Article

Randomized and Efficient Time Synchronization in Dynamic Wireless Sensor Networks: A Gossip-Consensus-Based Approach

Nan Xiong [1], Minrui Fei [1]* and Taicheng Yang [2]

[1] Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200072, China; nan0521@i.shu.edu.cn (N.X.)
[2] Department of Engineering and Design, University of Sussex, Brighton BN1 9QT, U.K.; taiyang@sussex.ac.uk (T.Y.)
* Correspondence: mrfei@staff.shu.edu.cn (M.F.); Tel.: +86-021-5633-1934

Academic Editor: name
Version August 4, 2018 submitted to Complexity;

Abstract: This paper proposes novel randomized gossip-consensus-based sync (RGCS) algorithms to realize efficient time correction in dynamic wireless sensor networks (WSNs). First, the unreliable links are described by stochastic connections, reflecting the characteristic of changing connectivity gleaned from dynamic WSNs. Secondly, based on the mutual drift estimation, each pair of activated nodes fully adjusts clock rate and offset to achieve network-wide time synchronization by drawing upon the gossip consensus approach. The converge-to-max criterion is introduced to achieve a much faster convergence speed. The theoretical results on the probabilistic synchronization performance of the RGCS are presented. Thirdly, a Revised-RGCS is developed to counteract the negative impact of bounded delays, because the uncertain delays are always present in practice and would lead to a large deterioration of algorithm performances. Finally, extensive simulations are performed on the MATLAB and OMNeT++ platform for performance evaluation. Simulation results demonstrate that the proposed algorithms are not only efficient for synchronization issues required for dynamic topology changes but also give a better performance in term of converging speed, collision rate, and the robustness of resisting delay, and outperform other existing protocols.

Keywords: wireless sensor networks; consensus time synchronization; gossip algorithms; changing connectivity; fast convergence; collisions; bounded delays;

1. Introduction

Ad hoc Wireless Sensor Networks (WSNs) without pre-existing infrastructures are composed of autonomous sensors [1], and the sensors are a variety of low cost, low power, sensing devices, which work cooperatively through ad hoc wireless communications. In WSNs, many kinds of missions [2,3], such as transmission scheduling, event sequencing, information fusion, and distributed filtering, rely heavily on a consistent notion of time to keep running. For instance, Internet of Things' perception layer provides a ubiquitous access to the network, in which the multiple terminals collaborate closely with each other, and accurate synchronized clocks determine whether a multi-terminal cooperation is accomplished or not. Therefore, timing protocols, which should concern with topologies, time-delays, low power consumption, etc., is the premise of dealing with services of ad hoc WSNs.

Two configurations of timing protocols are available: hierarchical and distributed. Cluster-based and tree-based protocols would both fall into the hierarchical configuration, because they perform a special operation; that is, some designated nodes are elected to be the reference (such as cluster head, root, master). The typical hierarchical timing protocols include the one-way message dissemination schemes [4], two-way message exchange schemes [5], flooding schemes [6], Pairwise Broadcast

Synchronization (PBS) [7], and Reference Broadcast Synchronization (RBS) [8]. Based on the message passing and filter methods, most of the works have build an estimation and tracking framework for the time synchronization problem, but these prototypes are not purely decentralized in nature. The hierarchy structure of network consists in the logical master-slave relationship which is maintained by actual infrastructure. The timing messages in this structure are delivered from top to bottom, while each node is labeled an equal identity tag in distributed configuration. In the absence of pre-existing infrastructure, the distributed configuration is obviously more suitable to WSNs.

Recently, based on distributed consensus algorithms [9,10] of multi-agent networks, consensus based approaches for designing distributed timing protocols [11–21] have received a lot of attention in WSNs. These protocols utilize local information to achieve global synchronization and easily adopt to WSNs' distributed sensor property with computational lightness. However, the common drawbacks of existing consensus based timing protocols are as follows:

(1) The changing connectivity is rarely considered [17]. In dynamic WSNs, it is not possible to deterministically forecast the activating timeslot of sensors due to the uncertainties of nodes, such as uncontrollable mobility, sleep scheduling, heterogeneous nodes with various coverage levels, etc. Radio signal fading caused by environmental turbulence can also severely corrupt the performance of wireless links. The nodes in the design of the consensus based timing protocols sequentially update their own clocks whenever they receive a timing message from a geographical neighbor [11]. Because the logical rate represents the slope of the linear logical clock model, each node i is required to build a fixed link with a same neighboring node in order to collect timing messages and estimate the relative drift. In the MTS [12], the authors assume that a larger B is used to keep link (i, j) constant during a time interval $[kB, (k+1)B]$. Although it enables the MTS to behave robustly to work against topology changes, in highly dynamic topology this assumption will be unrealistic. This is because at a timeslot t_1, node i can exchange timing messages with a neighbor only once to obtain clock reading. At next timeslot t_2, the link (i, j) will lost the connection. More generally, any pair of nodes "gossip" only once in one timeslot. Thus, the randomly changing connectivity between adjacent nodes renders efficient relative drift estimation challenging. This problem will directly affect the implementation of logical clock rate and offset compensation, and become more paramount as the change of connected relation of WSNs gets faster. On this issue the existing consensus based timing protocols can be classified into two categories: 1) deterministic synchronization, which relies on a fixed link, such as [11–18]; and 2) randomized synchronization, such as [19–21]. Brown et al. [19] investigated the transient consensus behavior of the Random Pairwise Consensus Synchronization (RPCS) algorithm. It is a pity that the reciprocal delay model is supposed to be symmetric. Strom et al. [20] proposed a randomized method to deal with the random access problem. The method proposed therein combines partial-update rule with complete-update rule. However, the partial-update rule is capable of compensating the offset in non-deterministic instants which results in incompletely compensating the clock rate. Bolognani et al. [21] proposed a randomized linear algorithm for the second-order consensus timing protocol. This algorithm is based on the average-value-based criterion, which have a slow speed of convergence.

(2) The speed of convergence may be relatively slow. A major concern is the lower convergence speed which increases the message complexity and consumes the limited power of the sensor nodes. However, many protocols [11,13–16,19–21] are proposed based on the average-value-based criterion, which need more iterations to achieve acceptable synchronization error.

(3) The delay model is unrealistic. Several protocols suppose the uplink delay and the downlink delay among two nodes are symmetric [13,19] or obey a statistical delay model [12,14], such as normal distribution, Gaussian distribution, etc. Worse still, the uncertainty regarding the delay is associated with the influence of exterior environment.

(4) The common broadcast period would give rise to the timing message collision as pointed out in publications [11,18,20], but without being handled. The existing consensus based timing protocols employ a deterministic communication protocol, in which each node is allowed to communicate with its neighboring nodes that are within its range in each deterministic timeslot. Due to the hidden node

problem [24], the collisions become too serious to be further ignored since a exposed node receives multiple timing messages from hidden nodes with common broadcast period during a same timeslot.

The critical fact that the real-world delays would change irregularly, and the more iterations and the collisions would waste energy, has a negative effect on synchronization performances.

Inspired by the rumor spreading of human groups and the way how epidemics spread, gossip algorithm [25,26] has been widely investigated in the information theory community for information dissemination. In the context of the multi-agent networks, randomized gossip algorithm is attractive for solving the distributed and stochastic consensus problems [27–29] due to its randomized behavior and asynchronous processing. Faced with the dynamic topology issue, the gossip consensus provides less conservatism and higher efficiency than the existing consensus approach. So they are perceived as uniquely suited for the inherent dynamics of ad hoc WSNs whose topology is randomly connected. The gossip consensus approach contains two dimensions: gossip interaction and gossip update. The gossip interaction mode is particularly well suited for applications into wireless peer-to-peer and ad hoc sensor networks, where a random pair of nodes is active at each iteration and out-sync itself is a prior fact. In addition, randomized gossiping may allow communication asynchronously at random times to lower the probability of message collisions. The gossip update policy is that each node has to share its output with one of neighboring nodes. That being said, privacy issues could easily apply to a node who participating in such updating. On the contrary, the existing consensus approach allow a node to share its output with multiple nodes, so privacy issues could discourage some nodes from participating in such updating. Therefore, under a more practical delay model, it is of great interest to study randomized gossip algorithm for achieving the generalized consensus timing protocol that has faster convergence speed and lower collision rate in dynamic WSNs.

Motivated by these, this paper presents an innovative randomized and efficient timing protocol for dynamic WSNs, including the mutual drift estimation, the clock rate and offset compensation. The key novelty of the contribution lies in randomized algorithm for the extension of the generalized consensus based timing protocol which implements probabilistic synchronization using randomized gossip-type interactions and updates. To achieve this goal, for the first time, we design the idea of the randomized activation of the synchronization links based on the pre-programmed Poisson process to activate a pair of nodes, and use the gossip-consensus-based approach with the converge-to-max criterion to fully adjust the clock rate and offset of random pairwise nodes. Moveover, by exploiting a least square based low-pass filter, a revised version is improved to cope with the bounded delays.

The major contributions of this paper are summarized below:

- We first ignore the time-delays and propose a novel Randomized Gossip-Consensus-based time Synchronization (RGCS) protocol by incorporating the idea of randomized gossip algorithm into the consensus based timing protocol. Compared to traditional mechanisms, the gossip consensus is achievable in a fully decentralized, randomized and asynchronous fashion, even in highly dynamic WSNs. The converge-to-max criterion is introduced to achieve finite-time convergence, since the gossip consensus is a not fast convergence algorithm. Then we prove that the expected logical clocks are synchronized with probability one (w.p.1), and provide a lower bound of fast finite-time convergence. In addition, we develop Multi-RGCS protocol based on the principle of the Edge-Coloring (EC) technique to save finite-convergence time.
- We consider the case where the delay obeys a realistic bounded model compared to the particular distribution communication delay model, and further propose a Revised-RGCS protocol to work against the uncertain bounded time-delays. The filter proposed therein is competent to deal with practical delays which could be a basic constraint in drift estimation issue over real-world WSNs.
- We conduct performance evaluations of the proposed protocols through extensive simulation experiments using the MATLAB and OMNeT++. Simulation results demonstrate that RGCS fully adjusts logical rate and offset to achieve network-wide synchronization for randomly connected WSNs, and Revised-RGCS gives a better performance in terms of collision rate, converging speed, and the robustness of resisting delays compared to other existing protocols.

The rest of this paper is organized as follows. In section 2, we review the related literatures and state the novelty with respect to the previous works. Section 3 formulates the time synchronization problem. Section 4 elaborates the proposed RGCS, Multi-RGCS, and Revised-RGCS. The simulation results are given in Section 5. Section 6 concludes the paper.

2. Related Work

In the absence of pre-existing infrastructures, distributed configuration is a promising paradigm for distributed WSNs. Within this context, tremendous research efforts have already been devoted to distributed timing protocols. Based on belief propagation, a fully distributed timing protocol [30] developing from the two-way sender-receiver synchronization scheme was used for joint estimation of rate and offset. Distributed asynchronous Clock Synchronization (DCS) [31] protocol was proposed for delay-tolerant networks. DCS can achieve global time synchronization among mobile nodes over intermittent connections with long delays, but the rate and offset are adjusted separately. Ahmed et al. [32] considered unreliable links to build asynchronous framework, but it is the same as RFA without rate compensation. An On-demand Time Synchronization Protocol (AOTSP) [33] was proposed with the advantages of weak spatial accumulative effect, low communication cost, and high scalability. However, AOTSP suffers from temporal accumulative effect on account of exchanging timestamps.

In recent years, distributed consensus concept [9,10] has become a hot topic in the distributed time calibration research. Distributed timing protocol based on the idea of the consensus algorithm was originally achieved with the Average TimeSynch (ATS) [11] protocol which fully compensates clock rate and offset. To accelerate convergence speed, the Maximum Time Synchronization (MTS) [12] protocol was developed by which the system trajectories of rate and offset are updated to achieve maximum-value under normal distribution delay model, and Saiah et al. [14] proposed Consensus based Multi-hop Time Synchronization (CMTS) protocol under Gaussian delay model. Panigrahi et al. [13] proposed a multi-objective evolutionary strategy based topological optimization for the consensus timing protocol to deal with a trade-off between the minimizing of delays and the selection of sync initiating nodes, but the delays are assumed to be symmetric. The other improved protocols include the Robust-ATS (RoATS) [15] and Least Square estimation based Time Synchronization (LSTS) [16] aiming at the delays. Recent works contributed by Tian et al. [17] presented a very generalized model of Consensus Based Time Synchronization (CBTS) algorithm without topological conditions. However, the RoATS, LSTS, and CBTS designed for distributed WSNs with delays are static scenarios and ignore the changing connectivity of dynamic WSNs. Brown et al. [19] investigated the transient consensus behavior of clock parameters in the Random Pairwise Consensus Synchronization (RPCS). Similar to DCS, RPCS adjusts rate and offset separately. He and Dong et al. investigated consensus timing protocol under message manipulation attacks [22] and sybil attacks [23], respectively. In order to address topology change issue, the MTS protocol suppose that link (i, j) should be keep constant within a time width B, however in highly dynamic topology, this assumption is unrealistic. Strom et al. proposed Random Broadcast based Distributed time Synchronization (RBDS) [20] protocol based on random access mechanism, which combines partial-update rule with complete-update rule. However, the partial-update rule is capable of compensating the offset in non-deterministic timeslot which results in incompletely adjusting the drift. Based on the gossip consensus, we relax the basic condition of the MTS (i.e., compared with the MTS, we don't need link (i, j) keep constant during a width of time $[kB, (k+1)B]$ to collect the clock states more than once) and also enable complete-rate and complete-offset compensation.

There has been a surge of activity in ad hoc WSNs using the idea of gossip protocols. Specific to time synchronization, Marechal et al. [34] proposed a distributed gossip protocol for only adjusting natural clock drifts. Ruggero et al. proposed a Proportional Integral (PI) synchronization controller [35] with gossip communication mode for time synchronization. The PI controller proposed therein takes advantage of the asynchronism of pairwise-exchange communication. Coupling-based internal Clock Synchronization (CCS) [36] which combines gossip paradigm with a nature-inspired approach

was proposed to cope with disturbance. The logical clock in the CCS is equal to the hardware clock plus an adjustment. This is a special case of our logical clock model which has both multiplicative and additive compensation. Joerg et al. [37] proposed a hybrid protocol for distributed microphones over a wireless network, in which integrated the two-way message exchange mechanism with gossiping technique. The idea of gossiping used therein mainly seeks a virtual master clock. Based on broadcast gossiping, Stankovic et al. [38] proposed two instrumental variable type distributed recursions for estimating parameters of calibration functions with a general noise assumption. A concern regarding distributed recursion is their excessive use of communication, while the nodes of randomly connected networks can only communicate with its immediate neighbor in an opportunistic manner.

In summary, some of the above algorithms only compensate offset or rate [29,32,34], and other algorithms compensate offset and rate separately [4,19,31,37]. Several algorithms reconstruct clock parameters and perform reverse reconstruction to align absolute time, so they usually introduce more computational complexity [4,32,37]. The distributed nature of the consensus based timing protocols is superior to other timing protocols, but the message collision rate is relatively high. In particular, node i announces a timing message when the hardware clock τ_i or logical clock T_i is such that there exists an integer Ψ satisfying $\tau_i(t) = n\Psi$ or $T_i(t) = n\Psi + \phi_i$, $n = 1, 2, 3, ..., N$. There is a high probability of a timing message collision event in algorithmic mechanisms when the hardware parameters or bias ϕ are closely for at least two unidentified and hidden nodes [11,12,18,20,23]. In addition, many algorithms [11,13–16,18–21,34] are still average-value-based algorithms, which have a slow speed of convergence. Worse still, under significant clock drifting, the average information will remove after few iterations with a slow speed of convergence. Hence, taking the above reasons into consideration, energy-efficient time synchronization for dynamic WSNs may not be achieved by these consensus based timing protocols. The major advantage of the gossip idea is that the iterative dynamics appear as randomized and asynchronous evolution, so that it is quite convenient for asynchronous network and topological changes. Nevertheless, the main drawbacks of the aforesaid gossip-based protocols have been analyzed from the compensation quantity, the excessive use of communication, etc.

Table 1. The comparison of representative distributed timing protocols

	Dynamic topology	Delay	Iterative way	Interference	Rate	Offset
MTS [12]	Yes	Yes	Max	High	Yes	Yes
RoATS [15]	No	Yes	Average	High	Yes	Yes
RPCS [19]	Yes	Yes	Average	High	Separate	Separate
RBDS [20]	Yes	No	Average	High	Incomplete	Yes
RFA [29]	No	No	-	Low	No	Yes
DCS [31]	Yes	Yes	Average	High	Separate	Separate
AOTSP [33]	No	Yes	-	Low	Yes	Yes
CCS [36]	No	Yes	Average	Low	No	Yes
Ours	Yes	Yes	Max	Low	Yes	Yes

So, taking a hybrid approach, we first proposed RGCS algorithm without considering the delays, which combines the advantage of randomized gossip algorithm and consensus based timesync. The algorithm fully adjusts rate and offset, and also fits into the dynamic topology of randomly connected WSNs with lower collision. The converge-to-max criterion was introduced to choose the coefficients properly in order to enable fast convergence. A Revised-RGCS algorithm was developed by adopting the least square based low-pass filter to counteract the impact of bounded delays. Summing up the works for comparison in this section, the features of representative distributed timing protocols are summarized in Table 1. Our time synchronization properties include:

- Rate and offset synchronization. Each clock rate and offset should be fully adjusted to achieve a common virtual clock.
- Compatible with dynamic topology changes. A time synchronization protocol has to explore the changing connectivity to design compensation rules.
- Energy-efficient. The number of message collisions should be small. Synchronization should be finished within a limited time, since the gossip consensus is a not fast converging algorithm.
- Robustness against bounded time-delays. The bounded convergence should be guaranteed even when bounded uncertain delays are present.

3. Problem Formulation

Suppose N sensor nodes of a WSN indexed by $i = 1, 2, ..., N$. Owing to the node dormancy or death and random failure of links, a successful contact between a pair of nodes i and j depends on the probability distribution of the setup of stochastic links. Thus, the topology of dynamic ad hoc WSNs is modeled as a time-varying graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$. $\mathcal{V}$ is the set of vertices, and the existence of stochastic links $\mathcal{E}(t) = \{e_{ij}(t)|i, j \in \mathcal{V}\}$ is determined independently. The set of node i's neighbors is denoted by $\mathcal{N}_i = \{j \in \mathcal{V}, e_{ij} \in \mathcal{E}(t)\}$. We use a notation (ij) to indicate e_{ij} which means node i and j happen to link together, and the undirected random graph is bidirectional.

Definition 1. A synchronization link (ij) or e_{ij} means that an arbitrary random node i is coupled with its geographical neighbor $j \in \mathcal{N}_i$ once to perform synchronous operation. Specifically, triggering node i sends a beacon to choose the triggered neighbor j and confirm the ID of the synchronization link e_{ij}, then they swap timing messages with each other. Due to the randomized activation of nodes, we need neighbor discovery protocol for nodes to discover each other when they are neighbors. Fortunately, a simple and useful neighbor discovery protocol can be used as in [39].

Assumption 1. When node $i(i \in \mathcal{V})$ is activated at time t, the cardinality $|\mathcal{N}_i(t)| \geq 1$.

Assumption 2. The delays of the lth uplink and downlink communication between node i and j is denoted by d_{ij} and d_{ji}. They are unequal and bounded by upper bound D; that is, $0 < d_{ij} < D$, $d_{ij} \neq d_{ji}$.

The oscillator generates the standard unit of frequency Ω. The clock provides its reading $\tau(t)$ of the elapsed absolute time t by accumulating the number of impulsive signals $\tau(t) = \kappa \int_{t_0}^{t} \Omega(\epsilon) d\epsilon + \tau(t_0)$, where κ is a oscillator frequency correction. The oscillation frequency is nearly invariable in a short time horizon, so the clock reading of the ith node possesses linear dynamic behavior such as

$$\tau_i(t) = a_i t + b_i, \tag{1}$$

where a_i, b_i is the hardware clock rate and offset, respectively. a_i determines the timing rate of clock and b_i represents the difference of nodes' clock reading. The logical clock is a linear affine function of $\tau_i(t)$. $\alpha_i(t)$ is a multiplicative compensation and $\beta_i(t)$ is an additive compensation, as follows

$$T_i(t) = \alpha_i(t)\tau_i(t) + \beta_i(t) = \tilde{\alpha}_i(t)t + \tilde{\beta}_i(t), \tag{2}$$

where $\tilde{\alpha}_i(t) = a_i\alpha_i(t)$, $\tilde{\beta}_i(t) = \alpha_i(t)b_i + \beta_i(t)$ are called the logical rate and offset, respectively. So, we have available quantity $\tau_i(t)$, and two variables $\alpha_i(t)$, $\beta_i(t)$ to be designed.

The objective is to design a randomized communication protocol and a gossip-consensus-based approach for coupled nodes $i, j \in \mathcal{V}$ announcing its timing messages and adjusting the logical rate and offset, such that the logical clocks $T_i(i \in \mathcal{V})$ is synchronized in the probability sense, as follows

$$\forall i, j \in \mathcal{V}, \exists \Pr\{\lim_{t\to\infty}[T_i(t) - T_j(t)] = 0\} = 1. \tag{3}$$

Finally, notation $\Pr, o, \mathcal{O}$ denote probability, infinitesimal of higher order and infinitesimal of the same order, respectively.

4. RGCS, Multi-RGCS, and Revised-RGCS Algorithms

4.1. RGCS Algorithm

4.1.1. Randomized Communication Protocol

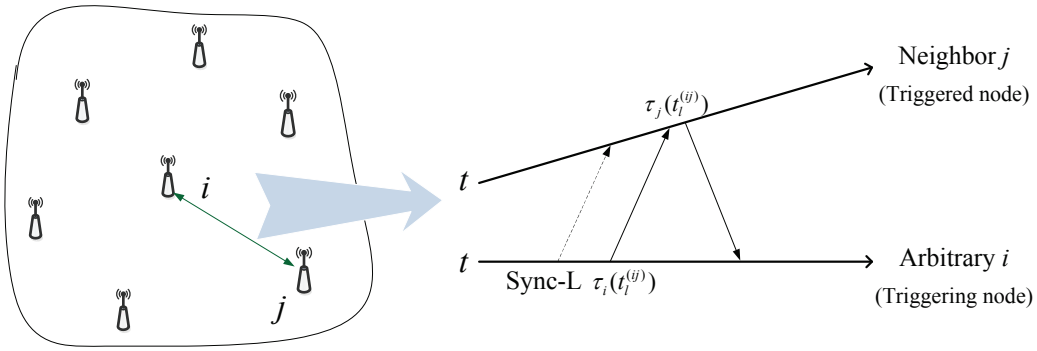

Figure 1. The Sync-L beacon and timing message exchanges in a dynamic wireless sensor network.

A randomly connected ad hoc WSN is specifically characterized in that the nodes are randomly activating and sleeping, and the RGCS algorithm is supposed to run independently in each individual node. Firstly, the Poisson process is configured to each node in order to generate the Synchronization Links (Sync-L) beacon which is sent to a node of its neighborhood by exploiting the broadcast nature of wireless communications. Thus, the randomized activation of $e_{ij}(i, j \in \mathcal{V})$ satisfies Poisson process with the constant intensity λ_{ij} such that $\Pr\{t^{(ij)}(l+\sigma) - t^{(ij)}(\sigma) = n\} = (\lambda_{ij}l)^n/n!e^{\lambda_{ij}l}, n = 0, 1...$, for $\forall l, \sigma \geq 0$. Some special circumstances are as follows, if $\lambda_{ij} = 0$, it means link e_{ij} is not activated forever; if $\lambda_{ij} = \infty$, it means link e_{ij} is activated infinitely in a given time interval. The Sync-L beacon between node i and j is confirmed, which is referred to the stochastic link activation event. Once the randomized activation of link e_{ij} takes place, by using the Media Access Control (MAC) layer time stamps, triggering node i pushes a multivariable message $[\alpha_i(t_l^{(ij)}), \beta_i(t_l^{(ij)}), \tau_i(t_l^{(ij)})]$ to node j. After that, the triggered node j pulls a multivariable message $[\alpha_j(t_l^{(ij)}), \beta_j(t_l^{(ij)}), \tau_j(t_l^{(ij)})]$ to node i. The overall procedure is illustrated in Figure 1.

Thus, accordingly, the set of activated synchronization links $\{\bigcup_{i=1,j=1}^{N} e_{ij}(t)\}$ are an identically and independently distributed (i.i.d) Poisson distribution. Suppose that function $\mathcal{F}(N)$ represents the amount of synchronization links of an underlying graph. When it is a complete graph, we can maximize $\mathcal{F}(N)$ as $N(N-1)/2$. Clearly, the set $\{\bigcup_{l=0}^{\infty}\bigcup_{i=1,j=1}^{N} t_l^{(ij)} | e_{ij} \in \mathcal{E}(t)\}$ satisfies global Poisson distribution with the intensity $\mathcal{F}(N)\lambda$, because all intensities have the same value. Each global gossip instant corresponds to an updating event. We denote $\Delta(l,(ij))$ as the inter-time between consecutive gossip instants for global Poisson process, and $\{\Delta(l,(ij))\}_{l\in N,(ij)\in\mathcal{E}(t)}$ is an i.i.d process with moments

$$\begin{cases} E[\Delta(l,(ij)] = \frac{1}{\mathcal{F}(N)\lambda}. \\ E[\Delta^2(l,(ij)] = \frac{2}{(\mathcal{F}(N)\lambda)^2}. \end{cases} \tag{4}$$

The intensity λ increased, whereas $\Delta(l,(ij))$ continuously decreased. The physical significance of λ_{ij} represents the intermittent of activation events of link e_{ij}, and the mathematical meaning of λ_{ij} represents the mean value of the occurrence of the activation events per unit time. In many scenarios, environmental-temperature variations will be slow to change around nodes. Hence, we can do that by increasing intensity λ_{ij} to enlarge the frequency of a link activation event in order to work against slowly changing drifts.

As shown in Figure 2, we illustrate how the proposed mechanism works under the hidden node problem. Node A is visible from node B, and node C is visible from node B too. However, node A and node B cannot sense with each other, because they are out of communication zone of each other. In the deterministic communication protocol (such as ATS, MTS, RoATS, etc.), each node transmits its timing messages periodically with a common period based on its own clock on chip. If node A and B have small difference in term of the hardware rate and offset, they will announce timing messages to node C simultaneously. Thus, the collisions will occur endlessly. In our proposed mechanism, each node announces a timing message asynchronously based on its pre-programmed Poisson process, and random pairwise nodes delivery timing messages in the uplink and downlink successively. So node A randomly gossip, then a sync timeslot is established between node A and C with probability $\Pr(t_l^{AC})$. The sync timeslot between node B and C is established with probability $\Pr(t_l^{BC})$. The Poisson intervals δ_{AC}, δ_{BC} are independent of each other. If the transmission delay is negligible, the width of the sync timeslot can be narrow such that the probability of collisions will further down. Hence, with an appropriate λ, the probability of collisions is lower than that in the deterministic communication.

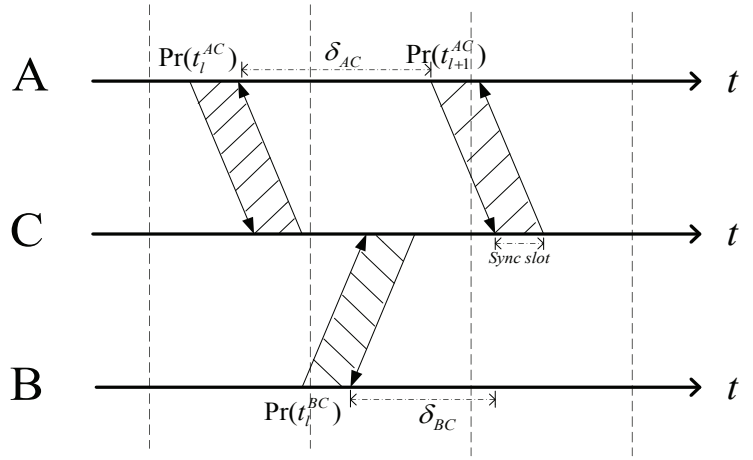

Figure 2. Illustration of the proposed randomized communication under the hidden node scene.

To defend against message manipulation attacks, secure consensus timing protocols should contain detecting and excluding outliers mechanism for the logical clock checking and hardware clock checking. From the analysis of logical clock checking mechanism in literature [22], we know that the bounded communication cycle which determined by $\{\Psi | \tau_i(t) = N\Psi, N = 1, 2, ..., \Psi > 0\}$ is a pivotal parameter of the secure protocols. However, the authors suppose Ψ is invariable under malicious modification with the purpose of preventing the attack nodes using outdated receiving information to cheat, i.e., it makes nodes to collect messages from neighboring nodes within a constant duration. The proposed gossip interaction mode which has less conservatism to topology changes can remove this limitation properly. Also, Ψ is the key parameter which contributes to represent the conforming relationship of Definition 1 in literature [23] to defend against sybil attack, and RGCS is robust to the distortion of bounded communication cycle.

4.1.2. Gossip Consensus Approach for Clock Rate and Offset Compensation

Let $\Delta S_i^{(ij)}$ be the difference of hardware clock readings τ_i of triggering node i in gossip instant $t_l^{(ij)}$ and $t_{l-1}^{(ij)}$, namely $\Delta S_i^{(ij)} = \tau_i(t_l^{(ij)}) - \tau_i(t_{l-1}^{(ij)})$. The main idea of the updating rule is to utilize the gossip consensus approach based on the interchange of the gossiping neighbors' messages. Triggered node j updates its logical rate $\tilde{\alpha}_j(t)$ and offset $\tilde{\beta}_j(t)$ by averaging them with the estimates of its gossiping neighbors, namely $\tilde{\alpha}_j(t_{l+1}^{(ij)}) = (1-\rho_1)\tilde{\alpha}_j(\tilde{t}_k^j) + \rho_1\tilde{\alpha}_i(t_l^{(ij)})$. Each node consists of estimating the logical rate with respect to virtual consensus rate, and local node j stores a new compensation α_j. Then by dividing a_j at both sides of above equation, we have

$$\alpha_j(t_{l+1}^{(ij)}) = (1-\rho_1)\alpha_j(\tilde{t}_k^j) + \rho_1(a_i/a_j)\alpha_i(t_l^{(ij)}). \tag{5}$$

where $\rho_1 \in (0,1]$. The constant ρ_1 defines the degree of change of compensation α in gossip instant $t_{l+1}^{(ij)}$. If ρ_1 is set to a small value, the average information will remove in few updates. Additionally, due to the discrete observations of the relative drift, we can increase ρ_1 to guarantee stationarity of the estimated α. Based on the above analysis, the converge-to-max criterion is applied to $\tilde{t}_k^j$. The time, $\tilde{t}_k^j$ indicates the previous gossip instant of arbitrary edges who contain node j just before activated edge e_{ij}'s lth gossip instant. Equation (5) counts the absolute time on two scales, i.e., updating usage round and communication round, to avoid confusion between the updating iteration and the gossiping communication. Scrupulously, $\tilde{t}_k^j$ is given as follows

$$\tilde{t}_k^j = max\{t_s^{(jp)}, t_r^{(qj)} | t_s^{(jp)} < t_l^{(ij)}, t_r^{(qj)} < t_l^{(ij)}; p,q = 1,...,n; s,r = 1,2,...\}. \tag{6}$$

When the Sync-L e_{ij} is activated in the next gossip instant $t_l^{(ij)}$, node i use $\tau_i(t_{l-1}^{(ij)})$ which be stored in flash memory of sensors to compute the number of drift, namely the relative drift $\alpha_{ji}(t_l^{(ij)})$ for node j is computed by

$$a_i/a_j = \alpha_{ji}(t_l^{(ij)}) = \Delta S_i^{(ij)}/\Delta S_j^{(ij)}. \tag{7}$$

During all iterations, each node i's storage is $O(|\mathcal{N}_i|)$ for all possible synchronization links. It implies that, even though the size of the network node increases, the storage complexities of the RGCS per node per iteration will not grow. This property ensures the scalability of the algorithm.

Substituting equation (7) into equation (5) yields the update equation of compensating α_j. Again, based on the converge-to-max criterion, we design the rate compensation iteration rule of random pairwise nodes. In this rule, the rate compensations of triggering node i, triggered node j, and silent nodes evolve as follows

$$\begin{cases} \alpha_j(t_{l+1}^{(ij)}) = max\{\alpha_j(\tilde{t}_k^j), \alpha_{ji}(t_l^{(ij)})\alpha_i(t_l^{(ij)})\}, \\ \alpha_i(t_{l+1}^{(ij)}) = max\{\alpha_i(\tilde{t}_k^i), \alpha_{ij}(t_l^{(ij)})\alpha_j(t_l^{(ij)})\}, \\ \alpha_{silent}(t_{l+1}^{(ij)}) = \alpha_{silent}(\tilde{t}_k^{silent}), \end{cases} \tag{8}$$

which means that random pairwise nodes i and j reach maximum logical clock together at a random gossip instant $t_{l+1}^{(ij)}$, and other nodes are silent. In equation (8), node i and j sequentially iterate its own estimated logical clock only once per gossiping interaction. Observing that whether $\tilde{t}_k^i$ and $\tilde{t}_k^j$ will be different or same depends on the contiguous activation of the Sync-L of global Poisson process.

After the rate compensation is applied, the random pairwise nodes compute the instantaneous estimated logical clock difference $T_i(t_l^{(ij)}) - T_j(t_l^{(ij)})$ and try to adjust its offset β in order to reduce the difference. Once again, the gossip consensus approach and the converge-to-max criterion are applied to local node i, j, and silent nodes for attaining offset compensations, as follows

$$\begin{cases} \beta_j(t_{l+1}^{(ij)}) = \beta_j(\tilde{t}_k^j) + \rho_2(T_i(t_l^{(ij)}) - T_j(t_l^{(ij)})), \\ \beta_i(t_{l+1}^{(ij)}) = \beta_i(\tilde{t}_k^i) + \rho_2(T_j(t_l^{(ij)}) - T_i(t_l^{(ij)})), \\ \beta_{silent}(t_{l+1}^{(ij)}) = \beta_{silent}(\tilde{t}_k^{silent}), \end{cases} \tag{9}$$

where $\rho_2 = 1$ and $T_i(t_l^{(ij)}) = \alpha_i(t_l^{(ij)})\tau_i(t_l^{(ij)}) + \beta_i(t_l^{(ij)})$. The flowchart of RGCS algorithm is shown in Figure 3(a), and the basic procedures are described as follows. Step 1. Randomly initialize each local clock τ_i. Step 2. Judge whether arbitrary node i activate its a neighbor or not, or is activated by its a neighbor or not. Step 3. Perform update equations (6), (7), (8), and (9). Step 4. Judge whether to meet the sync accuracy or not. Otherwise, return to Step 2.

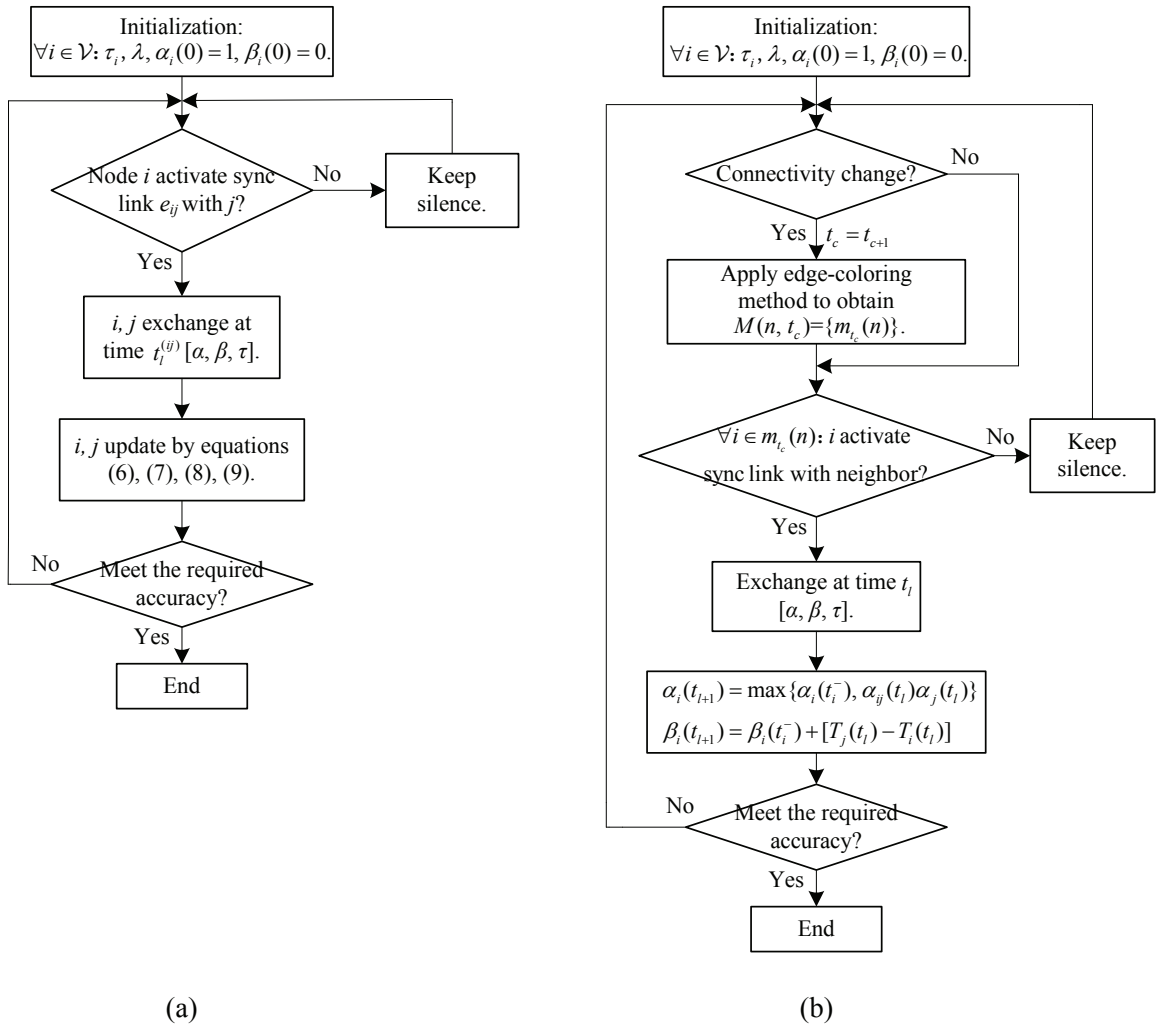

Figure 3. Flowchart of the proposed algorithms. (a) Randomized Gossip-Consensus-based time Synchronization (RGCS). (b) Multi-RGCS.

4.1.3. Convergence of RGCS

Substituting equation (2) into (3) yields an intuitive translation of the time synchronization issue:

$$\begin{cases} \Pr\{\lim_{t\to\infty} \tilde{\alpha}_i(t) = a_v\} = 1, \forall i \in \mathcal{V}, \\ \Pr\{\lim_{t\to\infty} \tilde{\beta}_i(t) = b_v\} = 1, \forall i \in \mathcal{V}, \\ \Pr[\tilde{\alpha}_i(t) - \tilde{\alpha}_j(t)] = o(1/t), \forall i, j \in \mathcal{V}, \end{cases} \tag{10}$$

where a_v and b_v are the parameters of consistent virtual clock. Equations (10) show that whether the clocks achieve synchronization or not depends not only on the convergence of logical rate and offset, but also on the convergence speed of the logical rate errors $\Pr[\tilde{\alpha}_i(t) - \tilde{\alpha}_j(t)]$, because errors will tend to diverge if the convergence speed to zero of $\Pr[\tilde{\alpha}_i(t) - \tilde{\alpha}_j(t)]$ is slower than $1/t$.

Theorem 1. Consider the rate and offset update equations given by Equation (6), (7), (8), (9), and there exist two variables $\Delta > 0$ and $l > 0$ such that the union graph $G(t_l^{(ij)}, \Delta(l, (ij))) = \bigcup_{ij} \bigcup_l \mathcal{G}(t_l^{(ij)})$ for $\forall e_{ij} \in \mathcal{E}(t)$ is connected with probability one. Then, equations (10) hold true.

Proof. Firstly, two function $\mathcal{V}_v(t)$ and $|\mathcal{V}_v(t)|$ are introduced, where $\mathcal{V}_v(t)$ is the set of nodes whose logical rate and offset are equal to a_v and b_v in instant t, and $|\mathcal{V}_v(t)|$ is the cardinality of set $\mathcal{V}_v(t)$.

Define $a_{max} = max\{a_i\}, \forall i \in \mathcal{V}$, let node v be the node whose clock rate is equal to a_{max}. According to the initial sets, there is at least one node in network whose logic clock rate and offset equal to a_v and b_v in initial instant, i.e., $\mathcal{V}_v(t) \neq \oslash$, $|\mathcal{V}_v(t)| \geq 1$. With loss of generality, assume node j has larger logical rate. If the link e_{ij} is activated at gossip time $t_l^{(ij)}$, node i will update its logical clock such that $a_i\alpha_i = a_j\alpha_j, \alpha_i b_i + \beta_i = \alpha_j b_j + \beta_j$. Thus, the logical rate of each node is less than or equal to a_{max} during the iteration of RGCS. For a pair of nodes i and j, $i, j \in \mathcal{V}_v$, there has $a_i\alpha_i = a_{max}$, $\alpha_i b_i + \beta_i = \beta_v$ and $a_j\alpha_j = a_{max}$, $\alpha_j b_j + \beta_i = \beta_v$. Thus, we can infer that $a_i\alpha_i$ or $a_j\alpha_j \geq a_k\alpha_k$ holds for $\forall k \in \mathcal{V}$. Therefore, a pair of nodes i and j in $\mathcal{V}_v$ will no longer update its logical clock and maintain its logical rate and offset during the latter iterations, which means that $|\mathcal{V}_v(t)|$ is nondecreasing.

The complement of a set $\mathcal{V}_v$ is defined as $\mathcal{V} - \mathcal{V}_v$, which means that the set of nodes are not in $\mathcal{V}_v$. If node k is not in the set $\mathcal{V}_v$, then node k is in $\mathcal{V} - \mathcal{V}_v$. Since all nodes' logical rate is less than or equal to a_{max} during the iteration of RGCS, so we have $a_k\alpha_k < a_v$ for $\forall k \in \mathcal{V} - \mathcal{V}_v$. Thus, if node $i \in \mathcal{V}_v$ is coupled with node k by Sync-L, then it follows form equation (8) and (9) that node k will update its logical clock such that $a_k\alpha_k = a_v$, $\alpha_k b_k + \beta_k = \beta_v$. Then, one obtains that $|\mathcal{V}_v(t^+)| = |\mathcal{V}_v(t)| + 1$ and $|\mathcal{V} - \mathcal{V}_v(t^+)| = |\mathcal{V} - \mathcal{V}_v(t)| - 1$, where t^+ is the finish time of the iteration. Hence, $|\mathcal{V}_v(t)|$ will strictly increase when node $i \in \mathcal{V}_v$ is coupled with node $j \in \mathcal{V} - \mathcal{V}_v$.

If $|\mathcal{V}_v(t)| = N$ in instant t, it implies that $\Pr\{\lim_{t\to\infty} \tilde{\alpha}_i(t) = a^c\} = 1$, $\Pr\{\lim_{t\to\infty} \tilde{\beta}_i(t) = b^c\} = 1$, for $\forall i \in \mathcal{V}$. Otherwise, we have $|\mathcal{V}_v(t)| < N$. Since graph $G(t_l^{(ij)}, \Delta(l, (ij)))$ is jointly connected with probability one, there is at least one link e_{ik} for $i \in \mathcal{V}_v$ and $k \in \mathcal{V} - \mathcal{V}_v$, which means that there has $\lambda_{ik} > 0$. The probability of the event that link e_{ik} is activated satisfies $1 - e^{-\lambda_{ik}(t^{'} - t)}$. Therefore, we have the probability of $|\mathcal{V}_v(t)| \geq |\mathcal{V}_v(t)| + 1$ being equal to $1 - e^{-\lambda_{ik}(t^{'} - t)}$. Thus, $|\mathcal{V}_v(t)|$ will strictly increase with probability one when $t \to \infty$. Hence, we have $\Pr\{\lim_{t\to\infty} |\mathcal{V}_v(t)| = n\} = 1$ which yields $\Pr\{\lim_{t\to\infty} \tilde{\alpha}_i(t) = a^c\} = 1$, $\Pr\{\lim_{t\to\infty} \tilde{\beta}_i(t) = b^c\} = 1$, for $\forall i \in \mathcal{V}$. Then, $\Pr[a_i\alpha_i(t) - a_j\alpha_j(t)]$ can approximate to zero when $t \to \infty$. Clearly, $\lim_{t\to\infty} \Pr[a_i\alpha_i(t) - a_j\alpha_j(t)]/(1/t) = 0$, $\Pr[\tilde{\alpha}_i(t) - \tilde{\alpha}_j(t)] = o(1/t)$ thus satisfied. The Theorem 1 thus proved.

In practice, considering the thrift energy to bigger extent, finite-time convergence of algorithms is very important in WSNs. Next, we will give the lower bound of the probability for the finite-time convergence of RGCS. For node i, j, let's assume W_{ij} represents the time cost for a link activation event. Then u_{ij}, $U_{ij} = \int_0^t u_{ij}(s)ds$ is the probability density function and probability distribution function of W_{ij}, respectively.

Theorem 2. Suppose the union graph $G(t_l^{(ij)}, \Delta(l, (ij))) = \bigcup_{ij} \bigcup_l \mathcal{G}(t_l^{(ij)})$ for $\forall e_{ij} \in \mathcal{E}(t)$ is connected with probability one. Then, we have $\Pr\{T_i(t) = \tau_v(t), i \in \mathcal{V}\} \geq \prod_{i\in\mathcal{V}} U_{vi}(t)$.

Proof. From the proof of Theorem 1, we can conclude that node i in $\mathcal{V} - \mathcal{V}_v$ will become a node in $\mathcal{V}_v$ after the link e_{vi} is activated. When $T_i(t) = \tau_v(t)$, there has $\Pr\{T_i(t) = \tau_v(t)\} = U_{vi}(t), i \in \mathcal{V}$. $\prod_{i\in\mathcal{V}} \Pr\{T_i(t) = \tau_v(t)\} \leq \Pr\{T_i(t) = \tau_v(t), i \in \mathcal{V}\}$ which yields $\Pr\{T_i(t) = \tau_v(t), i \in \mathcal{V}\} \geq \prod_{i\in\mathcal{V}} U_{vi}(t)$. The Theorem 2 thus proved.

4.2. Multi-RGCS Algorithm

Obviously, RGCS is obtained by the single-gossiping rule. Under this scenario, other silent nodes are situated in wait state, while only one synchronization link is activated in a timeslot. We expect that more coupling nodes are able to exchange and update states in a timeslot, or more than a neighboring node can overhear triggering node's messages (i.e. broadcast gossip manner). Based on their positive effect on the performance of the convergence rate, the Multi-gossiping version of RGCS (Multi-RGCS) is topic which is worth exploring. In fact, we can assume that more than two nodes $(i, j, ...)$ wake up in t_l simultaneously with probability $P(t_l) > 0$. So the single-gossiping version corresponds to specific implementation of above assumption in which if and only if two nodes i and j wake up.

Different from the foregoing single-gossiping, Multi-RGCS requires an additional precondition that the synchronization links can not have public vertex in the same slot. In other words, we need

to seek a multi-gossip sequence $M(n, t_c) = \{m_{t_c}(n)\}$, n is the color number, t_c is the switching time. So, the synchronization links are dyed different colors, and those links who have the same color are categorized as the group $m_{t_c}(n)$. This problem could be investigated by the idea of the Edge-Coloring (EC) [40]. The EC aims at an assignment of one color to each synchronization link such that no two synchronization links on the same node are assigned the same color. Here, we give a straightforward EC algorithm to realize Multi-RGCS as in Figure 3(b). Thereinto, the key procedures of obtaining the multi-gossip sequence are summarized as follows. For t_c do Step 1. Generate a spanning tree $\mathcal{H}$ of graph $\mathcal{G}(t_c)$; Step 2. Randomly pick a single edge in $\mathcal{H}$, find a non-adjacent edge to the former, find a nonstaining edge for coloring the third color, until it can not find any nonstaining edge. Step 3. The edges of different colors constitute the multi-gossip sequence $M(n, t_c) = \{m_{t_c}(1), m_{t_c}(2), ..., m_{t_c}(n)\}$. The probability characteristics of the set $\{t_l, l = 1, 2, ...\}$ is similar to the set $\{\bigcup_{l=0}^{\infty} \bigcup_{i=1,j=1}^{N} t_l^{(ij)}\}$, and the physical meaning of t_i^- refers to the previous gossip instant of any synchronization link involving node i. In Multi-RGCS, several nodes are involved in multi synchronization links $m_{t_c}(n)$ to increase the synchronization traffic and enhance the convergence speed.

4.3. Revised-RGCS Algorithm

In RGCS, we suppose that the delays is negligible. However, by considering various disturbance, which act as additive noise model (such as the delay), the actual timestamp is truly noisy, and the delay distribution is unknown. To deal with this issue, we proposed a least square estimation based low-pass filter against bounded delays (Assumption 2). As shown in Figure 4, the time-delay of the lth communication between node i and j is denoted by d_{ij} ($i \to j$, namely d_{ij} influences $\Delta S_j^{(ij)}$). d_{ij} is bounded by upper bound D; that is, $0 < d_{ij} < D$. d_{ji} ($j \to i$, namely d_{ji} influences $\Delta S_i^{(ij)}$) is the same definition as d_{ij}, but may be unequal.

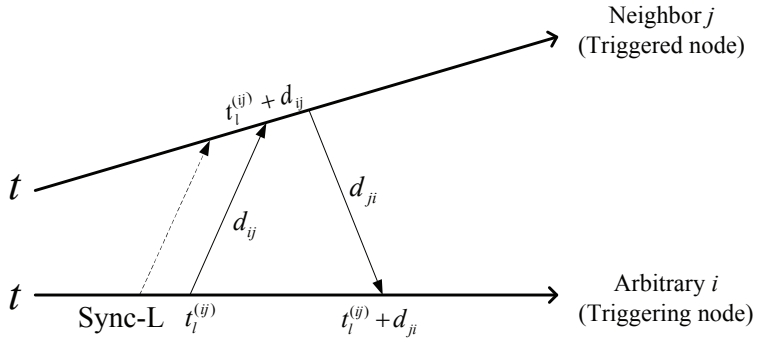

Figure 4. Illustration of gossip-consensus-based time synchronization with bounded delays.

Due to the symmetry, the following process takes the case of d_{ij}. Based on the definition of the relative drift and d_{ij}, we have a longspan estimator $\hat{\alpha}_{ij}(l)$ for $\alpha_{ij}(l)$ as follows

$$\begin{aligned} \hat{\alpha}_{ij}(l) &= \frac{\tau_j(t_l^{(ij)} + d_{ij}(l)) - \tau_j(t_0^{(ij)} + d_{ij}(0))}{\tau_i(t_l^{(ij)}) - \tau_i(t_0^{(ij)})} \\ &= \frac{a_j(\sum^l \delta_{ij}(l) + d_{ij}(l) - d_{ij}(0))}{a_i \sum^l \delta_{ij}} \\ &= \alpha_{ij}(1 + \hat{\xi}(l)), \end{aligned} \tag{11}$$

where $\hat{\xi}(l) = \frac{d_{ij}(l) - d_{ij}(0)}{\sum^l \delta_{ij}}$, δ_{ij} is the inter-time between consecutive gossip instants for link e_{ij}. Since d_{ij} is bounded by D, we have $|\hat{\xi}(l)| \leq \frac{D}{\sum^l \delta_{ij}}$. Hence, we know $\hat{\xi}(l)$ decays in a rate of $\mathcal{O}(1/\delta)$.

Let $\hat{\alpha}^*_{ij}(l)$ be the optimal estimator for $\hat{\alpha}_{ij}(l)$, and denote $e_l = \hat{\alpha}_{ij}(l)\tau_i(t_l^{(ij)}) - \hat{\alpha}^*_{ij}(l)\tau_i(t_l^{(ij)})$. A least square principle as $J = \sum_{l=1}^{l} e_l^2$ is employed. For calculating the recursion of $\hat{\alpha}^*_{ij}(l)$, we apply a partial derivative for $\hat{\alpha}^*_{ij}(l)$, namely

$$\frac{\partial J}{\partial \hat{\alpha}^*_{ij}(l)} = 0, \tag{12}$$

which yields a optimal estimator as follow

$$\hat{\alpha}^*_{ij}(l) = (1-\gamma^*(l))\hat{\alpha}^*_{ij}(l-1) + \gamma^*(l)\frac{\tau_j(t_l^{(ij)}) - \tau_j(t_0^{(ij)})}{\tau_i(t_l^{(ij)}) - \tau_i(t_0^{(ij)})}, \tag{13}$$

where the weighting parameter $\gamma^*(l) = \frac{\delta^2(1)+\delta^2(2)+...+\delta^2(l)}{\delta^2(1)+(\delta^2(1)+\delta^2(2))+...+(\delta^2(1)+\delta^2(2)+...+\delta^2(l))}$ and $\hat{\alpha}^*_{ij}(0) = 1$. Substituting equation (11) into (13) yields

$$\hat{\alpha}^*_{ij}(l) = \alpha_{ij}(1+\hat{\xi}^*(l)), \tag{14}$$

where $\hat{\xi}^*(l) = \frac{\delta(1)+(\delta(1)+\delta(2))+...+(\delta(1)+\delta(2)+...+\delta(l))(d_{ij}(l)-d_{ij}(0))}{\delta(1)+(\delta(1)+\delta(2))^2+...+(\delta(1)+\delta(2)+...+\delta(l))^2}$. Then by bounded d_{ij}, and we notice that, the denominator of $\hat{\xi}^*(l)$ is δ's quadratic term after accumulating, and the numerator of $\hat{\xi}^*(l)$ is δ's first degree term. It implies that the decay rate of $\hat{\xi}^*(l)$ is also $\mathcal{O}(1/\delta)$. This decay rate ensures that the proposed least square estimation based low-pass filter avoids the divergence condition of $\Pr[\tilde{\alpha}_i(t) - \tilde{\alpha}_j(t)]$. Hence, we can utilize $\frac{\hat{\alpha}^*_{ij}(l)}{(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{1/2}}$ to replace $\alpha_{ij}(t_l^{ij})$ and modify the rate updating rule of triggering node i of equations (6), there is

$$\alpha_i(t_{l+1}^{(ij)}) = (1 - \frac{\rho_1}{(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{1/2}})\alpha_i(\tilde{t}_k^i) + \frac{\rho_1\hat{\alpha}^*_{ij}(l)\alpha_j(t_l^{(ij)})}{(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{1/2}}. \tag{15}$$

The criterion for selecting the appropriate weighting parameter $(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{-1/2}$ is based on a piece-wise constant function, and it is chosen to be a decreasing factor, which contributes to restraining the negative effect of additive noise in stochastic approximation. Actually, the weighting parameter in this paper is a special case of the standard conditions in stochastic approximation methods: $\Sigma_l^{\infty}(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{-1/2} = \infty$ and $\Sigma_l^{\infty}(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{-1} < \infty$. Since $\alpha_{ij}(t_l^{(ij)})$ is an estimation of inverse relative drift $\alpha_{ji}(t_l^{(ij)})$, the rate updating rule of triggered node j is

$$\alpha_j(t_{l+1}^{(ij)}) = (1 - \frac{1-\rho_1}{(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{1/2}})\alpha_j(\tilde{t}_k^j) + \frac{(1-\rho_1)\alpha_i(t_l^{(ij)})}{\hat{\alpha}^*_{ij}(l)(\delta(1)+\delta(1)+\delta(2)+...+\delta(l))^{1/2}}. \tag{16}$$

Then, the clock offset compensations of local node i, j is updated as follows

$$\beta_i(t_{l+1}^{(ij)}) = \beta_i(\tilde{t}_k^i) + \rho_2(T_j(t_l^{(ij)}) - T_i(t_l^{(ij)} + d_{ji}(l))). \tag{17}$$

$$\beta_j(t_{l+1}^{(ij)}) = \beta_j(\tilde{t}_k^j) + \rho_2(T_i(t_l^{(ij)}) - T_j(t_l^{(ij)} + d_{ij}(l))). \tag{18}$$

The delays also have impact on $\tilde{t}_k^i$. The issue is caused by the fact that node j would not instantly receive its gossiping neighbor's states due to uncertain delay, and if delay d_{ij} satisfy following two conditions: (i) delay d_{ij} greater than inter-time Δ; (ii) node j joins next gossip averaging; $\tilde{t}_k^i$ will change and need to be modified. When $d_{(ij)} \neq d_{(ji)}$, we call asymmetric gossip, and that is a common case.

To clearly explain how delay influence the local updating equations, we might as well suppose link e_{ij} is activating. After Δ, there are three possibilities of adjacent Sync-L: $e_{ik}, e_{jh}, e_{kh}, \forall k, h \in \mathcal{V}$, if:

Case 1, $d_{ij} \geq \Delta(l,(ik)) + n\Delta(l,(kh)), n = 0,1,...$, then equations (16), (18) utilize $\hat{\alpha}^*_{ik}(l)$ and $\tilde{t}^i_k = max\{t^{(ik)}_s, t^{(ki)}_r | s,r = 1,2,...\}$. Case 2, $d_{ji} \geq \Delta(l,(jh)) + n\Delta(l,(kh)), n = 0,1,...$, then equations (15), (17) utilize $\hat{\alpha}^*_{jh}(l)$ and $\tilde{t}^j_h = max\{t^{(jh)}_s, t^{(hj)}_r | s,r = 1,2,...\}$.

The discriminant of Case 1 and Case 2 is easy to implement, because the comparison between the length of delay and the interval time of node activation depends on the local nodes themselves. The delays make Revised-RGCS slightly complex, but it will not damage the convergence. However, we should point out that the update of the logical rate is based on the longspan neighboring states $\tau_j(t^{(ij)}_l) - \tau_j(t^{(ij)}_0)$, so the nodes should set up a cached memory for forming buffer queue of the timing messages. Summarize the above process, a flow diagram of the Revised-RGCS algorithm is described in Figure 5.

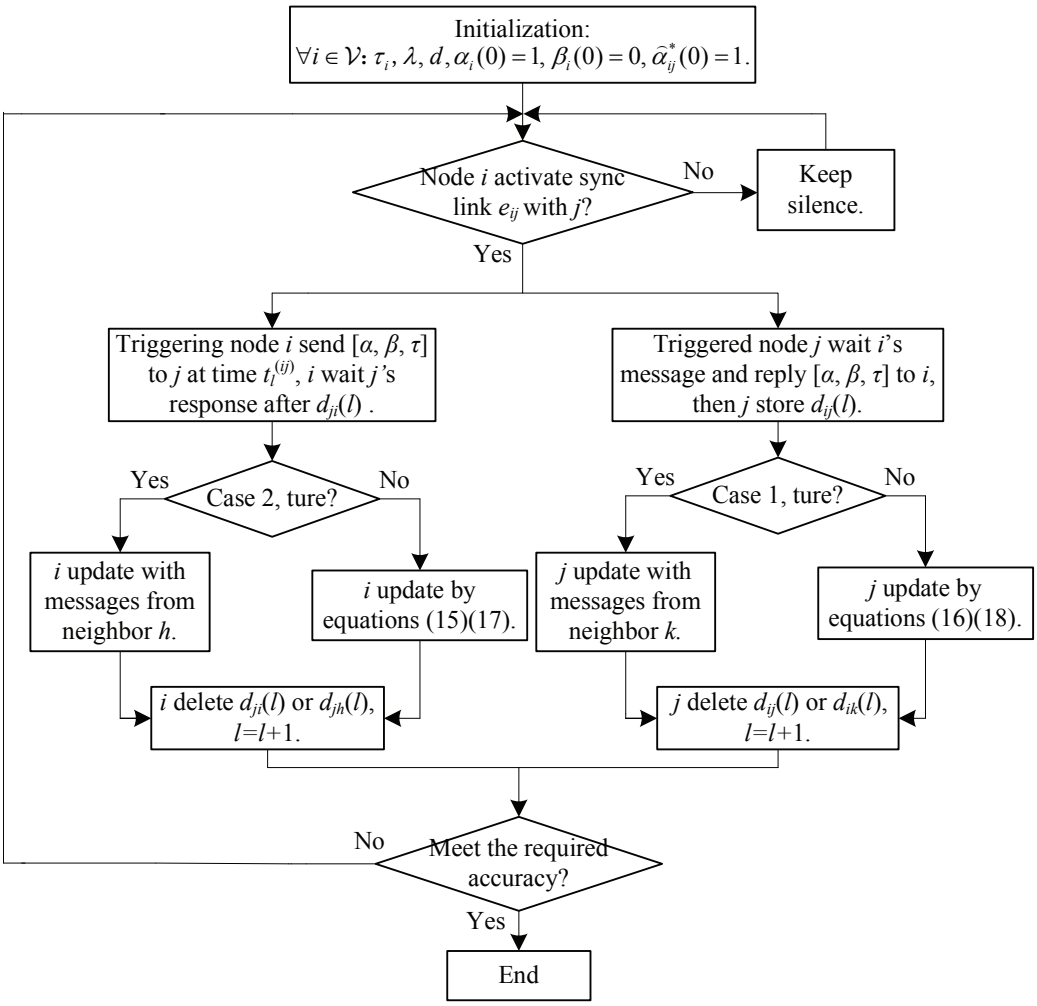

Figure 5. Flowchart of Revised-RGCS algorithm.

5. Simulation Studies

To verify the superiority of novel proposed algorithms, we carried out comparative simulations for RGCS and Multi-RGCS with delay-free case in the Matlab R2010b. Then, using the OMNeT++ 5.1 simulator, a more realistic simulator for WSNs, we verified the Revised-RGCS algorithm with delays.

5.1. Delay-Free Case

As shown in Table 1, DCS and RPCS are two appropriate reference algorithms, so we compared the performances of RGCS with them. All of them have been investigated numerically for emulating a randomly connected ad hoc WSN. The important parameter setting is described as follows. The local hardware rate is chosen from [0.999, 1.0001], and the local offset is chosen from [0, 0.002]. $\alpha_i(0) = 1$ and $\beta_i(0) = 0$. The number of nodes is 9, and the number of local nodes that can be accessed by other nodes is 3. The initial synchronization period is 10 s. The threshold of successful Sync-L is characterize by a Poisson process with intensity λ, and the intensity reflects the connection strength between individual nodes. The average number of Sync-L per time interval is 10. It is feasible that the joint topologies of simulation model of dynamic WSN are connected with probability one. The time unit is one second. The required accuracy is ± 1 millisecond. Because there are no reference nodes and random initial parameters, we employed a metric to evaluate the synchronization error of the algorithms: the maximum difference of the parameters between any two nodes in overall WSN [18–20], i.e. $\eta_{95\%}(t) \geq 95\%$, where $\eta(t) = \frac{\max\limits_{i,j}|X_i(0)-X_j(0)| - \max\limits_{i,j}|X_i(t)-X_j(t)|}{\max\limits_{i,j}|X_i(0)-X_j(0)|} \cdot 100\%$, for $\forall i, j \in \mathcal{V}$.

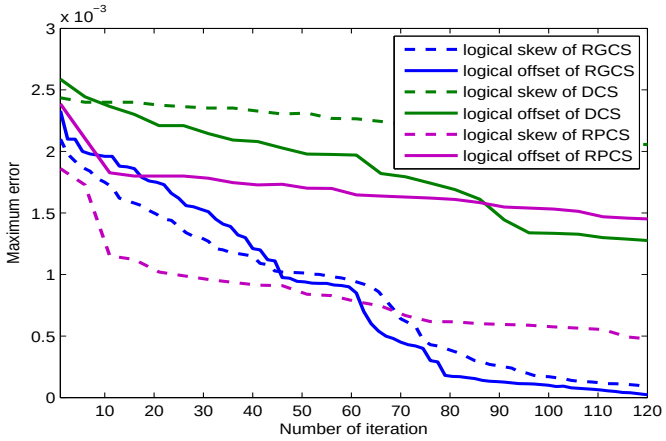

Figure 6. Comparison of maximum error of rate $\tilde{\alpha}_i(t)$ and offset $\tilde{\beta}_i(t)$ between RGCS algorithm and the reference algorithms.

Figure 6 shows the maximum error of logical rate $\tilde{\alpha}_i(t)$ and offset $\tilde{\beta}_i(t)$ under RGCS and DCS. In our algorithm, logical rate and offset synchronization can be achieved simultaneously. So, RGCS designed for randomly connected WSNs is an asynchronous and distributed configuration, and can make a complete compensation for the rate and offset. Due to the dense node density, DCS can not achieve rate compensation and offset compensation simultaneously. Because DCS devised for sparse node density to tolerate long delays leads to the synchronous mode for the update of the rate and offset. At the beginning of the compensation in RPCS, the rate of the rate compensation is faster than that in RGCS, because RPCS employs a standard frequency estimation technique to obtain an estimate of the pairwise drift. However, the descending rate of the offset compensation is relative slow, and the logical clock $T_i(t)$ could not be synchronized without the backward jumps phenomenon.

Figure 7 shows the maximum error of $T_i(t)$ of RGCS under $\lambda = 1, 5, 10$, from which it can be observed that it takes less time with $\lambda = 10$ to reach synchronization. This is because a larger intensity of Poisson process can decrease inter-time $\Delta(l, (ij))$ between consecutive gossip instants for any links. The results in Figure 6 validate the theoretical features of equation (3). So the descent velocity of the maximum error of logical clocks with $\lambda = 10$ becomes more remarkable.

Figure 8 shows the comparative simulation results with regard to single RGCS and Multi-RGCS. The green line represents the convergence of logical clocks in Multi RGCS. Falling speed of green line is faster than blue line (single RGCS), because more clocks exchange their states and come close to

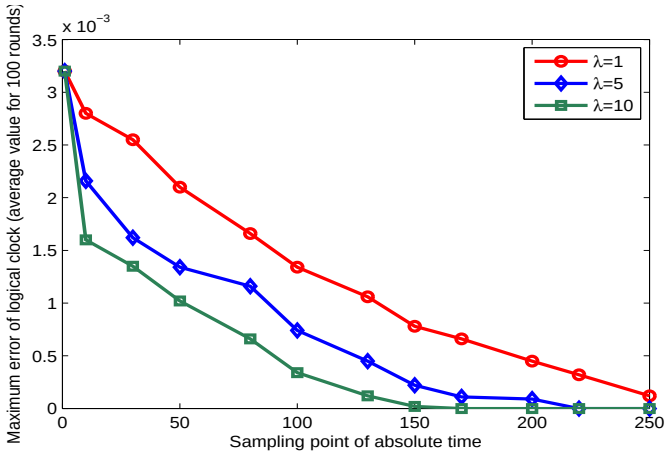

Figure 7. Comparison of maximum errors of logical clocks $T_i(t)$ of RGCS under different λ (the same initialization values of clocks for comparing intuitively).

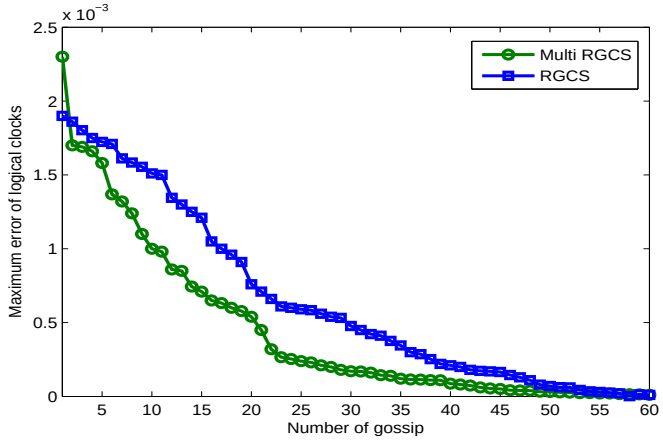

Figure 8. Comparison of maximum errors of $T_i(t)$ between single RGCS and Multi-RGCS.

the maximum-value at each gossip instant. Then, Figure 9 shows the relationship between different coloring edge's number ranging from $n = 5$ to $n = 12$ and 95th percentile of synchronization errors. The performance degrades as more coloring number increase, however it exhibits a approximative linear dependence, thus it refers to the amount of gossip as a function of the coloring number n. So we can involve this feature to suppress drift for different synchronization accuracies.

Based on the above simulation results and discussions, it can be seen that the paper has proposed a RGCS algorithm with superior performances compared to existing methods in term of dynamic adaptability and faster convergence speed. A edge-coloring algorithm can be applied to constitute a spanning tree so that Multi RGCS has faster convergence speed than that in RGCS.

5.2. Bounded Delay Case

OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. Based on MiXiM framework and Network Description (NED) files, we implement an ad hoc WSN on the OMNeT++ 5.1.1, and then run the Revised-RGCS algorithm on it. It has been simulated for WSNs of $N = 9$ clocks placed on a field of size 100m$\times$100m. Moving sensors indexed by bidimensional coordinate XY, namely node11, node12, node13, node21, node22, node23, node31, node32 and node33. For those mobile nodes, we used the Random Walk module to locate the coordinate position of the node. In this module, each node was made to move

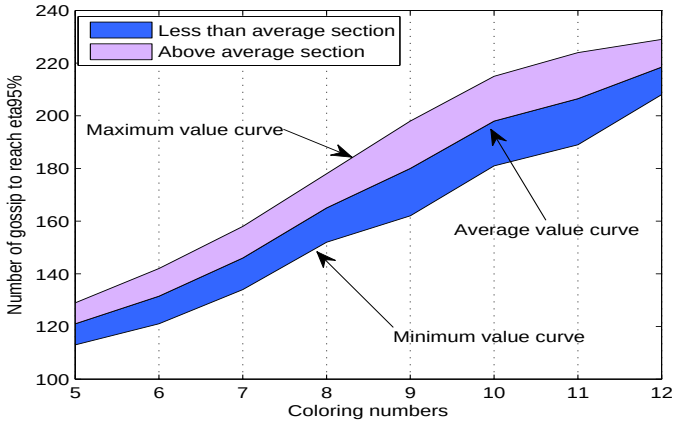

Figure 9. The relationship between the number of gossip and coloring number (simulate it for 10).

to next coordinate position at random. Note that the radios in a node could collect incident message sets with higher layer module operation decoupled by buffers, so deferred message will be valid. Summarize the key parameters into Table 2.

Table 2. Simulation Parameters

Module	Parameter and Data Value
Application	250 kbps data rate
Wireless channel	Bandwidth: 20 MHz, Data rate: 250 kbps Modulation type: BPSK
Radio	Sensitivity: -95 dBm, Noise floor: -100 dBm, Transmit power: 0 dBm, Mode: Ideal
Tunable MAC	Timestamp and default parameters
Communication radius	10m, 20m, 30m for every three nodes
Mobility model	Random walk (mobility update interval = 100 ms, speed = 5 m/sec)
Initial energy	28080 J
Field size	100m×100m
Simulation time	600 secs

The probability density distributions of time-delays d_{ij} and d_{ji} are shown in Figure 10. We carried on the statistics from 10 times simulations and obtain the distribution of them. It can be seen that the maximum delay is bounded which confirms the realistic bounded model. In reality, d_{ij} is often different from d_{ji} because the uplink and downlink between head and tail are independent of one another. Obviously, Revised-RGCS is an asymmetric gossip. In RPCS, the authors assume reciprocal propagation delays $d_{ij} = d_{ji}$ in each synchronization round. The delay assumption in this paper is more practical than that in the RPCS.

With the bounded delays, we compared the relative drift estimation method, which is based on low-pass filter, used in Revised-RGCS and that in RGCS. Figure 11 shows the results of relative drift estimation error $\hat{\xi}^*(l)$ of random pairwise nodes. It is observed that using equation (14), relative drift α_{ij} can be estimated accurately as $\hat{\alpha}^*_{ij}(l)$ (the blue line) will converge to the ideal value. Oppositely, the average estimate error is diverging in RGCS (the green line) without the low-pass filter. We see that the relative rate estimation under the bound delay can be obtained with Revised-RGCS, on which an effective timing protocol depends.

The simulation results of the Revised-RGCS, DCS and RPCS implementations are presented in Figure 12. It is observed from Figure 12 that Revised-RGCS takes about 20 synchronization rounds to reach within 10 ticks while RPCS does not achieve the expected error before 20 rounds. It is clear that

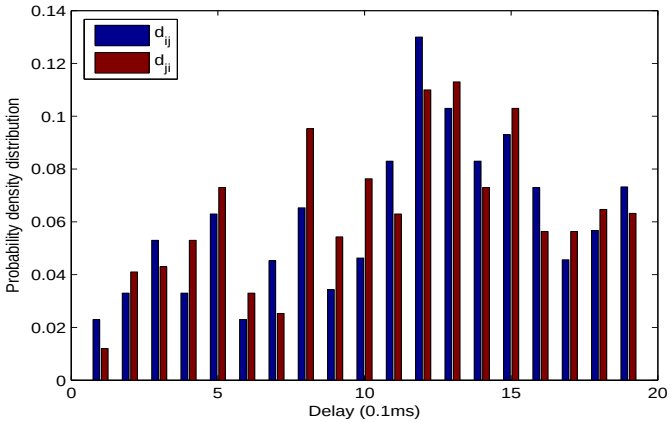

Figure 10. The difference of d_{ij} and d_{ji}.

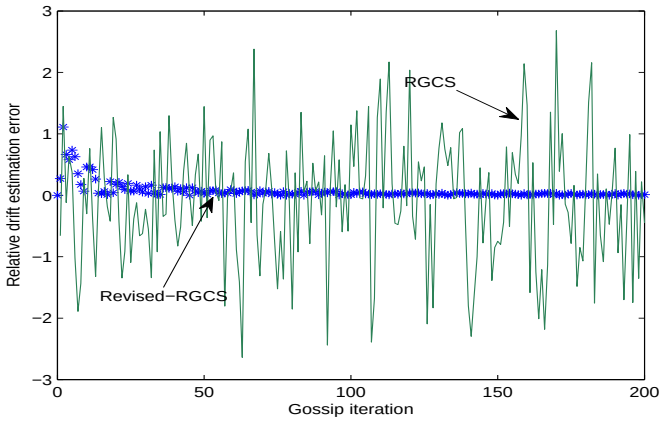

Figure 11. Comparison of relative drift estimation error between Revised-RGCS and RGCS.

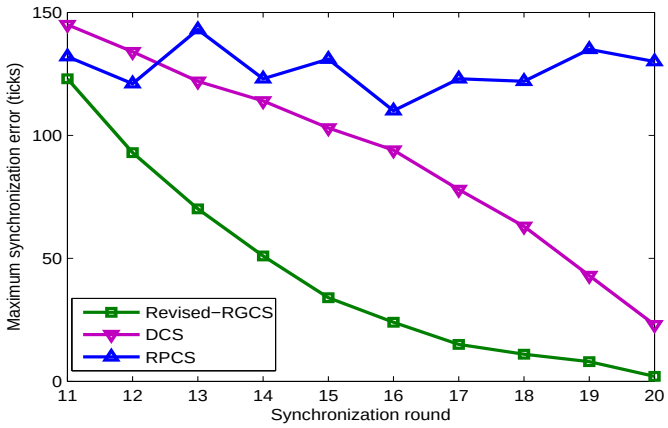

Figure 12. Comparison of maximum synchronization error (ticks) between Revised-RGCS algorithm and the current algorithms.

the Revised-RGCS can ensure that the sync error is bounded. In contrast, due to symmetric delay, the maximum synchronization error of the RPCS is ceaselessly growing. The results demonstrate that Revised-RGCS converges asymptotically while RPCS is diverging. Although the DCS algorithm can

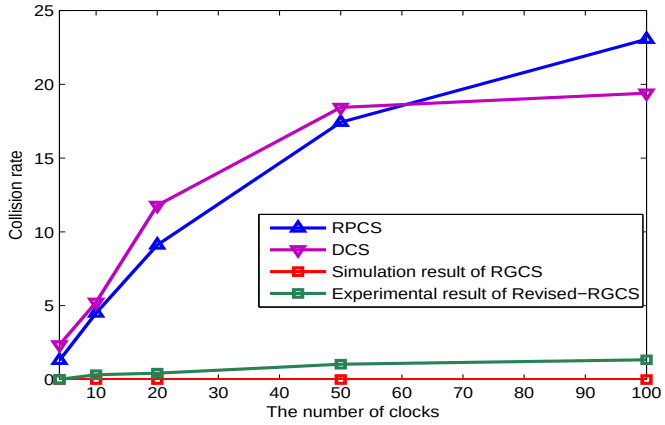

Figure 13. Comparison of Revised-RGCS, RGCS, DCS, and RPCS on collision rate as the number of clocks increasing.

reach time synchronization, but its convergence rate is relatively slow under the randomly connected scene. This is because-in DCS, the offset and rate compensation are initiated by the updated table information which is a weighted averages of the neighborhoods.

Most of the existing timing protocols employ a deterministic periodic synchronization scheme which results in message collisions, and they focus concentrate on which merely exploit a packet in a collision timeslot or directly discard redundant packets. These techniques are passive to a certain extent. The reason for this is that WSNs require a nontrivial collision detection scheme and invalid packets waste energy. In RGCS, we assume that every gossiping of a pair of nodes costs the same amount of energy $\mathrm{E_{eg}}$ and let $\mathrm{E_g}$ be the total energy cost for gossiping to the expected accuracy, then $\mathrm{E_s} = \mathrm{E_g} = \sum_{l=1;e_{ij}}^{N} \delta(l,(ij)) \times \mathrm{E_{eg}}$ for $\forall e_{ij} \in \mathcal{E}(t)$. As discussed in the Section 2, we know that, in most of distributed configuration protocols, the similar hardware clock states is the main reason behind the message collisions of hidden nodes. Hence, the number of clocks is considered to be an evaluation indicator. Figure 13 shows the timing message collision rate versus the number of clocks. The timing collision rate of RPCS and DCS protocols achieve 23.4%, 19.2% when the network scale is 100. In simulation, due to asynchronous pairwise policy and an appropriate λ, the collision rate of RGCS is null. In the Revised-RGCS, the collision rate is 2.1%, because the timing messages are delivered to the MAC layer of the nodes with long delays which collides with the Sync-L event at few instants.

Based on the above simulation results and discussions, the Revised-RGCS can guarantee time synchronization under realistic bounded delay model with low collision rate compared to the current algorithms in practice.

6. Conclusions

This paper presents a new randomized and energy-efficient time synchronization protocol called RGCS for dynamic WSNs with randomly changing connectivity. The protocol is based on distributed consensus timesync but incorporating with the gossip algorithm. Therefore, it is superior to existing protocols in a generalized randomization framework, and can be well adapted to the link variation of ad hoc WSNs dynamically. The new idea of randomized scheduling of the synchronization links is to lower the rate of collision, so the RGCS dramatically relieves the collisions phenomenon. Based on the converge-to-max criterion, the method has been deduced to accelerate the convergence speed. By combining the idea of the EC, Multi-RGCS has been proposed. Moreover, Revised-RGCS has been developed to specially restrain the impact of uncertain bounded delays. Extensive simulations have demonstrated the better performances of the proposed protocols.

The proposed EC method is rough, since the dynamic ad hoc WSNs need a global coordinator in order to produce a spanning tree at each changing moment of topology. It is worth investigating how to design pure distributed EC algorithm for Multi-RGCS. This will enhance the adaptability for fast changing networks. However, pure distributed edge-coloring is to be NP-Complete. In future work, we will focus on developing a solution to find the maximum clique of the sync edges.

Acknowledgments: This work was supported by the National Natural Science Foundation of China under grant No. 61633016.

Author Contributions: Nan Xiong and Minrui Fei made substantial contributions to the original ideas on RGCS, Multi RGCS, and Revised-RGCS. Nan Xiong carried out the simulations and wrote the paper. Taicheng Yang provided critical guidance and given quite a lot of suggestions during the research and paper revising.

Conflicts of Interest: The authors declare that there are no conflicts of interest.

Bibliography

1. Wu, F.-J.; Kao, Y.-F.; Tseng, Y.-C. From wireless sensor networks towards cyber physical systems. *Pervasive Mob. Comput.* **2011**, *7*, 397-413.
2. Mahmood, A.; Exel, R.; Trsek, H.; Sauter, T. Clock synchronization over IEEE 802.11 - a survey of methodologies and protocols. *IEEE Trans. Ind. Inf.* **2017**, *13*, 907-922.
3. Zhang, R.-H.; He, Z.-C.; Wang, H.-W.; et al. Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system. *IEEE Access* **2017**, *5*, 6649-6660.
4. Huang, P.-H.; Maulik, D.; Qiu, X.F.; et al. On the multihop performance of synchronization mechanisms in high propagation delay networks. *IEEE Trans. Comput.* **2009**, *58*, 577-590.
5. Du, J.; Wu, Y.-C. Distributed clock skew and offset estimation in wireless sensor networks: Asynchronous algorithm and convergence analysis, *IEEE Trans. Wirel. Commun.* **2013**, *12*, 5908–5917.
6. Yildirim, K.-S.; Kantarci, A. Time synchronization based on slow-flooding in wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 244–253.
7. Cao, X.-Y.; Yang, F.; Gan, X.-Y.; Liu, J.; et al. Joint estimation of clock skew and offset in pairwise broadcast synchronization mechanism. *IEEE Trans. Commun.* **2013**, *61*, 2508–2521.
8. Ren, F.-Y.; Lin, C.; Liu, F. Self-correcting time synchronization using reference broadcast in wireless sensor network. *IEEE Wirel. Commun.* **2008**, *15*, 79-85.
9. Meng, Y.; Li, T.; Zhang, J.-F. Finite-level quantized synchronization of discrete-time linear multi-agent systems with switching topologies. *SIAM J. Control Optim.* **2017**, *55*, 275-299.
10. Zong, X.-F; Li, T.; Zhang, J.-F. Consensus conditions of continuous-time multi-agent systems with additive and multiplicative measurement noises. *SIAM J. Control Optim.* **2018**, *56*, 19-52.
11. Schenato, L.; Fiorentin, F. Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica* **2011**, *47*, 1878-1886.
12. He, J.-P.; Cheng, P.; Shi, L.; Chen, J.-M. Time synchronization in WSNs: A maximum-value-based consensus approach. *IEEE Trans. Autom. Control* **2014**, *59*, 660-675.
13. Panigrahi, N.; Khilar, P. M. An evolutionary based topological optimization strategy for consensus based clock synchronization protocols in wireless sensor network. *Swarm Evol. Comput.* **2015**, *22*, 66-85.
14. Saiah, A.; Benzaid, C.; Badache, N. CMTS: Consensus-based multi-hop time synchronization protocol in wireless sensor networks. In Proceedings of the IEEE 15th International Symposium on Network Computing and Applications, Cambridge, United States, October 30, 2016 - November 2, 2016; pp. 232-236.
15. Garone, E.; Gasparri, A.; Lamonaca, F. Clock synchronization protocol for wireless sensor networks with bounded communication delays. *Automatica* **2015**, *59*, 60-72.
16. Tian, Y.-P. LSTS: A new time synchronization protocol for networks with random communication delays. In Proceedings of the 2015 IEEE 54th Annual Conference on Decision and Control Conference, Osaka, Japan, 15-18 December 2015; pp.7404-7409.
17. Tian, Y.-P.; Zong, S.-H.; Cao, Q.-Q. Structural modeling and convergence analysis of consensus-based time synchronization algorithms over networks: Non-topological conditions. *Automatica* **2016**, *65*, 60-75.

18. Lamonaca, F.; Gasparri, A.; Garone, E.; Grimaldi, D. Clock synchronization in wireless sensor network with selective convergence rate for event driven measurement applications. *IEEE Trans. Instrum. Meas.* **2014**, *63*, 2279-2287.
19. Brown, D. R.; Klein, A. G.; Wang, R. Monotonic mean-squared convergence conditions for random pairwise consensus synchronization in wireless networks. *IEEE Trans. Signal Process* **2015**, *63*, 988–1000.
20. Sun, W.-L; Strom, E.G.; Brannstrom, F.; Gholami, M.R. Random broadcast based distributed consensus clock synchronization for mobile networks. *IEEE Trans. Wireless Commun.* **2015**, *14*, 3378-3389.
21. Bolognani, S.; Carli, R.; Lovisari, E.; Zampieri, S. A randomized linear algorithm for clock synchronization in multi-agent systems. *IEEE Trans. on Autom. Control* **2016**, *61*, 1711-1726.
22. He, J.-P; Cheng, P.; Shi, L.; et al. SATS: Secure average-consensus-based time synchronization in wireless sensor networks. *IEEE Transaction on Signal Processing* **2013**, *61*, 6387-6400.
23. Dong, W.; Liu, X.-J. Robust and secure time-synchronization against sybil attacks for sensor networks. *IEEE Transactions on Industrial Informatics* **2015**, *11*, 1482-1491.
24. Katarzyna, K.-S. A survey of MAC layer solutions to the hidden node problem in ad-hoc networks. *Ad Hoc Networks* **2012**, *10*, 635-660.
25. Boyd, S.; Ghosh, A.; Prabhakar, B.; Shah, D. Randomized gossip algorithms. *IEEE Trans. Inf. Theory* **2006**, *52*, 2508-2530.
26. Dimakis, A.G.; Kar, S.; Moura, J.M.F.; et al. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE* **2010**, *98*, 1847-1864.
27. Lavaei, J.; Murray, R. M. Quantized consensus by means of gossip algorithm. *IEEE Trans. on Autom. Control* **2012**, *57*, 19-32.
28. Picci, G.; Taylor, T.-J. Almost sure exponential convergence to consensus of random gossip algorithms, *Int. J. Robust Nonlinear Control* **2013**, *23*, 1033–1045.
29. Franceschelli, M.; Giua, A.; Seatzu, C. Fast discrete consensus based on gossip for makespan minimization in networked systems. *Automatica* **2015**, *56*, 60-69.
30. Du, J.; Wu, Y.-C. Distributed clock skew and offset estimation in wireless sensor networks: Asynchronous algorithm and convergence analysis, *IEEE Trans. Wireless Commun.* **2013**, *12*, 5908–5917.
31. Choi, B. J.; Liang, H.; Shen, X.; Zhuang, W.-H. DCS: distributed asynchronous clock synchronization in delay tolerant networks. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 491–504.
32. Ahmed, S.; Xiao, F.; Chen, T.-W. Asynchronous consensus-based time synchronisation in wireless sensor networks using unreliable communication links. *IET Control Theory Appl.* **2014**, *8*, 1083–1090.
33. Huang, G.; Zomaya, A. Y.; Delicato, F. C.; Pires, P.F. An accurate on-demand time synchronization protocol for wireless sensor networks. *J. Parallel Distrib. Comput.* **2012**, *72*, 1332–1346.
34. Nicolas, M.; Jean-Benoit, P.; Jean-Marie, G. Fine synchronization for wireless sensor networks using gossip averaging algorithms. In Proceedings of the 2008 IEEE International Conference on Communications, Beijing, China, 19-23 May 2008; pp.4963-4967.
35. Saverio, B.; Ruggero, C.; Sandro, Z. A PI consensus controller with gossip communication for clock synchronization in wireless sensors networks. In Proceedings of the 1st IFAC Workshop on Estimation and Control of Networked Systems, Venice, Italy, 24-26 September 2009; pp.78-83.
36. Baldoni, R.; Corsaro, A.; Querzoni, L.; Scipioni, S.; Tucci Piergiovanni, S. Coupling-based internal clock synchronization for large-scale dynamic distributed systems. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *21*, 607–619.
37. Joerg, S.; Patrick, J.; Reinhold, H-U. A gossiping approach to sampling clock synchronization in wireless acoustic sensor networks. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing, Florence, Italy, 4-9 May 2014; pp.7575-7579.
38. Stankovic, M.S.; Stankovic, S.S.; Johansson, K.H. Asynchronous distributed blind calibration of sensor networks under noisy measurements. *IEEE Trans. Control Netw. Syst.* **2018**, *5*, 571-582.
39. Zhang, D.-S.; Tian, H.; Liu, Y.-H.; et al. Poster: Neighbor discovery with distributed quorum system. In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, Seattle, United States, 1-4 November 2011; pp.369-370.
40. Gandham, S.; Dawande, M.; Prakashc, R. Link scheduling in wireless sensor networks: Distributed edge-coloring revisited. *J. Parallel Distrib. Comput.* **2008**, *68*, 1122-1134.