skews were drawn by a normal distribution of standard deviation 100 ppm, i.e., $\alpha_p \sim \mathcal{N}(1, 10^{-4})$. Similarly the standard deviation for the clock offset is selected to 5 s, i.e., $\beta_p \sim \mathcal{N}(0, 5)$. A node is selected as pseudo-master if it has $k_p = 20$ updates in the standard operation.

We compare the performance of ICS with the consensus method from [6], referred to as ATS, and the conventional MF method from [2] using $K = 100$ measurements and $I = 4$ iterations, referred to as CMF. For evaluation, we depict 800 time intervals, where each node broadcasts one packet per interval. While ICS and ATS perform synchronization operations after each broadcast, CMF has to collect its measurements first. In Fig. 4, the RMSE of the clock phase (solid lines) and clock skew (dashed lines) are depicted. While all methods achieve similar accuracy for clock skew with the selected parameters, the differences in the clock phase accuracy are significant. ATS, which is limited by the communication delay, achieves an accuracy of 27.5 μs, while ICS and CMF achieve 0.23 μs and 0.05 μs, respectively. Thus, ICS clearly outperforms ATS while having a similar complexity and achieves results close to CMF that has major drawbacks for ad-hoc network implementations.

Finally, we assess the latency and the computational complexity. For ICS we define the latency as the number of intervals until all nodes are pseudo-master nodes in standard operation, i.e., k_p times the network diameter (which is 3 in this network). ATS has no latency, as all nodes update their clock parameters from the first packet on. For CMF, the latency is defined by the duration of the message passing phase, i.e., K plus I. For the considered network we have 60, 1 and 104 intervals of latency for ICS, ATS and CMF, respectively. The computational complexity is assessed in the number of mathematical operations per received packet for ICS and ATS, or per iteration in the message passing phase for CMF. The resulting

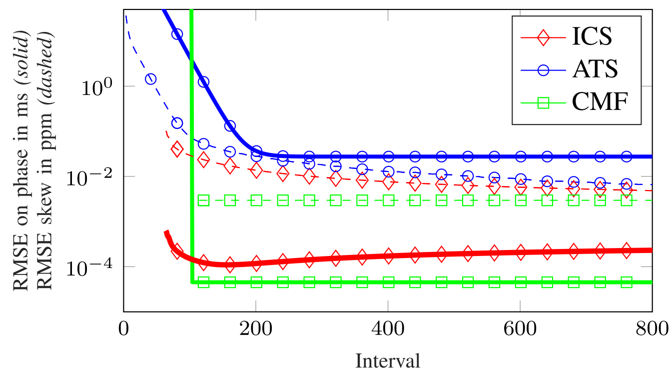

Fig. 4. Numerical results with RMSE on clock phase (solid lines) and clock skew (dashed lines).

numbers are shown in Table I. While ICS requires only slightly more operations per packet than ATS, both are advantageous to CMF in the message passing phase.

VI. CONCLUSION

While existing message passing based synchronization schemes yield excellent results, they have major shortcomings that prevent them from broad application. In order to mitigate those drawbacks, ICS proposes to use MF message passing on an extended factorization of the a posteriori function and a specific message schedule, in order to achieve a simplistic and still highly accurate synchronization method with drastic reduction in computational complexity. Hereby, nodes have to perform only simple computations, and take local decisions for operating in a standard operation or in a link initialization phase. The above-mentioned features make ICS suitable for resource constrained WSN nodes. Despite convergence was observed in numerical evaluations, a theoretical analysis has to be carried out in the future work.

REFERENCES

[1] Y.-C. Wu, Q. M. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 28, no. 1, pp. 124–138, Jan. 2011.

[2] B. Etzlinger, H. Wymeersch, and A. Springer, "Cooperative synchronization in wireless networks," *IEEE Trans. Signal Process.*, vol. 62, no. 11, pp. 2837–2849, Jun. 2014.

[3] B. Etzlinger, D. Bartel, W. Haselmayr, and A. Springer, "Mean field message passing for cooperative simultaneous ranging and synchronization," in *Proc. IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2013, pp. 583–586.

[4] R. Lim, B. Maag, and L. Thiele, "Time-of-flight aware time synchronization for wireless embedded systems," in *Proc. Eur. Conf. Wireless Syst. Netw. (EWSN)*, Feb. 2016, pp. 149–158.

[5] D. Zennaro, E. Dall'Anese, T. Erseghe, and L. Vangelista, "Fast clock synchronization in wireless sensor networks via ADMM-based consensus," in *Proc. 9th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw.*, May 2011, pp. 148–153.

[6] L. Schenato and F. Fiorentin, "Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, Sep. 2011.

[7] M. K. Maggs, S. G. O'Keefe, and D. V. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 12, no. 6, pp. 2269–2277, Jun. 2012.

[8] Y.-P. Tian, "Time synchronization in WSNs with random bounded communication delays," *IEEE Trans. Autom. Control*, vol. 62, no. 10, pp. 5445–5450, Oct. 2017.

[9] K. S. Yıldırım, R. Carli, and L. Schenato, "Adaptive proportional–integral clock synchronization in wireless sensor networks," *IEEE Trans. Control Sys. Technol.*, vol. 26, no. 2, pp. 610–623, Mar. 2018.

[10] Y. Xiong, N. Wu, Y. Shen, and M. Z. Win, "Cooperative network synchronization: Asymptotic analysis," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 757–772, Feb. 2018.