DCS: Distributed Asynchronous Clock Synchronization in Delay Tolerant Networks

Bong Jun Choi, *Student Member, IEEE,* Hao Liang, *Student Member, IEEE,*
Xuemin (Sherman) Shen, *Fellow, IEEE,* and Weihua Zhuang, *Fellow, IEEE*
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada
{bjchoi, h8liang, xshen, wzhuang}@bbcr.uwaterloo.ca

Abstract

In this paper, we propose a distributed asynchronous clock synchronization (DCS) protocol for Delay Tolerant Networks (DTNs). Different from existing clock synchronization protocols, the proposed DCS protocol can achieve global clock synchronization among mobile nodes within the network over asynchronous and intermittent connections with long delays. Convergence of the clock values can be reached by compensating for clock errors using mutual relative clock information that is propagated in the network by contacted nodes. The level of clock accuracy is depreciated with respect to time in order to account for long delays between contact opportunities. Mathematical analysis and simulation results for various network scenarios are presented to demonstrate the convergence and performance of the DCS protocol. It is shown that the DCS protocol can achieve faster clock convergence speed and, as a result, reduce energy cost by half for neighbor discovery.

Index Terms

Delay Tolerant Networks, clock synchronization, mobility, power management.

I. Introduction

The delay/disruption tolerant networks (DTNs) is characterized by frequent disconnections and long delays of links among nodes due to mobility, sparse deployment of nodes, attacks,

This paper was presented in part at IEEE ICC 2010 [1].

and noise, etc. Considerable research efforts have been devoted recently to DTNs to enable communications between network entities with intermittent connectivity [2–4].

Clock synchronization is an important requirement in DTNs for providing accurate timing information of data collected from physical environments as well as for energy conservation. In traditional multihop wireless networks, clock synchronization is required for collision-free transmissions in medium access control (MAC) such as TDMA and the superframe based protocols. Especially, accurate clock synchronization is crucial for energy efficient sleep scheduling mechanisms in DTNs [5–7] where nodes need to coordinately wake up at every beacon interval for an awake period to discover other nodes within a transmission range. Due to much larger inter-contact durations than contact durations, by more than an order of magnitude in many DTN scenarios, nodes consume a significant amount of energy in the neighbor discovery process, much more than that in infrequent data transfers [8]. Moreover, the energy required for the neighbor discovery increases if the clocks are not perfectly synchronized. In general, perfect clock oscillators do not exist, and relative clock errors are unavoidable. Therefore, nodes usually have loosely synchronized clocks and use extra awake periods, called guard periods, to compensate for uncertainty in clock accuracy, as illustrated in Fig. 1. In DTNs, an increase in the clock inaccuracy coupled with increases in the number of hops and the inter-contact durations [9] results in a need for large guard periods that cause significant energy consumptions. Therefore, clock synchronization is essential in DTNs for achieving high energy efficiency. In addition, synchronization protocols typically cannot rely on the Global Positioning System (GPS) which requires a large amount of energy and a line of sight to the satellites or reference nodes acting as centralized time servers.

While clock synchronization in multihop wireless networks is a well-studied problem, the new environment in DTNs presents a set of great challenges. In the traditional multihop wireless networks, nodes are assumed to be constantly connected. However, this assumption does not hold in DTNs which suffers from large inter-contact durations and infrequent message exchanges. Furthermore, clock synchronization may need to be performed asynchronously by each node due to opportunistic contacts. The simplest solution to this problem is to have a particular node, acting as a reference node, to broadcast its own clock value to all other nodes in the network. However, this approach is not robust to node failures. Also, there is a large overhead associated with the discovery and management of reference nodes due to a long duration between adjacent

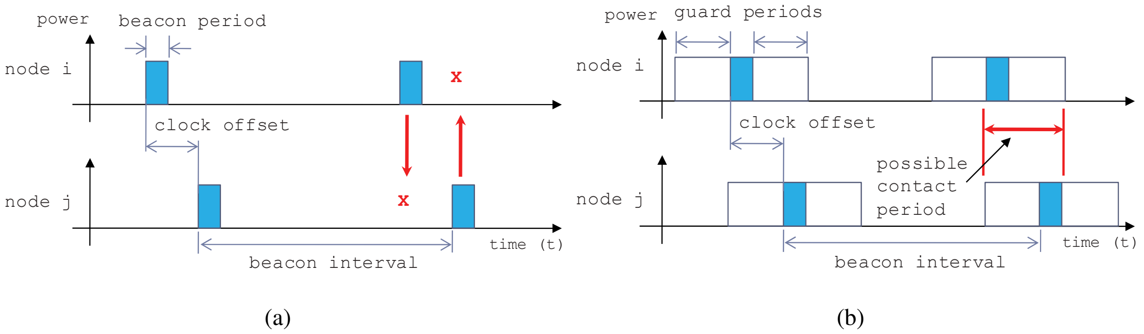


Fig. 1. Use of guard periods to compensate for clock inaccuracy in sleep scheduling: a) Without guard periods: contact not possible due to non-overlapping active periods; b) With guard periods: contact possible during additional active periods that allow overlapping active periods.

contacts and frequent network partitions. To address these challenges, we propose a distributed asynchronous clock synchronization (DCS) protocol for DTNs. The protocol is fully distributed, so that all nodes independently execute exactly the same algorithm without the need of reference nodes. Global clock synchronization is achieved by asynchronously compensating for clock errors using relative clock information exchange among nodes. The long inter-contact duration in DTNs is taken into account by introducing a weighting coefficient that represents the level of accuracy of propagated information. Analytical and simulated results are presented to evaluate the performance of the DCS protocol.

The remainder of this paper is organized as follows. Section II provides an overview of the related work. Section III describes the system model. Section IV presents the proposed distributed asynchronous clock synchronization protocol for DTNs. The performance analysis is presented in Section V. The numerical and simulation results are given in Section VI to evaluate the performance of the DCS protocol. Section VII draws the conclusions of this work.

II. Related Work

The Network Time Protocol (NTP) [10] has been widely used to synchronize computer clocks in the Internet. The NTP enables synchronization between the hierarchically arranged servers and clients. The clocks of servers are adjusted by trusted time references. However, the NTP is intended for connected Internet where the synchronization operation can be conducted between the reference node and clients continuously and frequently.

Recently, there has been extensive research on clock synchronization in multihop wireless networks. Existing protocols can be classified into two types, depending on whether or not there are reference nodes: reference based clock synchronization and distributed clock synchronization. In the reference based clock synchronization, non-reference nodes tune to the clock information distributed by reference nodes. Reference nodes are referred to as *roots* in tree based protocols [11, 12], *gateways* in cluster based protocols [13], or *time servers* in NTP based protocols [10, 14]. Conversely, in the distributed clock synchronization, all nodes in the network run the same distributed algorithm without a reference node. Global clock synchronization is reached by each node advancing to the faster clock [15, 16], averaging clock values of local nodes [17, 18], or gradually decreasing the clock error with neighboring nodes using a proportional controller [19]. However, independent of whether or not these protocols assume deployments of static or mobile nodes, they all require a network topology without frequent disconnections or long inter-contact durations.

In terms of DTNs, there have been some efforts for clock synchronization. The Timestamp Transformation Protocol (TTP) [9] solves the temporal ordering problem in sparse ad-hoc networks. The protocol does not synchronize clocks, but transforms message time-stamps at each node to its local time-stamp with some error bound as a message moves from hop to hop. Simulation results show that the clock inaccuracy increases linearly with time and the number of hops. The Double-pairwise Time Protocol (DTP) [14] provides time synchronization in DTNs with a modified NTP. The DTP achieves a clock estimation error lower than the NTP by explicitly estimating the relative clock frequency using back-to-back messages with a controllable interval in between. However, the DTP is a reference node clock synchronization that assumes at least one time server in the network, and it does not work in a distributed environment where there is no reference node to spread the correct reference clock information. The Asynchronous Diffusion (AD) protocol [17, 20] provides distributed clock synchronization by asynchronously averaging clock values with the contacted neighbors. There also have been some efforts to provide clock synchronization in underwater acoustic networks (UANs) using acoustic modems [21]. Our focus in this paper is on providing distributed clock synchronization in terrestrial networks. The research problems and solutions for terrestrial networks and UANs are different since the main source of delay is due to the sparse deployment and mobility nodes in terrestrial networks and due to the long propagation delay in UANs.

TABLE I

SUMMARY OF IMPORTANT SYMBOLS USED

Symbol	**Definition**
$C_i(t)$, $f_i(t)$	clock value and frequency of node i at time t
$C_{ij}(t)$, $f_{ij}(t)$	relative clock offset and skew between nodes i and j at time t
$C_{il}^T(t)$, $f_{il}^T(t)$, $w_{il}^T(t)$	relative clock offsets, relative skews, and weight coefficients stored in node i for node l at time t
λ	aging parameter
$N_i^T(t)$	set of node entries stored in node i at time t
$t_k^{i,j}$	kth contact time between node i and j
t_k	kth contact time between any pair of nodes

III. SYSTEM MODEL

A. Network Model

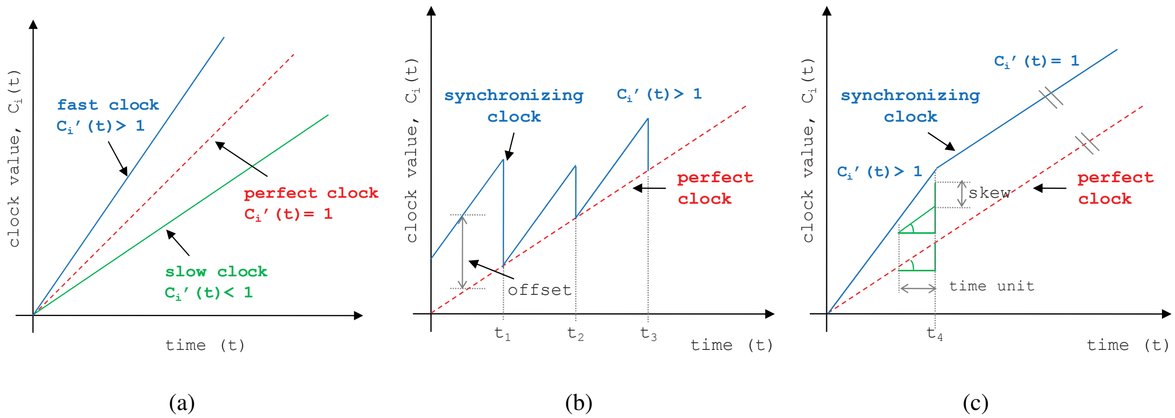
We consider a network represented by graph $G(t) = (V, E(t))$, where the vertex set V contains N mobile nodes and the edge set $E(t)$ is defined as the set of nodes in contact at time t. Due to frequent link disconnections and dynamic network topology, $E(t)$ varies with time. Contact schedules among nodes are not known in advance. At time t, two nodes i and j are connected, i.e., $(i, j) \in E(t)$, if they can successfully exchange connection setup messages. The set of contact times of node i and node $i \neq j$ are represented as $T_c^{i,j} = \{t_1^{i,j}, \ldots, t_k^{i,j}, \ldots\}$. Links are undirected and symmetric. Therefore, if $(i, j) \in E(t)$, we also have $(j, i) \in E(t)$. Upon each contact, nodes exchange and update their timing information. We assume a distributed communication topology where there are no special reference nodes such as roots or gateways, and all nodes execute exactly the same algorithm for clock synchronization. The procedure and modeling of clock value and frequency updates apply to all nodes in the network. As many symbols are used in this paper, Table I summarizes the important ones.

B. Clock Model

Each node maintains a logical software clock as a function of the hardware oscillator. The clock value of node i at time t is given by

$$C_i(t) = (1 + \xi_i) \int_{t_0}^{t} \omega_i(\tau) d\tau + C_i(t_0) \tag{1}$$

where ξ_i is a proportional coefficient of the node i oscillator, $\omega_i(\tau)$ is the frequency of the hardware oscillator at time τ, and $C_i(t_0)$ is the initial clock value at time t_0. The clock value



(a) (b) (c)

Fig. 2. Clock model: a) Clock error caused by clocks running at different rates; b) Offset compensation: clock value is tuned to the clock value of the perfect clock, but the clock offset increases with time due to the clock skew error; c) Skew compensation: clock frequency is tuned to the clock frequency of the perfect clock, but the clock offset does not match due to the clock offset error.

is incremented by an oscillator with frequency $f_i(t) = \frac{dC_i(t)}{dt}$. As illustrated in Fig. 2(a), the frequency of a perfect clock relative to Coordinated Universal Time (UTC) is $\frac{dC_i(t)}{dt} = 1$. However, the clock deviates from the perfect clock over time due to errors in clock frequency and changes in supply voltage, temperature, etc. We have $\frac{dC_i(t)}{dt} > 1$ for a fast clock, and $\frac{dC_i(t)}{dt} < 1$ for a slow clock.

For an inaccurate clock, the clock frequency value is represented as an estimate with a lower and a upper bound, i.e.,

$$1 - \rho \leq \frac{dC_i(t)}{dt} \leq 1 + \rho, \ \forall \ t \tag{2}$$

where ρ is a constant maximum clock frequency error specified by the hardware manufacturer, and the bounded error for each node is modeled by the uniform distribution [16]. Typical error for a quartz crystal oscillator is $\rho \in [10, 100]$ ppm, which corresponds to a 0.6 ms to 6 ms error in 60 s.

C. *Sources of Clock Synchronization Error*

In general, the frequency of quartz crystal oscillator of node i ($\omega_i(\tau)$) is a time-varying random variable. The randomness is due to short-term and long-term instabilities. Short-term instability is caused by environmental factors, such as changes in temperature, pressure, and supply voltage, whereas long-term instability is caused by the oscillator aging. We study the effect of time variant oscillator frequencies in Section VI-B-5. Furthermore, uncertainty in the message delays cause clock estimation errors. Usually, in a multihop wireless network, an accurate estimate of the message delay at each hop is critical for synchronization protocols since the end-to-end delay is comparable to the error caused by message delays. Components of a message delay include medium access time, transmission time, radio propagation time, and detection time. In DTNs, however, as the inter-contact duration increases and the frequency of message exchange decreases, the clock error induced by the inaccurate clock frequency increases, whereas the error related to the message delays remains constant [9]. Nonetheless, there exists uncertainty in message delays over various hardware interfaces and the wireless channel. The uncertainty in the message delays cause error when estimating clock information among nodes. As a result, the clock offset and the clock skew are estimated with some error bound. We study the effect of clock estimation errors in Section VI-B-6.

D. Clock Compensation

The goal of clock synchronization protocols is to maintain the minimum possible clock error among nodes in the network. As described in the clock model, the accuracy of a clock is measured by two parameters: clock value and clock frequency. Since there is no reference node in the distributed network, the prefect clock value and frequency are impossible to obtain. Instead, relative clock value and frequency between two nodes can be obtained by simple two-way exchange of time synchronization messages. The difference in clock value readings $C_{ij}(t) \triangleq C_j(t) - C_i(t)$ is called *relative clock offset*, and the difference in logical clock frequency $f_{ij}(t) \triangleq f_j(t) - f_i(t)$ is called *relative clock skew*. The process of minimizing the clock offset (skew) among nodes is referred to as *clock offset (skew) compensation*. As shown in Figs. 2(b) and 2(c), respectively, offset (skew) compensations are done at times t_1, t_2, and t_3 (t_4) to match the logical time (clock frequency) among nodes. Consequently, our goal is to minimize relative clock offset and relative clock skew among nodes in the network through compensations.

E. Performance Metric

The performance of clock synchronization protocols is evaluated by how quickly the clock values of different nodes can converge to a global average. The metrics used to describe the convergence are relative clock offset and relative clock skew. For a total number of N nodes in the network, the average relative clock offset and the average relative clock skew at time t are calculated by

$$C_{avg}(t) = \frac{1}{N(N-1)/2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |C_i(t) - C_j(t)| \tag{3}$$

$$f_{avg}(t) = \frac{1}{N(N-1)/2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |f_i(t) - f_j(t)| \, . \tag{4}$$

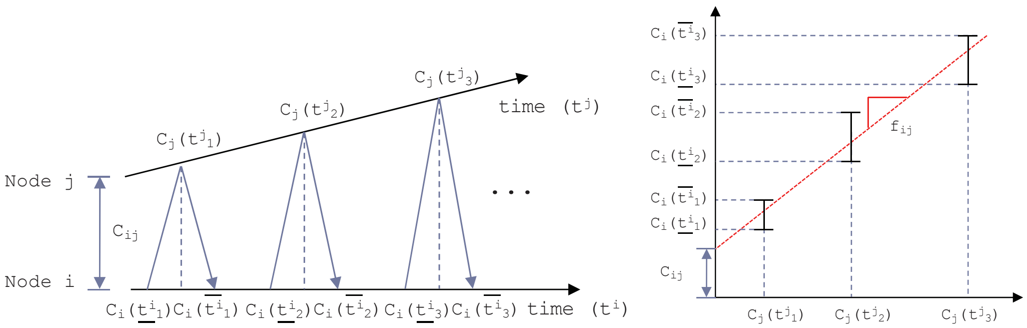
IV. Distributed Asynchronous Clock Synchronization Protocol

In this section, we propose a distributed asynchronous clock synchronization protocol for DTNs, which provides global clock synchronization with a distributed algorithm. The DCS protocol is designed for DTNs where finding or electing reference nodes is difficult and where connections are often delayed and disrupted among nodes due to mobility and a sparse node density.

The basic idea of the protocol is to utilize the relative clock information spread in the network to update clock values, rather than diffusing the information obtained from local neighbors in hop-by-hop fashion as in existing distributed clock synchronization protocols for multihop wireless networks. Each node independently manages a table that contains relative clock information. Upon each new contact, this information is exchanged with the contacted node and transformed to the compensated logical clock values. Each node uses the clock information in the table to asynchronously calculate the clock frequency and value that gradually approach their global averages. To account for a decrease in the accuracy of the propagated clock information, the contributing weights of the stored information used for the clock compensations are depreciated over time.

A. Clock Table Structure

We first introduce a structure of the clock table that contains the relative clock information. At time t, each node i contains a list of other nodes ($\forall\ l \in V, l \neq i$) in the network that it has contacted or obtained from contacted nodes. Each node entry in the list is identified by a unique identifier with following fields: relative clock offset ($C_{il}^T(t)$), relative clock skew ($f_{il}^T(t)$), and weight ($w_{il}^T(t)$) which represents the level of information accuracy of node l at node i. Initially, node i contains only its own information in the list as $C_{ii}^T(0)$, $f_{ii}^T(0) = 0$, and $w_{ii}^T(0) = 1$. The set of node entries in the table of node i ($N_i^T(t)$) increases when the node obtains information



(a) (b)

Fig. 3. Relative clock estimation: a) Exchanging time-stamps between node i and node j; b) Plotting time-stamped triples for clock skew estimation.

of a new node. Note that the table entries may become outdated with time and do not represent the up-to-date differences of the clock values and clock frequencies, i.e., it is possible that $C^T_{il}(t) \neq C_{il}(t)$ and $f^T_{il}(t) \neq f_{il}(t)$.

A large clock table size may degrade the performance of the network having limited resources (such as wireless sensor networks) and limited contact durations (such as vehicular networks). In such scenarios, the clock table overhead can be reduced by various table management techniques. For example, nodes can decide to store, compute, or exchange a certain maximum number of entries based on the performance requirement. A higher priority can be given to entries with higher weights.

B. Exchanging Clock Table Information

In the absence of a reference clock, although the actual clock frequencies $f_i(t^{i,j}_k)$ and $f_j(t^{i,j}_k)$ are impossible to obtain, a relative clock skew ($f_{ij}(t^{i,j}_k)$) can be estimated. When node i contacts node j at time $t^{i,j}_k$, i.e, $(i,j) \in E(t^{i,j}_k)$, they exchange series time-stamped triples $(C_i(\underline{t^i_m}), C_j(t^j_m), C_i(\overline{t^i_m}))$ for $m = 1, 2, \ldots$, as illustrated in Fig. 3. Here, $C_i(\underline{t^i_m})$ is the local time of node i when mth message is sent, $C_j(t^j_m)$ is the local time of node j when the mth message is received, and $C_i(\overline{t^i_m})$ is the local time of node i when the reply for the mth message is received from node j. Then, by plotting series of time-stamped triples, as shown in Fig. 3(b), $C_{ij}(t^{i,j}_k)$ and $f_{ij}(t^{i,j}_k)$ can be estimated by the following linear equation

$$C_i(t^i_m) = f_{ij}(t^{i,j}_k)C_j(t^j_m) - C_{ij}(t^{i,j}_k) \tag{5}$$

representing a line that passes through the bounded errors $C_i(\underline{t_m^i})$ and $C_i(\overline{t_m^i})$ [22]. More timestamps generate tighter estimation error bounds for the relative clock offset and the relative clock skew.

The update procedure is executed for both nodes i and j upon contact. Without loss of generality, we present the update procedure for node i here. Since the information of relative clock offset and relative clock skew between nodes i and j is newly obtained upon the contact, we update $C_{ij}^T(t_k^{i,j}) \leftarrow C_{ij}(t_k^{i,j})$ and $f_{ij}^T(t_k^{i,j}) \leftarrow f_{ij}(t_k^{i,j})$ for node i. Also, the associated weight values are reset as $w_{ij}^T(t_k^{i,j}) = 1$. Then, the clock information is updated in the table of node i for node $l \neq i, j$ if the information about node l received from the node j is more accurate, i.e., $w_{il}^T(t_k^{i,j}) \leq w_{jl}^T(t_k^{i,j})$. For node i, the updated table information after the exchange is

$$C_{il}^T(t_k^{i,j}) \leftarrow \begin{cases} C_{il}^T(t_k^{i,j}), & \text{if } w_{il}^T(t_k^{i,j}) \geq w_{jl}^T(t_k^{i,j}) \\ C_{ij}(t_k^{i,j}) + C_{jl}^T(t_k^{i,j}), & \text{otherwise} \end{cases} \tag{6}$$

$$f_{il}^T(t_k^{i,j}) \leftarrow \begin{cases} f_{il}^T(t_k^{i,j}), & \text{if } w_{il}^T(t_k^{i,j}) \geq w_{jl}^T(t_k^{i,j}) \\ f_{ij}(t_k^{i,j}) + f_{jl}^T(t_k^{i,j}), & \text{otherwise} \end{cases} \tag{7}$$

$$w_{il}^T(t_k^{i,j}) \leftarrow \begin{cases} w_{il}^T(t_k^{i,j}), & \text{if } w_{il}^T(t_k^{i,j}) \geq w_{jl}^T(t_k^{i,j}) \\ w_{jl}^T(t_k^{i,j}), & \text{otherwise.} \end{cases} \tag{8}$$

C. Clock Compensation

In DTNs, skew compensations are equally important as offset compensations. Since offset compensations cannot be done frequently in the network, even if two nodes start with the same time value, difference in their logical clock frequencies can result in a large clock offset over time. For instance, assuming that they have a relative clock skew of just 10 ppm, their time difference will be 0.6 ms after 60 s, and will further diverge to 36.0 ms after one hour.

Therefore, nodes i and j compensate for offset and skew errors using the updated table information. For node i, the compensated clock value and frequency are calculated using weighted averages as

$$C_i(t_k^{i,j}) \leftarrow C_i(t_k^{i,j}) + \frac{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) C_{il}^T(t_k^{i,j})}{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})}, f_i(t_k^{i,j}) \leftarrow f_i(t_k^{i,j}) + \frac{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) f_{il}^T(t_k^{i,j})}{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})} \tag{9}$$

and the clock offsets and skews in the table entries, except for $C_{ii}^T(t_k^{i,j})$ and $f_{ii}^T(t_k^{i,j})$, are updated as

$$C_{il}^T(t_k^{i,j}) \leftarrow C_{il}^T(t_k^{i,j}) - \frac{\sum_{l\in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) C_{il}^T(t_k^{i,j})}{\sum_{l\in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})}, f_{il}^T(t_k^{i,j}) \leftarrow f_{il}^T(t_k^{i,j}) - \frac{\sum_{l\in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) f_{il}^T(t_k^{i,j})}{\sum_{l\in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})}. \tag{10}$$

Note that, since w_{ii}^T represents the level of accuracy of its own clock information, w_{ii}^T is always one. Therefore, (9) includes the case for $l = i$ to account for w_{ii}^T. Also, by definition, since the relative clock offset and skew of a node to itself are both zero, C_{ii}^T and f_{ii}^T are initially assigned zero (see section IV-A) and remain unchanged.

While the updated values of offsets and skews do not change between contacts, the contributing weights in the table of node i are decreased between contacts to account for the decrease of the accuracy of clock information with time. Suppose two consecutive contacts of node i take place at times t and $t + \Delta t$, we have

$$w_{il}^T(t+\Delta t) = \begin{cases} 1, & \text{if } l = i \\ w_{il}^T(t)\lambda^{\Delta t}, & \text{otherwise} \end{cases} \tag{11}$$

where $\lambda \in [0, 1]$ is the aging parameter and Δt is the time elapsed in seconds between two consecutive contacts. Note that $\lambda = 1$ corresponds to propagating information without depreciating the weight over time, while $\lambda = 0$ corresponds to only making use of the information about the contacted node. The contact time for each node is unknown in advance, but the time difference between two contact times can be calculated upon each new contact. Although, the calculation of Δt may not be accurate due to clock errors of node itself, since the inter-contact durations in the DTNs are usually much larger than the error, the effect of the estimation error is negligible.

Note that a backward clock movement is not acceptable for some applications. Whereas the logical clock frequency can be changed immediately by applying the skew compensation, the time of a clock cannot run backward by applying a negative offset compensation. A common solution to this problem is to freeze the time of the node until the other node, having a slower time and applied with a positive offset compensation, reaches the same time [19]. This solution can be considered in implementing the DCS protocol to solve the backward clock movement problem.

D. Convergence Analysis

In this section, we show that the clocks running the DCS protocol converge to a common value. For the analysis, the identities of all N nodes in the network are assumed to be known. Let the set of contact times between any pair of nodes in the network be defined as $T_c = \bigcup_{i,j \in V} T_c^{i,j} = \{t_1, \ldots, t_k, \ldots\}$. Since $f_i(t_k)$ is constant between contacts for all $i \in V$ and $C_i(t_k)$ is dependent on $f_i(t_k)$, if $f_i(t_k)$ converges to a common value, the convergence of $C_i(t_k)$ simply follows the convergence proof of $f_i(t_k)$. Therefore, we focus on the convergence proof of the clock frequency, $f_i(t_k)$.

The clocks in the network converge if the relative clock skews converge to zero as

$$\lim_{t_k \to \infty} \max_{i,j \in V} |f_{ij}(t_k)| = 0. \tag{12}$$

or equivalently, for some constant value f_s, if

$$\lim_{t_k \to \infty} \max_{j \in V} f_j(t_k) = \lim_{t_k \to \infty} \min_{i \in V} f_i(t_k) = f_s. \tag{13}$$

The convergence proof can be simplified by the following lemma.

Lemma 1. *With the DCS protocol, the resulting frequencies of node i, after the updates using relative skews in the table ($f_{il}^T(t_k)$) and the outdated actual frequencies ($f_l(t_{il}^d(t_k))$), are the same, where $0 \leq t_{il}^d(t_k)$ represents the time when the frequency value of node l was recorded and observed by node i at time t_k, as illustrated in Fig. 4.*

Proof: According to (9), when node i contacts with node j, the update using the relative clock skews with respect to the underlying actual clock frequency can be calculated as

$$\begin{aligned} f_i(t_k) + \frac{\sum_{l=1}^N w_{il}^T(t_k) f_{il}^T(t_k)}{\sum_{l=1}^N w_{il}^T(t_k)} &= f_i(t_k) + \frac{\sum_{l=1}^N w_{il}^T(t_k)(f_l(t_{il}^d(t_k)) - f_i(t_k))}{\sum_{l=1}^N w_{il}^T(t_k)} \\ &= f_i(t_k) + \frac{\sum_{l=1}^N w_{il}^T(t_k) f_l(t_{il}^d(t_k))}{\sum_{l=1}^N w_{il}^T(t_k)} - \underbrace{\frac{\sum_{l=1}^N w_{il}^T(t_k) f_i(t_k)}{\sum_{l=1}^N w_{il}^T(t_k)}}_{f_i(t_k)} \\ &= \frac{\sum_{l=1}^N w_{il}^T(t_k) f_l(t_{il}^d(t_k))}{\sum_{l=1}^N w_{il}^T(t_k)}. \end{aligned} \tag{14}$$

Note that we have $f_{il}^T(t_k) = f_l(t_{il}^d(t_k)) - f_i(t_k)$ in (14) since the clock compensation is performed for both clock frequency $f_i(t_k^{i,j})$ in (9) and clock skews $f_{il}^T(t_k^{i,j})$ in (10), which preserves the recorded actual clock frequency $f_l(t_{il}^d(t_k))$ upon each contact. ■

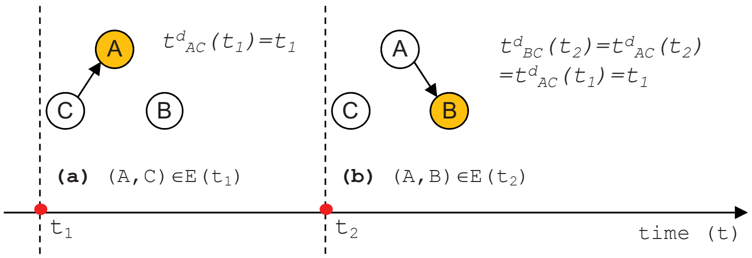


Fig. 4. An illustrative example of delayed information: (a) At time t_1, nodes A and C contact with each other, and node A obtains the clock frequency value of node C ($f_C(t_1)$) directly from node C; (b) At time t_2, nodes A and B contact with each other, and the clock frequency value of node C ($f_C(t^d_{BC}(t_2))$) is forwarded to node B via node A.

Let $F(t_k) = [f_1(t_k); \ldots; f_N(t_k)]^T$ represent the N state vector containing the clock frequencies at time t_k. Based on Lemma 1, the update procedure of the DCS protocol can be formulated as an consensus/agreement problem [23], where the updated frequencies after the kth update can be calculated as

$$f_i(t_k) = \sum_{l=1}^{N} a_{il}(t_k) f_l(t^d_{il}(t_k)), \forall\ i \in V. \tag{15}$$

Define $A(t_k)$ as an $N \times N$ matrix containing normalized weights $a_{il}(t_k) \in [0, 1]$, which is given by

$$a_{il}(t_k) = \begin{cases} w^T_{il}(t_k)/(\sum_{n=1}^{N} w^T_{il}(t_k)), & \text{if } \exists\ j \neq i,\ (i,j) \in E(t_k) \\ I(l = i), & \text{otherwise} \end{cases} \tag{16}$$

where $w^T_{il}(t_k)$ is the decaying weight defined in (11) and $I(\cdot)$ is an indication function which equals 1 if true and 0 otherwise. In (16), the first case corresponds to node i contacts with another node, while the second case occurs when there is no contact between node i and any other node, and thus there is no update for clock frequency of node i. Moreover, in (15), we have $t^d_{ii}(t_k) = t_k$ since each node can acquire the clock information of itself without delay.

Theorem 1. (Convergence of the DCS protocol) *The clock values using the DCS protocol converge to a common value under deterministic mobility scenarios, and converges to the value with probability under random mobility scenarios.*

Proof: Based on the consensus theorem [24, 25], if the following conditions are satisfied, ((1) the update weight matrices are row stochastic, (2) the network is strongly connected, and (3) the communication delay is bounded such that $t_k - B < t^d_{il}(t_k) \leq t_k$) the algorithm guarantees asymptotic consensus: Condition (1) is satisfied for the DCS since it can be easily verified that $A(t_k) \in \Re^{N \times N}$ and $A(t_k)\mathbf{1_N} = \mathbf{1_N} \ \forall \ t_k$, where $\mathbf{1_N} = [1; \ldots ; 1]^T$. Conditions (2) and (3) are satisfied depending on the mobility scenarios. The connectivity graph is strongly connected if there is a path from each vertex in the graph to every other vertex. In mobile networks, a virtual path exists from a source node to a destination node if messages can be forwarded using one or more mobile nodes acting as intermediate nodes, and the communication delay is the sum of inter-contact durations from the source node to the destination node. Thus, the network is strongly connected if there exists a forwarding path from each node in the graph to every other node, i.e., there is no isolated node. The delay is bounded ($t_k - t^d_{il}(t_k) < B$), if the sum of the inter-contact durations over the forwarding path is bounded. Specifically, in deterministic mobility scenarios, such as bus routes [8] and message ferries [26, 27], where the mobility is planned or controlled such that the forwarding paths are consistently available and nodes contact following certain schedule, the clock values *asymptotically converges*. On the other hand, in random mobility scenarios, such as random waypoint (RWP) and random direction (RD), the inter-contact duration is modeled by a probability distribution. As a result, the inter-contact duration is bounded with certain probability. Since an unbounded inter-contact duration leads to an unbounded delay, the DCS protocol *converges with probability* [28] in random mobility scenarios. ■

As the probability for the delay to exceed a bound B is low when B is large according to the analysis [29, 30], the probability for the DCS protocol to converge is high for most scenarios. Note that considering N instead of $N_i^T(t)$ in the analysis does not change the convergence result since the weight values, except for $i = j$, are all initialized to zero and they do not contribute to the updates as if their identities are unknown.

Based on the convergence proof of the DCS protocol, the convergence of the AD protocol can be also proved. The AD protocol is a special case of the DCS protocol having $t^d_{ij}(t_k) = t^d_{ji}(t_k) = t_k$ and $a_{il}(t_k)$ given by

$$a_{il}(t_k) = \begin{cases} 1/2, & \text{if } \exists \ j \neq i, \ (i,j) \in E(t_k) \text{ and } l = i, j \\ I(l = i), & \text{otherwise.} \end{cases} \tag{17}$$

Different convergence proofs for the AD protocol can be found in [17, 20, 24, 31].

Furthermore, since the information stored in the tables becomes less accurate with time and different table entries have different accuracy, a weighting mechanism with a tunable aging parameter (λ) in (11) is adopted in the DCS protocol to better utilize the table entries with higher accuracy. The effect of λ on the convergence speed will be shown by analysis and simulation in Subsection VI-A and Subsection VI-B, respectively.

V. Performance Analysis

In this section, a discrete time analysis is proposed to evaluate the performance of the DCS protocol. Different from the convergence analysis in Subsection IV-D, we consider a fixed time interval (τ) in this section since the analytical complexity is prohibitive to keep all contact histories. For analytical tractability, the following assumptions are made:

1) The movement of all mobile nodes are independent and the pairwise inter-contact duration is approximately exponentially distributed with an average value $1/\gamma$;
2) The density of mobile nodes is low and the probability for more than two mobile nodes to contact with each other simultaneously is negligible.

The assumption 1) holds for numerous random mobility models, such as in random waypoint (RWP) and random direction (RD) [29, 30, 32, 33], and real mobility traces [34, 35]. The performance analysis of the DCS protocol consists of two parts. In Subsection V-A, the table updating procedure is modeled. In Subsection V-B, the performance metrics in terms of the average relative clock offset and the average relative clock skew are evaluated by considering the updated table information.

A. Modeling of Table Updating Procedure

For analytical simplicity, we consider the clock value and clock frequency as the table entries for the DCS protocol, which are equivalent to the clock offset and clock skew since the accurate clock information of each node is available in the analytical model. Denote $X_{il}(\tau_k)$ and $Y_{il}(\tau_k)$ as the clock value and clock frequency, respectively, of node l recorded in the table of node i at time $\tau_k = k\tau$ ($k = 0, 1, 2, \cdots$), which are given by

$$X_{ii}(\tau_k) = C_i(\tau_k), \qquad\qquad Y_{ii}(\tau_k) = f_i(\tau_k) \tag{18}$$

$$X_{il}(\tau_k) = C_{il}^T(\tau_k) + C_i(\tau_k),\ Y_{il}(\tau_k) = f_{il}^T(\tau_k) + f_i(\tau_k),\ i \neq l. \tag{19}$$

Note that $X_{il}(\tau_k)$ and $Y_{il}(\tau_k)$ may not have the same values as $X_{ll}(\tau_k)$ and $Y_{ll}(\tau_k)$ since the table entries $C_{il}^T(\tau_k)$ and $f_{il}^T(\tau_k)$ may be outdated.

1) Table Updating Procedure of the Clock Frequency: The clock frequency $Y_{il}(\tau_k)$ of node l stored in the table of node i can be updated by (1) a direct contact with node l and (2) an indirect contact through some nodes other than l, or (3) remain unchanged without any contact.

First, $Y_{il}(\tau_k)$ can be updated by a direct contact with node l. Since the historic contact information is not available in the discrete time model, the approximated contributing weight is used for analytical tractability. The updated clock frequency of node i when it contacts node l within τ is given by

$$\tilde{Y}_{il}(\tau_{k+1}) = \frac{\sum_{h=1}^{N} w^T_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_{k+1}) Y_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_k)}{\sum_{h=1}^{N} w^T_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_{k+1})} \tag{20}$$

where $w^T_{nh}(\tau_{k+1})$ is the approximated contributing weight with respect to node h in the table of node n and $n^*_{ilh}(\tau_{k+1})$ denotes the node (either node i or node l) with a higher contributing weight with respect to node h ($h = 1, \cdots, N$). Here, the value of $w^T_{nh}(\tau_{k+1})$ is estimated based on the instantaneous clock value and clock frequency as follows

$$w^T_{nh}(\tau_{k+1}) = \begin{cases} 1, & \text{if } \; h = i \text{ or } l \\ \lambda^{T_{nh}(\tau_{k+1})}, & \text{otherwise} \end{cases} \tag{21}$$

where $T_{nh}(\tau_{k+1})$ is the approximated elapsed time since the clock information of node h was recorded. The value of $T_{nh}(\tau_{k+1})$ can be calculated based on the difference between the clock value divided by the difference between the clock frequency, and is given by

$$T_{nh}(\tau_{k+1}) = \begin{cases} \left| \frac{[X_{hh}(\tau_k)+\tau Y_{hh}(\tau_k)]-[X_{nh}(\tau_k)+\tau Y_{nn}(\tau_k)]}{Y_{hh}(\tau_k)-Y_{nn}(\tau_k)} \right|, & \text{if } Y_{hh}(\tau_k) \neq Y_{nn}(\tau_k) \\ \infty, & \text{otherwise.} \end{cases} \tag{22}$$

Also, the value of $n^*_{ilh}(\tau_{k+1})$ is determined based on the accuracy of the table entry as follows

$$n^*_{ilh}(\tau_{k+1}) = \arg\min_{n \in \{i,l\}} |[X_{nh}(\tau_k) + \tau Y_{nn}(\tau_k)] - [X_{hh}(\tau_{k+1}) + \tau Y_{hh}(\tau_k)]| \tag{23}$$

where the terms $\tau Y_{nn}(\tau_k)$ and $\tau Y_{hh}(\tau_k)$ are applied since the clock value of each node changes at a constant rate according to the clock frequency if there is no contact with other nodes during τ. Note that the table entries of clock values are updated according to the clock frequency of the node keeping the table.

Second, $Y_{il}(\tau_k)$ can be updated by an indirect contact with some nodes other than l that has the clock frequency of node l in its table. The updated clock frequency in the table of node i

with respect to node l, when node i contacts with node j $(j \neq i, l)$ within τ, is given by

$$\tilde{Y}_{ijl}(\tau_{k+1}) \;=\; Y_{(n^*_{ijl}(\tau_{k+1}))l}(\tau_k). \tag{24}$$

Third, $Y_{il}(\tau_k)$ remains unchanged if there is no contact between node i and any of the other $(N-1)$ nodes within τ.

Finally, the table updating procedure of the clock frequencies in the DCS protocol considering all three cases can be modeled as

$$Y_{il}(\tau_{k+1}) \;=\; \frac{1-P_0}{(N-1)}\tilde{Y}_{il}(\tau_{k+1}) + \sum_{\substack{j=1\\ j\neq i,l}}^{N} \frac{1-P_0}{(N-1)}\tilde{Y}_{ijl}(\tau_{k+1}) + P_0 Y_{il}(\tau_k), i \neq l \tag{25}$$

where $P_0 = \left(e^{-\gamma\tau}\right)^{N-1}$ is the probability that there is no contact between node i and any of the other $(N-1)$ nodes within τ. An approximation is made in the analysis that the probability for more than one contact between node i and any of the other $(N-1)$ nodes within τ is negligible. Since node i contacts with any of the other $(N-1)$ nodes with the same probability, the factor $\frac{1}{(N-1)}$ is applied.

2) Table Updating Procedure of the Clock Value: The clock value $X_{il}(\tau_k)$ of node l stored in the table of node i is updated also for the same three cases used in the modeling of the table updating procedure of the clock frequency. However, different from the table updating procedure of the clock frequency, the table entries of clock values change not only when two nodes contact with each other, but also over time according to the clock frequency.

First, the updated clock value of node i when it contacts node l within τ is given by

$$\tilde{X}_{il}(\tau_{k+1}) \;=\; \frac{\sum_{h=1}^{N} w^T_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_{k+1})\left[X_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_k) + \tau Y_{(n^*_{ilh}(\tau_{k+1}))(n^*_{ilh}(\tau_{k+1}))}(\tau_k)\right]}{\sum_{h=1}^{N} w^T_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_{k+1})} \tag{26}$$

where the term $\tau Y_{(n^*_{ilh}(\tau_{k+1}))(n^*_{ilh}(\tau_{k+1}))}(\tau_k)$ is applied to the numerator since the clock value changes according to the clock frequency during τ.

Second, the updated clock value in the table of node i with respect to node l, when node i contacts with node j $(j \neq i, l)$ within τ, is given by

$$\tilde{X}_{ijl}(\tau_{k+1}) = X_{(n^*_{ijl}(\tau_{k+1}))l}(\tau_k) + \tau Y_{(n^*_{ijl}(\tau_{k+1}))(n^*_{ijl}(\tau_{k+1}))}(\tau_k). \tag{27}$$

Third, the clock value of node l stored in node i when there is no contact between node i and any of the other $(N-1)$ nodes within τ is given by $X_{il}(\tau_k) + \tau Y_{ii}(\tau_k)$.

Finally, the table updating procedure of the clock values in the DCS protocol can be modeled as

$$X_{il}(\tau_{k+1}) = \frac{1-P_0}{(N-1)}\tilde{X}_{il}(\tau_{k+1}) + \sum_{\substack{j=1\\ j\neq i,l}}^{N} \frac{1-P_0}{(N-1)}\tilde{X}_{ijl}(\tau_{k+1}) + P_0\left[X_{il}(\tau_k) + \tau Y_{ii}(\tau_k)\right], i \neq l. \quad (28)$$

B. Evaluation of Performance Metrics

Based on the modeling of the table updating procedure, the update of the clock value and clock frequency are given by

$$X_{ii}(\tau_{k+1}) = \sum_{\substack{l=1\\ l\neq i}}^{N} \frac{1-P_0}{(N-1)}\tilde{X}_{il}(\tau_{k+1}) + P_0\left[X_{ii}(\tau_k) + \tau Y_{ii}(\tau_k)\right] \quad (29)$$

$$Y_{ii}(\tau_{k+1}) = \sum_{\substack{l=1\\ l\neq i}}^{N} \frac{1-P_0}{(N-1)}\tilde{Y}_{il}(\tau_{k+1}) + P_0 Y_{ii}(\tau_k) \quad (30)$$

where $\tilde{X}_{il}(\tau_{k+1})$ and $\tilde{Y}_{il}(\tau_{k+1})$ are given by (26) and (20), respectively.

Define the system state of the DCS protocol at time τ_k as $\mathbf{S}_{DCS}(\tau_k) = \{X_{il}(\tau_k), Y_{il}(\tau_k) | i, l = 1, \cdots, N\}$. For the updating procedure in (25), and (28)-(30), only the current system state $\mathbf{S}_{DCS}(\tau_k)$ is needed to obtain the next system state $\mathbf{S}_{DCS}(\tau_{k+1})$. Therefore, we can define an operation $F_{DCS}(\cdot)$ such that $\mathbf{S}_{DCS}(\tau_{k+1}) = F_{DCS}(\mathbf{S}_{DCS}(\tau_k))$. Given the initial clock values $X_{ii}(0) = C_i(0)$ and clock frequency $Y_{ii}(0) = f_i(0)$, and the initial values of table entries $X_{il}(0) = X_{ii}(0)$ and $Y_{il}(0) = Y_{ii}(0)$ ($i \neq l$), the system state of the DCS protocol at time τ_k can be calculated as

$$\mathbf{S}_{DCS}(k) = F^k_{DCS}(\mathbf{S}_{DCS}(0)) \quad (31)$$

where $F^k_{DCS}(\cdot)$ denotes applying the operation $F_{DCS}(\cdot)$ by k times. Then the performance metrics at time τ_k ($C_{avg}(\tau_k)$ and $f_{avg}(\tau_k)$) of the DCS protocol can be calculated. Based on the analytical model of the DCS protocol, we can also evaluate the performance of the AD protocol without considering the table updating procedure. The detailed derivation is presented in Appendix.

VI. Performance Evaluation

In this section, numerical and simulation results are presented to evaluate the performance of the proposed DCS protocol. The numerical results (Ana.) are based on the analytical model

TABLE II

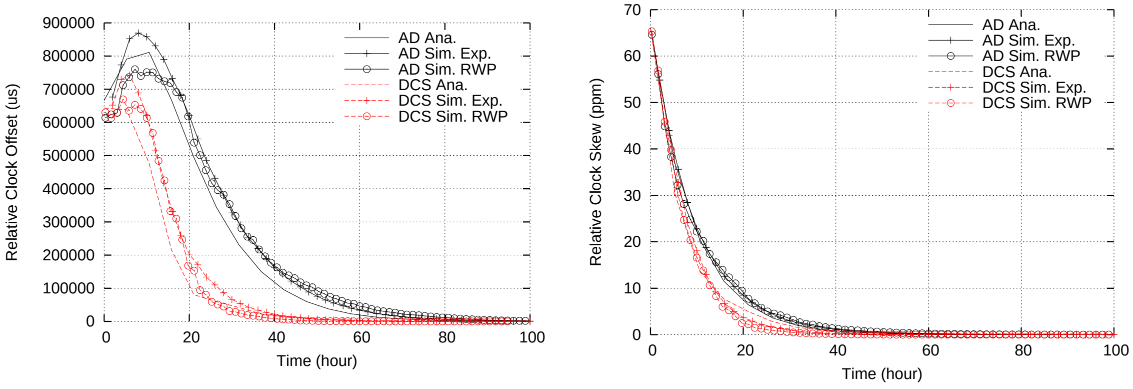
DEFAULT SYSTEM PARAMETER

Parameter	**Value**
Simulation Time	550 hours
Map Size ($M \times M$)	50 km × 50 km, 20 km × 20 km
Number of Nodes (N)	50
Node Speed (v)	Uniform$(0.5, 1.5)$ m/s
Pause Time	0 - 120 s
Radio Transmission Range (R)	250 m
Initial Skew ($C'_i(t_0)$)	Uniform$(-100, +100)$ ppm
Initial Offset ($C_i(t_0)$)	Uniform$(-10^6, +10^6)$ μs
Aging Constant (λ)	$1 - 10^{-5}$

presented in Section V, while the simulation results are obtained based on a discrete event-driven simulator using exponentially distributed inter-contact durations (Sim. Exp.) and the Opportunistic Network Environment (ONE) simulator [36] with an additional implementation of the clock synchronization mechanism (Sim. RWP). Note that in the analysis, contacts are assumed to be one-to-one for analytical tractability. However, since both simulators are event-driven, table updating procedure is performed whenever there is a new connection created between any pair of nodes, regardless of the number of simultaneous connections. If there are multiple new contacts, connections are created in the increasing order of the node ID. Still, chances of having multiple new connections in the same event interval is negligibly small for all nodes. Default system parameters are given in Table II.

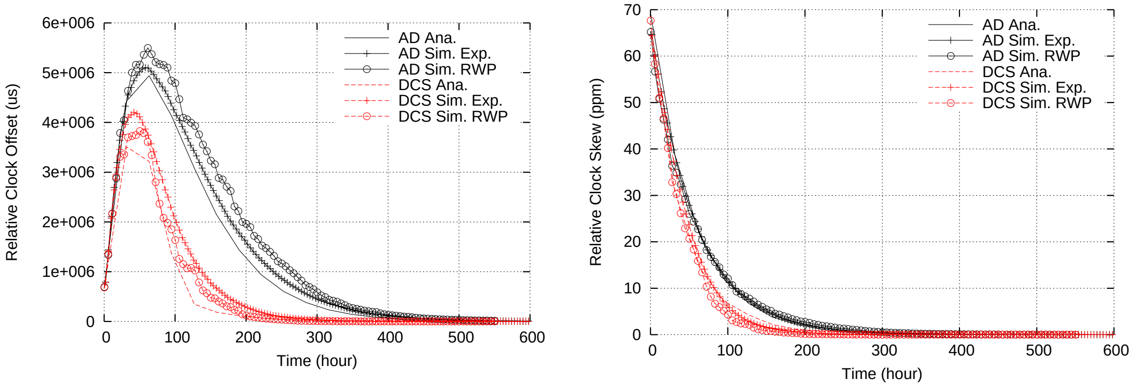
A. Numerical Results

For the RWP mobility model under consideration, the pairwise inter-contact duration is approximately exponentially distributed [30, 32]. Both theoretical and experimental results can be used to estimate the pairwise inter-contact rate γ. In this work, γ is estimated as the reciprocal of the average pairwise inter-contact duration obtained from simulation results. We consider two network coverage areas. The density of mobile nodes is configured to create a disconnected network with a node degree much smaller than 1 [37] and hence nodes are mostly disconnected. The pairwise inter-contact rate for $M = 20$ km and $M = 50$ km is $2.15 \times 10^{-6} s^{-1}$ and $4.05 \times 10^{-7} s^{-1}$, respectively. Since the clock values $X_{ii}(\tau_k)$ and frequencies $Y_{ii}(\tau_k)$ of the N



(a) (b)

Fig. 5. Convergence of clock offset and skew ($M = 20$ km): (a) Average relative clock offset; (b) Average relative clock skew.

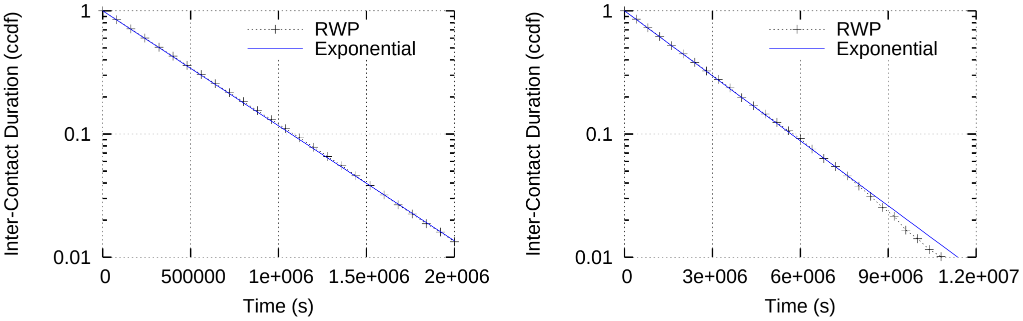


(a) (b)

Fig. 6. Convergence of clock offset and skew ($M = 50$ km): (a) Average relative clock offset; (b) Average relative clock skew.

nodes are updated simultaneously for each time period τ in (29) and (30), we consider $\tau = \frac{N}{\binom{N}{2}\gamma}$, where $\binom{N}{2}\gamma$ equals the average inter-contact rate for all combinations of pairwise contacts. Note that, during τ, N updates of clock values and frequencies take place on average.

1) Impact of Node Density: Figs. 5 and 6 show the average relative clock offset and skew for $M = 20$ km and $M = 50$ km, respectively. For both node densities, the clock offset of the DCS protocol converges faster than that of the AD protocol. The divergence in the beginning is due to large initial relative clock skews that produce large time differences during long delays between connections. The clock offsets gradually converge to zero as the clock skews converge exponentially with time as shown in Figs. 5(b) and 6(b). Since nodes have more contact opportunities among them at higher node densities, the convergence speed is faster and



(a)

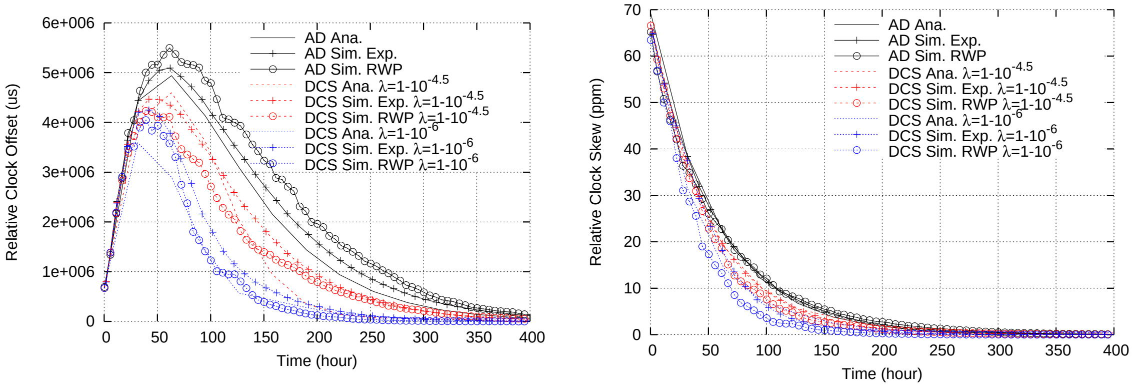
(b)

Fig. 7. Distribution of inter-contact duration: (a) $M = 20$ km, $\gamma = 2.15 \times 10^{-6} s^{-1}$; (b) $M = 50$ km, $\gamma = 4.05 \times 10^{-7} s^{-1}$.

the initial divergence of relative clock offset is smaller when $M = 20$ km.

Overall, for both the AD and DCS protocols, the analytical and simulation results, using the same average inter-contact durations, match well with each other. The estimation error in Ana. and Sim. Exp. in comparison with Sim. RWP are caused by the fact that the inter-contact durations of the RWP mobility model are only approximately exponentially distributed, as shown in Fig. 7. Similar observations have been obtained for the RD mobility model [32]. In addition, although, the Ana. result and the Sim. Exp. result of relative clock skew match well with each other, the Ana. result of relative clock offset is smaller than the Sim. Exp. result. The reason is that, the table entries in the discrete time analysis of Ana. are updated at the end of each discrete period (with duration τ), whereas the table entries in the event-driven simulator Sim. Exp. are updated at random moments. Consequently, considering the updated entries at the end of each period with constant clock frequencies, the Ana. result indicates smaller relative clock offset results by neglecting the divergence of relative clock offset in between the discrete periods.

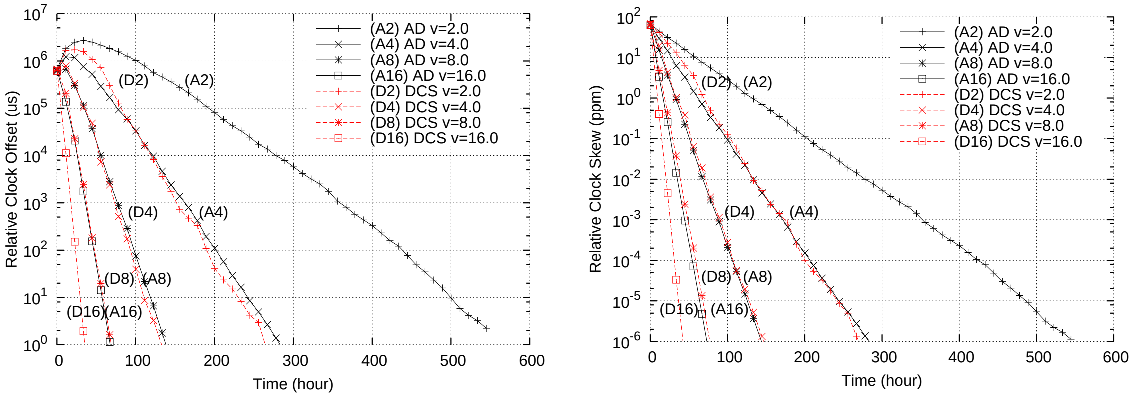
2) Impact of Aging Parameter: Fig. 8 shows the average relative clock offset and skew for the DCS protocol with respect to time t for various λ values. Although the elapsed time for the contributing weight calculation is approximated by (22) based on the instantaneous clock value and clock frequency, the proposed analytical model provides a good estimate of the two performance metrics for different λ values in all the scenarios. In practical applications, when the pair-wise inter-contact duration is approximately exponentially distributed, the proposed analytical model can be used as an efficient tool to facilitate the system performance estimation.



(a)

(b)

Fig. 8. Convergence under different aging parameters: (a) Average relative clock offset; (b) Average relative clock skew.



(a)

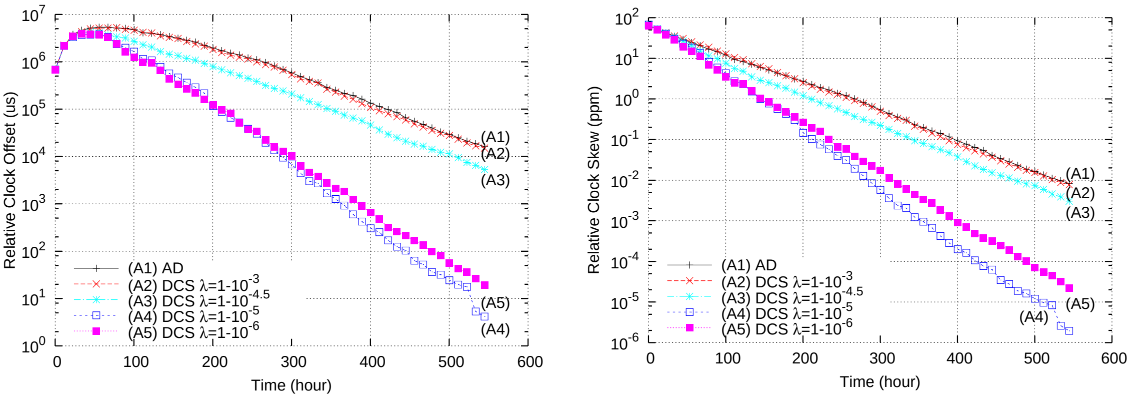
(b)

Fig. 9. Impact of node speed: (a) Average relative clock offset; (b) Average relative clock skew.

B. Simulation Results

The convergence of the DCS protocol is verified using the numerical results which match well with the simulation results. However, for the analytical tractability, the analytical model is derived only for random mobility models with some stochastic approximations. Here, the performance of the DCS protocols is further evaluated using extensive simulations under different node speeds, aging parameters, frequency changes, clock estimation errors, and mobility models.

1) Impact of Node Speed: Fig. 9 shows the impact of node speed v in m/s. For all the node speeds, the clock offset of the DCS protocol converges faster than that of the AD protocol. As the node speed increases, the convergence speed increases and the initial divergence of relative clock offsets decreases. This is because at higher node speeds, nodes have more contact opportunities



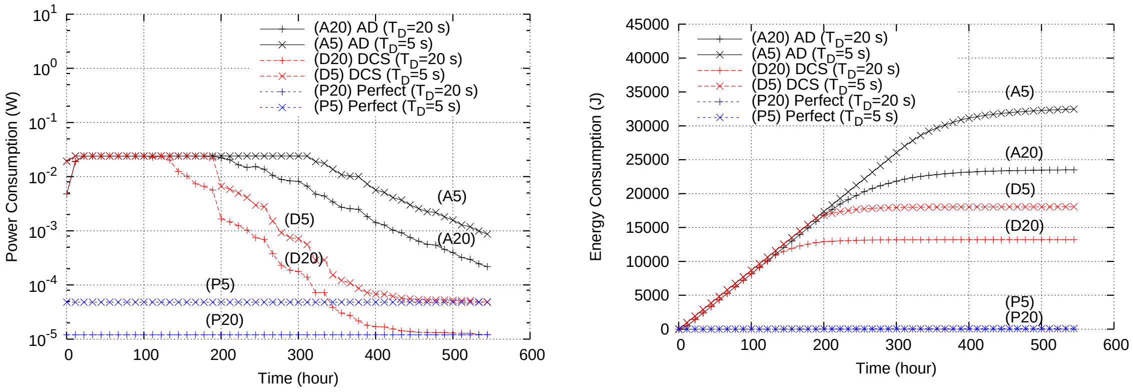
(a) (b)

Fig. 10. Impact of aging parameter: (a) Average relative clock offset; (b) Average relative clock skew.

among them.

2) Impact of Aging Parameter: The impact of the tuning parameter λ in (11) is shown in Fig. 10. For this specific scenario, $\lambda = 1 - 10^{-6}$ achieves the lowest relative clock offset until about 200 h, but at the end of the simulation, the average relative clock offsets of the DCS protocol with $\lambda = 0$ (same as the AD protocol), $\lambda = 1 - 10^{-3}$, $\lambda = 1 - 10^{-4.5}$, $\lambda = 1 - 10^{-5}$, and $\lambda = 1 - 10^{-6}$ are 13148 μs, 12632 μs, 4463 μs, 3 μs, and 13 μs, respectively. This result indicates that there exists some optimal λ value for each scenario. The impact of different aging parameter values can also be seen in Fig. 10(b) for the skew result. The aging parameter can be selected to effectively discard the information that becomes less accurate over time. If some nodes fail or become isolated so that they are unable to propagate their information to the whole network, the outdated information coupled with long inter-contact delays under the disconnected network topology can increase the average relative clock values. On the other hand, when $\lambda = 1 - 10^{-3}$, the DCS protocol operates similarly to the AD protocol since $w_{ij}^T(t)$ quickly approaches zero, and by the time a new contact is discovered, $C_{ij}^T(t)$ and $f_{ij}^T(t)$ have a negligible share in the compensation algorithm. However, how to analytically acquire the optimal values of λ for different scenarios is still an open problem.

3) Impact on Energy Consumption: In order to demonstrate the impact of synchronization error on energy consumption, simulation result for the average energy consumption in neighbor discovery is shown in Fig. 11. Each node uses a sleep schedule with a duty cycle T_D and awake periods with lengths $2C_{max}(t) + T_A \leq T_D$ where $C_{max}(t) = \max|C_{ij}(t)|, \forall\ i, j$, is the

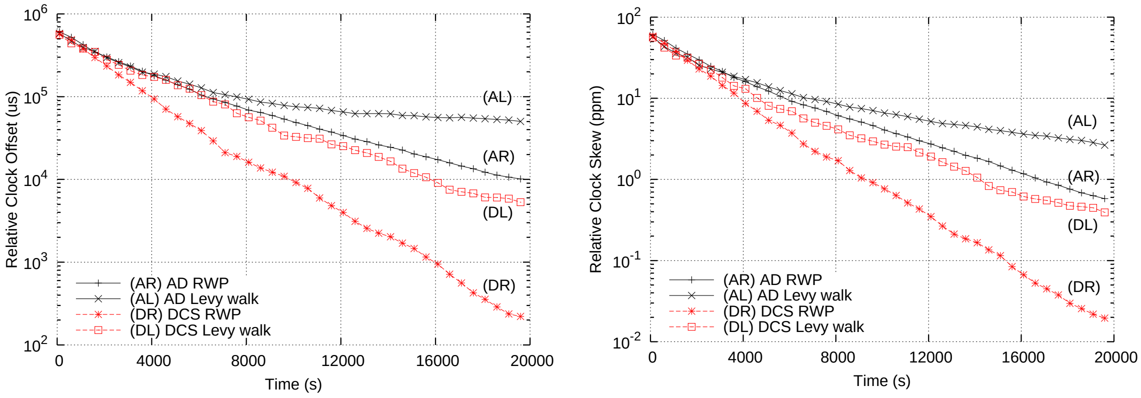


(a) (b)

Fig. 11. Impact on energy consumption: (a) Average power consumption; (b) Cumulative average energy consumption.

maximum relative clock offset and $T_A = 10$ ms is the minimum awake period required for exchanging connection setup messages. The power consumption model of sensor motes [38] is used where power consumptions during the idle mode and sleep mode are 24.0 mW and 0.03 mW, respectively. Fig. 11(a) shows a large power consumption difference between the cases with and without synchronization errors. For the AD and DCS protocols, the constant power consumption periods in the beginning of the simulation time are caused by nodes in constant awake mode as awake periods cannot exceed the duty cycle. As clock values converge, the energy consumption rate approaches the lower bound set by the perfect clock synchronization. In addition, energy consumption is higher for a shorter duty cycle due to more frequent awake periods. Since the AD protocol takes a longer time to converge than the DCS protocol, the overall energy consumption is reduced almost by half in the DCS protocol as shown in Fig. 11(b). The simulation results indicate how the synchronization error can significantly increase the energy consumption.

4) Impact of Mobility Model: Levy walk mobility model [39] closely resembles the movement length and pause time distributions of human walks. These distributions follow truncated power law distributions where a node is much more likely to move to locations closer to its current location than that in the RWP mobility model. For a fair comparison, we have chosen mobility parameters such that both mobility models have a comparable number of contacts during the



(a) (b)

Fig. 12. Impact of mobility model ($M = 5$ km): (a) Average relative clock offset; (b) Average relative clock skew.

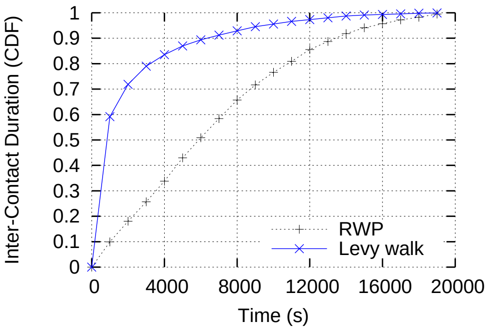
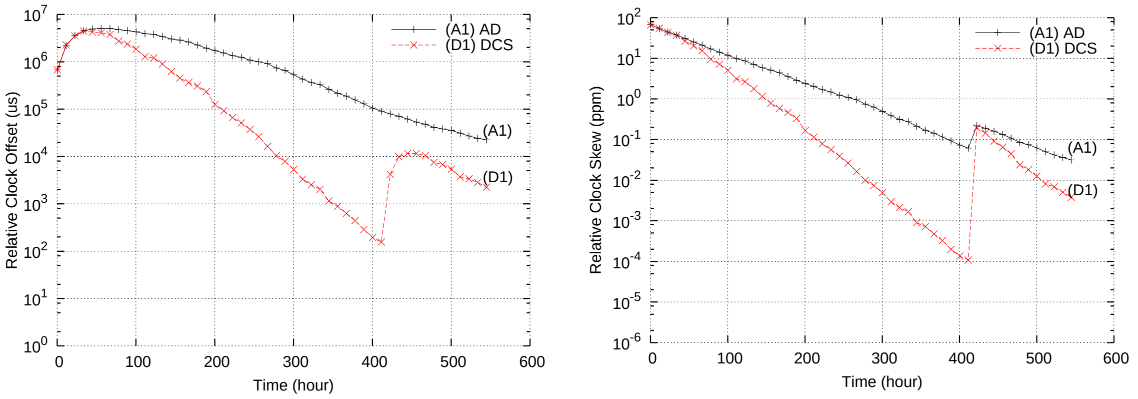


Fig. 13. Distribution of inter-contact duration ($M = 5$ km)

simulations[1]. The average number of contacts during the simulations for RWP and Levy walk mobility model are 1680 and 1682, respectively. Fig. 12(a) shows that the clock offset converges faster for RWP. Even though the total number of contacts is slightly higher for the Levy walk mobility, nodes for Levy walk have a higher probability to contact with the nodes that are closer to their current location (i.e., node mobility is less diffusive), as shown in Fig. 13. Therefore, the convergence speed depends on how evenly the probability of meeting different nodes is distributed.

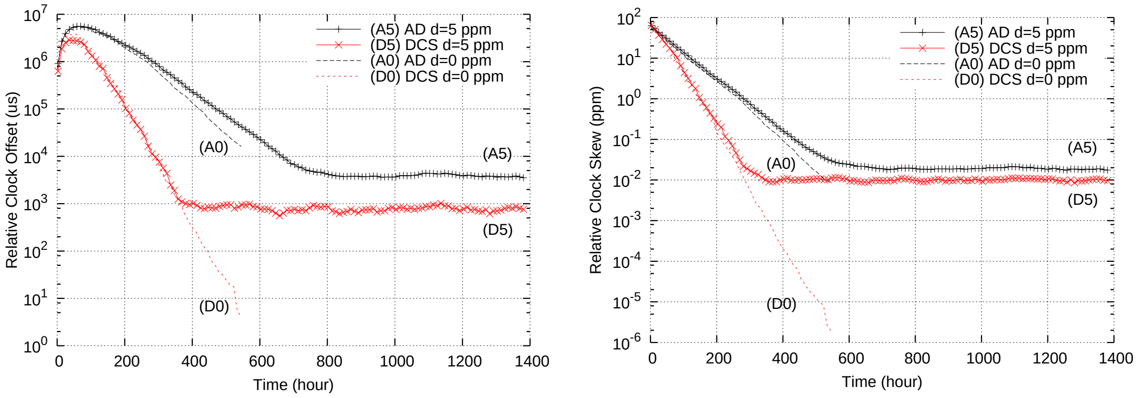
5) Impact of Clock Frequency Instability: We investigate the impact of the clock frequency instabilities on the clock convergence. First, the impact of short-term clock frequency instability

[1] Exact parameters used to generate the traces of Levy walk mobility patterns are as follows: power-law slope of flight length $\alpha = 0.4$, power-law slot of pause time $\beta = 0.5$, scale factor of flight length = 2.5, truncated flight length = 3000 m, and truncated pause time = 120 s. Please refer to [39] for the detailed description of the model and the choice of the parameters used in the trace generation.



(a) (b)

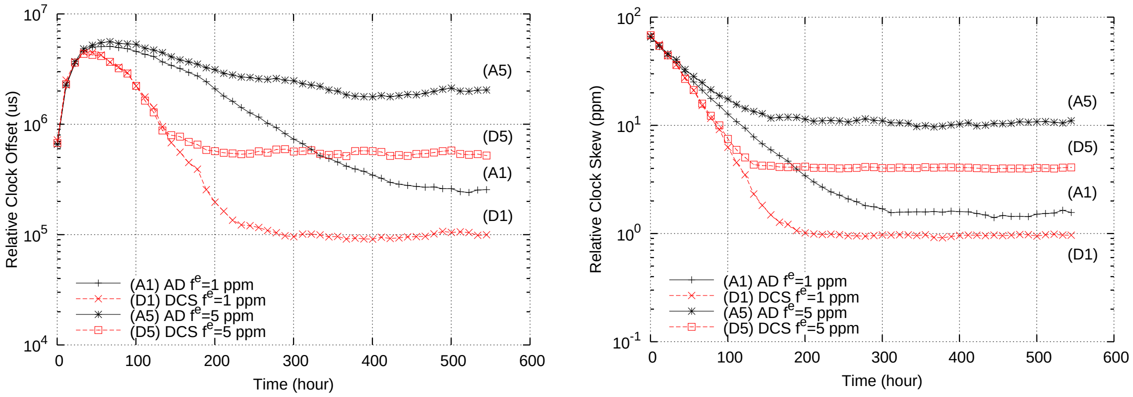
Fig. 14. Impact of short-term clock frequency stability due to a temperature change: (a) Average relative clock offset; (b) Average relative clock skew.



(a) (b)

Fig. 15. Impact of long-term clock frequency instability due to the oscillator aging: Average relative clock offset; (b) Average relative clock skew.

is studied. The clock frequency of half of nodes are changed at 420 h assuming a commonly used 32 kHz tuning forkcrystal with a known parabolic coefficient of -0.04 ppm/°C^2 [40] and a temperature change of -10 °C. As shown in Fig. 14, the relative clock values of the AD protocol are not significantly affected due to the fact that the additional clock frequency error is relatively small compared with the current error. Although the DCS protocol experiences a sudden increase in the clock error at 420 h, it is able to quickly recover from it and continue to converge exponentially. Second, the impact of long-term clock frequency instability due to oscillator aging is studied. Each node is assigned an aging rate (d) modeled as a uniform distribution with a



(a)

(b)

Fig. 16. Impact of relative clock estimation error: (a) Average relative clock offset; (b) Average relative clock skew.

bound of ±5 ppm per year [40]. As shown in Fig. 15, for both the AD and the DCS protocols, the convergence speeds are slightly slower than those without the oscillator aging (shown as dashed lines). Also, the relative clock offset and skew converge to some limit. This is because, even if the clock frequency can be adjusted upon each contact using the relative clock skew compensation, the clock frequency constantly deviates from the compensated clock frequency due to the uncontrollable frequency change caused by the aging. Numerically, after 400 h, the average relative clock offsets of the AD and the DCS protocols are 771 μs and 3539 μs, respectively.

6) Impact of Relative Clock Estimation Error: Due to the uncertainty in message delays and limited accuracy of the linear estimators, clock estimation error may exist. The relative offset estimation error (C^e) and the relative skew estimation error (f^e) are modeled as a uniform distribution with bounds of ±5 μs [19, 41, 42] and ±1 ppm [43], respectively. The bound of the relative skew estimation error corresponds to 1 percent of the maximum clock frequency error. As shown in Fig. 16, similar to the simulation results in Fig. 15, the relative clock offset and skew converge to some limit. Also, the limit is higher for the higher estimation error using ±5 ppm. Therefore, the clock convergence speed and the limit are sensitive to the clock estimation error. Various clock offset and skew estimation methods [43] can be used to improve the estimation error. For the same estimation error, the DCS outperforms the AD protocol since the effect of random error is mitigated by the clock information accumulated over multiple contacts.

VII. Conclusion

The clock synchronization is an essential requirement for efficient network protocol operations in DTNs. To achieve global clock synchronization in DTNs, we have proposed a distributed asynchronous clock synchronization protocol that uses the relative clock information spread among nodes. Analytical and simulation results demonstrate that the DCS protocol can achieve faster convergence speed than existing distributed asynchronous clock synchronization protocols under various network conditions. A smaller clock error from the DCS protocol can provide more accurate timing information in data collection from a physical environment and render sleep scheduling mechanisms more energy efficient.

Appendix
Performance Analysis of the AD Protocol

Since there is no table update in the AD protocol, the update of the clock value $X_{ii}(\tau_k)$ and clock frequency $Y_{ii}(\tau_k)$ of each node can be modeled as

$$\begin{aligned} X_{ii}(\tau_{k+1}) &= \sum_{\substack{l=1\\ l\neq i}}^{N} \frac{1-P_0}{(N-1)} \overline{X}_{il}(\tau_{k+1}) + P_0 \left[X_{ii}(\tau_k) + \tau Y_{ii}(\tau_k)\right] \qquad (32)\\ Y_{ii}(\tau_{k+1}) &= \sum_{\substack{l=1\\ l\neq i}}^{N} \frac{1-P_0}{(N-1)} \overline{Y}_{il}(\tau_{k+1}) + P_0 Y_{ii}(\tau_k), \qquad (33)\\ & \text{for } i = 1, \cdots, N \end{aligned}$$

where $\overline{X}_{il}(\tau_{k+1})$ is the average clock value between nodes i and l if they contact with each other within τ, given by

$$\overline{X}_{il}(\tau_{k+1}) = \frac{[X_{ii}(\tau_k) + \tau Y_{ii}(\tau_k)] + [X_{ll}(\tau_k) + \tau Y_{ll}(\tau_k)]}{2}, \qquad (34)$$

and $\overline{Y}_{il}(\tau_{k+1})$ is the average clock frequency between nodes i and l if they contact with each other within τ, given by

$$\overline{Y}_{il}(\tau_{k+1}) = \frac{Y_{ii}(\tau_k) + Y_{ll}(\tau_k)}{2}. \qquad (35)$$

Define the system state of the AD protocol at time τ_k as $\mathrm{S}_{AD}(\tau_k) = \{X_{ii}(\tau_k), Y_{ii}(\tau_k) | i = 1, \cdots, N\}$. Similar to the performance analysis of the DCS protocol, the operation $F_{AD}(\cdot)$ is also defined for the AD protocol. Given the initial clock value $X_{ii}(0) = C_i(0)$ and clock frequency $Y_{ii}(0) = f_i(0)$, the system state of the AD protocol at time τ_k can be calculated as

$$\mathbf{S}_{AD}(\tau_k) = F^k_{AD}(\mathbf{S}_{AD}(0)). \tag{36}$$

Then, the performance metrics at time τ_k, $C_{avg}(\tau_k)$ and $f_{avg}(\tau_k)$, of the AD protocol can be calculated.

REFERENCES

[1] B. J. Choi and X. Shen, "Distributed Clock Synchronization in Delay Tolerant Networks," in *Proc. IEEE ICC*, May. 2010.

[2] "Delay-Tolerant Networking Research Group (DTNRG)," http://www.dtnrg.org/.

[3] K. R. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *Proc. ACM SIGCOMM*, Aug. 2003.

[4] "Special Issue on Delay and Disruption Tolerant Wireless Communication Systems," *IEEE J. Select. Areas Commun.*, vol. 26, no. 5, pp. 745–836, June 2008.

[5] H. Jun, M. H. Ammar, and E. W. Zegura, "Power Management in Delay Tolerant Networks: A Framework and Knowledge-Based Mechanisms," in *Proc. IEEE SECON*, Sept. 2005.

[6] B. J. Choi and X. Shen, "Adaptive Asynchronous Clock based Power Saving Protocols for Delay Tolerant Networks," in *Proc. IEEE GLOBECOM*, Nov.–Dec. 2009.

[7] Y. Xi, M. Chuah, and K. Chang, "Performance Evaluation of a Power Management Scheme for Disruption Tolerant Network," *Lecture Notes in Computer Science*, vol. 12, no. 5–6, pp. 370–380, Dec. 2007.

[8] N. Banerjee, M. D. Corner, and B. N. Levine, "An Energy-Efficient Architecture for DTN Throwboxes," in *Proc. IEEE INFOCOM*, April 2007.

[9] K. Römer, "Time Synchronization in Ad Hoc Networks," in *Proc. ACM MobiHoc*, Oct. 2001.

[10] J. Burbank, "Network Time Protocol Version 4 Protocol and Algorithms Specification," IETF, Internet-Draft draft-ietf-ntp-ntpv4-proto-11, Sept. 2008, work in progress.

[11] W. Su and I. F. Akyildiz, "Time-Diffusion Synchronization Protocol for Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 13, no. 2, pp. 384–397, 2005.

[12] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-Sync Protocol for Sensor Networks," in *Proc. ACM SenSys*, Nov. 2003.

[13] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization Using Reference Broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[14] Q. Ye and L. Cheng, "DTP: Double-Pairwise Time Protocol for Disruption Tolerant Networks," in *Proc. IEEE ICDCS*, June 2008.

[15] J.-P. Sheu, C.-M. Chao, and C.-W. Sun, "A Clock Synchronization Algorithm for Multi-Hop Wireless Ad Hoc Networks," in *Proc. IEEE ICDCS*, June 2004.

[16] D. Zhou and T. H. Lai, "An Accurate and Scalable Clock Synchronization Protocol for IEEE 802.11-Based Multihop Ad Hoc Networks," *IEEE Trans. Parallel and Distributed System*, vol. 18, no. 12, pp. 1797–1808, 2007.

[17] Q. Li and D. Rus, "Global Clock Synchronization in Sensor Networks," *IEEE Trans. Computers*, vol. 55, no. 2, pp. 214–226, 2006.

[18] P. Sommer and R. Wattenhofer, "Gradient Clock Synchronization in Wireless Sensor Networks," in *Proc. ACM/IEEE IPSN*, April 2009.

[19] C. H. Rentel and T. Kunz, "A Mutual Network Synchronization Method for Wireless Ad Hoc and Sensor Networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 633–646, 2008.

[20] M. Sasabe and T. Takine, "A Simple Scheme for Relative Time Synchronization in Delay Tolerant MANETs," in *Proc. Int. Conf. Intelligent Networking and Collaborative Systems*, Nov. 2009.

[21] A. A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," in *Proc. IEEE INFOCOM*, April 2006.

[22] M. D. Lemmon, J. Ganguly, and L. Xia, "Model-based Clock Synchronization in Networks with Drifting Clocks," in *Proc. IEEE PRDC*, Dec. 2000.

[23] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1997.

[24] J. Wolfowitz, "Products of Indecomposable, Aperiodic, Stochastic Matrices," *Proceedings of the American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963. [Online]. Available: http://www.jstor.org/stable/2034984

[25] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis, "Convergence in Multiagent Coordination, Consensus, and Flocking," in *Proc. IEEE CDC-ECC*, Dec. 2005.

[26] W. Zhao, M. Ammar, and E. Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks," in *Proc. ACM MobiHoc*, May 2004.

[27] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks," in *Proc. IEEE SNPA*, April 2003.

[28] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Companies, 1991.

[29] T. Small and Z. Haas, "Quality of Service and Capacity in Constrained Intermittent-Connectivity Networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 7, pp. 803–814, July 2007.

[30] R. Groenevelt, P. Nain, and G. Koole, "The Message Delay in Mobile Ad Hoc Networks," *Perform. Eval.*, vol. 62, pp. 210–228, 2005.

[31] P. Denantes, F. Benezit, P. Thiran, and M. Vetterli, "Which Distributed Averaging Algorithm Should I Choose for my Sensor Network," in *Proc. IEEE INFOCOM*, April 2008.

[32] T. Spyropoulos, A. Jindal, and K. Psounis, "An Analytical Study of Fundamental Mobility Properties for Encounter-based Protocols," *Int. J. Auton. Adapt. Commun. Syst.*, vol. 1, no. 1, pp. 4–40, 2008.

[33] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," *IEEE/ACM Trans. Networking*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[34] K. Lee, Y. Yi, J. Jeong, H. Won, I. Rhee, and S. Chong, "Max-Contribution: On Optimal Resource Allocation in Delay Tolerant Networks," in *Proc. IEEE INFOCOM*, Mar. 2010.

[35] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, and L. M. Ni, "Recognizing Exponential Inter-Contact Time in VANETs," in *Proc. IEEE INFOCOM*, Mar. 2010.

[36] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Proc. ICST SIMUTools*, Mar. 2009.

[37] C. Bettstetter, "On the Minimum Node Degree and Connectivity of a Wireless Multihop Network," in *Proc. ACM MobiHoc*, June 2002.

[38] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance Measurements of Motes Sensor Networks," in *Proc. ACM MSWiM*, Oct. 2004.

[39] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the Levy-walk Nature of Human Mobility," in *Proc. IEEE INFOCOM*, April 2008.

[40] J. R. Vig, "Introduction to Quartz Frequency Standards," Army Research Laboratory, Research and Development Technical Report SLCET-TR-92-1 (Rev. 1), Oct. 1992.

[41] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The Flooding Time Synchronization Protocol," in *Proc. ACM SenSys*, Nov. 2004.

[42] J. Hill and D. Culler, "Mica: A Wireless Platform for Deeply Embedded Networks," *Micro, IEEE*, vol. 22, no. 6, pp. 12–24, Nov.-Dec 2002.

[43] E. Serpedin and Q. M. Chaudhari, *Synchronization in Wireless Sensor Networks: Parameter Estimation, Peformance Benchmarks, and Protocols*. Cambridge University Press, 2009.