Energy Efficient Protocols for Delay Tolerant Networks

by

Bong Jun Choi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2011

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The delay tolerant networks (DTNs) is characterized by frequent disconnections and long delays of links among devices due to mobility, sparse deployment of devices, attacks, and noise, etc. Considerable research efforts have been devoted recently to DTNs enabling communications between network entities with intermittent connectivity. Unfortunately, mobile devices have limited energy capacity, and the fundamental problem is that traditional power-saving mechanisms are designed assuming well connected networks. Due to much larger inter-contact durations than contact durations, devices spend most of their life time in the neighbor discovery, and centralized power-saving strategies are difficult. Consequently, mobile devices consume a significant amount of energy in the neighbor discovery, rather than in infrequent data transfers. Therefore, distributed energy efficient neighbor discovery protocols for DTNs are essential to minimize the degradation of network connectivity and maximize the benefits from mobility.

In this thesis, we develop sleep scheduling protocols in the medium access control (MAC) layer that are adaptive and distributed under different clock synchronization conditions: synchronous, asynchronous, and semi-asynchronous. In addition, we propose a distributed clock synchronization protocol to mitigate the clock synchronization problem in DTNs. Our research accomplishments are briefly outlined as follows:

Firstly, we design an adaptive exponential beacon (AEB) protocol. By exploiting the trend of contact availability, beacon periods are independently adjusted by each device and optimized using the distribution of contact durations. The AEB protocol significantly reduces energy consumption while maintaining comparable packet delivery delay and delivery ratio.

Secondly, we design two asynchronous clock based sleep scheduling (ACDS) protocols. Based on the fact that global clock synchronization is difficult to achieve in general, predetermined patterns of sleep schedules are constructed using hierarchical arrangements of cyclic difference sets such that devices independently selecting different duty cycle lengths are still guaranteed to have overlapping awake intervals with other devices within the communication range.

Thirdly, we design a distributed semi-asynchronous sleep scheduling (DSA) protocol. Although the synchronization error is unavoidable, some level of clock accuracy may be possible

for many practical scenarios. The sleep schedules are constructed to guarantee contacts among devices having loosely synchronized clocks, and parameters are optimized using the distribution of synchronization error. We also define conditions for which the proposed semi-asynchronous protocol outperforms existing asynchronous sleep scheduling protocols.

Lastly, we design a distributed clock synchronization (DCS) protocol. The proposed protocol considers asynchronous and long delayed connections when exchanging relative clock information among nodes. As a result, smaller synchronization error achieved by the proposed protocol allows more accurate timing information and renders neighbor discovery more energy efficient.

The designed protocols improve the lifetime of mobile devices in DTNs by means of energy efficient neighbor discoveries that reduce the energy waste caused by idle listening problems.

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Xuemin (Sherman) Shen, for his continuous encouragement, guidance, understanding, and support from the initial to the final stage of my Ph.D. study. Without his invaluable advice and deep insight, this thesis would not have been possible. He has become my role model for his professional knowledge and strong commitment toward the high quality research. More importantly, I have learned from him the highly positive and self motivated attitude. He has not only been my supervisor, but also the role model of integrity, respect, and responsibility. From the bottom of my heart, I sincerely thank Professor Shen for everything that I have achieved during the course of my Ph.D. study at University of Waterloo.

I would also like to thank Professor Liang-Liang Xie, Professor Fakhri Karray, Professor Liping Fu, and Professor Rong Zheng for serving on my thesis advisory committee. I appreciate their precious time and effort in reading my thesis and providing insightful comments and valuable suggestions that helped to improve the quality of the thesis.

I would like to express my great gratitude to Professor Weihua Zhuang for the knowledge of stochastic processes and valuable discussions in solving key problems. I am also thankful to Professor Pin-Han Ho for his guidance and support during the first and second year of the Ph.D. program.

I feel fortunate to work with many wonderful researchers in the Broadband Communications Research Group (BBCR). My deep appreciation goes to Hao Liang, Dr. Hany Samuel, Dr. Hangguan Shan, Rongxing Lu, Tom H. Luan, Dr. Lin X. Cai, and Dr. Haojin Zhu for their research collaborations. I am also grateful to Dr. Albert Wasef, Xigang Huang, Mohamed E. Mahmoud, Dr. Jiming Chen, Dr. Yanfei Fan, Dr. Ho Young Hwang, Dr. Preetha Thulasiraman, Dr. Anderson Cheng, Dr. Fen Hou, Dr. Mehri Mehrjoo, Dr. Xinhua Ling, and other friends and colleagues for their friendship, support, and advice.

I gratefully appreciate the financial support from the Ontario Graduate Scholarship (OGS) and numerous scholarships from the University of Waterloo. Many thanks to the kind and dedicated administration staff: Wendy Boles, Lisa Handel, Annette Dietrich, and Karen Schooley.

Finally, I would like to thank my parents and my wife. I thank my parents for their unconditional love and always believing in me. Their love and encouragement have been and will always

be the greatest source of inspiration and comfort in my life. I thank my wife Grace for being one with me. We have written the thesis together, and I look forward to a happy life together. I love you. Special thanks to my daughter Isabelle, who brings me the greatest joy in life.

To my dear parents, wife,
and parents-in-law

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACK	Acknowledgement
AMC	Adaptive Modulation and Coding
ARQ	Automatic Repeat Request
ATIM	Ad-hoc Traffic Indication Map
BS	Base Station
CAM	Constant Access Mode
CDS	Cyclic Difference Set
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DTN	Delay/Disruption Tolerant Network
EWMA	Exponential Weighted Moving Average
FEC	Forward Error Correction
GPS	Global Positioning System
IBSS	Independent Basic Service Set
IEEE	Institute of Electrical and Electronics Engineering
IETF	Internet Engineering Task Force
IP	Internet Protocol
LLC	Logical Link Control
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MN	Mobile Node
NS	Neighbor Sensitivity

PDA	Personal Digital Assistant
PHY	Physical
PSL	Power Saving Level
PSM	Power Saving Mode
QoS	Quality of Service
QPS	Quorum-based Power Saving Protocol
RFID	Radio Frequency Identification
RTS/CTS	Ready to Send/Clear to Send
RWP	Random Way Point
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
TSF	Time Synchronization Function
TTL	Time To Live
UTC	Coordinated Universal Time
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

List of Notations

α	weight value of the moving average
β	binary decision variable for contact discovery in the last beacon period
γ	pairwise inter-contact rate
λ	aging parameter
ρ	constant maximum clock frequency error specified by the hardware manufacturer
τ	average update interval of tables
ω_i	frequency of hardware oscillator
ξ_i	proportional coefficient of the node i oscillator
C_{avg}	average relative clock offset among all nodes in the network
$C_{h,m}(E_i)$	set i repeated to m slots with a shift h
C_i	clock value of node i
C_{ij}	relative clock offset between nodes i and j
C_{il}^T	relative clock offsets stored in node i for node l
E_i	list of active elements of set i
E	set of nodes contacted
f_{avg}	average relative clock skew among all nodes in the network
f_i	clock frequency of node i
f_{ij}	relative clock skew between nodes i and j
f_{il}^T	relative skews stored in node i for node l
h_{max}	maximum beacon interval coefficient
L_A	length of the ATIM window

L_f	length of a frame
L_s	length of a slot
M	length of edges on a square map
N	number of nodes in the network
$N_i^T(t)$	set of node entries stored in node i
n_i	number of slots in set i
n_{on}	number of awake slots within a frame
n_{off}	number of asleep slots within a frame
n_s	number of slots within a frame
NS_{ij}	neighbor sensitivity between set i and set j
P_0	probability that there is no contact within τ
P_c	contact probability
P_i	power saving level i
P_{tx}	power consumption in transmit mode
P_{rx}	power consumption in receive mode
P_{idle}	power consumption in idle mode
P_{sleep}	power consumption in sleep mode
R	active ratio
R_t	current contact discovery rate
$RCP\{E_i, E_j\}$	rotational closure probability between set i and set j
S	slope of contact discovery rate
$\mathbf{S}$	system state of the protocol
T_b^i	length of a beacon interval for node i
T_{CD}	contact duration
T_D	length of a duty cycle
T_{ICD}	inter-contact duration
$T_c^{i,j}$	set of contact times of node i and node j
T_c	set of contact times between any pair of nodes
$t_k^{i,j}$	kth contact time between node i and j
t_k	kth contact time between any pair of nodes
T_{min}	length of the minimum beacon interval

T_{nh}	approximated elapsed time since the clock information of node h was recorded
T_R	data exchange duration
T_s	length of a slope window used to calculate the slope
T_{SD}	slot delay duration
V	set of all nodes in the network
w_{il}^T	weight coefficients stored in node i for node l
X_{il}	clock value of node l recorded in the table of node i
$\tilde{X}_{il}$	updated clock value of node i when it contacts node l
$\tilde{X}_{ijl}$	updated clock value in the table of node i with respect to node l, when node i contacts with node j
Y_{il}	clock frequency of node l recorded in the table of node i
$\tilde{Y}_{il}$	updated clock frequency of node i when it contacts node l within τ
$\tilde{Y}_{ijl}$	updated clock frequency in the table of node i with respect to node l, when node i contacts with node j

Chapter 1

Introduction

1.1 Overview of Delay Tolerant Networks

Recently, considerable research efforts have been put to Delay/disruption Tolerant Networks (DTNs) [1–8] enabling communications between disconnected network entities. DTN architecture is characterized by frequent disruptions and long delayed connections due to mobility, sparse deployment of nodes, node failures, and noises, etc. Disruption-tolerant networking is becoming a more favorable term for DTNs since it applies to more general situations. Similar properties of DTNs are also shown in some wireless mobile network topics such as intermittently connected network, disconnected spare network, opportunistic communication, partially connected ad-hoc network, and sparse mobile ad hoc networks.

Figure 1.2 shows the basic concept of DTNs. Since an end-to-end connection does not exist between node A and node C, the message is routed to the destination node C by a store and forward mechanism. The message is stored in node A until it is forwarded to an intermediate forwarder node B. This store and forward process is repeated until the message arrives at the destination or discarded after a given amount of time to live (TTL) value.

DTNs have a broad range of potential applications including scenarios with high delay and scenarios with frequent disruptions and disconnections. Some potential application for DTNs include military battlefields [9], vehicular communications [5], deep space communications [10],

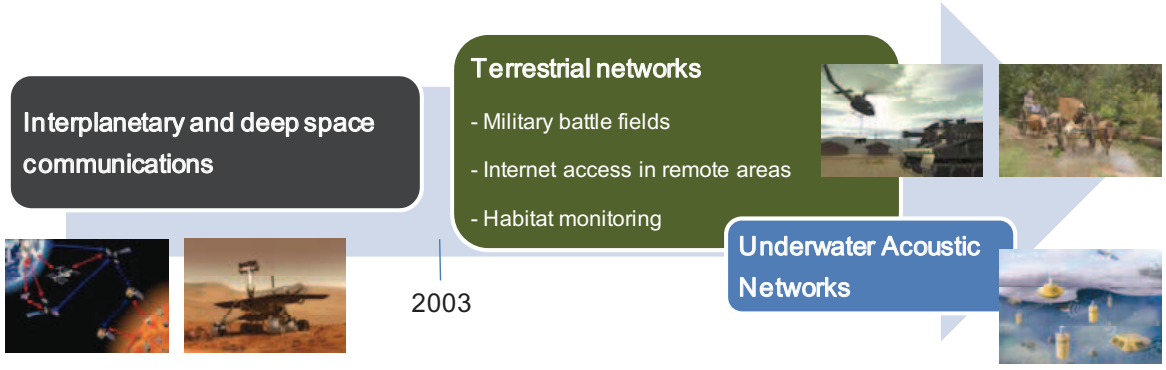


Figure 1.1: Examples of DTNs

habitat monitoring [11], and internet access in rural areas [12]. As there are many application scenarios for DTNs, the design of DTNs varies greatly depending on the application scenario. Contact schedules of nodes such as contact predictability and waiting times affect the design of DTNs. For example, while the contact schedules of orbital satellites are mostly predictable, the contact schedules of automobiles are much less predictable. Contact schedules in a decreasing order of predictability can be listed as follows: deep space, orbital satellite, transit bus, highway, human movement, random waypoint movement, and finally random movement. As well, capacity on contact affects the design of DTNs. Radio coverage range and bandwidth depend on the radio technology being used and contact duration determines the amount of transferable data. Additional network components such as fixed or mobile relay also introduce a new variable when designing DTN protocols. Consideration for resource availability is also important. Available storage capacity determines forwarding capabilities of intermediate nodes, and available energy supply determines the lifetime of communication devices.

1.2 Power Saving Protocols in Wireless Mobile Networks

The capacity of the battery determines how long the devices can be used before replacing or recharging the battery. Energy efficiency is a critical issue for mobile devices that are communi-

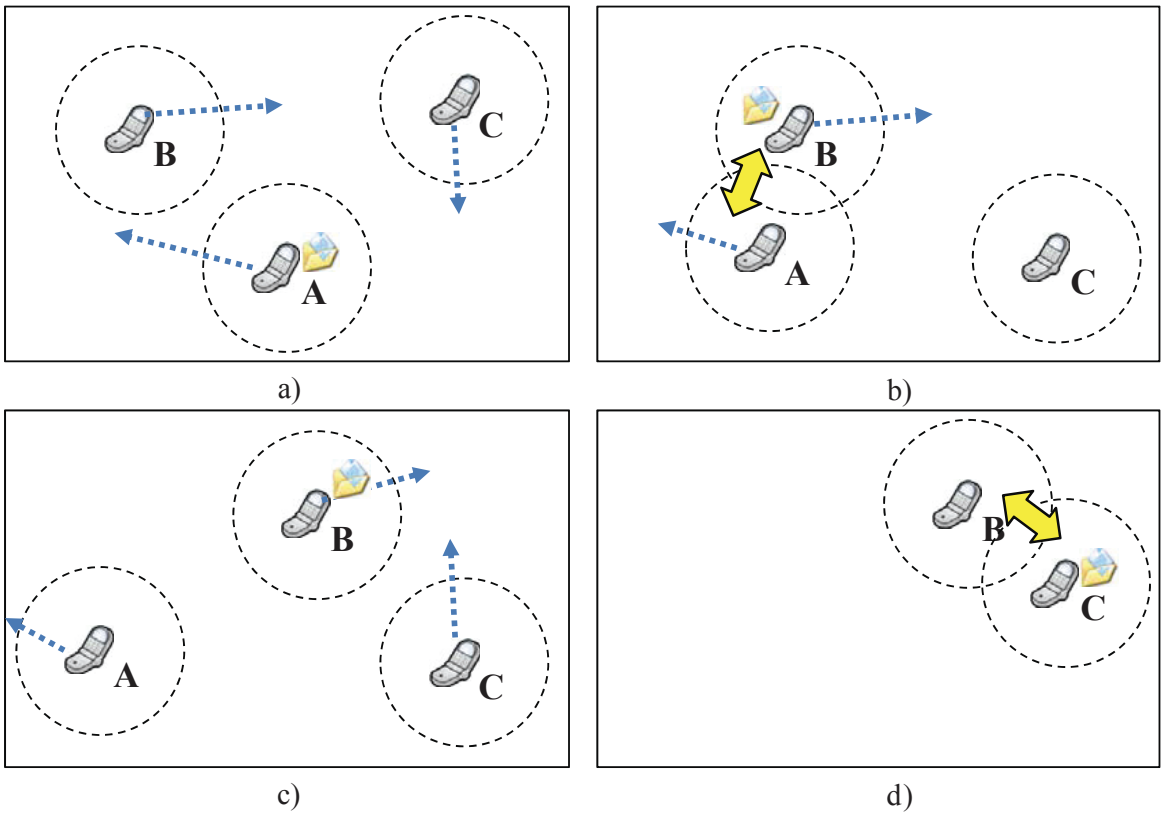


Figure 1.2: Store and forward mechanism of DTNs. A message is sent from a source node A to a destination node C via an intermediate node B.

cation oriented since considerable portion of total energy is consumed by the wireless network interface. It becomes even more important in scenarios where it is nearly impossible to recharge or replace batteries, such as in sensor networks. To enhance the energy efficiency, many power saving protocols have been proposed for various network architectures and at different layers of wireless network protocol stack [13, 14]. Some examples of power management techniques at different layers of protocol stack are outlined in Figure 1.3.

In particular, existing sleep scheduling protocols can be categorized into three types: on-demand wakeup, scheduled wakeup, and asynchronous wakeup. For on-demand wake up proto-

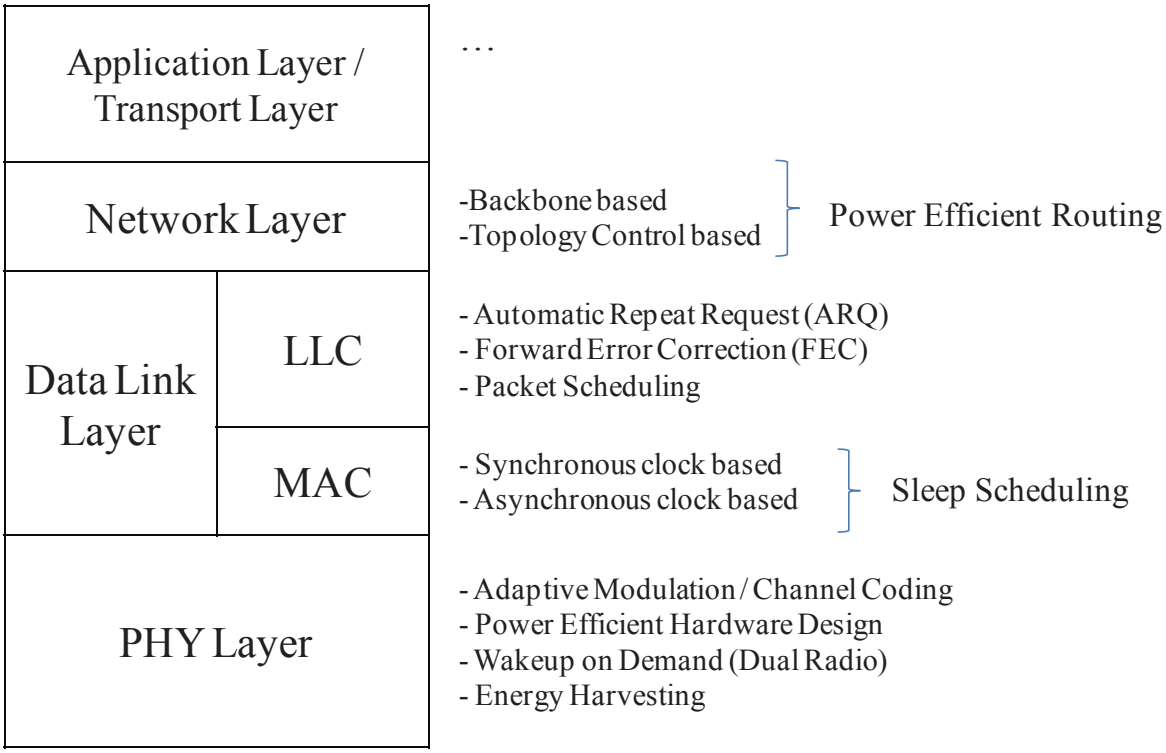


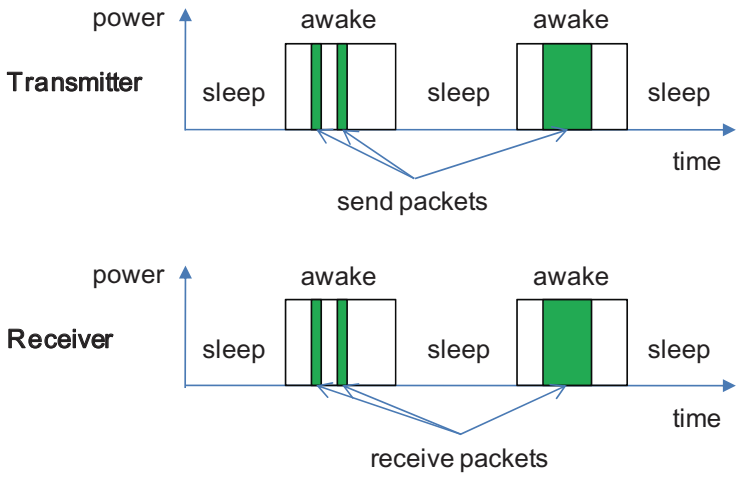
Figure 1.3: Power management techniques

cols, mobile nodes are equipped with a secondary low power radio module or a soft-state timer to wake up its radios to be ready for data exchange. For example, a separate signaling lower power radio [15], a RFID tag [16], or a soft-state timer [17] can be used to wake up the device for data exchanges. For scheduled wakeup protocol, nodes in the network wake up at synchronized intervals to communicate with each other. IEEE 802.11 PSM [18] is the most well known synchronous sleep scheduling protocol originally designed for single hop networks. As illustrated in Fig. 1.4a, wireless devices, called nodes, with synchronized clocks, periodically turn on or off their radio to save energy. This process is called a *duty cycling* or a *sleep scheduling*. For multihop wireless networks, nodes can cooperatively coordinate their wake schedules to increase the energy efficiency of the network while maintaining sufficient connectivity for required bandwidth [19]. However, these protocols assume that nodes are synchronized by global syn-

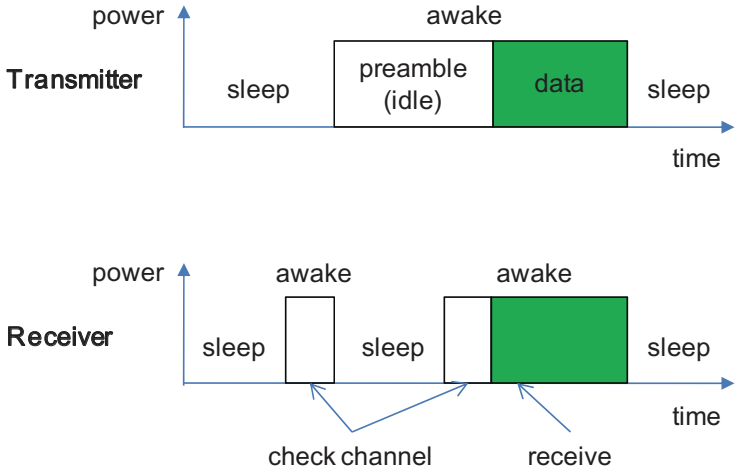
chronization algorithm. For asynchronous wakeup protocols [20–22], nodes connect with each other by waking up at predetermined time intervals that guarantee overlap with awake intervals of neighboring nodes, as illustrated in Fig. 1.4b. Asynchronous wakeup protocols do not require synchronized clocks, and they are suitable in scenarios where global clock synchronization is impossible or impractical to realize. Furthermore, there exists performance tradeoff between energy efficiency and neighbor discovery ratio for different cycle lengths of the sleep schedules. The performance can be further improved if each node can adaptively use a sleep schedule that optimizes the individual performance requirement. Adaptive asynchronous sleep scheduling protocols provide a set of sleep schedules with different cycle lengths that guarantee overlapping awake intervals between any pair of sleep schedules [23–28].

1.3 Research Motivations and Objectives

Unfortunately, many mobility scenarios in DTNs depend on mobile devices that have limited energy capacity, and the fundamental problem is that traditional power saving protocols are designed assuming well connected networks. Different from the traditional power saving protocols, it is not possible to coordinately schedule on and off periods in DTNs to save power. Also, previous researches in DTNs has mostly focused on DTN routing protocols, and these protocols assume continuous access mode (CAM) without sleep scheduling mechanisms where the radio is always turned on. For a sparsely connected network, time durations between contacts are generally much larger than contact durations for more than an order of a magnitude [29], which indicates that nodes spend most of their life time in the idle listening mode and centralized power saving strategies are difficult. Experimental studies in [30, 31] show that power consumption in an idle listening mode is almost as high as in a receiving mode. Consequently, a large amount of energy, over 95 percent of the total energy [32], is consumed by the idle listening mode searching for neighbors, rather than by infrequent data transfers. The idle listening problem seriously affects the lifespan of energy constrained devices in DTNs. Therefore, it is essential to have power saving protocols for DTNs that is distributed and effective at reducing energy waste in the idle listening mode in order to minimize the degradation of network connectivity and to maximize the benefits from mobility.



(a) Synchronous clock based



(b) Asynchronous clock based

Figure 1.4: Sleep scheduling protocols

1.4 Research Contributions

In this research, we intend to address the following research question:

"Given a sparsely deployed wireless mobile nodes having frequent disconnections, how can we design power saving protocols to provide extended lifetime of wireless mobile nodes and support the required delay and delivery performance by means of efficient sleep scheduling and clock synchronization?"

The research contributions and significance of this thesis are summarized as follows.

- **Synchronous Clock based Adaptive Sleep Scheduling for DTNs** [33] - We propose a distributed adaptive exponential beacon protocol by exploiting the intermittent connection characteristic of DTNs in synchronized clock based scenario. The beacon periods of nodes are independently adjusted depending on the trend of contact availability. The proposed protocol is optimized for different network environments using the probability distribution of contact durations. Simulation results show that energy waste from the idle listening problem can be effectively reduced by adaptive sleep scheduling protocols, and energy savings up to 35 percent are achieved compared with existing power saving protocols, while maintaining similar average packet delays and packet delivery ratios to that without a sleep scheduling. This work is presented in Chapter 3.

- **Asynchronous Clock based Adaptive Sleep Scheduling for DTNs** [34, 35] - We propose two asynchronous clock based sleep scheduling protocols that are distributed, adaptive, and energy efficient. Moreover, the sleep schedules can be constructed using simple systematic algorithms. We also discuss how the proposed protocols can be implemented in mobile devices for adapting to dynamic network conditions in DTNs. Theoretical analysis is given to demonstrate the energy efficiency and scalability of the proposed protocols. Simulation results show that the proposed protocols reduce the energy consumption in the idle listening mode up to 35 percent in comparison with other existing asynchronous clock based sleep scheduling protocols, and more than 90 percent compared with the protocol without

a sleep scheduling, while maintaining comparable packet delivery delay and delivery ratio. This work is presented in Chapter 4.

- **Semi-Asynchronous Clock based Sleep Scheduling for DTNs** [36] - We propose a distributed semi-asynchronous sleep scheduling protocol considering loosely synchronized clocks in sparse mobile wireless networks. The sleep schedules are constructed to guarantee contacts among distributed nodes by utilizing clock synchronization error information rather than assuming that clock synchronization is infeasible or impossible. Individual nodes using the proposed protocol not only can adjust to required connectivity but also to the estimated clock synchronization error to further enhance energy efficiency. Also, the proposed protocol can be optimized using the probability distribution of the clock synchronization error to maximize the energy efficiency. Using theoretical analysis and simulation results, we demonstrate that the proposed protocol can achieve higher energy efficiency than existing asynchronous sleep scheduling protocols. This work is presented in Chapter 5.

- **Distributed Asynchronous Clock Synchronization for DTNs** [37] - We propose a distributed asynchronous clock synchronization protocol for DTNs. Different from existing clock synchronization protocols, the proposed 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 is reached by compensating 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 proposed protocol. The proposed protocol can achieve faster clock convergence speed and, as a result, reduce energy cost due to neighbor discovery by half. This work is presented in Chapter 6.

1.5 Outline of the Thesis

This thesis is organized as follows. The system model of this research is presented in Chapter 2. The distributed adaptive sleep scheduling protocols for synchronous, asynchronous, and semi-asynchronous clocks are respectively given in Chapter 3, Chapter 4, and Chapter 5. The distributed asynchronous clock synchronization protocol is given in Chapter 6. Finally, conclusion and further research are given in Chapter 7. To better illustrate the relationship among the research accomplishments, the organizational flowchart of this thesis is depicted in Fig. 1.5.

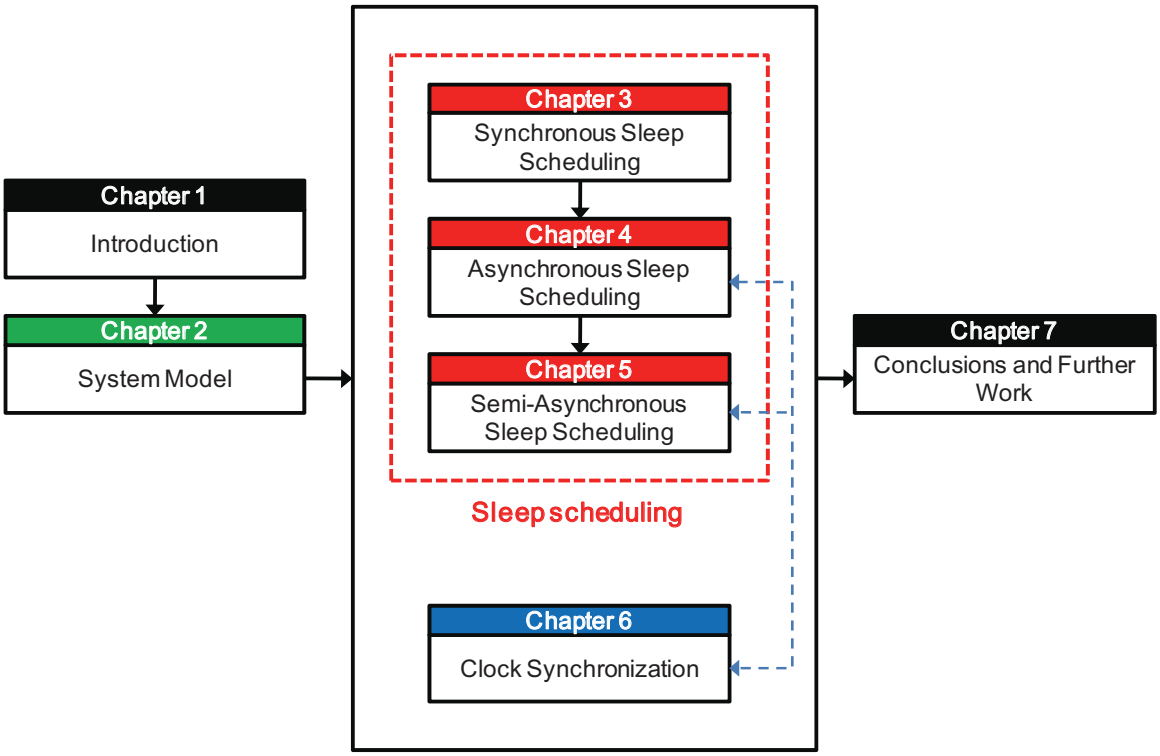


Figure 1.5: The organizational flow chart of the thesis. Solid (dotted) lines represent direct correlations (partial correlations) between two functional parties and arrows indicate the flow of communication.

Chapter 2

System Model

2.1 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 $(j, i) \in E(t)$. 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 protocols, upon each contact, nodes exchange and update their timing information. The procedure and modeling of clock value and frequency updates apply to all nodes in the network.

2.2 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

$$C_i(t) = (1 + \xi_i) \int_{t_0}^{t} \omega_i(\tau) d\tau + C_i(t_0), i = 1, 2, \ldots, N \tag{2.1}$$

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. Under stable external conditions (e.g., temperature and pressure), the oscillator frequency can be approximated by a time-invariant constant. The clock value is incremented by an oscillator with frequency $f_i(t) = \frac{dC_i(t)}{dt}$. As illustrated in Fig. 2.1a, 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, and so on. 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 an upper bound as

$$1 - \rho \leq \frac{dC_i(t)}{dt} \leq 1 + \rho, \ \forall \, t \tag{2.2}$$

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 [38]. 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. We also consider changes in the clock frequency due to oscillator aging of 5 ppm per year [39] in our simulations. We assume a clock resolution of 1 μs with 64 bits as in Time Synchronization Function (TSF) of IEEE 802.11 [18].

2.2.1 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

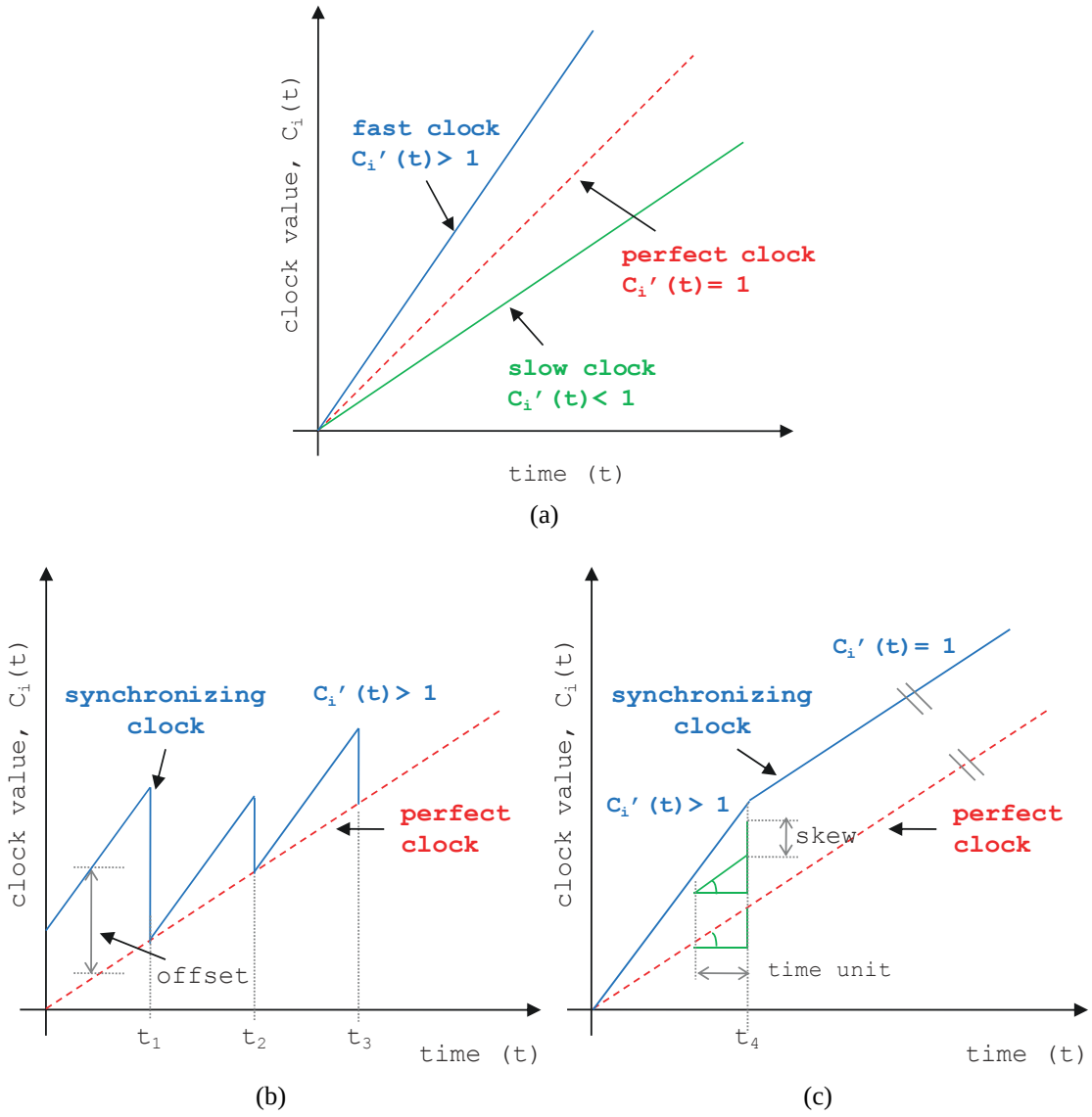


Figure 2.1: Clock model: a) Clock error caused by clocks running at different rates; b) *Offset compensation*: clock value is tuned to the clock value of the perfect clock, but the clock offset increases due to the clock skew error; c) *Skew compensation*: clock frequency is tuned to the clock frequency of the perfect clock, but the clock offset does not match due to the clock offset error.

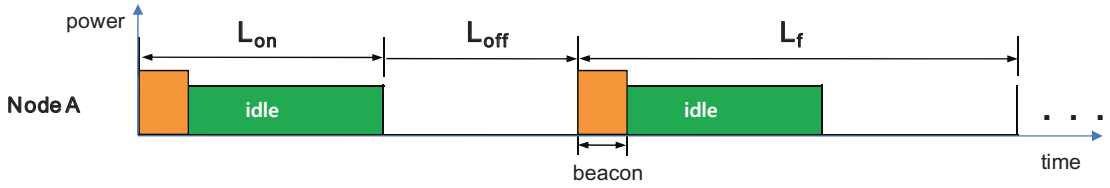
time variant oscillator frequencies in Section VI-B-5. Furthermore, uncertainty in the message delays cause clock estimation errors. Usually, in a multihop wireless network, an accurate estimate of the message delay at each hop is critical for synchronization protocols since the end-to-end delay is comparable 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 [40]. 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.

2.2.2 Clock Compensation

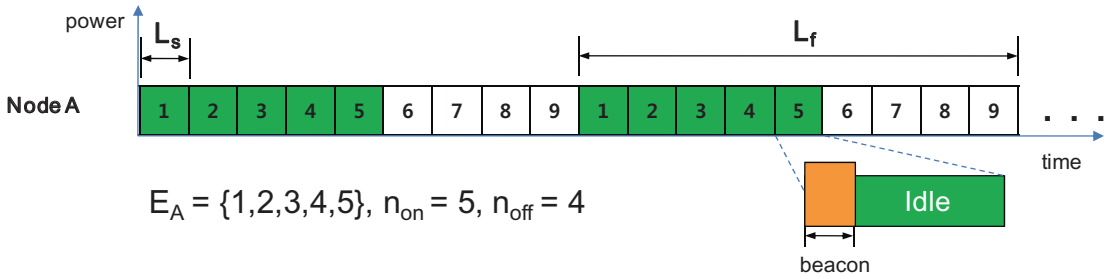
The goal of clock synchronization protocols is to maintain the minimum possible clock error among nodes in the network. As described in the clock model, the accuracy of a clock is measured by two parameters: clock value and clock frequency. Since there is no reference node in the distributed network, 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. 2.1b and 2.1c, 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.

2.3 Sleep Scheduling Model

Each node follows a predetermined combination of awake and asleep time intervals, called a *sleep schedule*, that is successively repeated. A complete cycle of the sleep schedule is called



(a) Non-slotted frame structure



(b) Slotted frame structure

Figure 2.2: Frame structure

a *frame*, and its length is denoted by L_f. A frame A may consist of unequal sized intervals as in [41] or equal sized intervals, called *slots*, forming a *set* as in [23, 25, 25, 26, 28, 42]. Here, E_A represents a list of awake slots within a frame A. For the protocols with unequal sized slots, as illustrated in Fig. 2.2a, let L_{on} and L_{off} respectively represent the duration of awake intervals and asleep intervals. Similarly, for the protocols with equal sized slots of lengths L_s dividing a frame into $n_s = L_f/L_s$ slots, as illustrated in Fig. 2.2b, let n_{on} and n_{off} respectively represent the number of awake slots and asleep slots. L_A is the length of the Announcement Traffic Indication Message (ATIM) window length required to exchange connection setup messages (Beacon-ATIM-ATIM Response) as defined in [18].

2.4 Power Consumption Model

There are four different modes of operation for the wireless radio interface: transmit, receive, idle, and sleep. The interface of a node is either actively transmitting beacons or messages in the transmit mode, actively receiving beacons or message in the receive mode, waiting for bea-

cons or messages in the idle mode, or turned off to save power in the sleep mode. In the sleep mode, nodes can save a considerable amount of power, for more than an order of a magnitude, but are unable to exchange beacons or messages. In particular, for sleep scheduling protocols, a well known power consumption model of 2.4 GHz IEEE 802.11 is used [30] where power consumptions in the transmit mode (P_{tx}), receive mode (P_{rx}), idle mode (P_{idle}), and sleep mode (P_{sleep}) are 1.3272 W, 0.9670 W, 0.8437 W, and 0.0664 W, respectively. For clock synchronization protocols, a well known power consumption model of sensor motes is used [43] where power consumptions in the transmit mode, receive mode, idle mode, and sleep mode are 42 mW, 36 mW, 24 mW, and 0.02 mW, respectively.

2.5 Performance Metrics

To evaluate the performance of our proposed protocols, certain performance measurements are necessary. In this thesis, we consider the following performance metrics.

2.5.1 Energy Consumption

Energy consumption is the amount of energy consumed measured in Joules ($J = W \cdot s$) by the operations of the wireless network interface.

2.5.2 Average Packet Delivery Delay

Average packet delay is the time delay of packets from source nodes to destination nodes. For multiple copy routing protocols, such as spray and wait routing protocol [7], which generates multiple copies of the same packet, the delay of the first arriving packet at the destination among multiple copies is considered.

2.5.3 Packet Delivery Ratio

Packet delivery ratio is the number of packets received by the destination nodes divided by the number of packets generated by the source nodes. For multiple copy routing protocols, duplicated copies of the same packet are not counted.

2.5.4 Active Ratio

The energy efficiency of sleep scheduling protocols is measured by the active ratio, R, which describes how often the node is in active mode. The R is defined for a given frame of length L_f with unequal sized slots as

$$\begin{aligned} R &= L_{on}/L_f \\ L_f &= L_{on} + L_{off} \end{aligned} \tag{2.3}$$

and for a given frame of length L_f with equal sized slots as

$$\begin{aligned} R &= n_{on}/n_s \\ n_s &= L_f/L_s = n_{on} + n_{off}. \end{aligned} \tag{2.4}$$

2.5.5 Neighbor Sensitivity

The required connectivity of sleep scheduling protocols is measured by the neighbor sensitivity, NS, which describes how well the node finds a neighboring node within its transmission range. The NS is defined as the worst-case delay for a node to detect a new node within its coverage. In asynchronous sleep schedules, the NS is determined by the minimum frame size that guarantees at least one overlapping slot between two different schedules.

2.5.6 Average Relative Clock Offset and Skew

The performance of clock synchronization protocols are 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

$$C_{avg}(t) = \frac{1}{N(N-1)/2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |C_i(t) - C_j(t)| \tag{2.5}$$

$$f_{avg}(t) = \frac{1}{N(N-1)/2} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} |f_i(t) - f_j(t)| \,. \tag{2.6}$$

Chapter 3

Adaptive Synchronous Sleep Scheduling Protocol in Delay Tolerant Networks

As discussed in Chapter 1, mitigating the idle listening problem is the key to enhance the performance of mobile devices in DTNs. Assuming that clocks of nodes are globally synchronized, nodes can cooperatively coordinate their sleep schedules to reduce energy waste due to the idle listening problem. The key issue in sleep scheduling protocols is to decide when to wake up from sleep to communicate with other mobile nodes. Although it is often difficult to know exactly when and how long contacts will be in advance in many DTN scenarios, changes in the contact rates are caused by changes in node densities and traffic requirements in different regions. The changes in the density are caused by dynamic movements of nodes. Even in RWP model, the spatial node distribution is not uniform [44]. Moreover, the density differences become more evident in non-random movement models. Therefore, non-adaptive protocols such as continuous access mode (CAM) or PSM may not be effective. For example, when there are very few nodes wanting to exchange data, turning on the radio frequently to check the presence of other nodes wastes energy without discovering useful contacts. On the other hand, when there are many nodes wanting to exchange data, it is desirable to wake up frequently to connect with other nodes.

Therefore, in this chapter, we focus on adaptive sleep scheduling protocol in DTNs that minimizes energy consumption while minimizing the performance degradation of average packet delay and packet delivery ratio. The contribution and significance of this research work [33] are as follows

- First, we propose a distributed adaptive exponential beacon period protocol for DTNs. The trend of contact frequency is used as a decision variable to reflect the network condition, and nodes independently adjust their beacon periods according to the decision variable to save energy.
- Second, we optimize the decision parameters for different network environments using the probability distribution of contact durations.
- Third, simulation results for various network scenarios are given to demonstrate the performance. The results show that the proposed protocol can quickly adapt to dynamic changes in the network topology, and achieves up to 35 percent energy saving in comparison with existing protocols while maintaining similar average packet delays and packet delivery ratios.

The remainder of the chapter is organized as follows. In section 3.1, existing power saving protocols are presented. In section 3.2, an adaptive exponential beacon protocol is proposed. In section 3.3, performance of our proposed protocol is evaluated using simulations. The chapter is concluded in section 3.4.

3.1 Related Work

3.1.1 Power Saving Protocols in Multihop Wireless Networks

Existing power saving protocols for different wireless environments, including IEEE 802.11, Mobile Ad-hoc Networks (MANET), Wireless Sensor Networks (WSN), and DTNs are presented in this section.

The most well known power saving strategy is the power save mode in 802.11 [18]. It is originally designed for well connected single hop networks. The standard also defines a power save mode in an Independent Basic Service Set (IBSS) architecture, also known as an ad-hoc network. Considering mobility of nodes, MANET is a type of wireless ad-hoc networks which consist of self-configuring network of mobile hosts connected by wireless links. The power saving protocols in MANET can be generally classified into four different categories. They are wake-up on-demand based [16, 17], coordinator based [19], synchronous clock based [45], and asynchronous clock based [20]. WSN is a wireless network with distributed autonomous sensors to cooperatively monitor physical or environmental conditions. A common objective of sensor MAC design is to maximize the network lifetime since replacing or recharging depleted batteries is impractical in most WSN scenarios. Some examples of contention based power saving protocols in WSN are SMAC [46] and T-MAC [47]. There are also TDMA based protocols such as TRAMA [48] and wake-up on-demand protocols such as PAMAS [15].

3.1.2 Power Saving Protocols in Delay Tolerant Networks

Existing power saving protocols in MANET and WSN assume dense deployment and low mobility of nodes. Therefore, they rely on well connected nodes with few disconnections and disruptions, and cannot be directly applied to DTNs. A knowledge based framework in sparsely connected DTNs is presented in [49]. In the zero knowledge scenario, assuming a synchronized clock, periodical beacon interval is determined by finding the maximum interval that satisfies the traffic load requirement. In the partial knowledge scenario, searching intervals are estimated using mean and variance values of contact durations and waiting times. However, such statistical information may not be readily available in many real life scenarios. A hierarchical power management scheme in [50] uses a secondary low power radio to discover communication opportunities and a primary high power radio to exchange data. An asynchronous power management protocol, called a context-aware power management (CAPM) [41], determines the duty cycle depending on the node density and traffic load requirements.

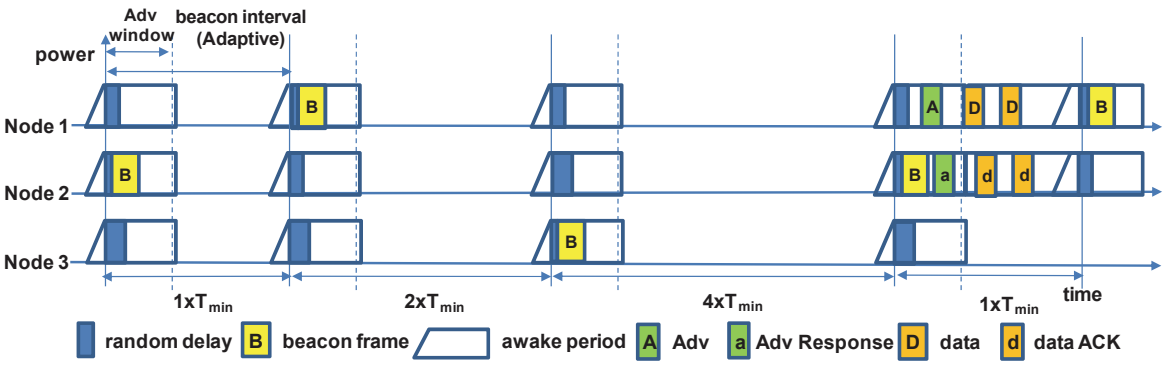


Figure 3.1: Exponential beacon period

3.2 Adaptive Exponential Beacon Protocol

3.2.1 Basic Mechanism

Adaptive exponential beacon protocol (AEB) dynamically adjusts beacon periods depending on the contact availability of nodes. At the beginning of each beacon interval, each node sends a short beacon after a random delay. If a node receives a beacon from another node, it checks its own queue to see if there are packets destined for the node that sent the beacon. If there are packets to be sent, the node returns an Advertisement (Adv) packet notifying the other node of future transmission. If there is no packet to be sent, the beacon is ignored. Adv packet is responded with an Advertisement Response (AdvResponse) packet. After the successful exchange of Adv and AdvResponse packets, a connection is established between the nodes, and data packet transmission is initiated. If no contact has been discovered during the Adv window, the node sleeps at the end of Adv window.

At the end of each beacon interval, each node checks if a contact has been discovered in the last beacon interval. If so, the node starts the beacon interval from the minimum beacon interval in the next beacon interval. Otherwise, the node doubles its beacon interval in the next beacon interval. At each beacon interval, the beacon period is exponentially increased to a maximum beacon period threshold.

The basic mechanism is illustrated in Figure 3.1. Mobile nodes 1, 2, and 3 are within radio

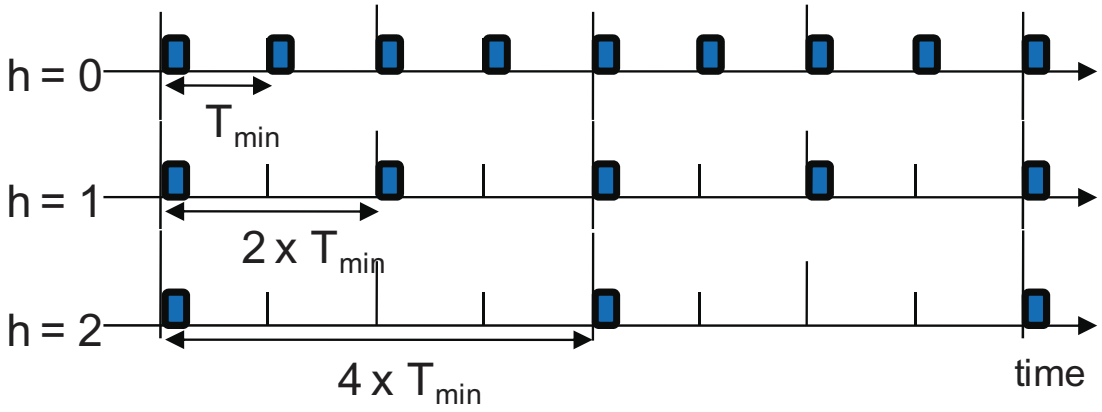


Figure 3.2: Adaptive exponential beacon interval

range of each other. In the first three intervals, nodes do not have any packet to exchange. Therefore, nodes sleep after each Adv window, and beacon period is doubled at each interval. In the fourth interval, some packets have been generated at node 1 and a connection is established between node 1 and 2. After the Adv window, data transmission is initiated. Since a connection has been established between node 1 and node 2, the beacon period is immediately set to the minimum value.

3.2.2 Beacon Interval Structure

The AEB protocol uses a set of exponentially increasing beacon intervals as shown in Figure 3.2. Every node independently chooses its beacon intervals, and must follow one of the interval set. A beacon interval for node i can be defined as follows

$$T_b^i = \{t_b^i \,|\, t_b^i = T_{min} \times 2^h,\ h = 0, 1, 2, 3, \ldots h_{max}\}, \tag{3.1}$$

where T_{min} is the minimum beacon interval.

3.2.3 Adjusting Beacon Periods

If a contact has been discovered, the beacon period of node i, T_b^i, is set to T_{min}, but otherwise incrementally doubled to the maximum beacon interval, h_{max}. Here, the choice of h_{max} is critical

to the performance of the protocol since tradeoffs exist between energy consumption and contact probability. Larger beacon intervals reduce energy consumption, but decrease the probability of contact. On the other hand, smaller beacon intervals increase the probability of contact, but the larger ratio of awake periods relative to the beacon period increases the energy consumption.

The value of h_{max} is determined by looking at the slope of contact discovery rate. First, an exponential weighted moving average (EWMA) is used to calculate contact discovery rate, R_t.

$$R_t = \alpha\,\beta + (1-\alpha)R_{t-1},\ \alpha = [0,1], \tag{3.2}$$

where α is the weight value for the moving average and β is the contact discovery value which is 1 if a contact has been discovered in the last beacon period and 0 otherwise.

Then, the slope of R_t can be defined as

$$S = \frac{1}{T_s}\sum_{j=t-T_s}^{t}\lambda_t, \tag{3.3}$$

where T_s is the slope window which indicates how many past beacon intervals are used for calculating the slope, and λ_t can be defined as

$$\lambda_t = \begin{cases} 1, & R_t > R_j \\ -1, & R_t < R_j \\ 0, & R_t = R_j, \end{cases} \tag{3.4}$$

where R_t is the current contact discovery rate.

If S is a positive value, a smaller value of h_{max} is used to allow the node to wake up more often. On the other hand, if S is a negative value, a larger value of h_{max} is used to allow the node to wake up less frequently to conserve energy. As shown in Table 3.1, the performance depends on the choice of h_{max}. Here, CAM represents a Constant Access Mode where nodes are always awake. The default parameters for simulations are outlined in Table 3.3.

3.2.4 Choosing h_{max}

Because of a low node degree in DTNs, nodes often spend much time in the disconnected mode. To illustrate, cumulative distribution functions (CDF) of inter-contact durations (T_{ICD})

Table 3.1: Impact of h_{max}

	Energy (J)	**Delay (s)**	**Delivery Ratio**
CAM	170025	537.18	0.9416
$h_{max} = 0$	72122	538.11	0.9415
$h_{max} = 1$	65136	546.56	0.9408
$h_{max} = 2$	58431	574.19	0.9376
$h_{max} = 3$	58135	705.67	0.9015

and contact durations (T_{CD}) are shown in Figure 3.4. T_{ICD} is the time between new contacts, and T_{CD} is the time between the beginning and the end of a contact. We can observe from both numerical [51, 52] and simulation results that T_{ICD} is generally much larger than T_{CD}. Therefore, we attempt to find the optimal h_{max} that achieves, for example, 98 percent of total possible contacts in radio transmission range, while saving as much energy as possible. This is directly associated with finding the contact duration of nodes in the network. We can find the h_{max} that allows nodes to discover certain percentage of total possible contacts, called a contact rate, using a CDF of contact duration as derived in [51] as below.

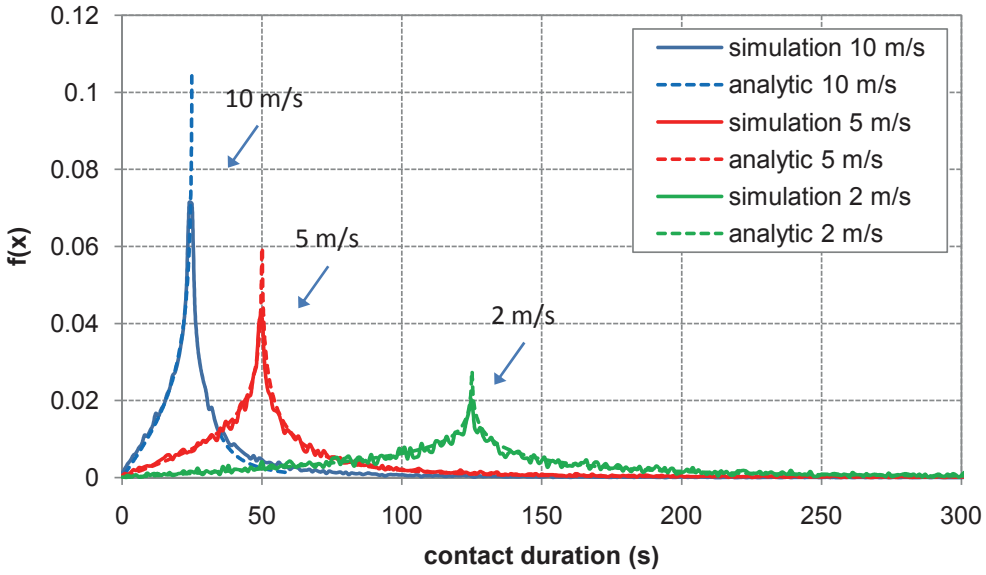
$$F(x) = \frac{1}{2} - \frac{r^2 - v^2x^2}{2rvx} \ln\left(\frac{r + vx}{\sqrt{|r^2 - v^2x^2|}}\right), \tag{3.5}$$

where r is the radio range of nodes and v is the average node velocity. The distribution area is assumed to be a unit square.

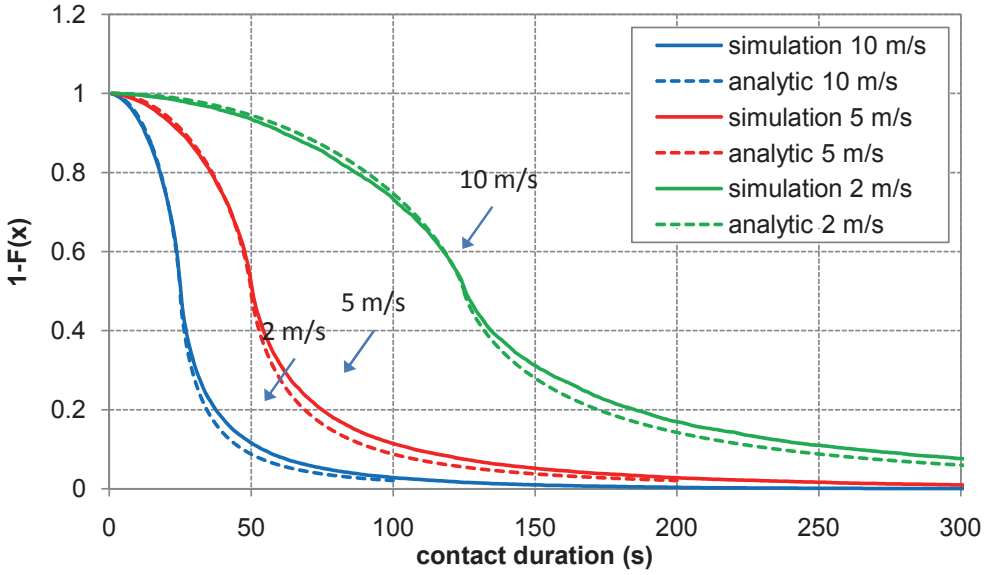
Figure 3.3a shows the pdf of contact duration for different node velocities. Faster moving nodes have shorter contact durations, and the contact duration is the highest at the expected value of contact duration at $E(X) = r/v$. In order to estimate the required contact duration, complementary cumulative distribution function (ccdf) is used. ccdf shows how often the random variable is above a particular level. In our case, as shown in Figure 3.3b, ccdf is the contact duration required to achieve a certain contact rate.

$$F_c(x) = P(X > x) = 1 - F(x). \tag{3.6}$$

Table 3.2 gives the contact duration values to achieve given contact rates for different node



(a)



(b)

Figure 3.3: Contact durations for different node velocities: (a) pdf; (b) ccdf.

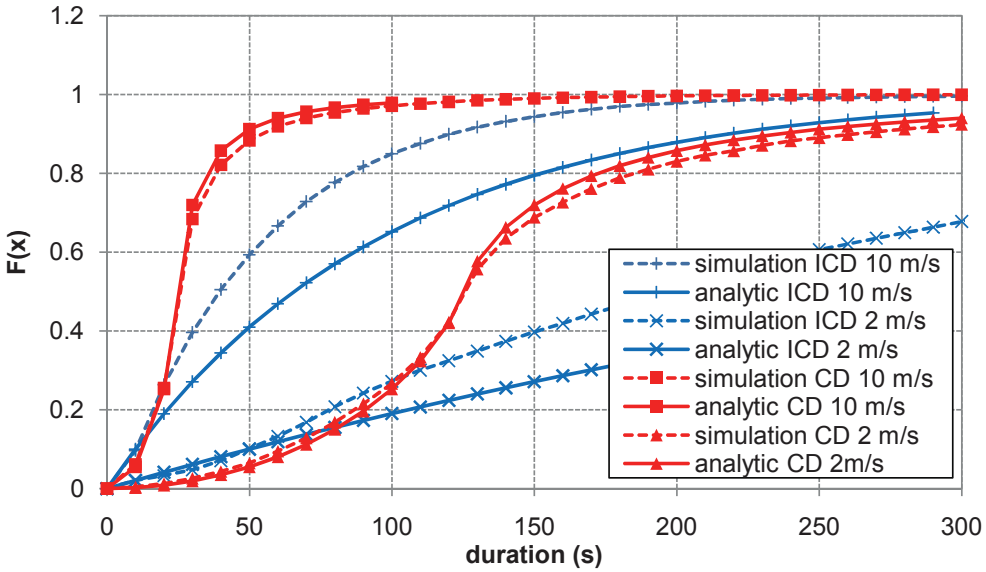


Figure 3.4: CDF of inter-contact duration and contact duration

Table 3.2: Contact duration (s) [numerical/simulation]

$1 - F(x)$	**Node speed (m/s)**		
	10	5	2
0.98	6.0 / 5.5	12.1 / 11.5	30.4 / 26.5
0.95	9.5 / 8.5	19.0 / 17.5	47.6 / 44.5
0.9	13.2 / 12.5	26.5 / 25.5	66.3 / 62.5
0.85	15.9 / 15.5	31.9 / 31.5	79.9 / 76.5

speeds. In addition, contact rates can be controlled by adjusting the value of h_{max}, and smaller values of h_{max} may be used in reality to cope with link quality estimation errors.

3.3 Performance Evaluation

3.3.1 Simulation Environment

The Network Simulator (*NS-2*) [53] is used for the simulation. DTN packet forward mechanism, and spray and wait routing protocol [54] are included in the simulator. Unless otherwise stated, default simulation parameters, as outlined in Table 3.3, are used. For each simulation setup, average values from 10 simulation results are used as the final result. Flows are created between source nodes and destination nodes that are randomly chosen. The energy model is taken from [30]. h_{max} values used for $S < 0$ are 3, 2, 2 and 1 respectively for nodes speeds of 2.5 m/s, 5.0 m/s, 7.5 m/s and 10.0 m/s. These values are chosen from Table 3.2 to achieve contact rates well above 98 percent.

3.3.2 Simulation Results

The performance of the AEB protocol is compared with previous synchronized clock based power saving protocols. In CAM, nodes are always awake. They are in the idle mode when not in transmit or receive mode. In PSM, as described in [50], nodes periodically wake up to broadcast beacons. In both PSM and CAM, a beacon period of 2.0 seconds is used. The performance metrics used in the simulation are energy consumption, average packet delivery delay, and packet delivery ratio. Energy consumption is the amount of energy consumed by the operations of the wireless radio. Average packet delay is the time delay of packets from source nodes to destination nodes. Since spray and wait routing protocol generates multiple copies of the same packet, the delay of the first arriving packet at the destination among multiple copies is considered. Packet delivery ratio is the number of packets received by the destination nodes divided by the number of packets generated by the source nodes. Simulation results for different topology sizes, node speeds, number of nodes, and traffic loads are shown in Figures 3.5 to 3.8, respectively. Also, the percentage of energy consumption of the proposed protocol compared with PSM is shown in the figure for clarity.

Table 3.3: Simulation parameters

Parameter	**Value**
Simulation Time	10000 s
Map Size	3000 m x 3000 m
Movement Model	RWP
Routing Protocol	Spray and Wait
MAC Protocol	CSMA/CA based
Spray size	2 duplicates
Node Parameters	
Number of Nodes	20
Node Speed	5 m/s
Pause Time	30 s
Radio Range	250 m
Bandwidth	2 Mbps
Flow Parameters	
Number of Flows	20
Packet Size	1000 bytes
Packet Generation Interval	10 s
Message Timeout	2000 s
Power Consumption Parameters	
Transmit Mode	1.3272 W
Receive Mode	0.9670 W
Idle Mode	0.8437 W
Sleep Mode	0.0664 W
Adaptive Beacon Parameters	
Minimum Beacon Period (T_{min})	2.0 s
Adv Window Size	0.4 s
Weight of EWMA (α)	0.1
Slope window (T_s)	16 beacon intervals

3.3.2.1 Impact of Topology Size

Impact of topology size is shown in Figure 3.5. As the simulation map size increases the total energy consumption decreases, average packet delay increases, and packet delivery ratio decreases. As the map size grows, nodes discover fewer contacts and nodes sleep more often but packet forwarding opportunities are reduced. CAM has little decrease in energy consumption

since nodes are in idle mode when they are not transmitting or receiving. The AEB protocol uses about 35 percent less energy than PSM and 75 percent less energy than CAM while maintaining similar average packet delay and packet delivery ratio as in CAM and PSM.

3.3.2.2 Impact of Node Speed

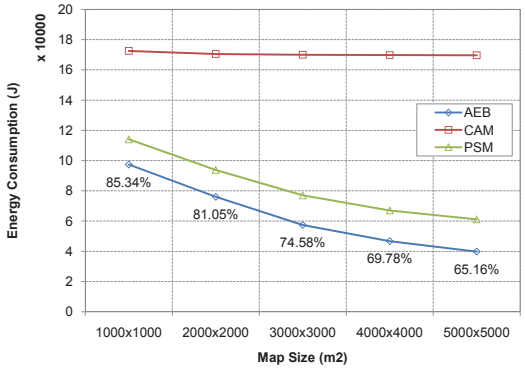
Impact of node speed is shown in Figure 3.6. As the node speed increases, the total energy consumption remains almost the same and average packet delay decreases. As the nodes move faster, the nodes discover more contacts which reduces the time packets spend in queues. There is only a little change in energy consumption since high node speeds also decrease the contact duration. The AEB protocol uses about 25 percent less energy than PSM while achieving similar results for average packet delay and packet delivery ratio as in CAM and PSM.

3.3.2.3 Impact of Number of Nodes

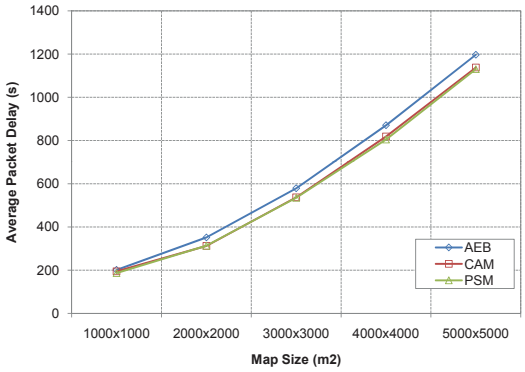
Impact of number of nodes is shown in Figure 3.7. As the number of nodes increases, the per node energy consumption increases slightly. The percentage of energy consumption of the proposed protocol is lower in lower number of nodes. This is because at lower node density, the proposed protocol has more chances to use larger beacon periods to save energy.

3.3.2.4 Impact of Traffic Load

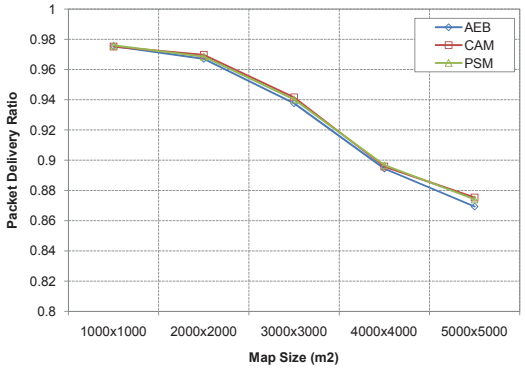
Impact of traffic load is shown in Figure 3.8. As the data arrival rate increases, the total energy consumption decreases and average packet delay increases slightly. As the data arrival rate increases, the nodes have smaller number of packets to forward. Therefore, they wake up less frequently and spend less energy in both transmission mode and idle mode. Average packet delay with moderate data arrival rate is smaller compared with the delay with low data arrival rate since more queued packets may be forwarded in a single contact with moderate data arrival rate.



(a)

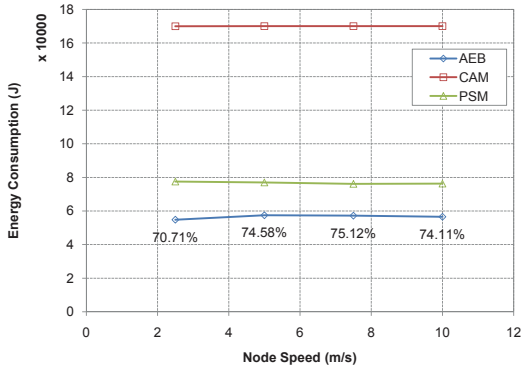


(b)

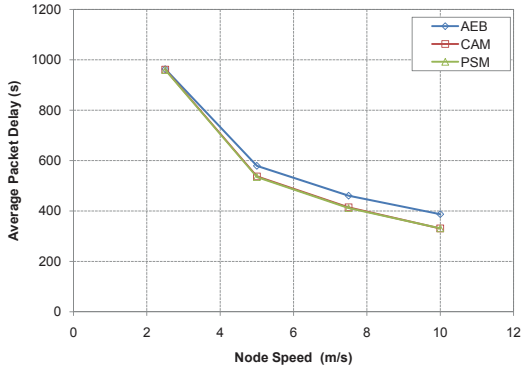


(c)

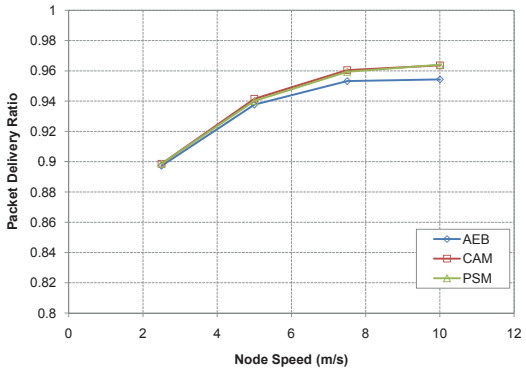
Figure 3.5: Impact of topology size: (a) Energy consumption; (b) Average packet delay; (c) Packet delivery ratio.



(a)

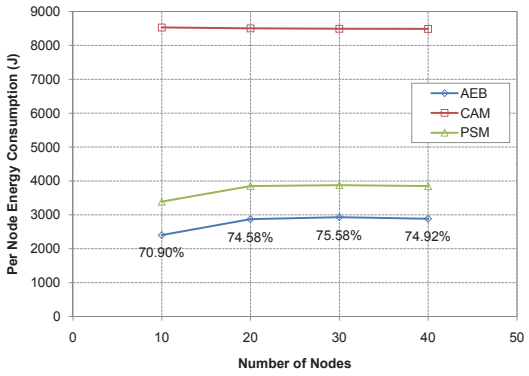


(b)

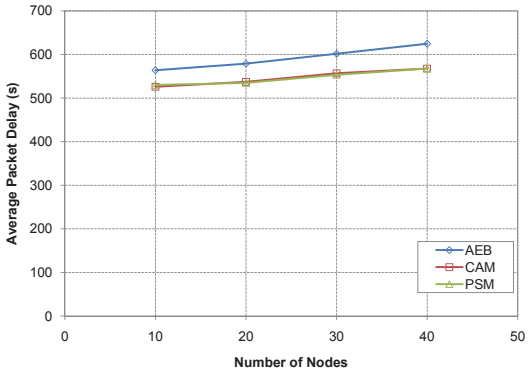


(c)

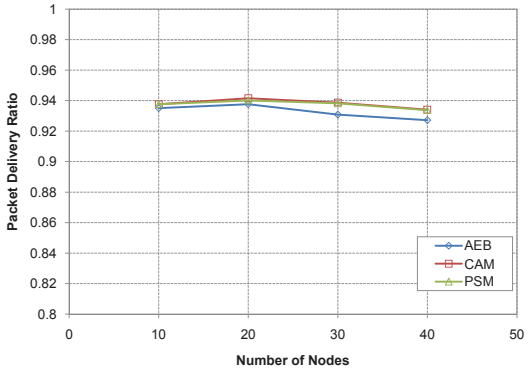
Figure 3.6: Impact of node speed: (a) Energy consumption; (b) Average packet delay; (c) Packet delivery ratio.



(a)

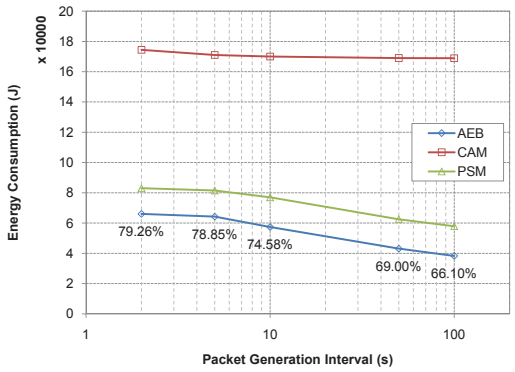


(b)

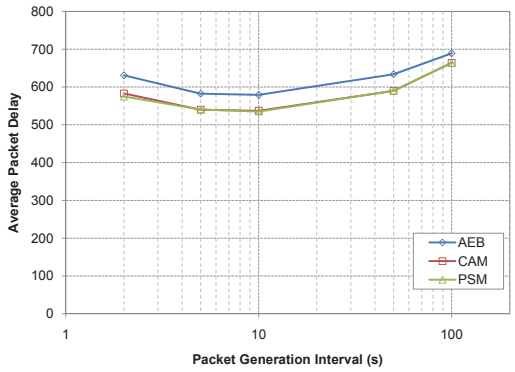


(c)

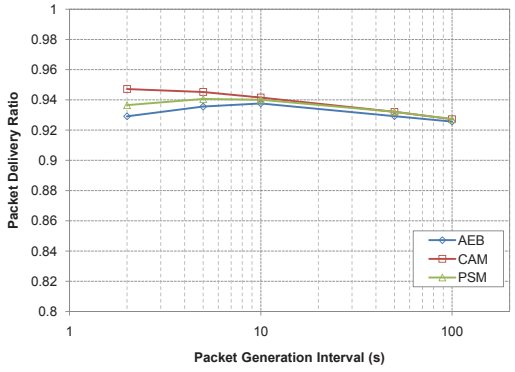
Figure 3.7: Impact of number of nodes: (a) Energy consumption; (b) Average packet delay; (c) Packet delivery ratio.



(a)



(b)



(c)

Figure 3.8: Impact of traffic load: (a) Energy consumption; (b) Average packet delay; (c) Packet delivery ratio.

3.4 Summary

In this chapter, a new adaptive synchronous clock based sleep scheduling protocol for DTNs is proposed. The AEB protocol exploits intermittent connection characteristic and varying contact rates in DTNs. By adjusting the beacon periods depending on the trend of contact availability and using tunable parameters for different network environments, considerable amount of energy can be saved without greatly sacrificing the average packet delay and packet delivery ratio. Performance of the AEB protocol is compared with other synchronous clock based protocols in DTNs using spray and wait routing protocol in RWP mobility model. Simulation results under various network conditions indicate that the proposed protocol consumes 65 to 80 percent of energy compared with PSM and 20 to 60 percent of energy compared with CAM, while keeping the average packet delay and delivery ratio comparable with CAM.

Chapter 4

Adaptive Asynchronous Sleep Scheduling Protocols in Delay Tolerant Networks

As presented in Chapter 3, nodes can adaptively adjust the frequency of beacons depending on the contact availability to reduce the energy waste cause by the idle listening problem. However, in reality, perfect clock oscillators do not exist, and clock reading errors among nodes are common. Moreover, due to large inter-contact durations and infrequent message exchanges in DTNs, the global clock synchronization is difficult. In addition, synchronization protocols cannot typically assume to rely on the Global Positioning System (GPS) that requires a large amount of energy and a line of sight to the satellites.

Therefore, in this chapter, we focus on asynchronous clock based sleep scheduling protocols in DTNs, meaning that the protocols do not require global clock synchronization among nodes and consider intrinsic characteristics of DTNs. Further, the sleep schedules are distributed and adaptive, meaning that each node can independently and asynchronously chose the power saving level depending on its current condition. The contribution and significance of this research work [34, 35] are as follows

- First, we propose adaptive asynchronous sleep scheduling protocols for DTNs, which pro-

vide multiple levels of power saving and are energy efficient under intermittent connectivity for minimizing energy consumption. Different duty cycle lengths are created by simple systematic algorithms that hierarchically arrange cyclic difference sets.

- Second, several implementation issues considering the unique characteristics of DTNs and approaches to maximize energy efficiency are discussed. We discuss the details of the asynchronous frame structure, neighbor discovery process, and adaptive power saving levels.

- Third, both theoretical analysis and simulation results are given to evaluate the performance of the proposed protocols in comparison with other existing adaptive asynchronous sleep scheduling protocols. Theoretical analysis results demonstrate that the proposed protocols are more energy efficient while guaranteeing required connectivity. Simulation results verify that the proposed protocols, while maintaining similar packet delivery delay and delivery ratio, can reduce the energy waste in the idle listening mode up to 35 percent.

The remainder of this chapter is organized as follows. We first present a comprehensive survey of existing asynchronous sleep scheduling protocols in multihop wireless networks in Section 4.1. After presenting preliminaries of adaptive asynchronous sleep scheduling protocols in Section 4.2, we propose two new adaptive asynchronous sleep scheduling protocols for DTNs in Section 4.3, and discuss implementation issues in Section 4.4. The performance of the proposed protocols is evaluated using theoretical analysis in Section 4.5 and simulations in Section 4.6. Finally, the chapter is concluded in Section 4.7. Summary of important symbols used in this chapter is given in Table 4.1 for easy reference.

4.1 Related Work

In this section, asynchronous sleep scheduling protocols in Mobile Ad-hoc Networks (MANETs) and DTNs are presented. They can be classified into non-adaptive or adaptive protocols depending on their ability to provide multiple power saving levels.

Table 4.1: Summary of important symbols used

Symbol	Definition
E_i	list of awake elements of set i
n_i	number of slots in set i
L_s	length of a slot
L_f	length of a frame ($= L_s \times n_i$)
L_A	length of the ATIM window
$C_{h,m}(E_i)$	set i repeated to m slots with a shift
$RCP\{E_i, E_j\}$	rotational closure probability between set i and set j
T_{ICD}	inter-contact duration
T_{CD}	contact duration
T_{SD}	slot delay duration
T_R	data exchange duration
P_i	power saving level i
P_c	contact probability
R_i	active ratio of set i
NS_{ij}	neighbor sensitivity between set i and set j

4.1.1 Non-Adaptive Asynchronous Sleep Scheduling Protocols

Quorum-based Protocol (QPS) [20] is the first work in the asynchronous sleep scheduling protocols. Assuming equal sized intervals, called *slots*, QPS guarantees at least two overlapping awake intervals between two nodes while being awake $(2\sqrt{n} - 1)$ slots out of n slots. This is achieved by each node using an awake schedule set that is formed by arbitrarily choosing a row and a column from a square n space, as shown in Figure 4.1a.

More energy efficient sleep schedules are constructed in Cyclic Difference Set Based Protocol (CDS) [22, 42] by using cyclic difference sets in combinatorial mathematics [55]. The construction of difference sets is a projective plane problem with parameters (v, k, λ)=$(q^2+q+1, q+1, 1)$ where a set consisting of v total slots constructed with k awake slots guarantees at least one overlapping awake slot. The difference set used by the CDS provides the theoretically minimum awake slot ratio. As shown in Figure 4.1b, one overlapping slot is guaranteed between any two

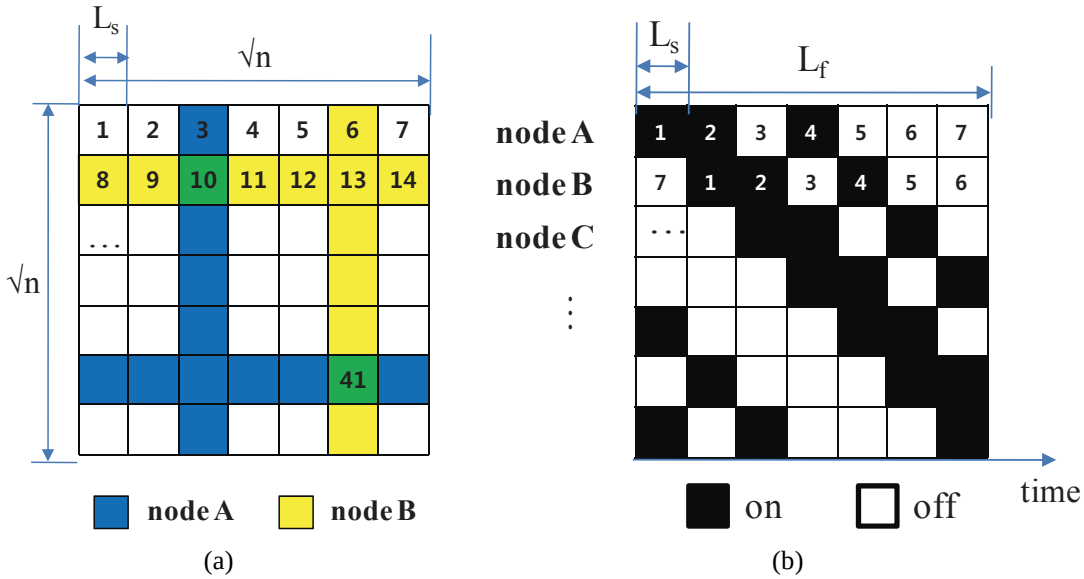


Figure 4.1: Asynchronous sleep scheduling protocols: (a) Quorum based ($n = 49$), the schedules of A and B have overlapping awake slots at slots 10 and 41; (b) Cyclic difference set based ($v = 7$), the awake slot 2 of schedule A is overlapping with the awake slot 1 of schedule B.

sets. In addition, CQPM [56] that has similar concepts to QPS and CDS is also independently proposed.

Table 4.2 gives the summary of non-adaptive asynchronous sleep scheduling protocols and their properties including the active ratio and the neighbor sensitivity.

4.1.2 Adaptive Asynchronous Sleep Scheduling Protocols

The drawback of QPS and CDS is their failure to guarantee overlapping awake slots between sets with different sizes. In other words, nodes may not be able to discover each other if sets with different sizes are used. Therefore, they cannot adjust to network dynamics, such as traffic load, topology, or node mobility to further optimize performance. To overcome this problem, several adaptive asynchronous sleep scheduling protocols have been proposed. There exists a tradeoff

Table 4.2: Summary of non-adaptive asynchronous sleep scheduling protocols

Protocol	**Year**	**Active Ratio**	**Neighbor Sensitivity**	**Remarks**
QPS [20, 21]	2002	$R_{QPS(\sqrt{n})} = (2\sqrt{n}-1)/n$	$L_s \times (n-\sqrt{n}+1)$	The first work in asynchronous sleep scheduling. Requires n to be a perfect square.
CDS [22, 42]	2005	$R_{CDS(n)} \approx \sqrt{n}/n$	$L_s \times n$	Achieves the theoretically optimal active ratio. Requires $n = q^2+q+1$.
CQPM [56]	2006			Concurrent independent research similar to [22] and [24].

between the delay and the energy efficiency. For both QPS and CDS, lower ratios of awake slot can be achieved in larger sets. However, as the size of a set increases, the required minimum number of slots to guarantee at least one overlapping awake slot between two different sets also increases. For example, in CDS, the active ratio of $\frac{q+1}{q^2+q+1}$ is achievable with at least q^2+q+1 slots per frame, and in QPS, the active ratio of $\frac{2n-1}{n^2}$ is achievable with at least n^2 slots. Adaptive protocols exploit this tradeoff to increase energy efficiency by adjusting sets sizes depending on the network condition.

Based on QPS, AQPS [24] and AQEC [23] allow different nodes to use individual n's to provide an adaptive power saving. However, as in QPS, each n must be a perfect square, and active slot ratio of sets are twice the optimal active slot ratio provided by CDS [24]. In DTNs, CAPM [41] determines the sleep schedule based on known node densities and traffic load requirements. Although the determined sleep schedules are fixed and not adaptive, we classify CAPM as an adaptive protocol since the sleep schedule of CAPM that achieves the lowest active slot ratio is the same as the sleep schedule of AQEC, as will be explained in Section 4.5.1.

Based on CDS, AAPM [25] provides an adaptive power saving by constructing a collection of pairs, called an *AA-quorum space*. Although AAPM achieves active slot ratios close to the optimal active ratio, the eligible values in the *AA-quorum space* must be a prime number. Moreover, we will show in Section 4.5.1 that AAPM is only effective for small n's ($n \leq 47$).

Based on *cyclic read-write coteries* used for managing replicated data, ACQ [26, 27, 57]

constructs two levels of power saving for clustered ad-hoc networks: a symmetric-quorum (s-quorum) and an asymmetric-quorum (a-quorum). ACQ allows the nodes using the s-quorum to discover other nodes using the s-quorum or the a-quorum. However, the nodes using the a-quorum cannot discover other nodes using the a-quorum.

Most recently, HQS [28] provides a generalization of QPS that constructs sets with arbitrary cycle lengths using a projection concept. At least one overlap of awake interval between any two set is guaranteed within a projected cycle length. Two algorithms, Extended Grid HQS (EGHQS) and Difference Set HQS (DSHQS), are proposed. Although they are adaptive, we show in Section 4.5.1 that the active slot ratio of EGHQS is close to AQEC and the active ratio of DSHQS is close to the optimal active ratio for small n's, such as $n \leq 25$. The projection algorithms used in HQS become inefficient as n increases. Therefore, HQS is more suitable for static and densely deployed ad-hoc networks that require fine-tunable cycle lengths.

Table 4.3 gives the summary of adaptive asynchronous sleep scheduling protocols including the active ratio and the neighbor sensitivity.

Table 4.3: Summary of adaptive asynchronous sleep scheduling protocols

Protocol	**Year**	**Active Ratio**	**Neighbor Sensitivity**	**Remarks**
AQPS [24]	2005	$\geq \sqrt{2n}/n$	$L_s \times n$	Adaptive version of QPS [20, 21] based on a torus system [58].
AQEC [23]	2006	$R_{QEC(\sqrt{n_i})} = (2\sqrt{n_i} - 1)/n_i$ where $0 \leq n_i \leq n_j$	$L_s \times (n_j - \sqrt{n_j} + 1)$	Similar to AQPS [24]. Requires n to be a square of an integer.
CAPM [41]	2007	$R_{CAPM(\sqrt{n})} \geq (2\sqrt{n} - 1)/n$	$L_s \times (n - \sqrt{n} + 1)$	Frame structure similar to AQEC [20].
ACQ [26]	2007	a-quorum: $R_{ACQ(n)} = p/n$, s-quorum: $R_{ACQ(n)} = (\phi + q - 1)/n$ where $1 \leq \phi \leq n$, $p = \lceil n/\phi \rceil$, $q = \lceil (n+1)/2\phi \rceil$	$L_s \times n$	Overlap not guaranteed among a-quorums.
AAPM [25]	2007	$R_{AAPM(n_i)} \approx \sqrt{n_i}/n_i$ where $0 \leq n_i \leq n_j$	$L_s \times n_j$	Adaptive space is limited to prime n for $n \leq 47$.
EGHQS [28]	2008	$R_{EGHQS(n_i)} = (\phi_i + q_i - 1)/n_i$ where $0 \leq n_i \leq n_j \leq n_{d-1}$, $q_i = \lfloor n_i/\phi_i \rfloor$, $\phi_i = \min\{\lfloor \sqrt{n_i} \rfloor, \lceil \sqrt{(n_{d-1}+1)/2} \rceil\}$	$L_s \times (n_j + \phi_i - 1)$	Similar to AQEC [23], but allows any n using a projection algorithm.
DSHQS [28]	2008	$R_{DSHQS(n_i)} = (\phi + q_i - 1)/n_i$ where $0 \leq n_i \leq n_j \leq n_{d-1}$, $\phi = \lceil \sqrt{(n_{d-1}+1)/2} \rceil$, $q_i = \lceil (n_i + 1)/2\phi \rceil$	$L_s \times (\lfloor (n_i - 1)/2 \rfloor + n_j + \phi - 1)$	Adaptive version of CDS [22] using a projection algorithm.
EACDS	2009	$R_{EACDS(i)} = (k_I (k_E)^i)/(v_I (v_E)^i)$, $n_i = v_I (v_E)^i$ where $0 \leq i \leq j$	$L_s \times n_j$	I and E are difference sets.
MACDS	2009	$R_{MACDS(i)} = (k_I k_{Mi})/(v_I v_{Mi})$, $n_i = v_I (v_{Mi})$ where $0 \leq i \leq j$	$L_s \times n_j$	I and M_i are difference sets. Active ratio close to the optimal.

4.2 Preliminaries

In this section, we present theoretical basis and necessary conditions for constructing adaptive asynchronous sleep scheduling protocols.

4.2.1 Cyclic Difference Set

In the CDS protocol, the cyclic difference set [55, 59, 60] in combinatoric mathematics is used to construct the basic sleep schedule. The *difference set* is designed such that for v total slots, called a *block*, with k "on" slots, there are exactly λ overlapping "on" slots among k different blocks. If the set of blocks is a square matrix with each of k elements appearing once in every block, this design is called a *symmetric design* as defined below.

Definition 1. (Symmetric Design) *A set with v elements and k subset elements $D : a_1, \ldots, a_k \pmod v$ is called a symmetric (v, k, λ)-difference set if and only if the sets $B_i : a_{1+i}, a_{2+i}, \ldots, a_{k+i} \pmod v$, $i = 0, \ldots, v-1$ are a cyclic (v, k, λ) block design.*

The blocks of a symmetric design are a cyclic shift of each other, and the difference set satisfying the symmetric design property is called a *cyclic difference set* as formally defined below.

Definition 2. (Cyclic Difference Set) *A set with v elements and k subset elements $D : a_1, \ldots, a_k \pmod v$ is called a (v, k, λ)-difference set if for every $d \neq 0 \pmod v$ there are **exactly** λ ordered pairs (a_i, a_j), $a_i, a_j \in D$ such that $a_i - a_j = d \pmod v$.*

The cyclic shift property of cyclic difference sets allows overlapping slots under any cyclic time shifts. An example using the cyclic difference set $(7, 3, 1)$ is illustrated in Figure 4.2.

The construction of cyclic difference sets follows the Singer's Theorem [61] as given in Eq. 4.1.

$$(v, k, \lambda) = \left(\frac{q^{n+2} - 1}{q - 1}, \frac{q^{n+1} - 1}{q - 1}, \frac{q^n - 1}{q - 1} \right) \tag{4.1}$$

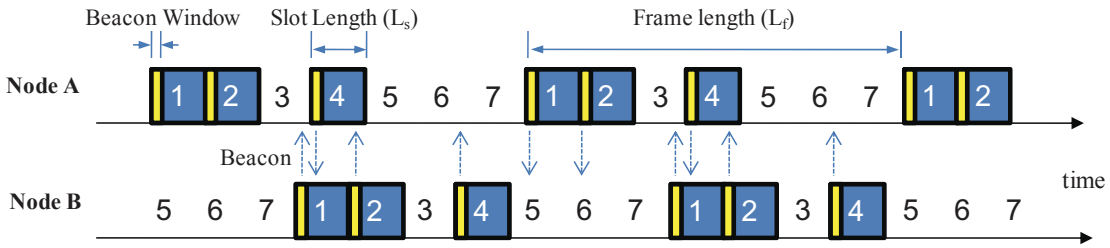


Figure 4.2: An example of overlap of slots between two nodes using $E_{(7,3,1)}$=$\{1, 2, 4\}$ under cyclic clock shifts. Slot 4 of node A is partly overlapping with slots 1 and 2 of node B.

From Eq. 4.1, a cyclic difference set exists for some prime power q and some positive integer n. For a special case with $n = 1$, the difference set becomes a projective plane problem. Then, the Eq. 4.1 becomes

$$(v, k, \lambda) = (q^2 + q + 1, q + 1, 1) \tag{4.2}$$

The possible cyclic difference sets with $\lambda = 1$ using the Singer's theorem is given in Table 4.4. Note that for any particular set of parameters v, k, and λ satisfying Definition 2, there are no other difference sets having these parameters.

The cyclic difference set design problem in combinatoric is formulated for asynchronous sleep scheduling protocols in CDS [22, 42]. A row of a block of length v, representing a frame of length L_f consisting of v slots of length L_s, have overlapping intervals combined to the length of a slot within L_f even if the slot boundaries are not synchronized under time shifts. The advantage of CDS in comparison with other asynchronous sleep scheduling protocols is that it constructs frames with the lowest active ratio. However, the drawback is that the sleep schedules are not adaptive. They are designed for ad-hoc network with high node degree and low mobility, and its non-adaptive structure is unsuitable for DTNs which have long delayed intermittent connections and dynamic mobility. An adaptive structure is suggested in CDS [42] using *superposition* of sets to reduce beacon pollution. However, a super-frame of length v^2 is created from a frame of length v, and the difference in frame sizes becomes too large to be practical for a large v.

Table 4.4: Possible cyclic difference sets ($\lambda = 1$)

v	**k**	λ	**Set Sequence**	**Active Ratio**
7	3	1	1 2 4	0.429
13	4	1	1 2 4 10	0.308
21	5	1	1 2 5 15 17	0.238
31	6	1	1 2 4 9 13 19	0.194
57	8	1	1 2 4 14 33 37 44 53	0.140
73	9	1	1 2 4 8 16 32 37 55 64	0.123
91	10	1	1 3 7 8 19 22 32 55 64 72	0.110
			etc.	

4.2.2 Rotational Closure Property

Since nodes are not synchronized to wakeup periodically in an asynchronous clock based environment, an important requirement of asynchronous sleep scheduling protocols is the *rotational closure property* [24] that is satisfied between two sets if they have at least one overlapping slot within a cycle length of n for any cyclic shifts. This property allows two nodes to discover each other even under different time shifts. In order to describe this requirement, we define a *rotational closure probability* which represents the probability of existence of at least one awake slot between two different schedules for all possible cyclic shifts. RCP between sets E_i and E_j for $n_i \leq n_j$ is formulated as

$$RCP\{E_i, E_j\} = \left(\sum_{h=0}^{n_i} \theta_h \right) / n_i \tag{4.3}$$

$$\theta_h = \begin{cases} 1 & \text{if } C_{h,n_j}(E_i) \cap C_{0,n_j}(E_j) \neq \emptyset \\ 0 & \text{if } C_{h,n_j}(E_i) \cap C_{0,n_j}(E_j) = \emptyset \end{cases}$$

where $C_{h,m}(E)$ is an *extended cyclic set* which represents a set E repeated to length m with a cyclic shift h. If $RCP\{E_i, E_j\} = 1$, then E_i and E_j are called *rotational sets*, and the rotational closure property is satisfied between the two sets. If $RCP\{E_i, E_j\} < 1$, set E_i and set E_j are not rotational sets, indicating that there exist cyclic shift cases where two sets do

not have an awake overlapping slot. Note that, for two same sets, $RCP = 1$, but for two different sets, $RCP \leq 1$. If two different sets have $RCP = 1$, then they are called *adaptive sets*, otherwise, they are *non-adaptive sets*. For example, between $(21, 5, 1)$ and $(13, 4, 1)$, $C_{0,21}(21, 5, 1) \cap C_{9,21}(13, 4, 1) = \{1, 2, 5, 15, 17\} \cap \{6, 10, 11, 13, 19\} = \emptyset$ and $C_{0,21}(21, 5, 1) \cap C_{6,21}(13, 4, 1) = \{1, 2, 5, 15, 17\} \cap \{3, 7, 8, 10, 16, 20, 21\} = \emptyset$. Therefore, for all possible shifts, $RCP\{(21, 5, 1), (13, 4, 1)\} = 11/13$, and they are non-adaptive sets. For example, $C_{5,30}(E_{(15,5,1)})$ = $\{6, 7, 8, 9, 13, 21, 22, 23, 24, 28\}$ has no common awake slot with $C_{0,30}(E_{(30,6,1)})$ = $\{1, 2, 3, 4, 5, 10\}$. Therefore, they are non-adaptive sets, i.e., $RCP\{E_{(15,5,1)}, E_{(30,6,1)}\} < 1$.

4.3 Adaptive Cyclic Difference Set System

In well connected traditional ad-hoc networks, the cycle lengths of adaptive sleep schedules need to be relative small and fine-grained to provide short end-to-end delays [28]. However, in DTNs with sparsely connected nodes, the cycle lengths are relative large, up to several seconds in many DTN scenarios, since small cycle lengths waste energy without discovering many more contacts [41, 50]. Also, the main source of energy waste that needs to be reduced is caused by the idle listening problem. Therefore, in this section, we propose Adaptive Cyclic Difference Set (ACDS) protocol that constructs two different types of sleep schedules suitable for DTNs. ACDS is designed with multiple power saving levels that are energy efficient for large cycle lengths.

4.3.1 Construction of Exponential ACDS

In this subsection, Exponential ACDS (EACDS) is proposed. The basic strategy is to use hierarchical arrangements of sets. As shown in Figure 4.3, a difference set called an initial set at power saving level 1 (P_1 = I = (v_I, k_I, λ_I)) is scaled by another difference set called an exponential set (E = (v_E, k_E, λ_E)) to create a hierarchical set with power saving level 2 (P_2). The hierarchical set can be scaled again with E to create yet a higher level hierarchical set P_3 which provides higher energy efficiency than P_2 at the cost of lower contact opportunities. The scaling is done by the Kronecker product, also called a direct product, denoted by $\otimes$. It operates on two matrices of arbitrary size resulting in a block matrix as defined below.

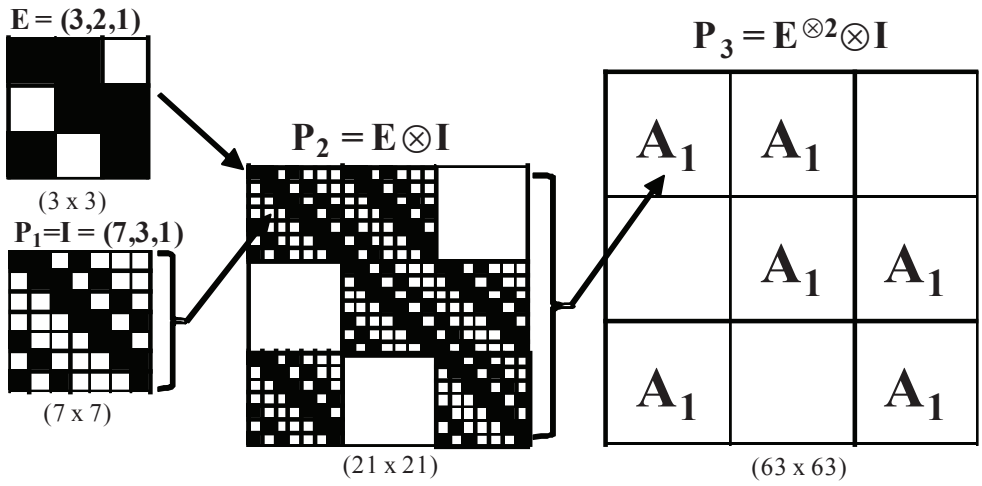


Figure 4.3: Construction of exponential adaptive CDS

Definition 3. (Kronecker Product) *Let $A = (a_{ij})$ be a $m \times n$ matrix and $B = (b_{kl})$ be a $p \times q$ matrix. Then, the Kronecker product of A and B is the $mp \times nq$ block matrix $C = (c_{\alpha\beta}) = a_{ij}b_{kl}$ where $\alpha = p(i-1)+k$ and $\beta = q(j-1)+l$.*

$$C = A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \tag{4.4}$$

In our design, A is the scaling set (E) and B is the initial set (I). Let nth Kronecker power of A be the n-fold Kronecker product of A with itself, i.e.,

$$A^n = \prod_n \otimes A \tag{4.5}$$

Then, the hierarchical sets can be expressed as $P_1 = I$, $P_2 = E \otimes I$, and $P_3 = E^2 \otimes I$.

For each block of size $v = v_I v_E^n$, EACDS guarantees that there is at least one overlapping slot between different blocks of size less than or equal to v as proved in Theorem 1.

Theorem 1. (Rotational Closure Property of the Exponential Hierarchical Design) *Given two sets $P_i = E^{i-1} \otimes I$ and $P_j = E^{j-1} \otimes I$ where $i \leq j$. For $n = v_I v_E^j$, $C_{a,n}(P_i) \cap C_{b,n}(P_j) \neq \emptyset$, $\forall a, b : 0 \leq a, b \leq n$.*

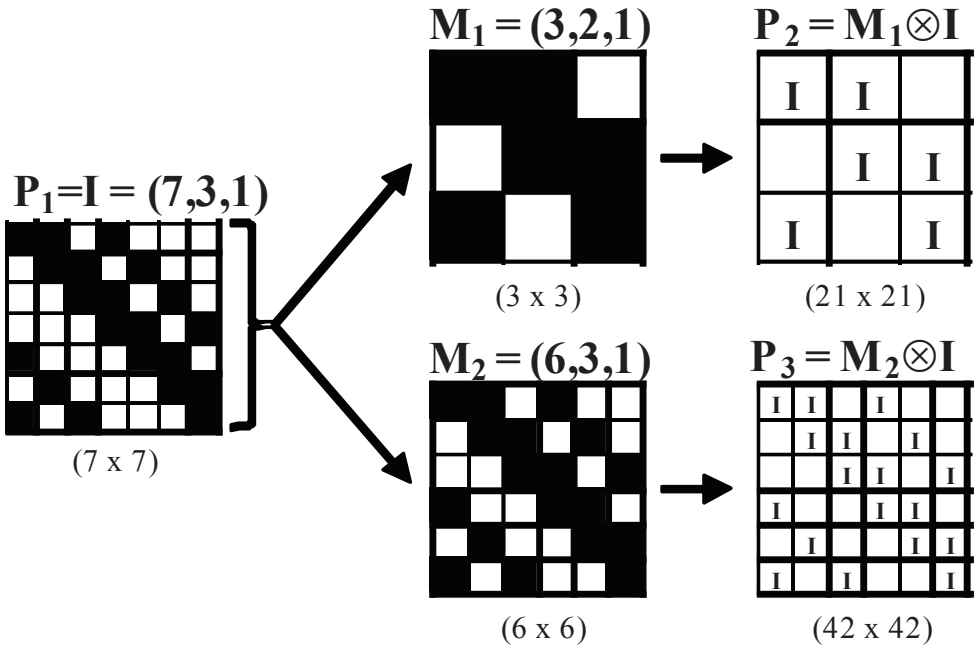


Figure 4.4: Construction of multiplicative adaptive CDS

Therefore, the progressive hierarchial structure of EACDS always guarantees the rotational closure property between the different levels of hierarchy as well as between the same level.

4.3.2 Construction of Multiplicative ACDS

In this subsection, multiplicative ACDS (MACDS) is proposed. The basic strategy is to use hierarchical arrangements of sets as in EACDS, but a multiplier set ($M = (v_M, k_M, \lambda_M)$) is used instead of an exponential set (E). As shown in Figure 4.4, the initial set ($P_1 = I$) is scaled by a multiplier set (M_1) to create a hierarchical set ($P_2 = M_1 \otimes I$) and scaled by another multiplier set (M_2) to create another hierarchical set ($P_3 = M_2 \otimes I$). M_1 and M_2 are in the same hierarchical level but have different n's. For the frame of size $n = v_I v_M$, there is at least one overlapping slot between different sets with sizes less than or equal to v. This is because the multiplier sets belong to a *rotational sets group* defined below:

Definition 4. (Rotational Set Group) *A group of difference sets,* $M = \{M_1 = (v_{M1}, k_{M1}, \lambda_{M1}),$

$M_2 = (v_{M2}, k_{M2}, \lambda_{M2}), \cdots, M_i = (v_{Mi}, k_{Mi}, \lambda_{Mi})\}$ *is called a rotational set group if* $RCP\{M_i, M_j\} = 1$ *for all* i *and* j.

For example, M_1 = $(3, 2, 1)$ and M_2 = $(6, 3, 1)$ belong to the same *rotational set group* since there is at least one overlapping slot between M_1 and M_2 within 6 slots.

The following corollary give the rotational closure property among hierarchical sets generated using the *rotational set group* on I.

Corollary 1. (Rotational Closure Property of the Multiplication Hierarchical Design) *Given two sets* $P_i = M_{i-1} \otimes I$ *and* $P_2 = M_{j-1} \otimes I$ *where* $v_{Mi} \leq v_{Mj}$ *and* $RCP\{M_i, M_j\} = 1$*, there exists an overlapping interval of at least* v_I *in* v_{Mj}*. Therefore, similar to proof for Theorem 1, we can prove that for* $n = v_I v_{Mj}$*,* $C_{a,n}(P_i) \cap C_{b,n}(P_j) \neq \emptyset$*,* $\forall a, b : 0 \leq a, b \leq n$.

The sets satisfying the above condition can be included in the same rotational set group. However, finding the rotational set group is not straight forward. Same cyclic difference set is a rotational set. However, two different cyclic difference sets are not necessarily rotational sets. For example, (15,5,1) = {1,2,3,4,8} repeated with +5 shift ($C_{5,30}(15, 5, 1)$ = $\{6, 7, 8, 9, 13, 21, 22, 23, 24, 28\}$) has no overlap with $(30, 6, 1) = \{1, 2, 3, 4, 5, 10\}$. In this case, two nodes will never be able to discover each other. In addition, the sets in a group may not be necessary multiple of one another. For example, $(7, 3, 1)$ and $(13, 4, 1)$ are rotational sets. The groups can be found by an exhaustive search.

From the example, the candidates for rotational sets are not limited to cyclic difference sets. The relaxed cyclic difference set [62] as defined below can also be used to construct sleep schedules. Different from the normal cyclic difference set, the relaxed cyclic difference sets [62] provide cyclic difference sets for any $n_s = v$ as shown in Table 4.5.

Definition 5. (Relaxed Difference Set) *A set with* v *elements and* k *subset elements* $D : a_1, \ldots, a_k \pmod v$ *is called a relaxed* (v, k)*-difference set if for every* $d \neq 0 \pmod v$ *there are at least* λ *ordered pairs* $(a_i, a_j), a_i, a_j \in D$ *such that* $a_i - a_j = d \pmod v$.

In MACDS, the combination of (v_M, k_M, λ_M) is chosen from the rotational set group. For example, M_1 = $(3, 2, 1)$, M_2 = $(6, 3, 1)$, M_4 = $(12, 4, 1)$, and M_5 = $(24, 6, 1)$ can be grouped

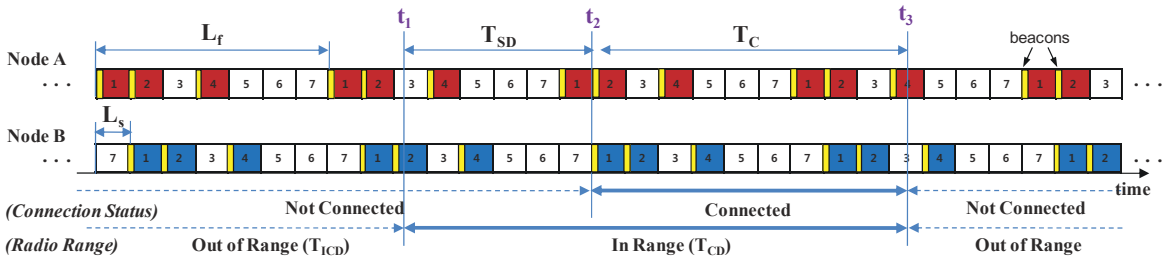


Figure 4.5: Neighbor discovery in DTNs. At t_1, nodes A and B come within range of each other. At t_2, after T_{SD}, awake slots of each node overlap with each other, and they are able to exchange connection setup messages. At t_3, they move out of range with each other and are disconnected.

together in the same rotational set group, and adaptive power saving levels with $I = (57, 8, 1)$ can be constructed as $P_1 = (57, 8, 1)$, $P_2 = (171, 16, 1)$, $P_3 = (342, 24, 1)$, $P_4 = (684, 32, 1)$, and $P_5 = (1368, 48, 1)$. Comparing EACDS and MACDS, MACDS is more energy efficient, while EACDS is more useful in scenarios where many power saving levels are needed. In addition, it is possible to use a combination of the two proposed protocols.

4.4 Implementation Issue

In this section, we discuss implementation issues considering sparse node density and mobility in DTNs, including frame structure, neighbor discovery process, data exchange process, and optimizing energy efficiency.

4.4.1 Asynchronous Frame Structure

As shown in Figure 4.5, we follow the same frame structure as in CDS [22, 42] where in each node, a frame of length L_f consisting of multiple slots of length L_s is consecutively repeated. The combination of awake and sleep slots are determined by the set constructed in Section 4.3. In order to discover other neighboring nodes, each node transmits a beacon message at the beginning of each awake slot. If the beacon is heard by another node, a connection can be established between the two nodes by exchanging connection set up messages.

Table 4.5: Possible relaxed cyclic difference sets ($\lambda = 1$)

v	**k**	λ	**Set Sequence**
4	3	1	1 2 3
5	3	1	1 2 3
6	3	1	1 2 4
			...
12	4	1	1 2 4 8
			...
24	6	1	1 2 3 4 8 16
			...
48	8	1	1 2 3 6 10 21 27 37
49	8	1	etc.

4.4.1.1 Minimum Slot Length

The slot length is directly associated with the energy efficiency ratio. In order to minimize energy consumption, the number of slots per frame should be maximized. However, the length of each slot has to be long enough to allow exchange of connection setup messages between contacted nodes. The length of each slot must be at least twice the length of the ATIM window:

$$L_s \geq 2L_A \tag{4.6}$$

This condition ensures nodes to exchange necessary setup information with other nodes upon contact. For example, if the minimum L_A is 10 ms, then L_s has to be at least 20 ms. This necessary condition is explained in Theorem 2. Additionally, L_s may possibly be shortened, if partial exchange of connection setup messages can trigger the activation of the following slot.

Theorem 2. (Minimum Slot Length of Asynchronous Sleep Scheduling Protocol) *The slot length should be at least* $2 \times L_A$ *to guarantee an overlapping active interval that is long enough for the connection establishment.*

4.4.2 Neighbor Discovery in DTNs

Neighbor discovery is responsible for finding other nodes in the network. In asynchronous sleep scheduling protocols, a connection can be established between two nodes within transmission range during common awake slots. Although the exact location of the overlap is not known, the overlap is guaranteed with an upper bound NS which represents the worst-case delay for a node to detect a new node in its coverage.

In traditional multihop wireless networks with high node degree, NS determines the required one-hop delay as suggested in CDS [22]. However, in DTNs, with mobile nodes having limited transmission range, a connection between two nodes can be established only if two nodes in motion have a sufficient common awake interval to exchange messages before they move out of range of each other. As illustrated in Figure 4.5, due to frequent disconnections and long delays between connections, nodes are often isolated, and alternate between an inter-contact duration and a contact duration, represented by T_{ICD} and T_{CD}, respectively. T_{ICD} is determined by how sparsely the nodes are deployed in a given topology, and T_{CD} is determined by how fast the nodes are moving with given radio transmission range. Since T_{ICD} is much larger than T_{CD} [29, 32, 33], nodes are often disconnected and one-hop delays are negligible compared to T_{ICD}. Therefore, L_f is chosen depending on the NS that achieves the required contact probability rather than the required one-hop delay.

Contact probability is the ratio of successful connections among mobile nodes given a certain L_f, and it is dependent on the connection duration and the NS. The amount of time required for a node to have an overlapping slot with another node in range is called a slot delay duration, represented by $T_{SD} \leq NS$. Due to mobility of nodes and $T_{SD} \geq 0$, the actual connection duration, represented by T_C, is less than or equal to T_{CD}. Let T_R represent the required time duration to exchange all data packets between two nodes. Then, the condition $T_{CD} \geq (T_{SD} + T_R)$ must be satisfied to guarantee successful transmission of all packets. If $NS > T_{CD}$, two nodes might miss each other. On the other hand, if NS is too small, then nodes will consume unnecessary power during idle listening periods.

The achievable contact probability given a NS can be estimated using the CDF of T_{CD} under random waypoint (RWP) model as in [51] or assuming a certain probability distribution of T_{CD} as in [63]. Figure 4.6 shows the minimum T_{CD} required to achieve a certain contact probability

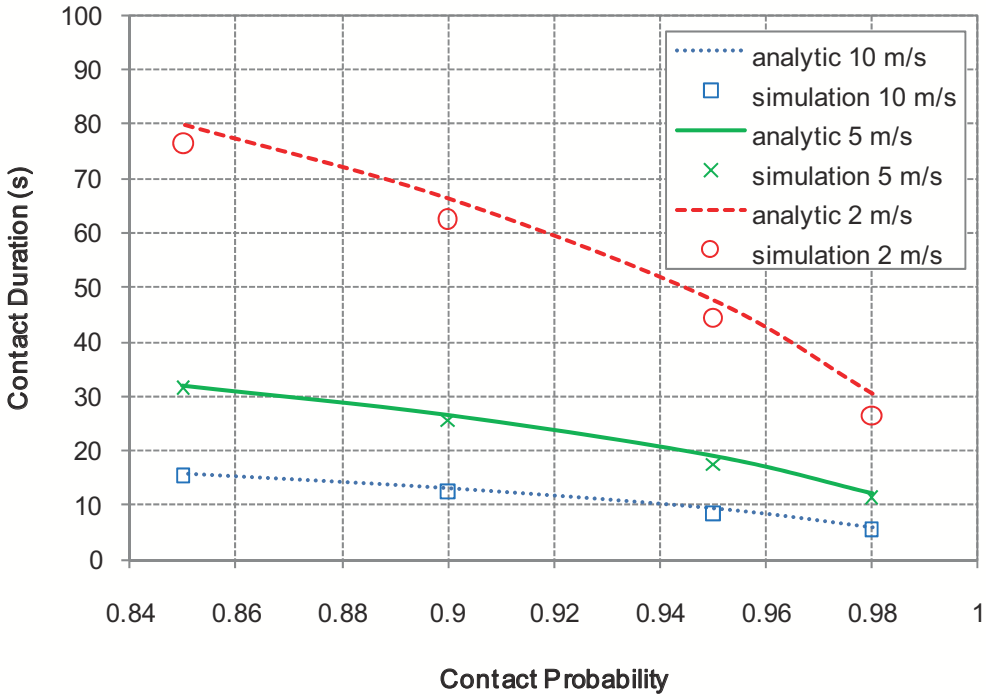


Figure 4.6: Required contact duration for given contact probability under different node velocities. 20 mobile nodes with 250 m radio transmission range moving according to RWP movement model within 3000 m by 3000 m area.

under RWP model. For instance, the NS of nodes moving at 10 m/s needs to be less than $T_{CD} = 8$ s to achieve the contact probability above 0.95.

4.4.3 Adaptive Power Saving Levels

Multiple power saving levels (PSLs) have frames with different lengths that provide a tradeoff between connectivity and energy efficiency. While well connected traditional ad-hoc networks require fine-grained multiple PSLs with relatively small frame sizes to minimize delays at each hop [28], sparsely connected DTNs require multiple PSLs, though they may be less fine-grained, to minimize idle listening periods. We define PSLs in ACDS as follows. The level with the initial set (I) is called PSL 1 (P_1). Here L_f of P_1 is chosen so that it achieves the maximum

required connectivity. Then, PSLs are labeled in an increasing order of frame lengths. Higher PSLs provide higher energy efficiency at the cost of lower contact opportunities.

Since we consider opportunistic contacts, contact times and durations are only known with some probability distribution [52, 63, 64]. As explained in the previous section, under sparse density of nodes, the delivery performance depends on the contact probability between nodes. Consider that nodes can approach each other at any angle, the probability distribution of T_{CD} is a function of speed and transmission range of nodes [51]. Therefore, given a fixed transmission range, PSLs can be selected based on the node speed.

4.4.3.1 Sleep Schedules of PSLs

We provide an example to show how sleep schedules of PSLs can be constructed given a range of node speeds v_{min} and v_{max} and a required contact probability P_c for a given routing protocol. In order to estimate the required range of NS, $\min(T_{CD})$ and $\max(T_{CD})$ are calculated between nodes moving at v_{max} and v_{min}, respectively, using a known mobility distribution or a mathematical approximation [51]. Since $T_{CD} \geq (T_{SD} + T_R)$, $\min(NS) = (\min(T_{CD}) - T_R)$ and $\max(NS) = (\max(T_{CD}) - T_R)$. If $\min(NS) = 8$ s and $\max(NS) = 80$ s, n_i for P_1 is $n_1 = \min(NS)/L_s = 400$. Next, we construct the sleep schedules using MACDS. A cyclic difference set which satisfies $v_I \leq n_1$ is chosen as $I = P_1$. Finally, $I = \{381, 20, 1\}$ and $M = \{M_1, \ldots, M_8\} = \{(v_{M1}, k_{M1}, 1), \ldots, (v_{M8}, k_{M8}, 1)\} = \{(3, 2, 1), (4, 3, 1), (5, 3, 1), (6, 3, 1), (7, 3, 1), (8, 4, 1), (9, 4, 1), (10, 5, 1)\}$[1] can be used to produce multiple PSLs $\{P_1, \ldots, P_9\} = \{(381, 20, 1), \ldots, (3810, 100, 1)\}$ having frame lengths from 7.62 s to 76.2 s.

4.4.3.2 Selecting PSLs

A moving node can adaptively select the highest possible P_i that achieves desired contact probability. In addition, PSLs of the proposed protocol can also be integrated with existing searching/probing protocols. For example, PSLs can be selected using statistical information of

[1]The exact parameters of sets are $(381, 20, 1) = \{1, 2, 20, 29, 97, 119, 152, 154, 177, 203, 241, 255, 291, 297, 301, 308, 338, 362, 367, 370\}$, $(3, 2, 1) = \{1, 2\}$, $(4, 3, 1) = \{1, 2, 3\}$, $(5, 3, 1) = \{1, 2, 3\}$, $(6, 3, 1) = \{1, 2, 4\}$, $(7, 3, 1) = \{1, 2, 4\}$, $(8, 4, 1) = \{1, 2, 3, 5\}$, $(9, 4, 1) = \{1, 2, 4, 5\}$, and $(10, 5, 1) = \{1, 2, 3, 4, 6\}$.

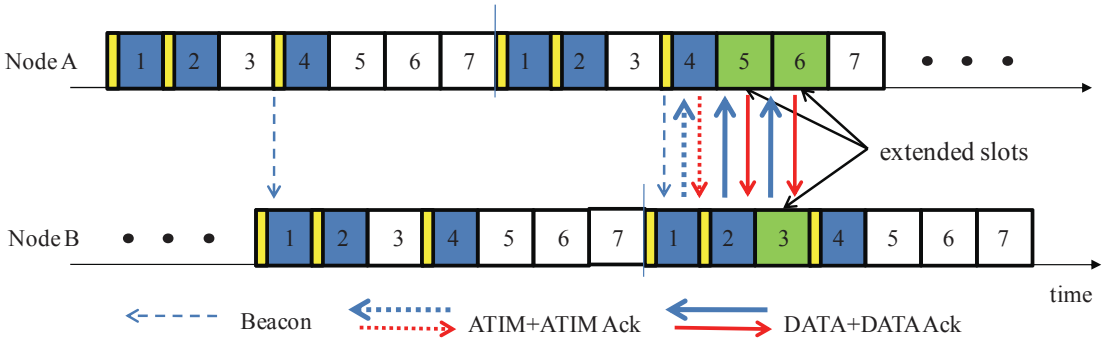


Figure 4.7: Slot extension mechanism

contact duration and waiting times as in [49, 63], and can also adaptively selected by exploiting self-similarity of contact arrival rate as in [33, 63]. The minimum amount of time that a node needs to be awake to discover other nodes in order to achieve required contact probability is called an optimal searching/probing interval, and it is equivalent to NS in this work.

4.4.4 Data Delivery using Asynchronous Frames

After exchanging connection setup messages, some extra awake slots may be needed for data exchange. As illustrated in Figure 4.7, if a node is involved in sending or receiving connection setup messages or data packets in the current slot, the succeeding slot also becomes an awake slot. The extension continues until the slot is not involved in any message exchange. This mechanism is called a *slot extension*. The number of extended slots is small for low traffic loads and large for high traffic loads. Therefore, the slot extension effectively minimizes the idle periods and the energy consumption.

4.4.5 Compensating Synchronization Errors

In general, wireless devices are equipped with clocks that use hardware oscillators. However, imperfection of clock oscillators and external factors, such as temperature variation, supply voltage variation, and aging cause time inaccuracies, can cause time drifts. Typical error for a quartz crystal oscillator is between 10 and 100 ppm which corresponds to 10 to 100 μs of error per

second. Consider that each node maintains a logical clock C. At time t_1, the time difference of two nodes, called a relative time drift, is $C_{ij}(t_1) = |C_i(t_1) - C_j(t_1)| \geq 0$. $C_{ij}(t_1)$ is equivalent to a constant time shift, and the rotational closure property for a constant time shift δ is proved in Section 4.3. However, if $C_{ij}(t_1)$ varies during a contact, the rotational closure property, which assumes a constant slot length L_s, does not hold. The additional time shift error needs to be compensated. The compensation can be accomplished using the following mechanism. If the maximum time drift error is ε ppm, each awake slot additionally becomes active before and after its original awake slot for $\varepsilon(t-t_1)$ where t_1 is the time of synchronization. Therefore, each frame can compensate for the maximum time drift error and guarantees an overlapping period to be at least L_s.

4.5 Performance Analysis

In this section, the performance of the proposed protocols in comparison with existing adaptive asynchronous sleep scheduling protocols is analyzed. In particular, AQEC [23] and HQS [28] are considered for the direct comparison. Other existing adaptive asynchronous sleep scheduling protocols, CAPM [41], AAPM [25] and ACQ [26], are not considered in the comparison for the following reasons: CAPM shares the similar frame structure with AQEC, and the sleep schedule of CAPM that achieves the lowest active ratio is the same as the sleep schedule of AQEC; AAPM is limited to small set sizes, since the collection of sets that satisfy the rotational closure property is rare for $n > 50$, as shown in Figure 4.8; ACQ is limited to two power saving levels and a-quorums fail to discover each other, as previously explained in Section 4.1. The active ratio (R) and the neighbor sensitivity (NS), as defined in 2.5, are compared in this section.

4.5.1 Comparison of Active Ratio

As a baseline for comparisons, the theoretical optimal active ratio provided by CDS [22] is given as

$$R_{CDS} = \frac{n_{on}}{n_s} = \frac{k}{v} = \frac{q+1}{q^2+q+1} \approx \frac{1}{\sqrt{n_s}} \tag{4.7}$$

where q is a non-negative integer.

AQPS [24] and AQEC [23] are based on QPS, and achieve active ratios approximately twice the optimal active ratio as

$$R_{QPS} = \frac{n_{on}}{n_s} = \frac{2\sqrt{n}-1}{n} = \frac{2\sqrt{n_s}-1}{n_s} \approx \frac{2}{\sqrt{n_s}} \tag{4.8}$$

CAPM [41] uses a frame structure that consists of equally spaced multiple short slots that are L_s long and L_c apart and one large slot that is L_c long. For given L_f and L_s, the active ratio of CAPM is

$$R_{CAPM} = \frac{L_s\left(\frac{L_f}{L_c}-1\right)+L_c}{L_f} \tag{4.9}$$

R_{CAPM} is minimized when $L_c = \sqrt{L_f L_s}$. If we assume that the frame is divided into n slots of an equal length,

$$\min(R_{CAPM}) = \frac{L_{on}}{L_f} = \frac{2L_s\sqrt{n}-1}{L_s n} \tag{4.10}$$

Since $L_f = n \times L_s$ and the length of each row is $L_c = \sqrt{n} \times L_s$ in AQEC, the sleep schedule of CAPM that achieves the lowest active ratio is the same as the sleep schedule of AQEC.

EGHQS and DSHQS presented in HQS [28] construct sets with arbitrary cycle lengths that guarantee rotational closure property between any two sets. In EGHQS, for any set i and set j with lengths less than or equal to the largest set n_{d-1}, $0 \le n_i \le n_j \le n_{d-1}$, the sets are constructed by containing ϕ continuous elements followed by $q_i - 1$ interspaced elements with mutual distances less than or equal to ϕ, where $\phi_i = \min\{\lfloor\sqrt{n_i}\rfloor, \lceil\sqrt{(n_{d-1}+1)/2}\rceil\}$ and $q_i = \lfloor n_i/\phi_i \rfloor$. R_{EGHQS} is calculated for two different cases depending on the value of ϕ_i. CASE 1: If $\phi_i = \lfloor\sqrt{n_i}\rfloor$,

$$\begin{aligned} R_{EGHQS} &= \frac{n_{on}}{n_s} = \frac{\left\lfloor\sqrt{n_i}\right\rfloor + \left\lfloor\frac{n_i}{\lfloor\sqrt{n_i}\rfloor}\right\rfloor - 1}{n_i} \\ &\approx \frac{2\sqrt{n_i}-1}{n_i} \approx \frac{2}{\sqrt{n_i}} = R_{AQEC} \end{aligned} \tag{4.11}$$

CASE 2: If $\phi_i = \lceil\sqrt{(n_{d-1}+1)/2}\rceil$, and let $n_{d-1} = kn_i$ where $k \ge 1$ represents the size ratio

between the set i and the largest set $(d-1)$,

$$\begin{aligned} R_{EGHQS} &= \frac{n_{on}}{n_s} = \frac{\left\lceil \sqrt{\frac{kn_i+1}{2}} \right\rceil + \left\lfloor \frac{n_i}{\left\lceil \sqrt{(kn_i+1)/2} \right\rceil} \right\rfloor - 1}{n_i} \\ &\approx \frac{k+2}{\sqrt{2k}} \frac{\sqrt{n_i}}{n_i} \geq \frac{2}{\sqrt{n_i}} = R_{AQEC} \end{aligned} \tag{4.12}$$

Therefore, the active ratios of EGHQS and AQEC are approximately the same when $\lfloor \sqrt{n_i} \rfloor \leq \lceil \sqrt{(n_{d-1}+1)/2} \rceil$ but larger in EGHQS than in AQEC when otherwise.

In DSHQS, for any set i and set j, $0 \leq n_i \leq n_j \leq n_{d-1}$, the sets are constructed by containing ϕ continuous elements followed by $q_i - 1$ interspaced elements with mutual distances less than or equal to ϕ, where $\phi = \lceil \sqrt{(n_{d-1}+1)/2} \rceil$ and $q_i = \lceil (n_i+1)/2\phi \rceil$. Let $n_{d-1} = kn_i$. Then,

$$\begin{aligned} R_{DSHQS} &= \frac{n_{on}}{n_s} = \frac{\left\lceil \sqrt{\frac{kn_i+1}{2}} \right\rceil + \left\lceil \frac{n_i+1}{2\left\lceil \sqrt{(kn_i+1)/2} \right\rceil} \right\rceil - 1}{n_i} \\ &\approx \frac{k+1}{\sqrt{2k}} \frac{\sqrt{n_i}}{n_i} \geq \frac{1}{\sqrt{n_i}} = R_{CDS} \end{aligned} \tag{4.13}$$

Therefore, the projection algorithm used in HQS achieves active ratio close to the optimal active ratio for small k, but becomes inefficient as n_{d-1} increases. Particularly, $R_{DSHQS} > R_{AQEC}$ for $k > 6$.

ACDS provides multiple power saving levels using hierarchically arranged difference sets. Two proposed protocols, EACDS and MACDS, guarantee at least one overlap of awake slot between any two set within the larger set (n_j). The active ratios of EACDS and MACDS, respectively, are

$$R_{EACDS} = \frac{n_{on}}{n_s} = \frac{k_I (k_E)^p}{v_I (v_E)^p} = \frac{1}{\sqrt{n_s}} \frac{k_I (k_E)^p}{\sqrt{v_I (v_E)^p}} \tag{4.14}$$

$$p = \log_{v_E}(n_s/v_I), n \in \mathbb{Z}^+$$

$$R_{MACDS} = \frac{n_{on}}{n_s} = \frac{k_I k_M}{v_I v_M} = \frac{1}{\sqrt{n_s}} \frac{k_I k_M}{\sqrt{v_I v_M}} \tag{4.15}$$

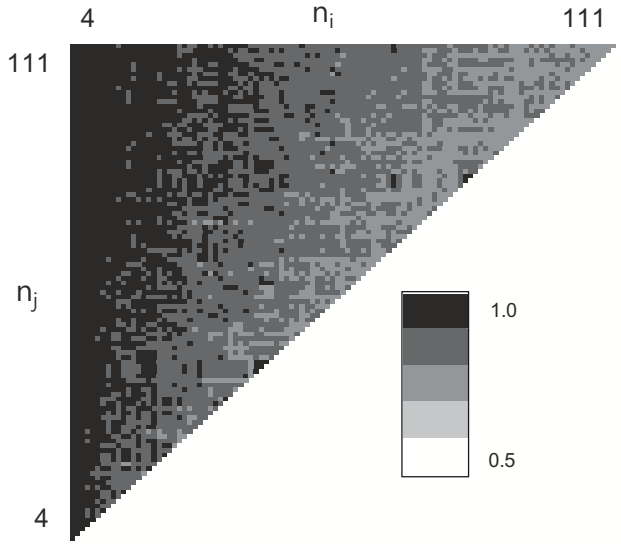


Figure 4.8: RCP between all pairs of relaxed difference sets for $n_i < n_j \leq 111$

where v_I and k_I are respectively v and k of the initial set, v_E and k_E are respectively v and k of the exponential set, and v_M and k_M are respectively v and k of the multiplier set from the rotational set group. The active ratios of EACDS and MACDS are close to the optimal active ratio since a multiplication of parameters from two difference sets still generates a new difference set with a active ratio close to $1/\sqrt{n_i}$.

4.5.2 Comparison of Neighbor Sensitivity

From Table 4.3, NS is different for different protocols with the same n_i. For all protocols, NS between two different sets having n_i and n_j for $n_i \leq n_j$ is about n_j. In other words, two sets of different frame lengths have at least one overlapping awake slot for the duration of the larger frame length. NS is exactly n_j for EACDS and MACDS, $(n_j - \sqrt{n_j} - 1)$ for AQEC, $n_j + \phi - 1$ for EGHQS, and $(\lfloor (n_i - 1)/2 \rfloor + n_j + \phi - 1)$ for DSHQS. As the number of slots increases, the effect of n_j for AQEC decreases exponentially but the effect of n_i for DSHQS increases linearly.

Overall comparison of active ratios with respect to NS is shown in Figure 4.9. For a fair comparison, the largest set is $n_{max} = 1368$. In EACDS, I = $(57, 8, 1)$ and E = $(3, 2, 1)$, and in

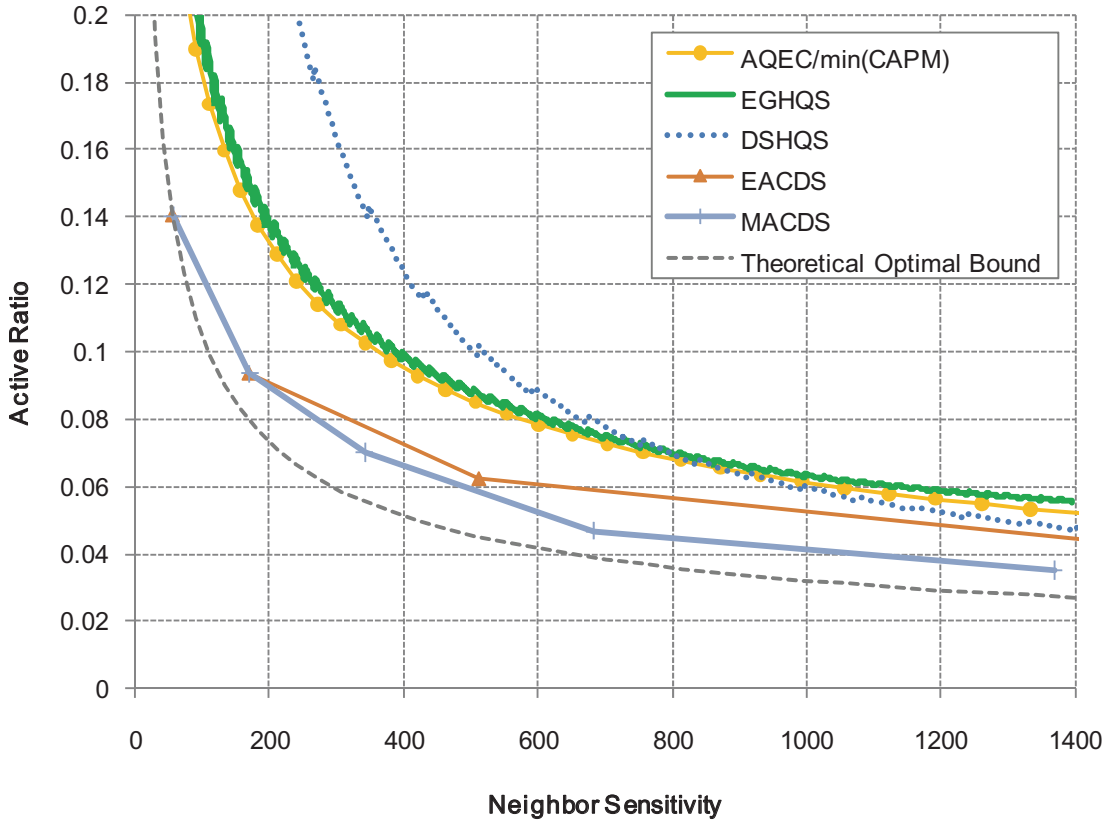


Figure 4.9: Active ratios of different adaptive asynchronous protocols

MACDS, I = $(57, 8, 1)$ and M from the Section 4.3 are used. It can be seen that the active ratios of all protocols decrease as NS increases. The theoretical lower bound achieved by CDS is used as a reference. Note that the bound is only achievable for non-adaptive sets in CDS. The active ratios of AQEC, CAPM, EGHQS, and DSHQS are all approximately twice the optimal bound. The active ratios of EACDS and MACDS are closest to the optimal active ratio. The active ratio of DSHQS is higher than MACDS and even higher in lower n's. The higher active ratio in DSHQS is due to the fact that every set $n_i \leq n_{max}$ requires at least $\phi = \lceil\sqrt{(n_{max}+1)/2}\rceil$ awake slots, which increases as n_{max} increases. In conclusion, although the set sizes of EACDS and MACDS are less fine-grained than other protocols, given required NS, the active ratios of existing protocols are similar while the active ratios of EACDS and MACDS are significantly smaller. Moreover, both EACDS and MACDS become more effective as the initial set size, v_I,

increases.

4.6 Simulation Results

In this section, the performance of the proposed protocols is evaluated by simulations. The Network Simulator (*NS-2*) [53], with additional implementations of the packet forwarding mechanisms in DTNs, the spray and wait routing protocol [54], and the asynchronous sleep scheduling protocol, is used for the simulation. The density of mobile nodes is configured to produce a sparse network with a contact ratio between 0.01 and 0.1 as demonstrated in real life experiments [63, 65, 66]. Flows are created between source nodes and destination nodes that are randomly chosen. Packets are generated at every packet generation interval throughout the simulation. The power consumption model is taken from [30] for 802.11 2 Mbps card. Each slot is at least twice the ATIM window size of 10 ms as explained in Section 4.4. Each node, once connected with another node, remains awake until the data exchange is finished. The summary of parameters used in the simulation are given in Table 4.6. Only the results for MACDS are shown for simplicity.

4.6.1 Impact of Node Speed

The performance of different PSLs using MACDS under various node speeds is given in this section to demonstrate the effect of different nodes speeds on the selection of PSLs. All nodes move at the same speed and packet generation interval is 60 s. The performance is compared with the constant access mode (CAM), where the node is in the idle mode when it is not in transmit or receive mode. The sleep schedule parameters used for PSLs in MACDS are shown on Table 4.7. Beacon period of 1.0 s is used in CAM to match the frame length of P_1 in MACDS which provides the highest contact probability in MACDS.

4.6.1.1 Energy Consumption

As shown in Figure 4.10a, when the node speed increases, the energy consumption increases slightly. Faster moving nodes have more opportunities to contact other nodes, and more packets

Table 4.6: Simulation parameters

Parameter	**Value**
Simulation Time	20000 s
Map Size	5000 m x 5000 m
Movement Model	Random Waypoint
Routing Protocol	Spray and Wait [54]
MAC Protocol	CSMA/CA based
Spray size	2 duplicates
Node Parameters	
Number of Nodes	20
Node Speed	1.0-32.0 m/s
Pause Time	30 s
Radio Range	250 m
Bandwidth	2 Mbps
Flow Parameters	
Number of Flows	20
Beacon Frame Size	40 bytes
Data Packet Size	1000 bytes
Packet Generation Interval	5.0-80.0 s
Message Timeout	10000 s
Power Consumption Model [30]	
Transmit Mode (P_{tx})	1.3272 W
Receive Mode (P_{rx})	0.9670 W
Idle Mode (P_{idle})	0.8437 W
Sleep Mode (P_{sleep})	0.0664 W

can be exchanged between nodes. As expected, nodes using higher PSLs achieve higher energy savings due to lower active ratios. The energy consumption in P_5 is 50 percent of P_1 and 11 percent of CAM, respectively.

To provide a more detailed evaluation of the energy consumption, the breakdown of energy consumption by different power states is given in Figure 4.11. It can be seen that the ratio of

Table 4.7: Sleep scheduling parameters for MACDS

PSL	n_s	n_{on}	M	L_f (s)
P_1	57	8		1.14
P_2	171	16	$(3, 2, 1)$	3.42
P_3	342	24	$(6, 3, 1)$	6.84
P_4	684	32	$(12, 4, 1)$	13.68
P_5	1368	48	$(24, 6, 1)$	27.36

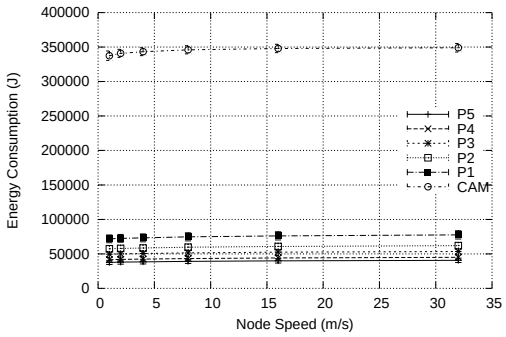
energy consumption in transmit and receive modes are much lower than that in idle and sleep modes. Nodes spend much more time waiting for contacts than actually being connected with other nodes. There is a slight increase of energy consumption in transmit and receive modes at higher node speeds due to the increase in packet transmission opportunities. In addition, the ratio of idle energy consumption in P_5 is lower than that in P_1. The result shows that the major source of energy saving in higher PSLs is due to the reduction of energy consumption in the idle mode.

4.6.1.2 Average Packet Delivery Delay

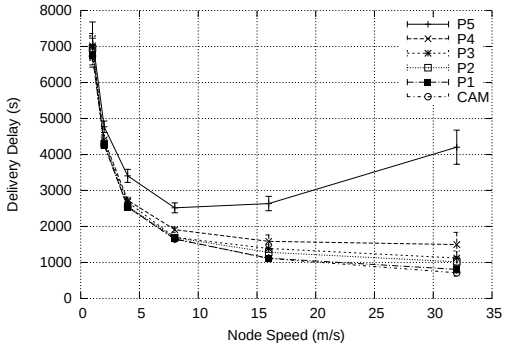
As shown in Figure 4.10b, the average packet delivery delay initially decreases as node speed increases, but increases at high node speeds. Slower moving nodes have longer inter-contact durations, and messages are stored for longer periods. Therefore, for lower node speeds, the delay is constrained by the moving speed of nodes. On the other hand, faster moving nodes,have shorter inter-contact durations, but they also have shorter contact durations available for neighbor discovery. Therefore, for higher node speeds, the effect of contact probability on the delivery delay is greater. Also, we can see that higher PSLs have higher delays and are more sensitive to the increase in node speeds. Longer sleep intervals decrease the contact probability, and consequently, increase the average packet delay.

4.6.1.3 Packet Delivery Ratio

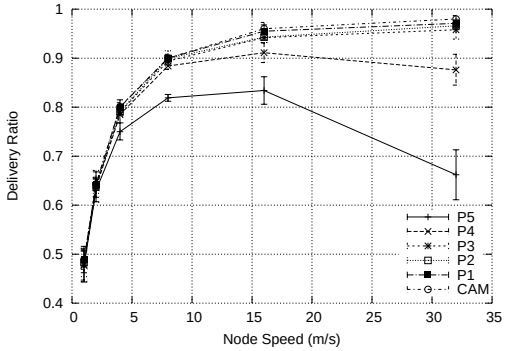
As shown in Figure 4.10c, when the node speed increases, the delivery ratio initially increases, but decreases at high node speeds. At low node speeds, nodes do not have many contact



(a)



(b)



(c)

Figure 4.10: Impact of node speed: (a) Energy consumption; (b) Average packet delay; (c) Packet delivery ratio.

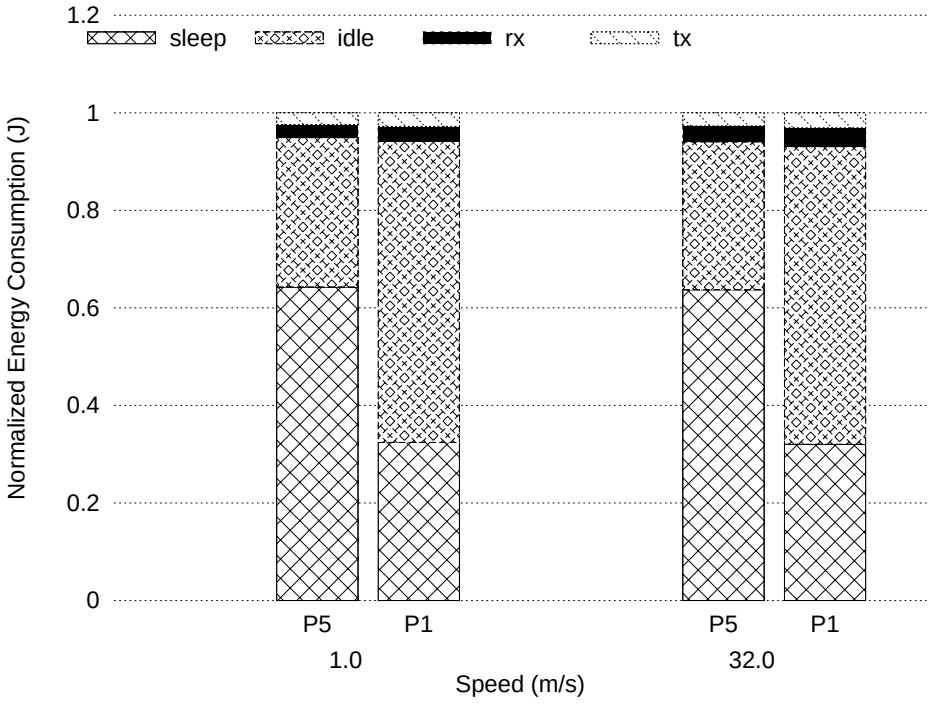


Figure 4.11: Breakdown of energy consumption for different node speeds

opportunities with other nodes in the network due to large inter-contact durations to forward packets to destination nodes within the message timeout or the simulation time. Also large inter-contact durations occur at high node speeds due to low contact probability.

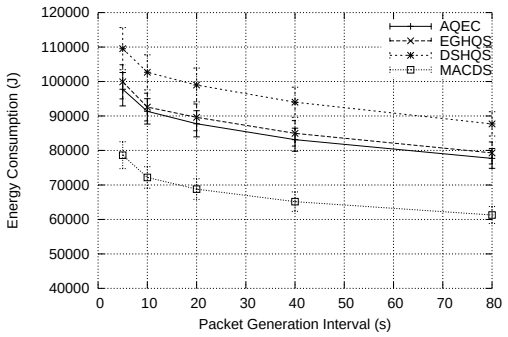
Overall, the simulation results in this subsection demonstrate that a significant amount of energy can be saved by the sleep scheduling protocol and there exist a tradeoff between different PSLs. Slower moving nodes with longer contact durations can use larger frames with lower active ratios to further reduce energy consumption. For example, P_5 consumes about 50 percent of energy consumed in P_1 while maintaining the delay and delivery ratio close to P_1. However, for fast moving nodes, lower PSLs should be used to limit the performance degradation to the required level. Therefore, PSLs should be carefully chosen to optimize the performance as discussed in Section 4.4.

Table 4.8: Parameters for asynchronous sleep scheduling protocols

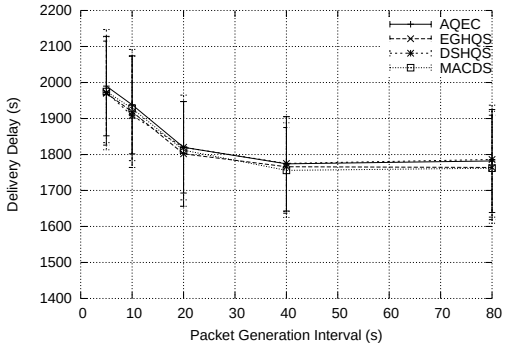
Protocol	Group	NS	n_s, n_{on}	L_f(s)
AQEC	1	91	100, 19	2.0
	2	601	625, 49	12.20
EGHQS	1	100	92, 18	1.84
	2	600	577, 47	11.54
DSHQS	1	100	58, 16	1.16
	2	601	392, 28	7.84
MACDS	1	100	100, 12	2.0
	2	600	600, 36	12.00

4.6.2 Impact of Traffic Load

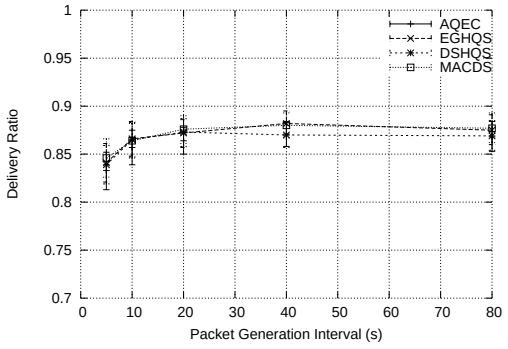
In order to demonstrate the effectiveness of the adaptive sleep scheduling protocols, different groups of nodes with different node speeds are simulated under various traffic loads. The performance of MACDS is compared with AQEC and HQS. Two groups of nodes with different node speeds employ different power saving levels. As demonstrated by the previous simulation results, slower moving nodes can use longer frames to save more energy while maintaining contact ratios similar to that without a power saving mode. Required NS to achieve contact probability above 0.95 for nodes moving at 10.0 m/s and 1.0 m/s are 2.0 s and 12.0 s, respectively. For a fair comparison, a common L_s = 20 ms is used, and sleep scheduling parameters of each protocol are assigned to achieve similar NS. The exact parameters used in the simulation are shown in Table 4.8. Here, n_{d-1} is used for EGHQS and DSHQS are 600 and 400, respectively. Note that NS is not exactly the same for each protocol. n_s of AQEC must be a perfect square, n_s of DSHQS and EGHQS can be any positive integer but some value of NS may not be available for all positive integers, and n_s of EACDS and MACDS are multiples of the initial set size.



(a)



(b)



(c)

Figure 4.12: Impact of traffic load: (a) Energy consumption; (b) Average packet delay; (c) Packet delivery ratio.

4.6.2.1 Energy Consumption

As shown in Figure 4.12a, when the packet generation interval decreases, the overall energy consumption increases. The breakdown of energy consumption in Figure 4.13 shows that at higher traffic loads, the energy consumptions in the transmit mode and the receive mode increase, and the energy consumptions in the sleep mode decrease. The largest energy is consumed in the idle listening mode and the second largest energy is consumed in the sleep mode. MACDS achieves the lowest energy consumption. Compared with the best performing protocol, MACDS consumes 25 percent less total energy and about 35 percent less idle energy.

4.6.2.2 Average Packet Delivery Delay and Packet Delivery Ratio

Packet delay is higher and delivery ratio is lower for higher traffic loads due to 1) limited radio resources and 2) packets generated near the end of simulation time not having sufficient time to forward packets to the destination. The packet delay and the delivery ratio, as shown respectively in Figs. 4.12b and 4.12c, are almost the same in all protocols. This is due to sparsely connected nodes achieving having NS values.

4.6.3 Effect of Power Consumption Model

From the theoretical analysis, the proposed protocols effectively reduce energy waste due to the idle listening problem by achieving low active ratios. However, due to non-zero power consumption in the sleep mode, the contribution of energy consumption by the sleep mode to the overall energy consumption grows significantly as the number of slots increases as shown in Figure 4.14. As the number of slots per frame increases, the ratio of awake slots ($\frac{n+1}{n^2+n+1}$), corresponding to P_{idle}, exponentially decreases, whereas the ratio of asleep slots ($\frac{n^2}{n^2+n+1}$), corresponding to P_{sleep}, remains relatively constant. Therefore, the energy efficiency of the proposed protocols should be higher if a power consumption model having a smaller P_{sleep}/P_{idle} ratio is assumed. For example, if the power consumption model of Berkeley motes [67] with $P_{sleep}/P_{idle} < 0.01$ is used, the amount of energy saving in MACDS compared with the existing protocol further increases from 25 to 30 percent.

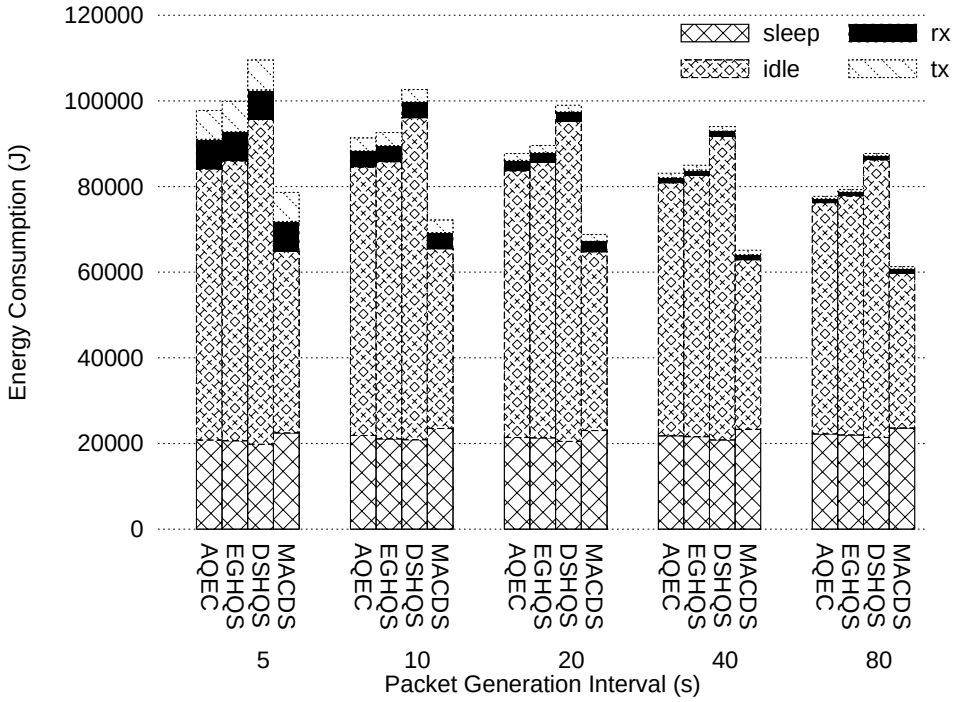


Figure 4.13: Breakdown of energy consumption for different traffic loads

4.7 Summary

In this chapter, energy efficient adaptive asynchronous sleep scheduling protocols, based on hierarchical arrangements of cyclic difference sets, have been proposed for DTNs. Also, implementation issues to maximize energy efficiency in frame structure, neighbor discovery, and message exchange have been discussed. Theoretical analysis of active ratio and neighbor sensitivity has demonstrated the effectiveness of the proposed protocols, especially for large sleep schedules. Simulation results have shown that the proposed protocols, in comparison with other existing protocols, significantly reduce energy waste in the idle listening mode. Furthermore, the advantage of the proposed protocols becomes more evident in scenarios using lower sleep power consumption ratios. In addition, multiple PSLs of the proposed protocols are easily applicable to other application scenarios with different network characteristics, such as traffic loads and transmission range, etc.

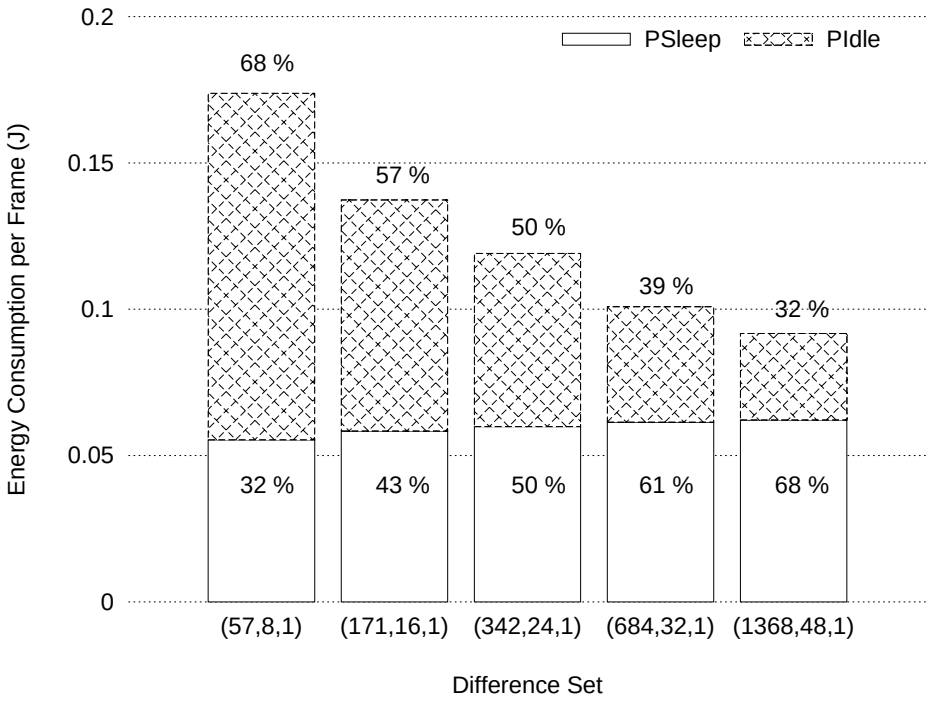


Figure 4.14: Effect of power consumption model ($P_{sleep}/P_{idle} = 0.078$)

Chapter 5

Distributed Semi-Asynchronous Sleep Scheduling Protocol for Delay Tolerant Networks

Synchronous sleep scheduling protocols are only applicable in scenarios where frequent resynchronization to the time servers is possible and practical. Without a perfect clock, nodes need to be awake for additional intervals, called guard intervals, to compensate for the clock synchronization error. Moreover, the length of the guard intervals increases with time and number of hops, and may be comparable to the data exchange time [37, 40, 68]. On the other hand, asynchronous sleep scheduling protocols are robust to any clock synchronization error. However, they may not be energy efficient in scenarios where the clock synchronization error can be bounded. It is possible to achieve and practical to assume that some level of clock synchronization among nodes using existing clock synchronization algorithms [37, 69–71] or by duty cycling of GPS [72, 73]. Since underlying clock synchronization error is ignored in asynchronous sleep schedules protocols, if the consecutively repeated frames are longer than the clock synchronization error bound, awake intervals existing outside the error bound are unnecessary for the neighbor discovery and wasting energy. Although the length of consecutively repeated frames can be adaptively adjusted to meet the connectivity requirement, they are not adaptive to the degree of the clock synchronization error. Although it is difficult to attain highly accurate synchro-

nized clocks, it is possible to achieve some level of synchronization among nodes, using existing clock synchronization algorithms [37, 69–71] or by duty cycling of GPS [72, 73]. Therefore, in this chapter, we propose a semi-asynchronous sleep scheduling protocol, which utilizes the clock synchronization error information and adaptively adjusts to the clock synchronization error. The proposed protocol can bridge the gap between the synchronous and asynchronous sleep scheduling protocols by providing flexibility to synchronous sleep scheduling protocols by allowing loosely synchronized clocks, and achieving higher energy efficiency than asynchronous sleep scheduling protocols by constructing sleep schedules having awake intervals only within the clock synchronization error bound. The contribution and significance of this research work [36] are as follows

- First, we design an energy efficient sleep scheduling protocol under loosely synchronized clocks. The sleep schedules are constructed such that they are distributed and adaptive to time varying clock synchronization error.
- Second, we optimize the proposed protocol using the probability distribution of the clock synchronization error.
- Third, we evaluate the performance of the proposed protocol by theoretical analysis and define conditions under which the proposed protocol outperforms the optimal asynchronous sleep scheduling protocol.
- Lastly, we verify the analysis results by simulations under different network conditions and clock synchronization algorithms.

The remainder of this chapter is organized as follows. We present a comprehensive survey of existing sleep scheduling protocols for different synchronization assumptions in Section 5.1. We propose a sleep scheduling protocol with loosely synchronized clock error in Section 5.2. The performance of the proposed protocol is evaluated using theoretical analysis in Section 5.3 and simulations in Section 5.4. After discussions in section 5.5, the chapter is concluded in Section 5.6. Summary of important symbols used in this chapter is given in Table 5.1 for easy reference.

Table 5.1: Summary of important symbols used

Notation	Description
C_i	local time of node i
L_s	length of a slot
L_f	length of a frame
L_g	length of a guard interval
ε	synchronization error
ε_{max}	synchronization error bound
T_{SI}	length of a search interval
T_{CD}	contact duration
T_{ICD}	inter-contact duration
n_f	$\lceil L_f/L_s \rceil$
n_g	$\lceil L_g/L_s \rceil$
h	$\lceil \varepsilon/L_s \rceil$
$n_{\max}$	$\lceil \varepsilon_{max}/L_s \rceil$

5.1 Related Work

5.1.1 Clock Synchronization Protocol

Network Time Protocol (NTP) [74] has been widely used in the wired Internet to synchronize clocks of computers that can be frequently synchronized with the hierarchically arranged servers. Furthermore, there have been numerous research on clock synchronization in multihop wireless networks. In the reference node based protocols, nodes tune their clocks to the clock information broadcasted from the reference node [69, 71, 75, 76]. Nodes can also achieve a global clock synchronization among the nodes in the network without reference nodes by adjusting their clocks in a distributed manner using the relative clock information [38, 70, 77–80].

5.1.2 Sleep Scheduling Protocol

As shown in Fig. 5.1a, assuming accurate clock information provided by clock synchronization algorithms, synchronous sleep scheduling protocols periodically turn on or off their radio using the process called a *duty cycling* to save energy [15, 18, 19, 81]. However, these protocols

are only effective when the synchronization error can be maintained within several microseconds. The tight bound of error usually requires frequent resynchronization with time servers. Therefore, they are unsuitable in scenarios where the resynchronization is too costly or the time servers are not readily available.

As shown in as shown Fig. 5.1b, asynchronous sleep scheduling protocols can overcome the difficulty in the clock synchronization by designing predetermined cyclic combinations of awake and sleep periods that guarantee overlapping awake periods between different schedules for any time shifts within some time bounds. Asynchronous sleep scheduling protocols are proposed for many different networks: ad hoc networks in [24, 42, 82, 83]; wireless sensor networks in [84, 85]; clustered ad hoc networks in [26]; vehicular ad hoc networks in [27]; and delay tolerant networks in [35, 41]. Asynchronous sleep scheduling protocols are robust to unbounded clock synchronization errors. However, they simply avoid the clock synchronization error by constructing sleep schedules that are consecutively repeated without any sleep periods in between. There also has been some hybrid sleep scheduling protocols for clustered topology where intra-cluster nodes use a synchronous sleep scheduling protocol and inter-cluster nodes use an asynchronous sleep scheduling protocol [86, 87].

5.2 Distributed Semi-Asynchronous Sleep Scheduling Protocol

In this section, we propose a distributed semi-asynchronous sleep scheduling protocol (DSA) that operates under a distributed network topology and requires loosely synchronized clocks. Different from the existing sleep scheduling protocols, semi-asynchronous sleep scheduling protocols allow neighbor discovery among nodes in the network that have different clock synchronization errors. We define conditions that semi-asynchronous sleep scheduling protocols must satisfy and construct a new semi-asynchronous sleep schedule.

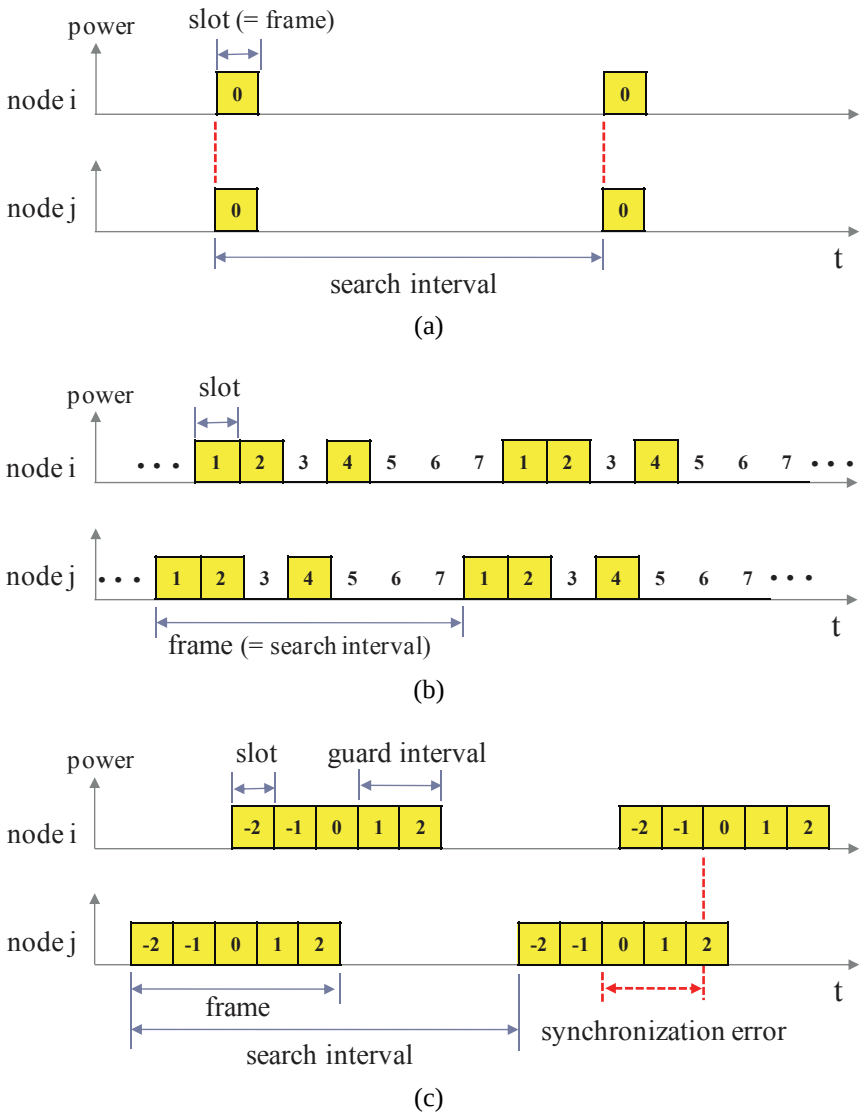


Figure 5.1: Different types of sleep scheduling protocols. a) Synchronous: perfectly aligned frames are periodically repeated; b) Asynchronous: (7,3,1) cyclic difference set based frames are repeated consecutively; c) Semi-asynchronous: frames containing guard intervals due to synchronization error are separated by sleep periods.

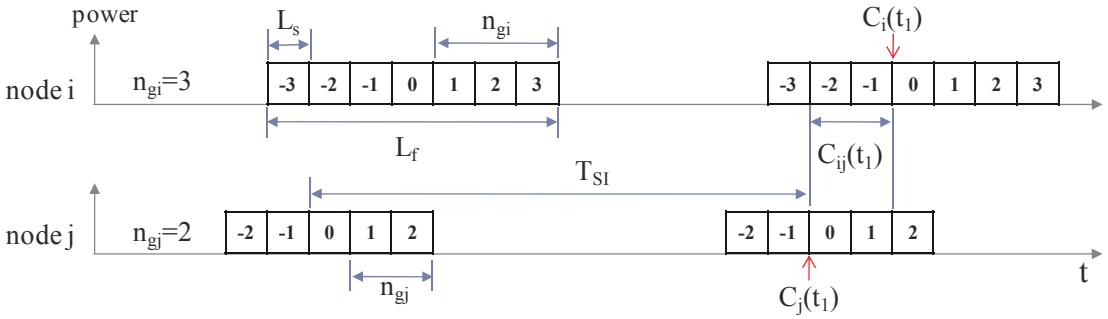


Figure 5.2: Frame structure. Due to clock synchronization error $C_{ij}(t_1)$, the reference slots are not synchronized. In order to guarantee a neighbor discovery, frames of node i (E_i) and node j (E_j) require n_{gi} and n_{gj} guard slots, respectively.

5.2.1 Problem Definition

Mobile nodes form a distributed network topology where a node is not assigned either as a sender or a receiver. They communicate under loosely synchronized clocks where the clock synchronization error (ε) for each node with respect to the perfect clock follows a probability distribution $f(\varepsilon)$. The probability distribution usually follows a normal distribution [68, 69, 88] as

$$\varepsilon \sim \mathcal{N}(0, \sigma^2). \tag{5.1}$$

The upper error bound for the clock error ε_{max} is calculated such that the confidence interval of clock error is above certain probability P_{err} as

$$P_{\text{err}} = Pr(|\varepsilon| \leq \varepsilon_{\text{max}}) = \text{erf}\left(\frac{\varepsilon_{\text{max}}}{\sqrt{2}\sigma_\varepsilon}\right) \tag{5.2}$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ and σ_ε is the standard deviation of clock synchronization error. Therefore, the ε_{max} can be determined for a given σ_ε and the required P_{err}. For example, if $\sigma_\varepsilon = 50$ ms and $P_{\text{err}} = 0.95$, ε_{max} is estimated to be 100 ms.

As shown in Fig. 5.2, we consider a frame of length L_f consisting of n equally sized slots of length L_s. The slots are active or inactive following the predetermined schedule, called a sleep schedule. In each awake slot, nodes are in the awake mode. In each asleep slot, nodes are in the sleep mode. In the case without a synchronization error, each node only needs to use

a reference slot, labeled 0. In the case with synchronization errors, additional awake intervals, called guard intervals, $L_g = \varepsilon$ are needed before and after the reference slot to compensate for the unsigned synchronization error. Therefore, a sleep scheduling frame has a length of $2(n_g + 1)L_s$ where $n_g = \lceil L_g/L_s \rceil$. For the neighbor discovery, a beacon is transmitted at the beginning of slot 0. After the beacon, two nodes can establish a connection by exchanging connection setup messages. Succeeding slots become active for data exchanges. The frame is repeated every search interval (T_{SI}). A longer T_{SI} decreases the energy consumption but increases the probability of missed contacts. Our focus is on designing a sleep schedule for given T_{SI} and the distribution of ε. T_{SI} is configured so that the neighbor discovery probability is above certain value to guarantee the required performance.

Let E_{n_g} represent a set describing the positions of awake slots within a sleep scheduling frame for n_g. Due to the clock synchronization error, the sleep schedule of a node forms a *shift schedule* as defined below.

Definition 6. (Shift Schedule) *Given $n_g \in \mathbb{N}_0$ and a universal set $U = \{-n_g, -n_g+1, \ldots, n_g-1, n_g\}$, let $E_{n_g} \in U$ and $E_{n_g} \neq \emptyset$. We define $S(E_{n_g}, h) = \{-n_g + h, \ldots, n_g + h\}$ a shift schedule of E_{n_g}.*

For example, if $E_2 = \{-2, -1, \ldots, 2\}$, then $S(E_2, -2) = \{-4, -3, \ldots, 0\}$.

A connection can be established between two nodes if their shift schedules satisfy the *shift intersection property* as defined below.

Definition 7. (Shift Intersection Property) *Given $E_{n_{g1}}$ and $E_{n_{g2}}$, the shift intersection property is satisfied if and only if $S(E_{n_{g1}}, h_1) \cap S(E_{n_{g2}}, h_2) \neq \emptyset \; \forall \; h_1, h_2 : h_1 \in \{-n_{g1}, \ldots, n_{g1}\}$ and $h_2 \in \{-n_{g2}, \ldots, n_{g2}\}$.*

In order to describe the shift intersection property, we define a *shift intersection probability* which represents the probability of having at least one overlapping slot between two different sets for any time shifts within the maximum possible time shifts due to the clock synchronization errors from $-(n_{g1} + n_{g2})$ to $(n_{g1} + n_{g2})$ as

$$P_{\text{shift}}\{E_{n_{g1}}, E_{n_{g2}}\} = \frac{1}{2(n_{g1} + n_{g2}) + 1} \sum_{\check{h}=-n_{g1}-n_{g2}}^{n_{g1}+n_{g2}} \beta_h \tag{5.3}$$

where $\check{h} = h_2 - h_1$ is a relative shift between the two sets and β_n is a binary decision variable determined as

$$\beta_{h_r} = \begin{cases} 1 & \text{if } S(E_{n_{g1}}, \check{h}) \cap S(E_{n_{g2}}, 0) \neq \emptyset \\ 0 & \text{if } S(E_{n_{g1}}, \check{h}) \cap S(E_{n_{g2}}, 0) = \emptyset. \end{cases} \tag{5.4}$$

If $P_{\text{shift}} = 1$, then the shift intersection property is satisfied between $E_{n_{g1}}$ and $E_{n_{g2}}$. Otherwise, $E_{n_{g1}}$ and $E_{n_{g2}}$ do not satisfies the shift intersection property. Note that the *rotational closure property* [24] or the *shift invariant property* [42] in asynchronous sleep scheduling protocols guarantee overlapping slots between frames that are repeated back-to-back without any gaps in between as $T_{SI} = L_f$. However, the *shift intersection property* in semi-asynchronous sleep scheduling protocols considers gaps that exist between periodically repeated frames when $T_{SI} > L_f$.

The energy efficiency of a sleep scheduling protocol is measured by an active ratio (R) that represents the proportion of awake periods within a search interval as

$$R = n_{\text{on}} L_s / T_{SI} \tag{5.5}$$

where n_{on} is the number of awake slots. The active ratio of a protocol without a sleep scheduling is 1. Therefore, our goal is to design a sleep schedule E_{n_g} that minimizes R while maximizing P_{shift} among frames having different n_g ranging from 0 to $n_{\text{max}} = \lceil \varepsilon_{\text{max}} / L_s \rceil$.

5.2.2 Construction of the DSA

The simplest solution to the shift intersection property is using a set of schedules having continuous guard intervals. Here, awake intervals are simply extended with respect to the estimated clock synchronization error. We call this sleep schedule a continuous guard interval (CGI). The CGI can be viewed as an extension of synchronous sleep scheduling protocols where additional awake slots are placed around the reference slot. The sleep schedule of the CGI contains only the awake slots as below

$$E_{n_g} = \{-n_g, -n_g + 1, \ldots, n_g - 1, n_g\}. \tag{5.6}$$

The proof of the shift intersection property of the CGI is given in Theorem 3.

Theorem 3. (Shift Intersection Property of the CGI) *Given two CGI sets* $E_{n_{g1}}$ *and* $E_{n_{g2}}$ *where* $0 \leq n_{g1} \leq n_{g2}$, $S(E_{n_{g1}}, h_1) \cap S(E_{n_{g2}}, h_2) \neq \emptyset \; \forall \; h_1, h_2 : h_1 \in \{-n_{g1}, \ldots, n_{g1}\}, h_2 \in \{-n_{g2}, \ldots, n_{g2}\}$.

As shown in the proof, at least the slot 0 of $E_{n_{g1}}$ and that of $E_{n_{g2}}$ are always overlapping. However, except for the maximum relative shift ($\pm(n_{g1} + n_{g2})$), there are more than one overlapping slot. Moreover, when $(n_{g1} - n_{g2}) \leq \check{h} \leq (n_{g2} - n_{g1})$, every slot of $E_{n_{g1}}$ is overlapping with $E_{n_{g2}}$. Since these redundant overlapping slots are unnecessary for the neighbor discovery, if they can be avoided, the energy efficiency can be improved.

The average active ratio for the CGI considering the error distribution is calculated as

$$R_{\text{CGI}} = \sum_{h=-n_{\max}}^{n_{\max}} f(h) \, (2|h| + 1) \, \frac{L_s}{T_{SI}} \tag{5.7}$$

where $f(h)$ is the probability distribution of the clock synchronization error for the given h.

Next, we present the sleep schedule of the DSA which improves the energy efficiency of the CGI by constructing sleep schedules with a smaller number of awake intervals. The sleep scheduling of the DSA is constructed using a minimum guard interval c as below

$$\begin{aligned} E_{n_g,c} = \{&-n_g, -n_g + 1, \ldots, -n_g + c - 2, -n_g + c - 1, \\ &- n_g + 2c - 1, -n_g + 3c - 1, \ldots, -n_g + c \left\lfloor \frac{2n_g + 1}{c} \right\rfloor - 1, n_g\}. \end{aligned} \tag{5.8}$$

In other words, $E_{n_g,c}$ consists of c consecutive awake slots followed by awake slots that are c slots apart and an awake slot on the last slot. Every node uses the same c and adaptively chooses $E_{n_g,c}$ according to the individual clock synchronization error n_g. The beacon in the DSA is sent every c slots. Note that the DSA is same as the traditional approach when $c = 1$. As in CGI, the DSA guarantees the shift intersection property, but with smaller number of awake slots. The proof of the shift intersection property of the DSA is given in Theorem 4.

Theorem 4. (Shift Intersection Property of the DSA) *Given two DSA sets* $E_{n_{g1},c}$ *and* $E_{n_{g2},c}$ *where* $0 \leq n_{g1} \leq n_{g2}$, $S(E_{n_{g1},c}, h_1) \cap S(E_{n_{g2},c}, h_2) \neq \emptyset \; \forall \; h_1, h_2 : h_1 \in \{-n_{g1}, \ldots, n_{g1}\}, h_2 \in \{-n_{g2}, \ldots, n_{g2}\}$.

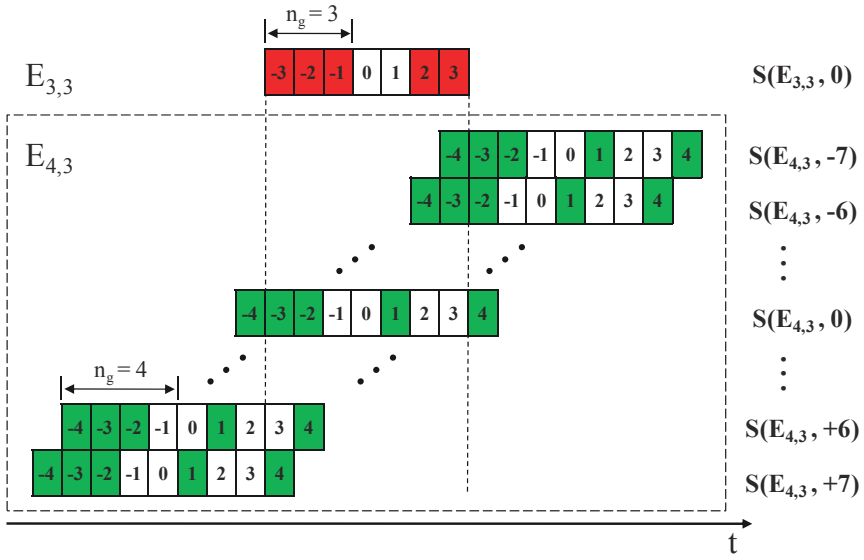


Figure 5.3: Illustration of the shift intersection property between $E_{4,3}$ and $E_{3,3}$. White and colored slots represent asleep and awake time slots, respectively. $E_{4,3}$ and $E_{3,3}$ have at least one overlapping awake slot for all relative shifts from $-3-4$ to $3+4$.

Fig. 5.3 illustrates the shift intersection property of the DSA between two different sets having different clock synchronization errors. The shift intersection property holds between any two different sets having the same c for all $n_g \geq 0$.

The average active ratio of the DSA is calculated as

$$R_{\text{DSA}} = \sum_{h=-n_{\max}}^{n_{\max}} f(h) \left(c + \left\lceil \frac{2|h|+1}{c} \right\rceil - 1 \right) \frac{L_s}{T_{SI}}. \tag{5.9}$$

5.2.3 Protocol Optimization

As shown in Fig. 5.4, there is a tradeoff between different values of c. A smaller c will provide higher energy efficiency for nodes with a small error but lower energy efficiency for nodes with a larger error. Therefore, optimizing the value of c can maximize the energy efficiency for the given $n_{\max}$. The value of c can be optimized given the probability distribution of the clock synchronization error ε. We choose $\varepsilon_{\max}$ in (5.2) so that $P_{\text{err}} \geq 0.95$. The optimum value for c can be calculated by solving the equation of R_{DSA} in (5.9).

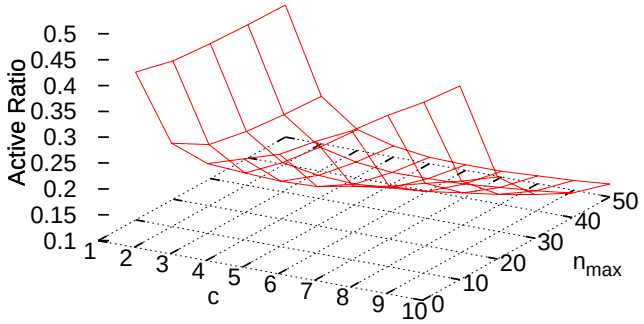
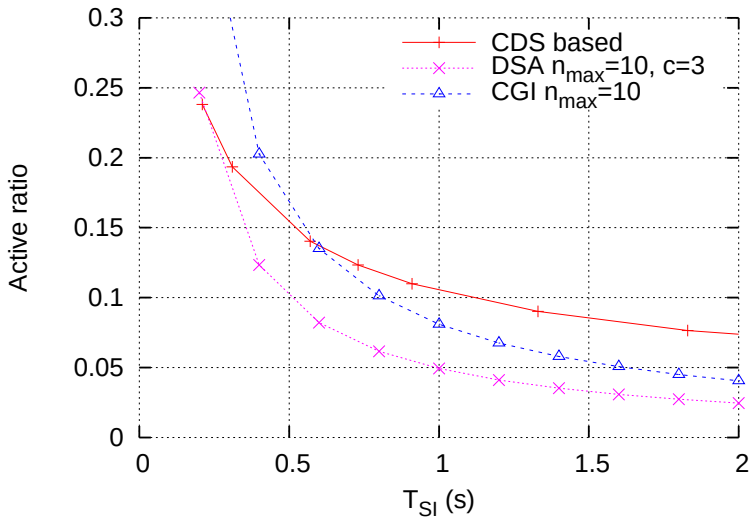


Figure 5.4: Average active ratio for $n_{\max} = T_{SI}$

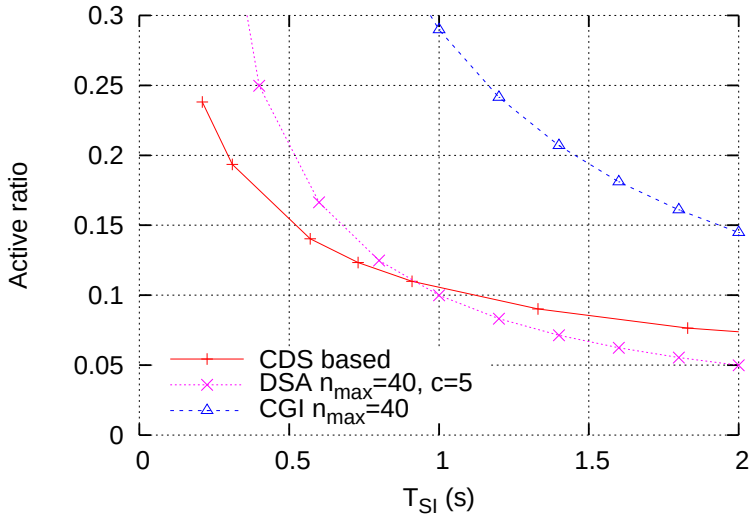
5.3 Performance Analysis

5.3.1 Comparison of Active Ratio

We compare the performance of the DSA with the existing asynchronous protocols in terms of energy efficiency using a theoretical analysis of the active ratio. For the DSA, the value of c that minimizes the active ratio for the given $n_{\max}$ is used. As shown in Fig. 5.5, the DSA achieves lower active ratios than the CGI for all T_{SI} since a smaller number of awake slots are used in the DSA for the construction of sleep schedules. For both the DSA and the CGI, the active ratio is higher for a larger $n_{\max}$ since longer guard intervals are required to compensate for a larger variance of the clock synchronization error. Moreover, the active ratio of DSA is compared with the cyclic difference set based asynchronous sleep scheduling protocol (CDS) [42] which provides the theoretically optimal active ratio of asynchronous sleep scheduling protocols. The active ratio of the CDS can be approximated by $1/\sqrt{T_{SI}}$. The comparison result indicates that the DSA can achieve a lower active ratio for a smaller $n_{\max}$. The reason is that, for a longer T_{SI}, the CDS requires a larger number of awake slots, whereas the DSA uses the same number of awake slots, as long as the clock synchronization error remains the same. The theoretical result can be



(a)



(b)

Figure 5.5: Analytical comparison of average active ratio: (a) $n_{\max} = 10$; (b) $n_{\max} = 40$.

Table 5.2: Simulation parameters

Parameter	**Value**
Simulation Time	86400 s (1 day)
Map Size	5000 m x 5000 m
Movement Model	Random Waypoint
Node Parameters	
Number of Nodes	50
Node Speed	1.0 m/s
Pause Time	0 - 300 s
Radio Transmission Range	100 m
Sleep Scheduling Parameters	
Search Interval (T_{SI})	1.0 s
ATIM Window Length (L_s)	10 ms
Clock Model	
Initial Skew Bound (ρ_i)	Uniform$(-50, +50)$ ppm
Initial Offset Bound ($C_i(t_0)$)	Uniform$(-10^6, +10^6)$ μs
Power Consumption Model [43]	
Transmission Mode (P_{tx})	42 mW
Receive Mode (P_{rx})	36 mW
Idle Mode (P_{idle})	24 mW
Sleep Mode (P_{sleep})	0.02 mW

used to determine the condition under which the DSA outperforms the existing asynchronous sleep scheduling protocols. For example, when $n_{\text{max}} = 40$ and $T_{SI} > 1.0$ s, it is more energy efficient to use the DSA, otherwise, use the CDS.

5.4 Simulation Results

In this section, we evaluate the performance of the DSA with the existing asynchronous protocols in terms of energy efficiency using simulation results. In particular, the DSA is compared with CGI and CDS[1]. We have additionally implemented clock synchronization algorithms and

[1] T_{SI} is smaller in CDS due to the limitation of possible CDS sets. The sleep scheduling parameters used for the CDS are chosen so that the cycle lengths do not exceed the T_{SI} used in other protocols. The exact parameters used

sleep scheduling protocols in the Opportunistic Network Environment (ONE) simulator designed for DTN protocols [89]. The default simulation parameters are outlined in Table 5.2.

5.4.1 Reference Node Based

We consider a scenario where the perfect clock information is only intermittently available. Nodes synchronize clocks with the accurate clock information broadcasted by a reference node. The reference node can obtain accurate clock information from the GPS. All nodes are initially synchronized with the reference node. However, due to imperfect clocks, the clock synchronization errors of non-reference nodes with respect to the perfect clock of the reference node increase at 50 ppm. The clock synchronization is achieved by non-reference nodes when each node contacts with the reference node and tunes its times to the reference node. When two non-reference nodes contact with each other, they compare their last contact times with the reference node and both follow the clock value of the node with more recent contact time with the reference node.

5.4.1.1 Impact of Node Speed

The degree of the clock synchronization error is affected by various network conditions, such as stability of clock oscillator, resynchronization interval, and node density. In this subsection, we investigate the impact of the clock synchronization error by changing the node speed, thus changing the clock information propagation speed in the network. Due to the sparse density of mobile nodes, the information propagation speed largely depends on the mobility of nodes. As expected, Fig. 5.6a shows that the clock synchronization error is smaller in the the network with higher node speeds. The clock offset fluctuates and cannot be perfectly synchronized since the perfect clock information is only intermittently available. Fig. 5.6b shows the comparison of energy consumption. To demonstrate the impact of the loosely synchronized clock on energy consumption, we show the energy consumption during the neighbor discovery process without data exchanges. In comparison with the CGI, the DSA consumes less than quarter of the energy. Although a larger number of beacons are sent in the DSA, the DSA uses less number of awake

in the simulations are $(91, 10, 1)$, $(381, 20, 1)$, $(1407, 38, 1)$, and $(2863, 54, 1)$, which correspond to T_{SI} equal to 0.9, 3.81, 14.07, and 28.63 s, respectively.

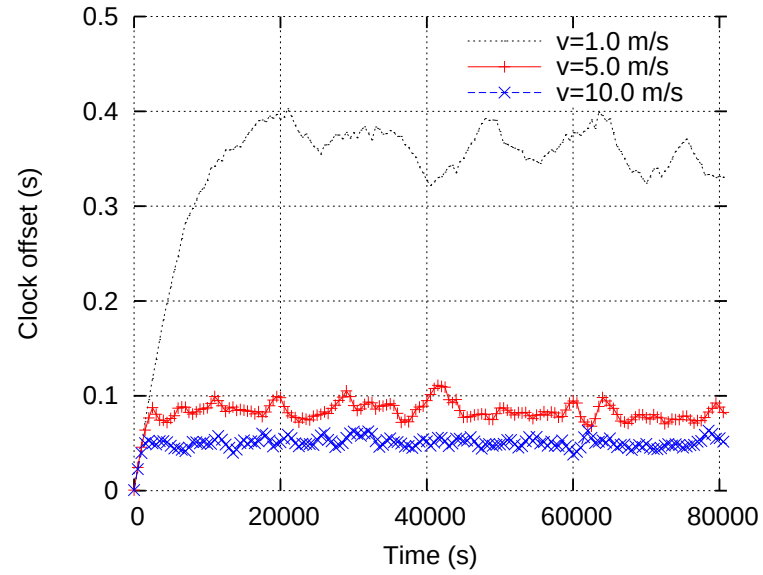
slots, especially for large clock synchronization errors. As a result, the effect of additional beacons is overshadowed by the greater number of awake slots in the CGI. In comparison with the CDS, the DSA consumes more energy when the node speed is 1.0 m/s, but less energy when the node speed is greater than 5.0 m/s. In accordance with the observation in the performance analysis, the CDS is more energy efficient than the DSA when the clock synchronization error is comparable to the T_{SI}. For example, when $T_{SI} = 1.0$ s and the clock synchronization error is within 0.2 s, the DSA is more energy efficient than the CDS.

5.4.1.2 Impact of Search Interval

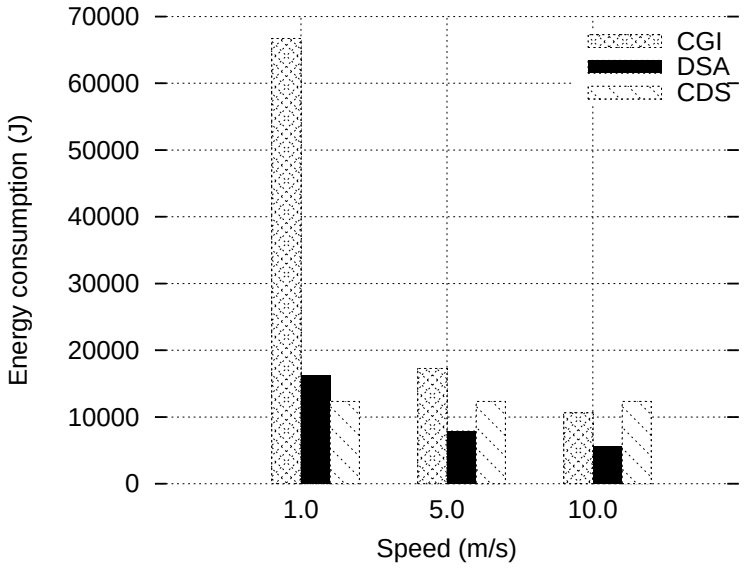
Fig. 5.7a shows the impact of search interval (T_{SI}) on the average clock offset. As shown in Fig. 5.8, due to the insufficient contact durations between mobile nodes, contact ratio decreases as T_{SI} increases. As a result, opportunities for the clock synchronization operation also decrease. Fig. 5.7b shows the comparison of energy consumption. Although, a longer T_{SI} produces a larger clock offset, since the frames are repeated less frequently, the total energy consumption decreases. Nevertheless, since the contact ratio also affects the packet delivery delay and ratio, there exists a tradeoff between the energy consumption and the packet delivery performance depending on T_{SI}. Comparing the energy consumption between protocols, the DSA consumes less energy than the CDS for a large T_{SI}. For the given clock synchronization error, as T_{SI} increases, the frame length to T_{SI} ratio also increases. Since the gaps between the frames consist of asleep periods, the energy efficiency of the DSA increases as the frame length to T_{SI} ratio increases.

5.4.2 Distributed Clock Synchronization Scenario

This scenario represents applications that do not rely on GPS or a reference node for the clock synchronization. Here, reference nodes do not exist and each untethered node synchronize clocks cooperatively by maintaining a table of parameters that relates the local clock of each node to the local clock of every other node in the network. Due to the imperfect clocks, the initial offset and skew are set to $U(-10^6, +10^6)$ μs and $U(-50, +50)$ ppm, respectively. When two nodes contact with each other, their clock values are averaged [70, 90]. For simplicity, we

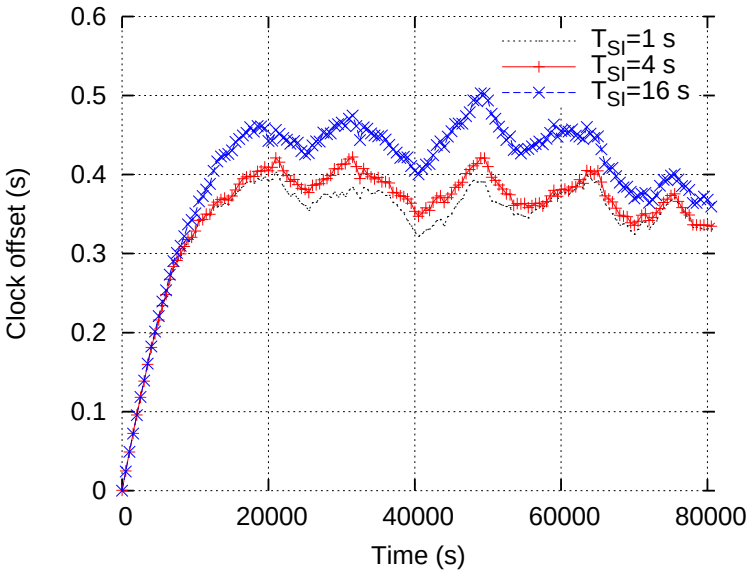


(a)

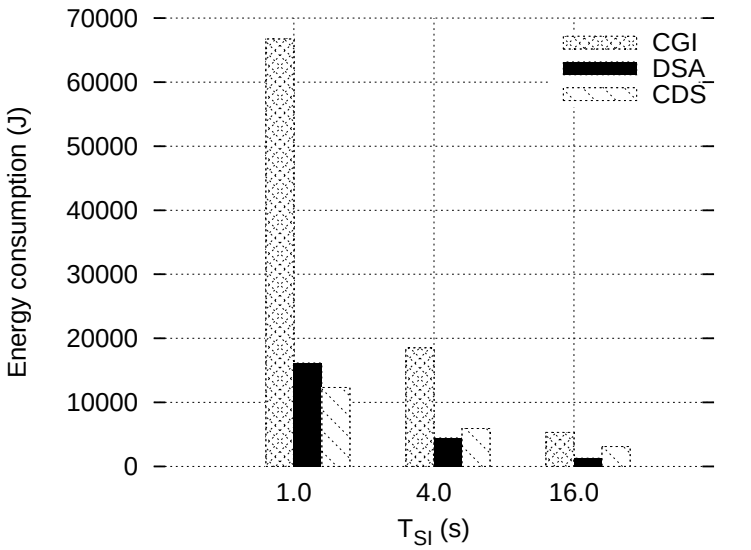


(b)

Figure 5.6: Impact of node speed: (a) Average $C_{ij}(t)$; (b) Energy consumption.



(a)



(b)

Figure 5.7: Impact of search interval: (a) Average $C_{ij}(t)$; (b) Energy consumption.

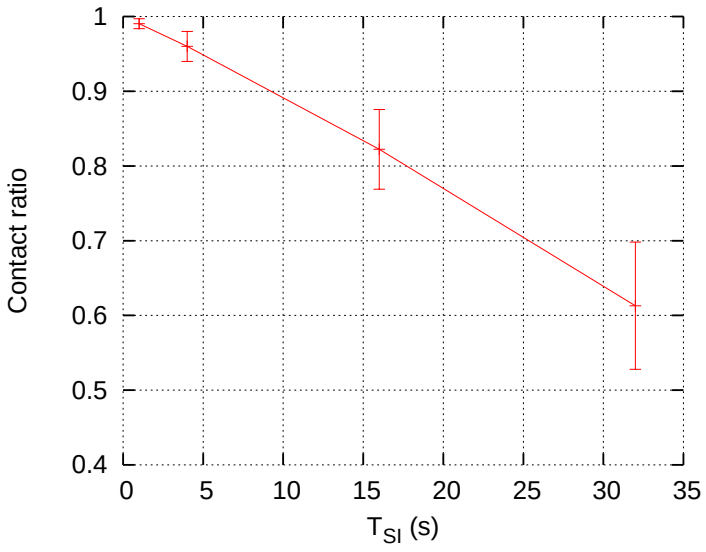


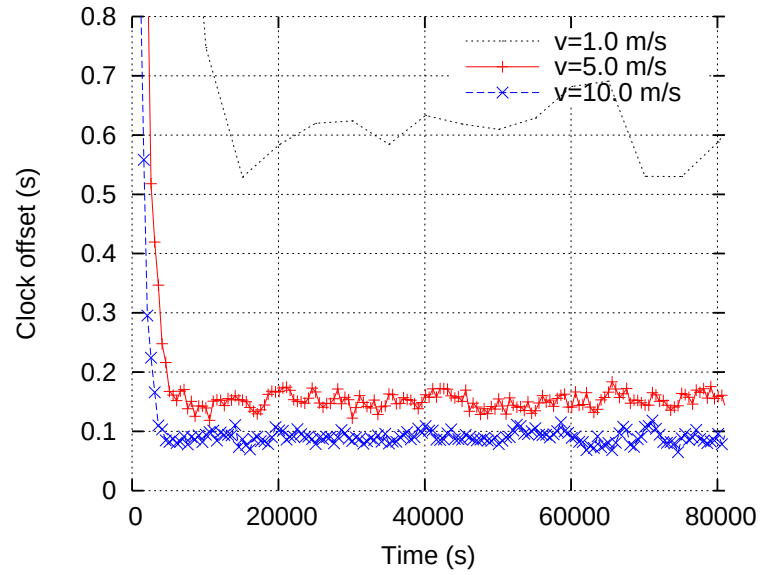
Figure 5.8: Contact ratio for different T_{SI}

assume that time varying relative offset can be known among nodes. Each node chooses ε equal to the maximum relative clock offset.

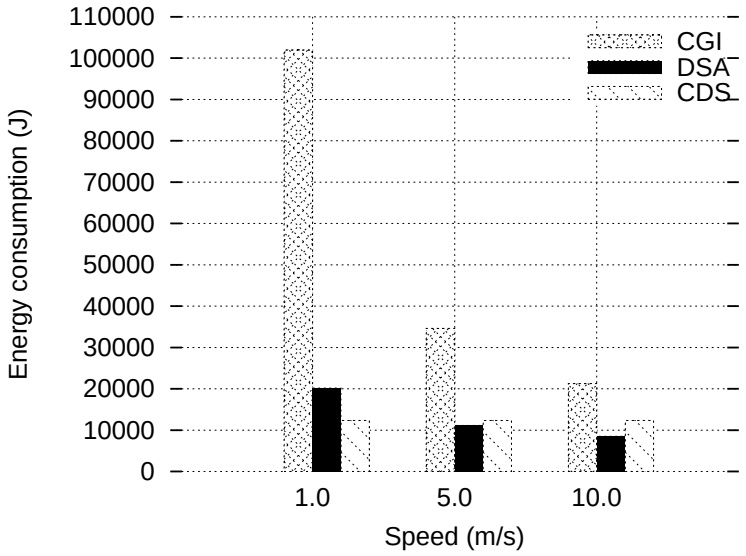
Fig. 5.9a shows the clock offset for different speeds. The clock offset does not converge due to the underlying clock skew. As shown in Fig. 5.10, the variance of the clock synchronization error is smaller for the network with faster moving nodes since they have greater number of contact opportunities. Similar to the previous simulation results, as shown in Fig. 5.9b, the DSA is always more energy efficient than the CGI, and becomes more energy efficient than the CDS as the clock synchronization error decreases.

5.5 Discussion

We observe that, in general, synchronous sleep scheduling protocols and asynchronous sleep scheduling protocols respectively provide the highest energy efficiency for small synchronization errors and large synchronization errors. In addition, we have confirmed using analytical and simulation results that in between the two conditions, where the synchronization error can be



(a)



(b)

Figure 5.9: Impact of node speed: (a) Maximum $C_{ij}(t)$; (b) Energy consumption.

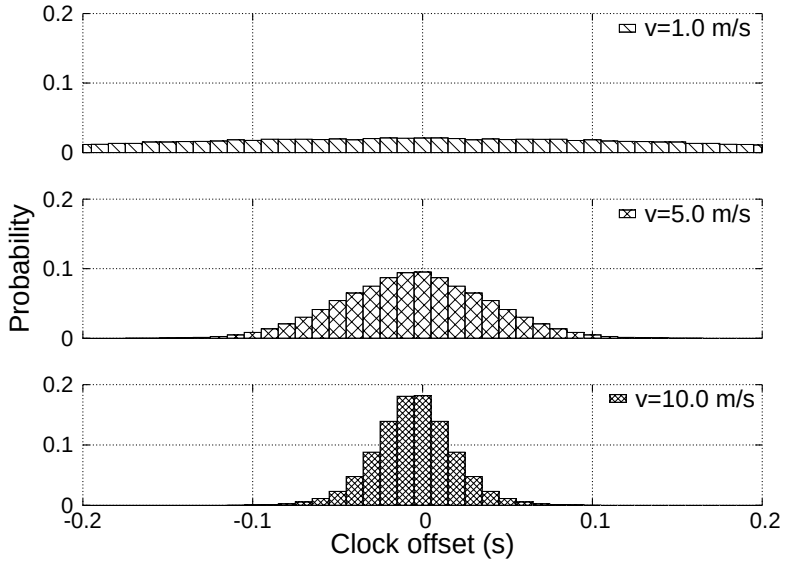


Figure 5.10: Distribution of relative clock synchronization error

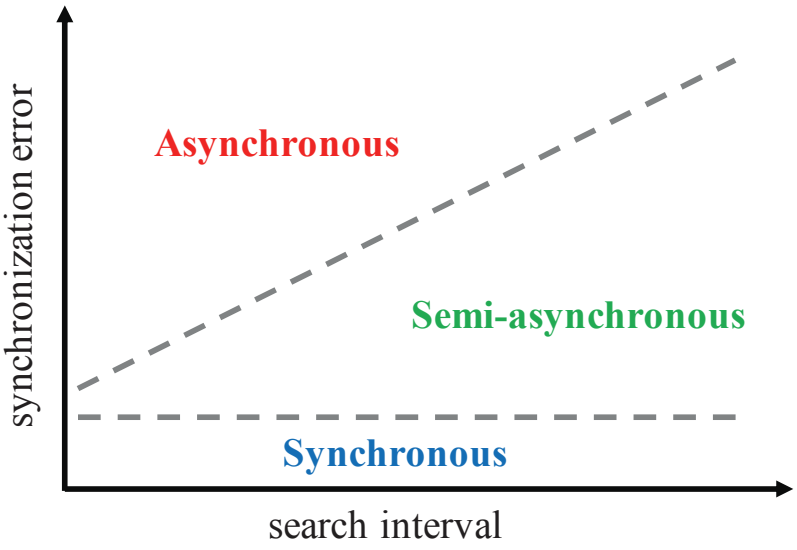


Figure 5.11: Guideline for selecting the type of sleep scheduling protocol

bounded and the search interval (or the duty cycle) is larger than the synchronization error, the DSA provides the highest energy efficiency. Fig. 5.11 shows the guideline for selecting the type of sleep scheduling protocol depending on the synchronization error and the search interval.

5.6 Summary

In this chapter, a distributed semi-asynchronous sleep scheduling protocol (DSA) considering loosely synchronized clocks in sparse mobile wireless multihop networks has been proposed. Individual nodes in the distributed network topology using the DSA can adaptively adjust sleep schedules to the estimated clock synchronization error to minimize energy waste from guard intervals. Given the probability distribution of the clock synchronization errors, the energy efficiency of the DSA can be optimized. The results from theoretical analysis and simulations demonstrate that DSA can significantly reduces energy cost in comparison with existing asynchronous sleep scheduling protocols when clock are loosely synchronized.

Chapter 6

DCS: Distributed Asynchronous Clock Synchronization Protocol in Delay Tolerant Networks

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 [34, 41, 49] 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 [32]. 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. 6.1. In DTNs, an increase in the clock inaccuracy coupled with increases in the number of hops and the inter-contact durations [40]

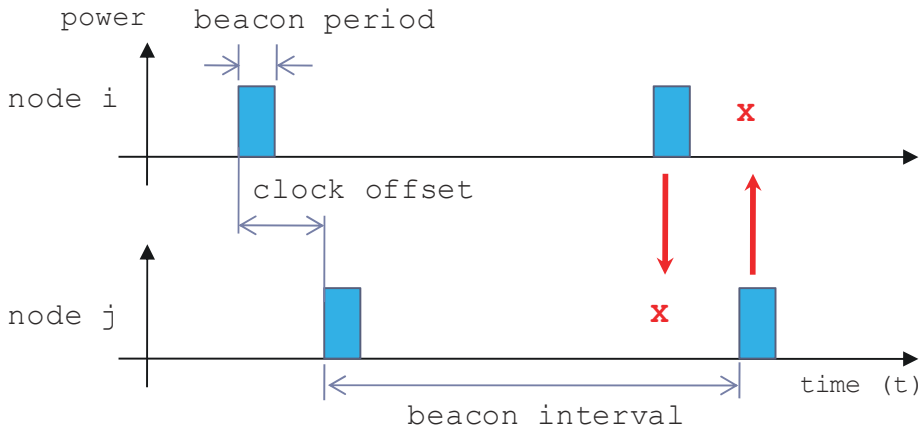
results in a need for large guard periods that cause significant energy consumptions. Therefore, clock synchronization is essential in DTNs for achieving high energy efficiency. In addition, synchronization protocols typically cannot rely on the Global Positioning System (GPS) that requires a large amount 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 contacts and frequent network partitions. Therefore, in this chapter, we focus on clock synchronization protocols in DTNs that are distributed and asynchronous. The contribution and significance of this research work [37] are as follows

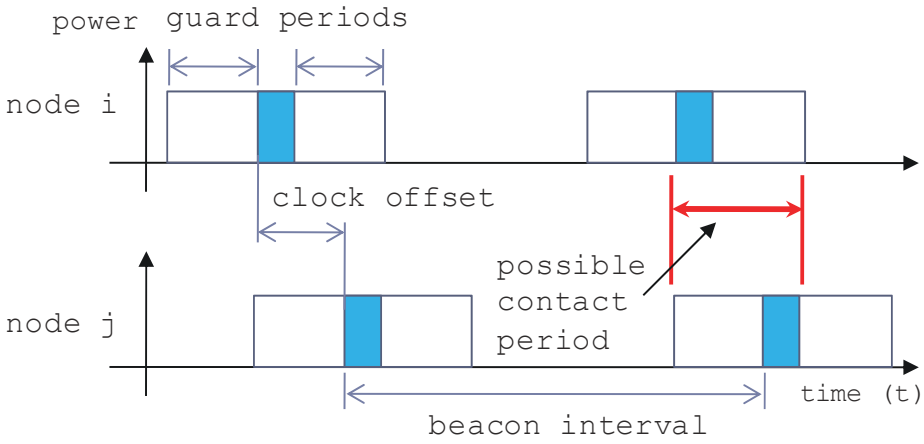
- First, we propose a distributed asynchronous clock synchronization (DCS) protocol for DTNs. Different from existing clock synchronization protocols, the proposed DCS protocol can achieve global clock synchronization among mobile nodes within the network over asynchronous and intermittent connections with long delays. The protocol is fully distributed, so that all nodes independently execute exactly the same algorithm without the need of reference nodes.

- Second, we propose a discrete time analysis to evaluate the performance of the DCS protocol. Considering exponentially distributed inter-contact rates, the proposed analytical model can be implemented as an efficient tool to facilitate the system performance estimation.

- Third, both mathematical analysis and simulation results for various network scenarios are given to demonstrate the convergence and performance of the DCS protocol. The DCS



(a)



(b)

Figure 6.1: 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.

Table 6.1: 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

protocol can achieve faster clock convergence speed and, as a result, reduce energy cost due to neighbor discovery by half.

The remainder of this chapter is organized as follows. Section 6.1 provides an overview of the related work. We present the proposed distributed asynchronous clock synchronization protocol for DTNs in Section 6.2. The performance analysis in presented in Section 6.3. The performance evaluation using numerical and and simulation results is presented in Section 6.4. The conclusions of this chapter is given in Section 6.5. Summary of important symbols used in this chapter is given in Table 6.1 for easy reference.

6.1 Related Work

6.1.1 Network Time Protocol

The Network Time Protocol (NTP) [74] is 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.

6.1.2 Clock Synchronization in Multihop Wireless Networks

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 [75, 76, 88, 91], *gateways* in cluster based protocols [69], or *time servers* in NTP based protocols [71, 74]. 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 [38, 78], averaging clock values of local nodes [70, 79, 80], or gradually decreasing the clock error with neighboring nodes using an exponential weighted moving average [92] or a proportional controller [77]. However, independent of whether or not these protocols assume deployments of static or mobile nodes, they all require strongly connected network topology with a high node degree.

6.1.3 Clock Synchronization in Delay Tolerant Networks

In terms of DTNs, there have been some efforts for clock synchronization. The Timestamp Transformation Protocol (TTP) [40] solves the temporal ordering problem and real-time issues, called *sensor fusion* [93, 94], 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) [71] 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 [70, 90] provides distributed clock synchronization by asynchronously averaging clock values with the contacted neighbors.

However, the AD is inefficient in DTNs where the connection is dynamic and often limited to just few neighbors. There also have been some efforts to provide clock synchronization in underwater acoustic networks (UANs) [95–99] using acoustic modems characterized by long propagation delays. Our focus in this work 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.

6.2 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.

6.2.1 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 table is identified by a unique identifier with the 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 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 $C_{il}^T(t) \neq C_{il}(t)$ and $f_{il}^T(t) \neq f_{il}(t)$.

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.

6.2.2 Exchanging Clock Table Information

In the absence of a reference clock, although the actual clock frequencies $f_i(t_k^{i,j})$ and $f_j(t_k^{i,j})$ are impossible to obtain, a relative clock skew ($f_{ij}(t_k^{i,j})$) can be estimated. When node i contacts node j at time $t_k^{i,j}$, i.e, $(i,j) \in E(t_k^{i,j})$, they exchange series time-stamped triples $(C_i(\underline{t_m^i}), C_j(t_m^j), C_i(\overline{t_m^i}))$ for $m = 1, 2, \ldots$, as illustrated in Fig. 6.2. Here, $C_i(\underline{t_m^i})$ is the local time of node i when mth message is sent, $C_j(t_m^j)$ is the local time of node j when the mth message is received, and $C_i(\overline{t_m^i})$ 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. 6.2b, $C_{ij}(t_k^{i,j})$ and $f_{ij}(t_k^{i,j})$ can be estimated by the following linear equation

$$C_i(t_m^i) = f_{ij}(t_k^{i,j}) C_j(t_m^j) - C_{ij}(t_k^{i,j}) \tag{6.1}$$

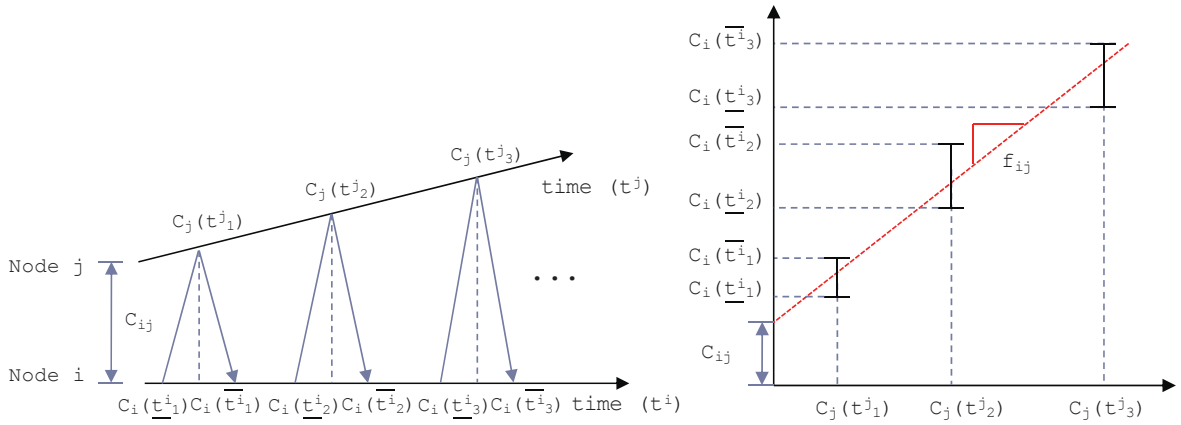


Figure 6.2: Relative clock estimation: a) Exchanging time-stamps between node i and node j; b) Plotting time-stamped triples for clock skew estimation.

representing a line that passes through the bounded errors $C_i(\underline{t^i_m})$ and $C_i(\overline{t^i_m})$ [100]. More time-stamps 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 $C^T_{ij}(t^{i,j}_k) \leftarrow C_{ij}(t^{i,j}_k)$ and $f^T_{ij}(t^{i,j}_k) \leftarrow f_{ij}(t^{i,j}_k)$ for node i. Also, the associated weight values are reset as $w^T_{ij}(t^{i,j}_k) = 1$. 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., $w^T_{il}(t^{i,j}_k) \leq w^T_{jl}(t^{i,j}_k)$. For node i, the updated table information after the exchange is

$$C^T_{il}(t^{i,j}_k) \leftarrow \begin{cases} C^T_{il}(t^{i,j}_k), & \text{if } w^T_{il}(t^{i,j}_k) \geq w^T_{jl}(t^{i,j}_k) \\ C_{ij}(t^{i,j}_k) + C^T_{jl}(t^{i,j}_k), & \text{otherwise} \end{cases} \tag{6.2}$$

$$f^T_{il}(t^{i,j}_k) \leftarrow \begin{cases} f^T_{il}(t^{i,j}_k), & \text{if } w^T_{il}(t^{i,j}_k) \geq w^T_{jl}(t^{i,j}_k) \\ f_{ij}(t^{i,j}_k) + f^T_{jl}(t^{i,j}_k), & \text{otherwise} \end{cases} \tag{6.3}$$

$$w_{il}^T(t_k^{i,j}) \leftarrow \begin{cases} w_{il}^T(t_k^{i,j}), & \text{if } w_{il}^T(t_k^{i,j}) \geq w_{jl}^T(t_k^{i,j}) \\ w_{jl}^T(t_k^{i,j}), & \text{otherwise.} \end{cases} \tag{6.4}$$

6.2.3 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

$$C_i(t_k^{i,j}) \leftarrow C_i(t_k^{i,j}) + \frac{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) C_{il}^T(t_k^{i,j})}{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})} \tag{6.5}$$

$$f_i(t_k^{i,j}) \leftarrow f_i(t_k^{i,j}) + \frac{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) f_{il}^T(t_k^{i,j})}{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})} \tag{6.6}$$

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

$$C_{il}^T(t_k^{i,j}) \leftarrow C_{il}^T(t_k^{i,j}) - \frac{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) C_{il}^T(t_k^{i,j})}{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})} \tag{6.7}$$

$$f_{il}^T(t_k^{i,j}) \leftarrow f_{il}^T(t_k^{i,j}) - \frac{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j}) f_{il}^T(t_k^{i,j})}{\sum_{l \in N_i^T(t_k^{i,j})} w_{il}^T(t_k^{i,j})}. \tag{6.8}$$

Note that, since w_{ii}^T represents the level of accuracy of its own clock information, w_{ii}^T is always one. Therefore, (6.5) 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

$$w_{il}^{T}(t + \Delta t) = \begin{cases} 1, & \text{if } l = i \\ w_{il}^{T}(t)\lambda^{\Delta t}, & \text{otherwise} \end{cases} \tag{6.9}$$

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 [77]. This solution can be considered in implementing the DCS protocol to solve the backward clock movement problem.

6.2.4 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, $f_i(t_k)$.

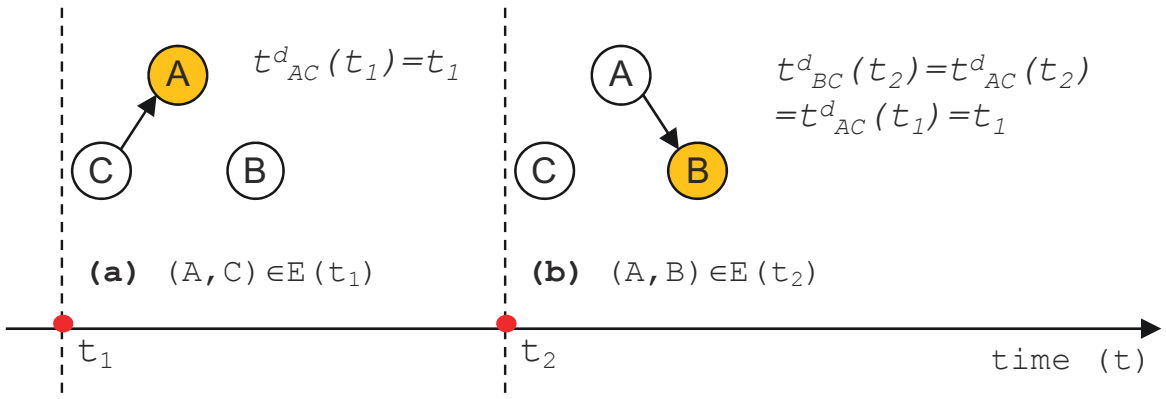


Figure 6.3: An illustrative example of delayed information: (a) At time t_1, nodes A and C contact with each other, and node A obtains the clock frequency value of node C ($f_C(t_1)$) directly from node C; (b) At time t_2, nodes A and B contact with each other, and the clock frequency value of node C ($f_C(t^d_{BC}(t_2))$) is forwarded to node B via node A.

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

$$\lim_{t_k \to \infty} \max_{i,j \in V} |f_{ij}(t_k)| = 0. \tag{6.10}$$

or equivalently, for some constant value f_s, if

$$\lim_{t_k \to \infty} \max_{j \in V} f_j(t_k) = \lim_{t_k \to \infty} \min_{i \in V} f_i(t_k) = f_s. \tag{6.11}$$

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 ($f^T_{il}(t_k)$) and the outdated actual frequencies ($f_l(t^d_{il}(t_k))$), are the same, where $t^d_{il}(t_k) \geq 0$ 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. 6.3.*

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 [101], where the updated frequencies after the kth update can be calculated as

$$f_i(t_k) = \sum_{l=1}^{N} a_{il}(t_k) f_l(t^d_{il}(t_k)), \forall\, i \in V. \tag{6.12}$$

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

$$a_{il}(t_k) = \begin{cases} w_{il}^T(t_k)/(\sum_{n=1}^{N} w_{il}^T(t_k)), & \text{if } \exists\, j \neq i, (i,j) \in E(t_k) \\ I(l = i), & \text{otherwise} \end{cases} \tag{6.13}$$

where $w_{il}^T(t_k)$ is the decaying weight defined in (6.9) and $I(\cdot)$ is an indication function which equals 1 if true and 0 otherwise. In (6.13), 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 (6.12), we have $t_{ii}^d(t_k) = t_k$ since each node can acquire the clock information of itself without delay.

Theorem 5. (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.*

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 $t_{ij}^d(t_k) = t_{ji}^d(t_k) = t_k$ and $a_{il}(t_k)$ given by

$$a_{il}(t_k) = \begin{cases} 1/2, & \text{if } \exists\, j \neq i, (i,j) \in E(t_k) \text{ and } l = i, j \\ I(l = i), & \text{otherwise.} \end{cases} \tag{6.14}$$

Different convergence proofs for the AD protocol can be found in [70, 90, 102, 103].

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 (6.9) 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 6.4.1 and Subsection 6.4.2, respectively.

6.3 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 6.2.4, 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) [104–107], and real mobility traces [108, 109]. The performance analysis of the DCS protocol consists of two parts. In Subsection 6.3.1, the table updating procedure is modeled. In Subsection 6.3.2, 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.

6.3.1 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

$$X_{ii}(\tau_k) = C_i(\tau_k), \qquad Y_{ii}(\tau_k) = f_i(\tau_k) \tag{6.15}$$

$$X_{il}(\tau_k) = C_{il}^T(\tau_k) + C_i(\tau_k),\ Y_{il}(\tau_k) = f_{il}^T(\tau_k) + f_i(\tau_k),\ i \neq l. \tag{6.16}$$

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.

6.3.1.1 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

$$\tilde{Y}_{il}(\tau_{k+1}) = \frac{\sum_{h=1}^{N} w^T_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_{k+1}) Y_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_k)}{\sum_{h=1}^{N} w^T_{(n^*_{ilh}(\tau_{k+1}))h}(\tau_{k+1})} \tag{6.17}$$

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

$$w^T_{nh}(\tau_{k+1}) = \begin{cases} 1, & \text{if } h = i \text{ or } l \\ \lambda^{T_{nh}(\tau_{k+1})}, & \text{otherwise} \end{cases} \tag{6.18}$$

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

$$T_{nh}(\tau_{k+1}) = \begin{cases} \left| \frac{[X_{hh}(\tau_k) + \tau Y_{hh}(\tau_k)] - [X_{nh}(\tau_k) + \tau Y_{nn}(\tau_k)]}{Y_{hh}(\tau_k) - Y_{nn}(\tau_k)} \right|, & \text{if } Y_{hh}(\tau_k) \neq Y_{nn}(\tau_k) \\ \infty, & \text{otherwise.} \end{cases} \tag{6.19}$$

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

$$n^*_{ilh}(\tau_{k+1}) = \arg \min_{n \in \{i,l\}} |[X_{nh}(\tau_k) + \tau Y_{nn}(\tau_k)] - [X_{hh}(\tau_{k+1}) + \tau Y_{hh}(\tau_k)]| \tag{6.20}$$

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

$$\tilde{Y}_{ijl}(\tau_{k+1}) \quad = \quad Y_{(n^*_{ijl}(\tau_{k+1}))l}(\tau_k). \tag{6.21}$$

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

$$\begin{aligned} Y_{il}(\tau_{k+1}) \quad &= \quad \frac{1-P_0}{(N-1)}\tilde{Y}_{il}(\tau_{k+1}) + \sum_{\substack{j=1 \\ j\neq i,l}}^{N} \frac{1-P_0}{(N-1)}\tilde{Y}_{ijl}(\tau_{k+1}) \\ &+ \quad P_0 Y_{il}(\tau_k), i \neq l \end{aligned} \tag{6.22}$$

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.

6.3.1.2 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

$$\tilde{X}_{il}(\tau_{k+1}) = \frac{\sum_{h=1}^{N} {w^T}_{(n^*_{ilh}(\tau_{k+1}))^h}(\tau_{k+1})[X_{(n^*_{ilh}(\tau_{k+1}))^h}(\tau_k) + \tau Y_{(n^*_{ilh}(\tau_{k+1}))(n^*_{ilh}(\tau_{k+1}))}(\tau_k)]}{\sum_{h=1}^{N} {w^T}_{(n^*_{ilh}(\tau_{k+1}))^h}(\tau_{k+1})} \tag{6.23}$$

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

$$\tilde{X}_{ijl}(\tau_{k+1}) = X_{(n^*_{ijl}(\tau_{k+1}))l}(\tau_k) + \tau Y_{(n^*_{ijl}(\tau_{k+1}))(n^*_{ijl}(\tau_{k+1}))}(\tau_k). \quad (6.24)$$

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

$$\begin{aligned} X_{il}(\tau_{k+1}) &= \frac{1-P_0}{(N-1)}\tilde{X}_{il}(\tau_{k+1}) + \sum_{\substack{j=1 \\ j\neq i,l}}^{N} \frac{1-P_0}{(N-1)}\tilde{X}_{ijl}(\tau_{k+1}) \\ &+ P_0\left[X_{il}(\tau_k) + \tau Y_{ii}(\tau_k)\right], i \neq l. \end{aligned} \quad (6.25)$$

6.3.2 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

$$X_{ii}(\tau_{k+1}) = \sum_{\substack{l=1 \\ l\neq i}}^{N} \frac{1-P_0}{(N-1)}\tilde{X}_{il}(\tau_{k+1}) + P_0\left[X_{ii}(\tau_k) + \tau Y_{ii}(\tau_k)\right] \quad (6.26)$$

$$Y_{ii}(\tau_{k+1}) = \sum_{\substack{l=1 \\ l\neq i}}^{N} \frac{1-P_0}{(N-1)}\tilde{Y}_{il}(\tau_{k+1}) + P_0 Y_{ii}(\tau_k) \quad (6.27)$$

where $\tilde{X}_{il}(\tau_{k+1})$ and $\tilde{Y}_{il}(\tau_{k+1})$ are given by (6.23) and (6.17), respectively.

Define the system state of the DCS protocol at time τ_k as

$$\mathbf{S}_{DCS}(\tau_k) = \{X_{il}(\tau_k), Y_{il}(\tau_k) | i, l = 1, \cdots, N\}. \quad (6.28)$$

For the updating procedure in (6.22), and (6.25)-(6.27), 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

$$\mathbf{S}_{DCS}(k) = F^k_{DCS}(\mathbf{S}_{DCS}(0)) \tag{6.29}$$

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 A

6.4 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 presented in Section 6.3, 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 [89] 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 6.2.

Table 6.2: Default system parameters

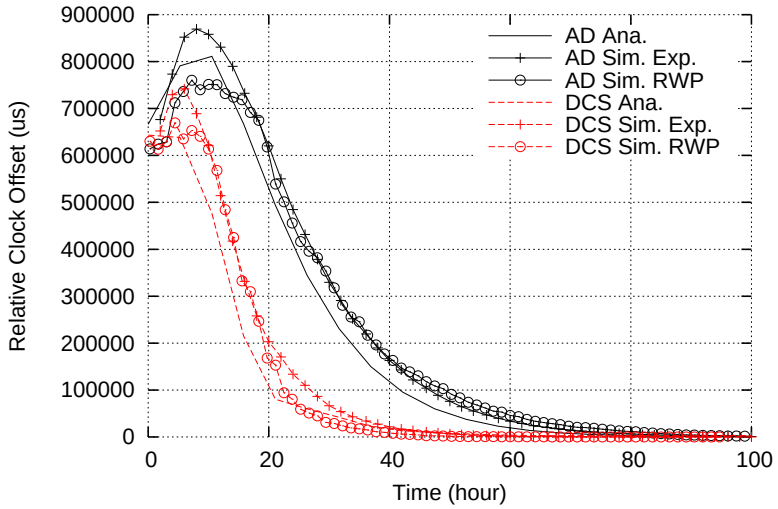
Parameter	**Value**
Simulation Time	550 hours
Map Size ($M \times M$)	50 km $\times$ 50 km, 20 km $\times$ 20 km
Number of Nodes (N)	50
Node Speed (v)	Uniform$(0.5, 1.5)$ m/s
Pause Time	0 - 120 s
Radio Transmission Range (R)	250 m
Initial Skew ($C_i'(t_0)$)	Uniform$(-100, +100)$ ppm
Initial Offset ($C_i(t_0)$)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

6.4.1 Numerical Results

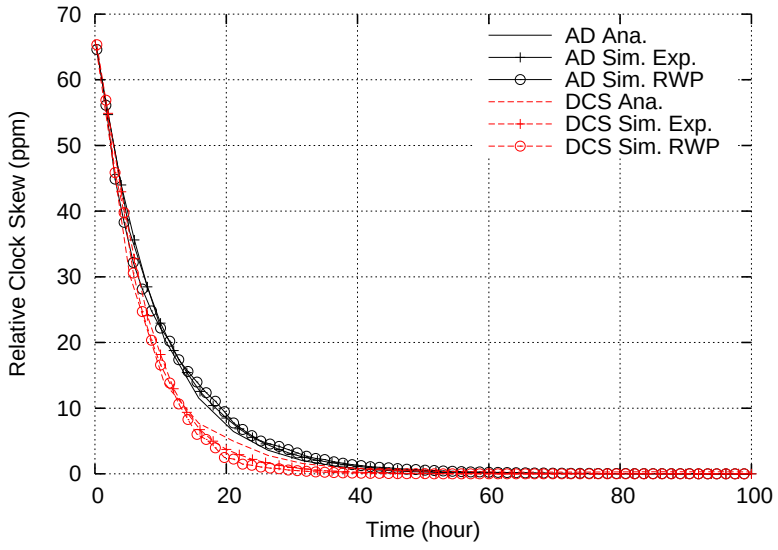
For the RWP mobility model under consideration, the pairwise inter-contact duration is approximately exponentially distributed [104, 105]. Both theoretical and experimental results can be used to estimate the pairwise inter-contact rate γ. In this work, γ is estimated as the reciprocal of the average 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 [110] 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 nodes are updated simultaneously for each time period τ in (6.26) and (6.27), 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.

6.4.1.1 Impact of Node Density

Figs. 6.4 and 6.5 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

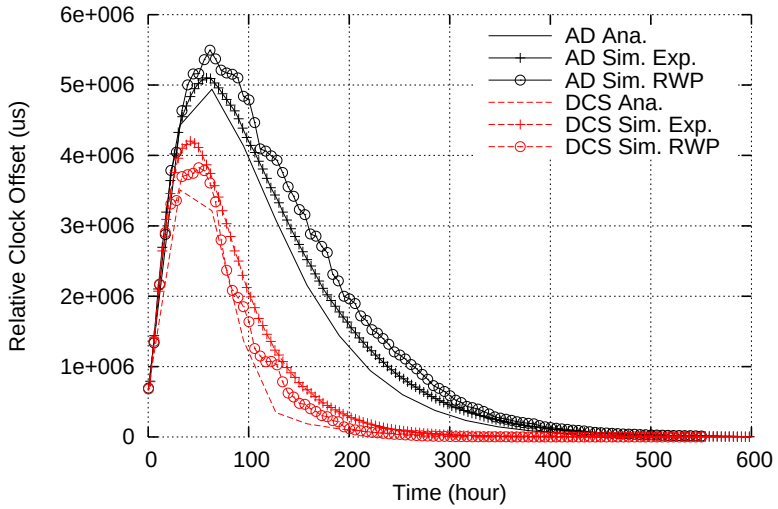


(a)

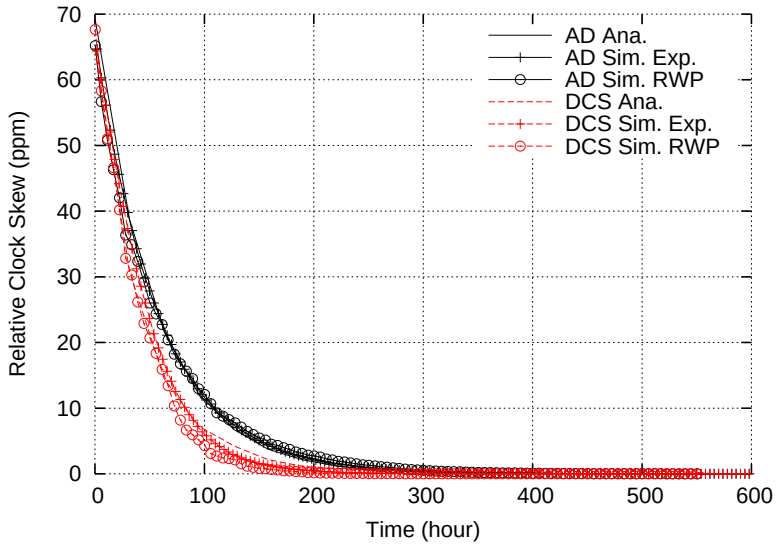


(b)

Figure 6.4: Convergence of clock offset and skew ($M = 20$ km): (a) Average relative clock offset; (b) Average relative clock skew.

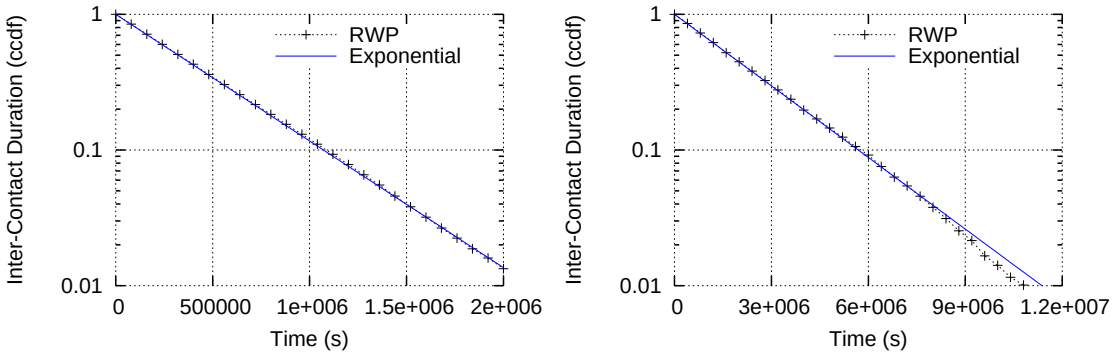


(a)



(b)

Figure 6.5: Convergence of clock offset and skew ($M = 50$ km): (a) Average relative clock offset; (b) Average relative clock skew.



(a) (b)

Figure 6.6: Distribution of inter-contact duration: (a) $M = 20$ km, $\gamma = 2.15 \times 10^{-6} s^{-1}$; (b) $M = 50$ km, $\gamma = 4.05 \times 10^{-7} s^{-1}$.

offsets gradually converge to zero as the clock skews converge exponentially with time as shown in Figs. 6.4b and 6.5b. Since nodes have more contact opportunities among them at higher node densities, the convergence speed is faster and 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. 6.6. Similar observations have been obtained for the RD mobility model [105]. 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.

6.4.1.2 Impact of Aging Parameter

Fig. 6.7 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 (6.19) 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.

6.4.2 Simulation Results

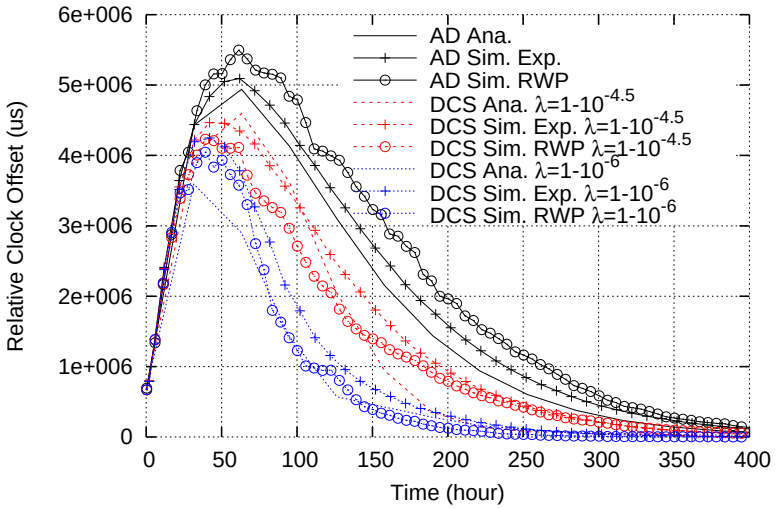
The convergence of the DCS protocol is verified using the numerical results which match well with the simulation results. However, for the analytical tractability, the analytical model is derived only for random mobility models with some stochastic approximations. Here, the performance of the DCS protocols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

6.4.2.1 Impact of Node Speed

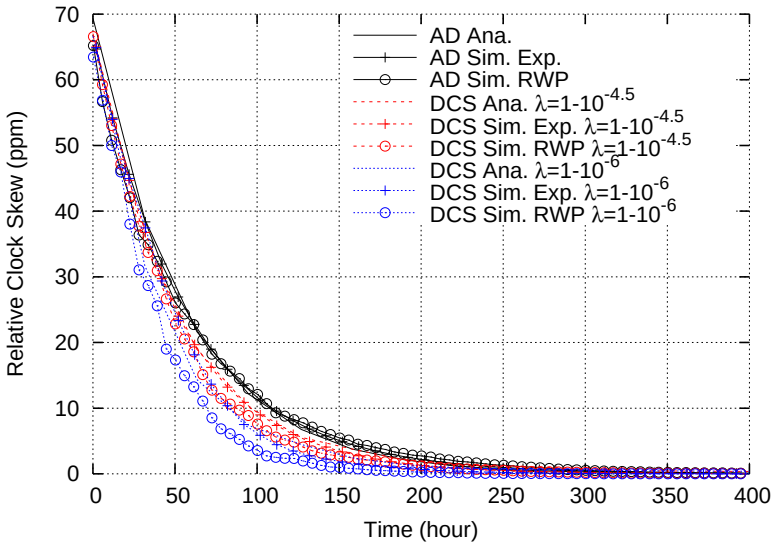
Fig. 6.8 shows the impact of node speed v in m/s. For all the node speeds, the clock offset of the DCS protocol converges faster than that of the AD protocol. As the node speed increases, the convergence speed increases and the initial divergence of relative clock offsets decreases. This is because at higher node speeds, nodes have more contact opportunities among them.

6.4.2.2 Impact of Aging Parameter

The impact of the tuning parameter λ in (6.9) is shown in Fig. 6.9. 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

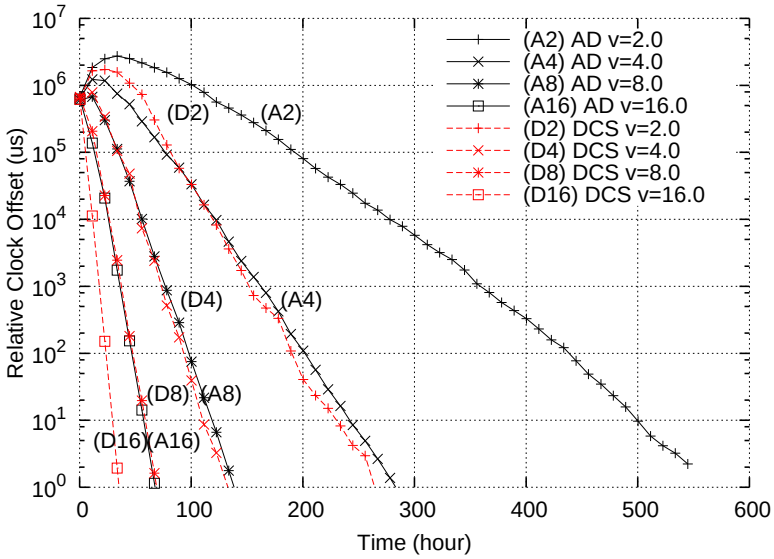


(a)

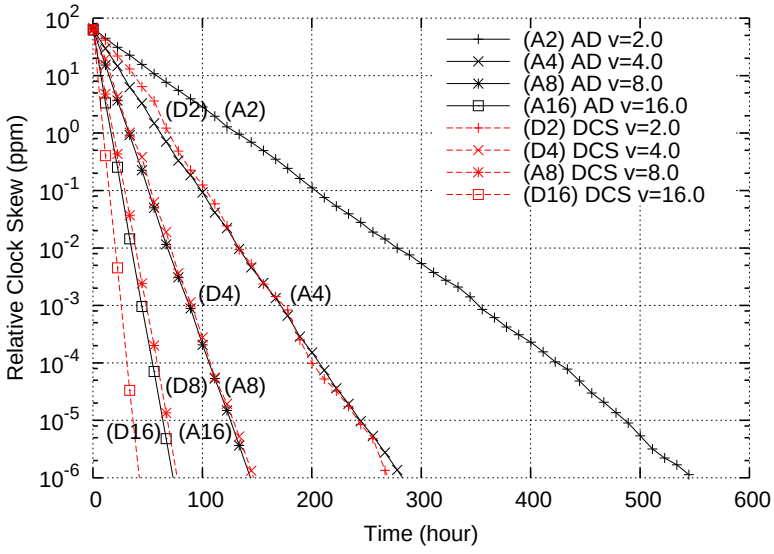


(b)

Figure 6.7: Convergence under different aging parameters: (a) Average relative clock offset; (b) Average relative clock skew.

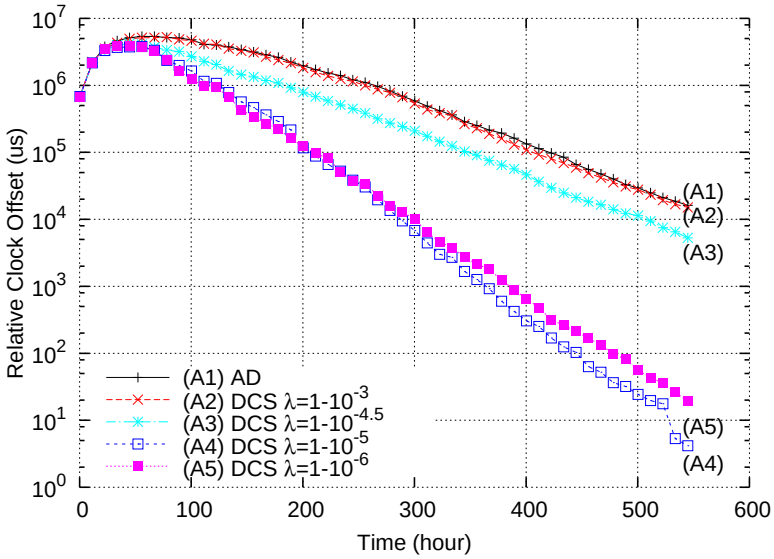


(a)

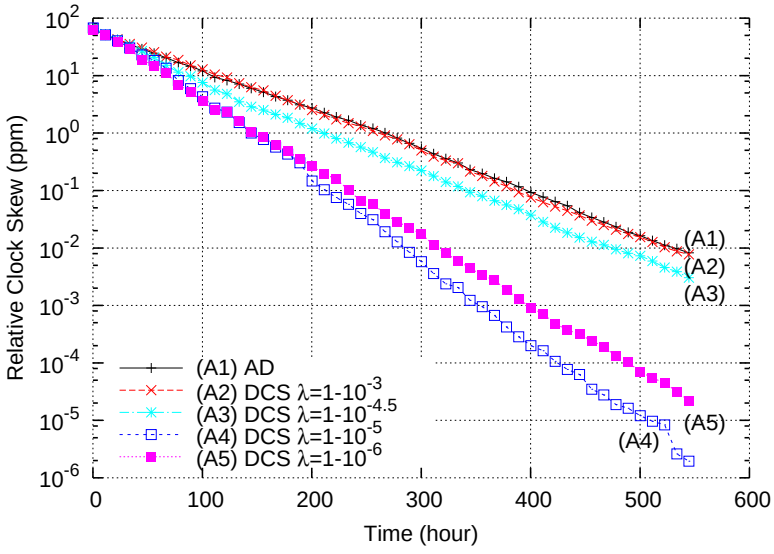


(b)

Figure 6.8: Impact of node speed: (a) Average relative clock offset; (b) Average relative clock skew.



(a)



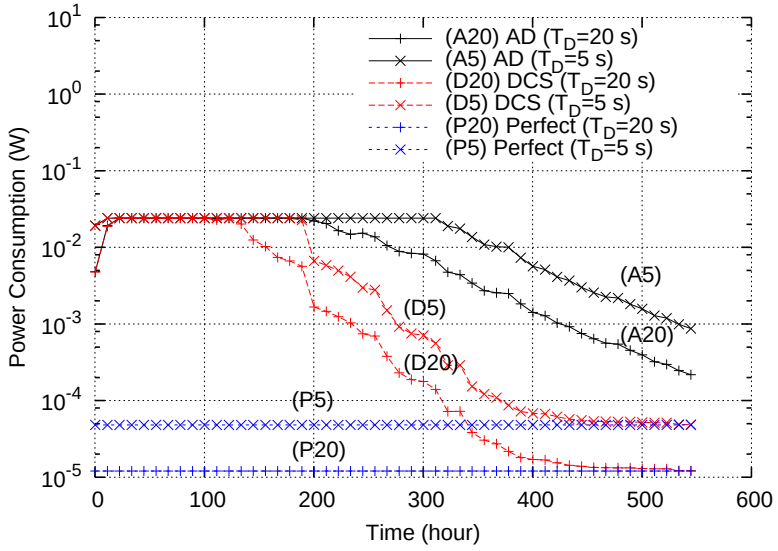
(b)

Figure 6.9: Impact of aging parameter: (a) Average relative clock offset; (b) Average relative clock skew.

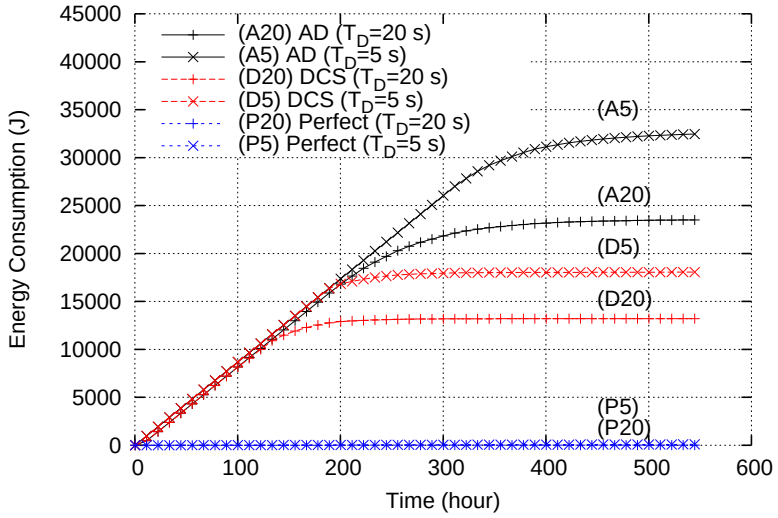
μs, 4463 μs, 3 μs, and 13 μs, respectively. This result indicates that there exists some optimal λ value for each scenario. The impact of different aging parameter values can also be seen in Fig. 6.9b for the skew result. The aging parameter can be selected to effectively discard the information that 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.

6.4.2.3 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. 6.10. 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 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 [43] is used where power consumptions during the idle mode and sleep mode are 24.0 mW and 0.03 mW, respectively. Fig. 6.10a 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. 6.10b. The simulation results indicate how the synchronization error can significantly increase the energy consumption.

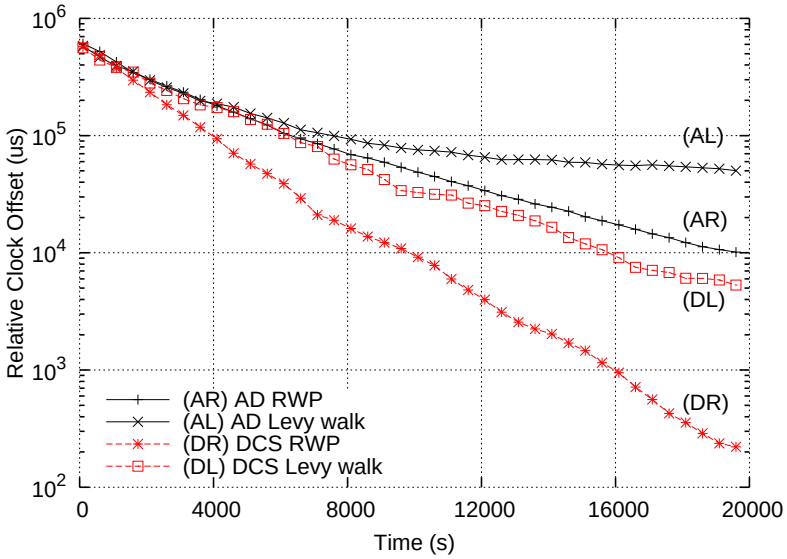


(a)

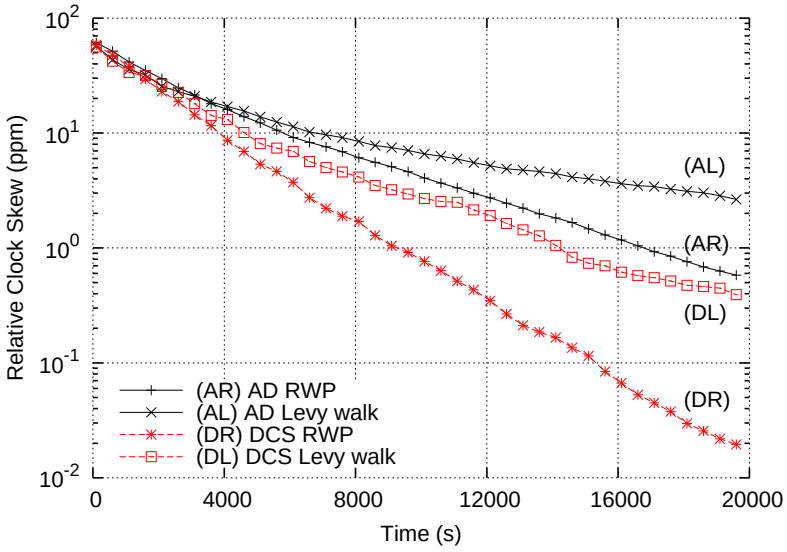


(b)

Figure 6.10: Impact on energy consumption: (a) Average power consumption; (b) Cumulative average energy consumption.



(a)



(b)

Figure 6.11: Impact of mobility model ($M = 5$ km): (a) Average relative clock offset; (b) Average relative clock skew.

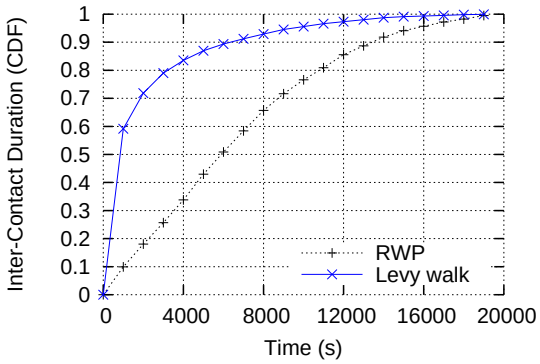


Figure 6.12: Distribution of inter-contact duration ($M = 5$ km)

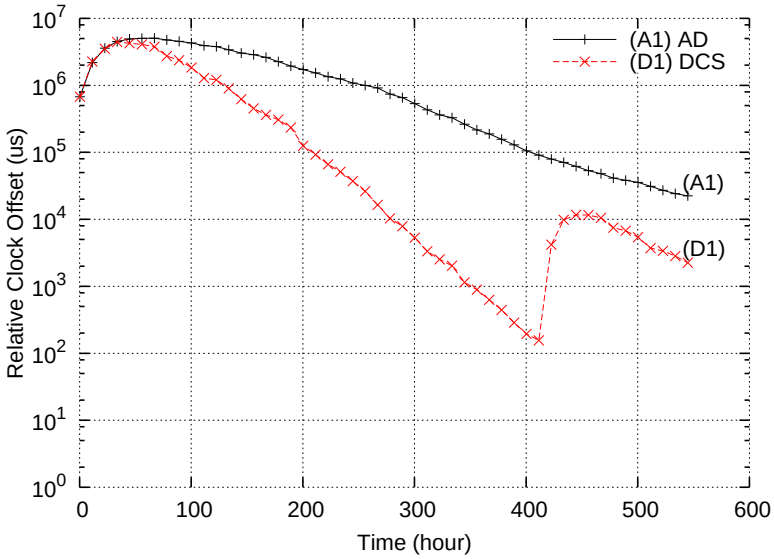
6.4.2.4 Impact of Mobility Model

Levy walk mobility model [64] 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 simulations[1]. The average number of contacts during the simulations for RWP and Levy walk mobility model are 1680 and 1682, respectively. Fig. 6.11a 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. 6.12. Therefore, the convergence speed depends on how evenly the probability of meeting different nodes is distributed.

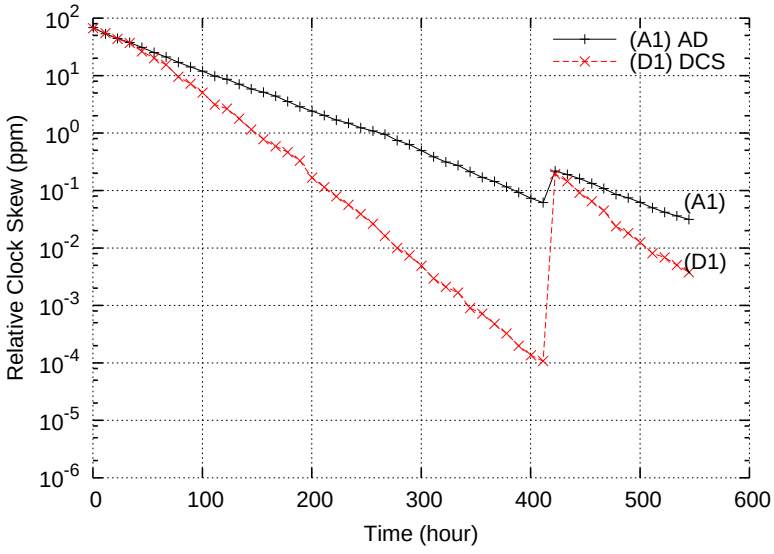
6.4.2.5 Impact of Clock Frequency Instability

We investigate the impact of the clock frequency instabilities on the clock convergence. First, the impact of short-term clock frequency instability is studied. The clock frequency of half of

[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 [64] for the detailed description of the model and the choice of the parameters used in the trace generation.

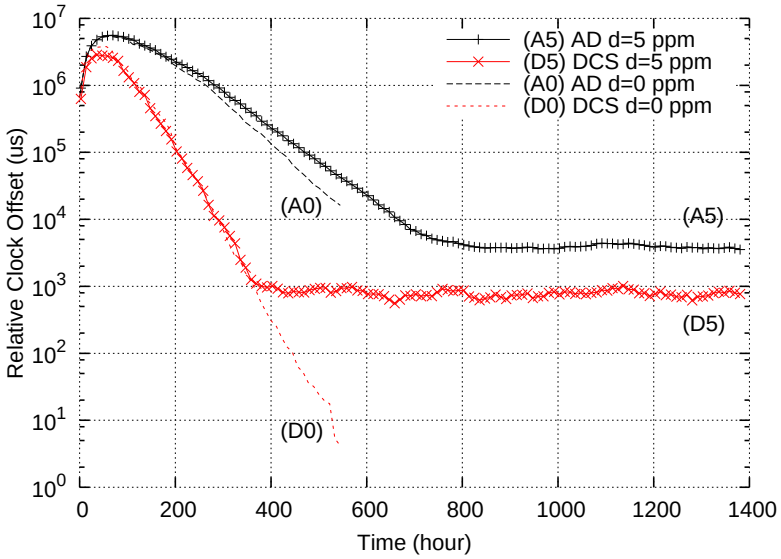


(a)

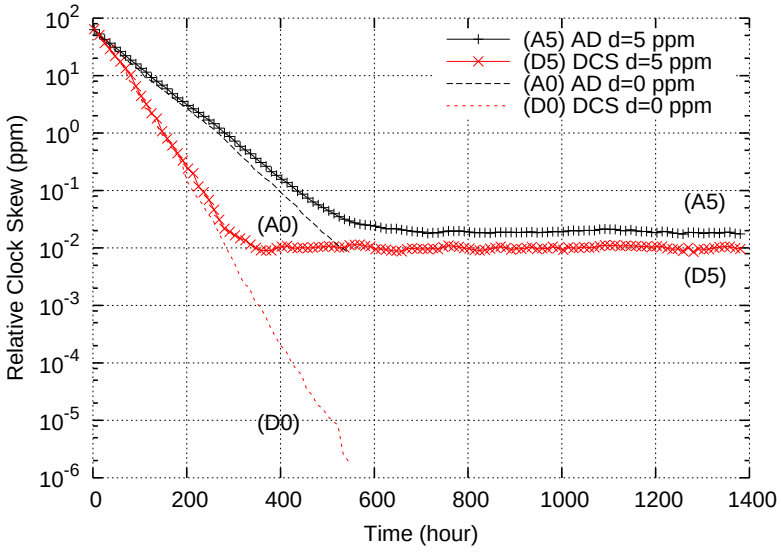


(b)

Figure 6.13: Impact of short-term clock frequency stability due to a temperature change: (a) Average relative clock offset; (b) Average relative clock skew.



(a)



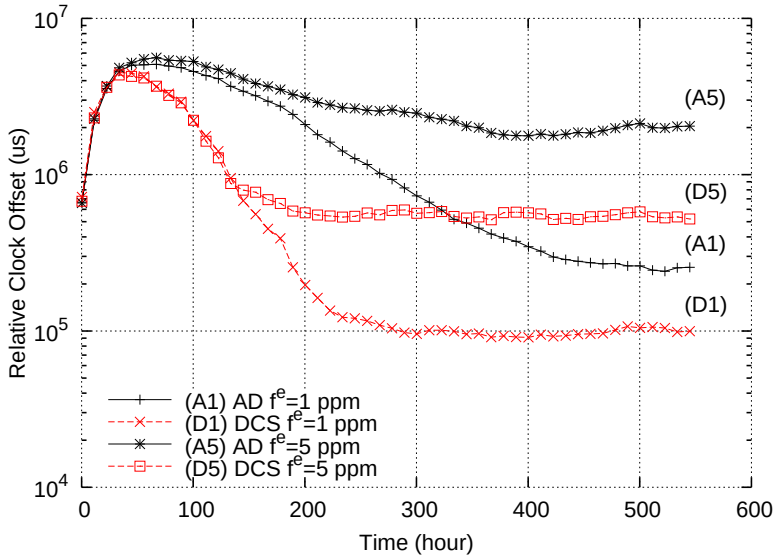
(b)

Figure 6.14: Impact of long-term clock frequency instability due to the oscillator aging: (a) Average relative clock offset; (b) Average relative clock skew.

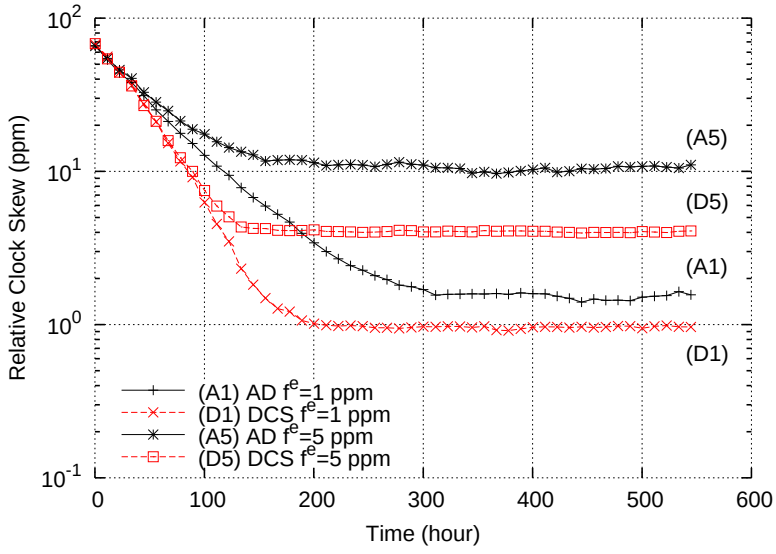
nodes are changed at 420 h assuming a commonly used 32 kHz tuning forkcrystal with a known parabolic coefficient of -0.04 ppm/$^{\circ}C^2$ [39] and a temperature change of -10 $^{\circ}C$. As shown in Fig. 6.13, the relative clock values of the AD protocol are not significantly affected due to the fact 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 bound of ± 5 ppm per year [39]. As shown in Fig. 6.14, 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 1400 h, the average relative clock offsets of the AD and the DCS protocols are 771 μs and 3539 μs, respectively.

6.4.2.6 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 [75, 77, 111] and ± 1 ppm [112], respectively. The bound of the relative skew estimation error corresponds to 1 percent of the maximum clock frequency error. As shown in Fig. 6.15, similar to the simulation results in Fig. 6.14, the relative clock offset and skew converge to some limit. Although the expected value of the estimation error is zero, the relative value of the error can be non-zero. 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 [112] 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.



(a)



(b)

Figure 6.15: Impact of relative clock estimation error: (a) Average relative clock offset; (b) Average relative clock skew.

6.5 Summary

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.

Chapter 7

Conclusions and Further Research

7.1 Conclusions

This thesis aims at energy efficient MAC protocol design and analysis in DTNs, including adaptive sleep scheduling protocols under synchronous, asynchronous, and semi-asynchronous clocks, and a distributed clock synchronization. In this chapter, we conclude this thesis by summarizing our contributions and proposing further research.

- In Chapter 3, an adaptive sleep scheduling protocol based on synchronous clocks is proposed to reduce the energy waste due to idle listening problem and extend the lifetime of energy capacity limited mobile devices. The exponential beacon periods are adaptively adjusted depending on the metric that captures the trend of contact availability under intermittent connections. The tunable parameters in the metric are optimized for different network conditions. Simulation results show that a considerable amount of energy is saved while maintaining similar average packet delays and packet delivery ratios to that without a sleep scheduling.

- In Chapter 4, an adaptive sleep scheduling protocol based on asynchronous clocks are proposed to consider the fact that global clock synchronization is often difficult to achieve due to imperfect clock oscillators and large inter-contact durations among nodes in the DTNs.

The sleep schedules, systematically constructed based on hierarchical arrangements of particular cyclic difference sets, allow multiple power saving levels that can be independently selected by each node in the network. Implementations issues, including frame structure, adaptive neighbor discovery, and message exchange, have been discussed. Theoretical analysis is given to demonstrate the energy efficiency and scalability of the proposed protocols in comparison with existing protocols. Although the power saving levels of the proposed protocols are less fine grained than that of other existing protocols, they are more energy efficient since each level can achieve the same neighbor sensitivity with lower active ratio. Simulation results show that energy waste due to idle listening can be reduced up to 35 percent in comparison with existing protocols (where the theoretical achievable bound is 50 percent).

- In Chapter 5, a distributed semi-asynchronous sleep scheduling protocol is proposed considering loosely synchronized clocks in sparse mobile wireless networks. Individual nodes using the proposed protocol can adjust to required connectivity and to the estimated clock synchronization error. Using theoretical analysis and simulation results, we demonstrate that the proposed protocol can achieve higher energy efficiency than existing asynchronous sleep scheduling protocols under certain conditions. The DSA can provide flexibility to existing synchronous sleep scheduling protocols, where frequent resynchronization is necessary, and higher energy efficiency than existing asynchronous sleep scheduling protocols, where the distribution of synchronization error is not considered.

- In Chapter 6, a distributed asynchronous clock synchronization protocol is proposed to reduce clock synchronization error and reduce energy waste from additional guard intervals. Asynchronously and intermittently exchanged relative clock information spread among nodes is compensated using weighted averages. The weights used in the clock compensations are assigned so that they form row stochastic matrices in order to guarantee clock convergence and are depreciated with respect to time in order to account for long delays. The formulated analytical models match well with the simulation results and can be used as efficient tools to estimate the system performance. Simulation results demonstrate that the proposed protocol achieves faster clock convergence and reduce energy consumption due to neighbor discovery by half.

7.2 Further Research

This thesis focuses on energy efficient protocol designs and analysis in various DTNs. There are many relevant research issues that are worth further investigation.

- **Joint Routing and MAC Protocol for Energy Efficiency** - Different DTN routing protocols have been designed to tackle routing problems for different application scenarios. Unfortunately, they assume an always on radio without power saving protocols. Energy efficiency of DTN routing protocols not only depend on the number of message exchanged among nodes but also on the amount of idle listening intervals. Energy efficient MAC layer protocols can be specifically designed for different DTN routing protocols.

- **Fine-Grained Construction of Asynchronous Sleep Scheduling** - The multiple power saving levels of the proposed adaptive asynchronous protocols are less fine grained than existing protocols, since the duty cycles of power saving levels are limited to the multiples of the fixed initial set. More fined grained protocol, perhaps with higher construction complexity, can be designed to provide more power saving options for nodes.

- **Adaptive Semi-Asynchronous Sleep Scheduling** - The synchronization error distribution may change over time. The performance of the proposed semi-asynchronous protocol can be further improved if the the protocol can be adaptively adjust its parameters depending on the time varying synchronization error distribution.

- **Accurate Clock Error Estimation Algorithm** - The performance of the proposed semi-asynchronous sleep scheduling protocol depends on the accuracy of the clock synchronization error distribution. An accurate estimation of the distribution allows optimization of the sleep scheduling parameter c and the lengths of adaptive adjusted guard intervals.

- **Refined Clock Compensation Algorithms** - More intelligent clock compensation algorithm can be designed by using more information about the network, such as mobility characteristic and accuracy of exchanged clock information. Weights can be calculated depending on the node's influence (e.g. contact rates) or reputation (e.g. relative accuracy) to further improve the convergence speed. Also, an estimator, such as MMSE (minimum

mean square error), can be used to calculate the accuracy of clock information instead of decaying weights with respect to time.

- **Optimization of the Aging Parameter** - The convergence speed of the proposed distributed clock synchronization algorithm depends on the choice of the aging parameter. Closed form expression for the analytical model can be derived to optimize the aging parameter. Also, the aging parameter can be adaptively adjusted to further enhance the performance under dynamic network environments.

- **Realistic Mobility Models** - The performance of the protocols is evaluated under random mobility models. In order to better reflect the real life scenarios, different mobility models, such as trace-driven mobility, vehicular mobility, and predicted mobility, can be considered.

APPENDICES

Appendix A

Supplementary

A.1 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

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

$$Y_{ii}(\tau_{k+1}) = \sum_{\substack{l=1 \\ l\neq i}}^{N} \frac{1-P_0}{(N-1)} \overline{Y}_{il}(\tau_{k+1}) + P_0 Y_{ii}(\tau_k), \quad \text{(A.2)}$$

$$\text{for } i = 1, \cdots, N$$

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

$$\overline{X}_{il}(\tau_{k+1}) = \frac{[X_{ii}(\tau_k) + \tau Y_{ii}(\tau_k)] + [X_{ll}(\tau_k) + \tau Y_{ll}(\tau_k)]}{2}, \quad \text{(A.3)}$$

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

$$\overline{Y}_{il}(\tau_{k+1}) = \frac{Y_{ii}(\tau_k) + Y_{ll}(\tau_k)}{2}. \quad \text{(A.4)}$$

Define the system state of the AD protocol at time τ_k as $\mathbf{S}_{AD}(\tau_k) = \{X_{ii}(\tau_k), Y_{ii}(\tau_k) | i = 1, \cdots, N\}$. 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

$$\mathbf{S}_{AD}(\tau_k) = F_{AD}^k(\mathbf{S}_{AD}(0)). \tag{A.5}$$

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

Appendix B

Proofs of Theorems, Corollaries, and Lemmas

Theorem 1. (Rotational Closure Property of the Exponential Hierarchical Design) *Given two sets $P_i = E^{i-1} \otimes I$ and $P_j = E^{j-1} \otimes I$ where $i \leq j$. For $n = v_I v_E^j$, $C_{a,n}(P_i) \cap C_{b,n}(P_j) \neq \emptyset$, $\forall a, b : 0 \leq a, b \leq n$.*

Proof. *CASE 1-Perfect Alignment of Slot Boundaries: For i = j = 1, two sets are cyclic difference sets with set I. Therefore, there is at least one overlapping slot in $n_j = v_I$. For i = 1 and j = 0, $P_2 = E \otimes I$ and $P_1 = I$. I can be viewed as is a slot of $E \otimes I$ with a length v_I. Then, there is at least one overlapping slot of length v_I in $n_j = v_I v_E$. Since I is a cyclic difference set, there is at least one overlapping slot within v_I. For i = j = 1, two sets are cyclic difference sets with set E with slots of I. Then, there is at least one overlapping slot of length v_I in $n_j = v_I v_E$, and consequently at least one overlapping slot within v_I. This can be proved for larger i and j by recursion.*

CASE 2-Imperfect Alignment of Slot Boundaries: Suppose that the difference in the clock shift of two set is δ, and length of a slot is L_s. Since P_i and P_j are cyclic difference sets, for an imperfect alignment of slots of δ, there exist overlapping sections $(L_s - \delta)$ and δ. The combination of $(L_s - \delta)$ and δ is equal to L_s. □

Corollary 1. (Rotational Closure Property of the Multiplication Hierarchical Design) *Given*

two sets $P_i = M_{i-1} \otimes I$ and $P_2 = M_{j-1} \otimes I$ where $v_{Mi} \leq v_{Mj}$ and $RCP\{M_i, M_j\} = 1$, there exists an overlapping interval of at least v_I in v_{Mj}. Therefore, similar to proof for Theorem 1, we can prove that for $n = v_I v_{Mj}$, $C_{a,n}(P_i) \cap C_{b,n}(P_j) \neq \emptyset$, $\forall a, b : 0 \leq a, b \leq n$.

Theorem 2. (Minimum Slot Length of Asynchronous Sleep Scheduling Protocol) *The slot length should be at least $2 \times L_A$ to guarantee an overlapping active interval that is long enough for the connection establishment.*

Proof. *By Theorem 1, rotational sets always have overlapping active intervals of duration $(L_s - \delta)$ and δ for the duration of the larger frame. Since $\max\{(L_s - \delta), \delta\} \geq L_s/2$ for $\delta = (0, L_s)$, nodes can exchange connection setup messages in either $(L_s - \delta)$ or δ if $L_A \leq L_s/2$.* □

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 $t_{il}^d(t_k) \geq 0$ 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. 6.3.*

Proof. *According to (6.6), when node i contacts node j, the update using the relative clock skews with respect to the underlying actual clock frequency can be calculated as*

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

Note that we have $f_{il}^T(t_k) = f_l(t_{il}^d(t_k)) - f_i(t_k)$ in (B.1) since the clock compensation is performed for both clock frequency $f_i(t_k^{i,j})$ in (6.6) and clock skews $f_{il}^T(t_k^{i,j})$ in (6.8), which preserves the recorded actual clock frequency $f_l(t_{il}^d(t_k))$ upon each contact. □

Theorem 3. (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 [103, 113], 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 $t_k - B < t_{il}^d(t_k) \leq t_k$) 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 ($t_k - t_{il}^d(t_k) < B$), if the sum of the inter-contact durations over the forwarding path is bounded. In particular, in deterministic mobility scenarios, such as bus routes [32] and message ferries [114, 115], 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 *[116] 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 [104, 107], 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.

Theorem 4. (Shift Intersection Property of the CGI) *Given two CGI sets $E_{n_{g1}}$ and $E_{n_{g2}}$ where $0 \leq n_{g1} \leq n_{g2}$, $S(E_{n_{g1}}, h_1) \cap S(E_{n_{g2}}, h_2) \neq \emptyset\ \forall\ h_1, h_2 : h_1 \in \{-n_{g1}, \ldots, n_{g1}\}, h_2 \in \{-n_{g2}, \ldots, n_{g2}\}$.*

Proof. *Since $\{0\} \in S(E_{n_{g1}}, h_1)\ \forall\ h_1$ and $\{0\} \in S(E_{n_{g2}}, h_2)\ \forall\ h_2$, there exists $E \in S(E_{n_{g1}}, h_1) \cap$*

$S(E_{n_{g2}}, h_2)$ *such that* $\{0\} \in E \; \forall \; h_1, h_2$. $\square$

Theorem 5. (Shift Intersection Property of the DSA) *Given two DSA sets* $E_{n_{g1},c}$ *and* $E_{n_{g2},c}$ *where* $0 \leq n_{g1} \leq n_{g2}$, $S(E_{n_{g1},c}, h_1) \cap S(E_{n_{g2},c}, h_2) \neq \emptyset \; \forall \; h_1, h_2 : h_1 \in \{-n_{g1}, \ldots, n_{g1}\}, h_2 \in \{-n_{g2}, \ldots, n_{g2}\}$.

Proof. *Let* $E^*_{n_i,c} = \{-n_i, -n_i + 1, \ldots, -n_i + c - 2, -n_i + c - 1\}$ *and* $E^\circ_{n_i,c} = \{-n_i, -n_i + \tilde{c} - 1, -n_i + 2\tilde{c} - 1, \ldots, n_i\}$ *where* $\tilde{c}$ $(\leq c)$. *Here* $E^*_{n_i,c} \cup E^\circ_{n_i,c} = E_{n_i,c}$. *Let* n_s *represent the number of overlapping slots, including both awake and asleep slots, between the two sets.*

CASE 1-$n_s \geq c$ *(*$|\check{h}| \leq (n_{g2} + n_{g1} - c)$*): Since there are more than* c *overlapping slots, there exists* $Q_1 \in S(E_{n_{g1},c}, h_1)$ *and* $P_2 \in S(E_{n_{g2},c}, h_2)$ *such that (1)* $Q_1 = S(E^*_{n_{g1},c}, h_1)$ *and* $Q_2 = S(E^\circ_{n_{g2},c}, h_2)$ *when* $\check{h} < 0$ *or (2)* $Q_1 = S(E^\circ_{n_{g1},c}, h_1)$ *and* $Q_2 = S(E^*_{n_{g2},c}, h_2)$ *when* $\check{h} \geq 0$. *For (1) and (2), there are* c *awake slots in* Q_1 *and at least one awake slot in* Q_2 *within the range of* $[-n_{g1} + h_1, n_{g1} + h_1 + c - 1]$ *and* $[-n_{g2} + h_2, n_{g2} + h_2 + c - 1]$, *respectively. Therefore,* $Q_1 \cap Q_2 \neq \emptyset$.

CASE 2-$n_s < c$ *(*$(n_{g2}+n_{g1}-c) < |\check{h}| \leq (n_{g2}+n_{g1})$*): For the maximum shift (*$|\check{h}_{max}| = (n_{g2}+n_{g1})$*),* $\{0\} \in S(E_{n_{g1},c}, h_1)$ *for* $h_1 = \{n_{g1}, -n_{g1}\}$ *and* $\{0\} \in S(E_{n_{g2},c}, h_2)$ *for* $h_2 = \{n_{g2}, -n_{g2}\}$. *Therefore, one awake slot at slot 0 is overlapping as* $S(E_{n_{g1},c}, \pm n_{g1}) \cap S(E_{n_{g2},c}, \mp n_{g2}) = \{0\}$. *For* $(|\check{h}_{max}| - x)$ *shift, there are* x *overlapping slots. Since, the partial overlapping occurs between the opposite edges of the frame,* $(-n_{g2} + h_2 + x)$ *overlaps with* $(n_{g1} + h_1)$ *when* $\check{h}_{max} > 0$ *and* $(n_{g2} + h_2)$ *overlaps with* $(-n_{g1} + h_1 + x)$ *when* $\check{h}_{max} < 0$. *Therefore,* $E_{n_{g1},c} \cap E_{n_{g2},c} \neq \emptyset$. $\square$

References

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

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

[3] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, “Delay-Tolerant Networking Architecture,” RFC 4838 (Informational), Apr. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4838.txt

[4] A. Vahdat and D. Becker, “Epidemic Routing for Partially Connected Ad Hoc Networks,” Duke University, Tech. Rep. CS-200006, April 2000.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networking,” in *Proc. IEEE INFOCOM*, April 2006.

[6] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic Routing in Intermittently Connected Networks,” *Lecture Notes in Computer Science*, vol. 3126, pp. 239–254, Sept. 2004.

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient Routing in Intermittently Connected Mobile Networks: The Multi-Copy Case,” *IEEE/ACM Trans. Networking*, vol. 16, no. 1, 2008.

[8] ——, “Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case,” *IEEE/ACM Trans. Networking*, vol. 16, no. 1, pp. 63–76, 2008.

[9] R. Malladi and D. P. Agrawal, “Current and Future Applications of Mobile and Wireless Networks,” *Commun. ACM*, vol. 45, no. 10, pp. 144–146, 2002.

[10] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-Tolerant Networking: An Approach to Interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, 2003.

[11] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2, pp. 20–41, 2001.

[12] S. Jain, K. Fall, and R. Patra, "Routing in a Delay Tolerant Network," in *Proc. ACM SIGCOMM*, Aug.–Sept. 2004.

[13] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen, "A Survey of Energy Effcient Network Protocols for Wireless Networks," *Wireless Networks*, vol. 7, pp. 343–358, 2001.

[14] H. Karl, "An Overview of Energy-Efficiency Techniques for Mobile Communication Systems," Telecommunication Networks Group, Technical University Berlin, Tech. Rep., Sept. 2003.

[15] S. Singh and C. S. Raghavendra, "PAMAS-Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 5–26, 1998.

[16] C. F. Chiasserini and R. R. Rao, "A Distributed Power Management Policy for Wireless Ad Hoc Networks," in *Proc. IEEE WCNC*, Sept. 2000.

[17] R. Zheng and R. Kravets, "On-demand Power Management for Ad Hoc Networks," in *Proc. IEEE INFOCOM*, Mar.–April 2003.

[18] "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (2007 revision)," IEEE Computer Society, June 2007.

[19] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Wirel. Netw.*, vol. 8, no. 5, pp. 481–494, 2002.

[20] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, “Power-saving protocols for IEEE 802.11-based Multi-hop Ad Hoc Networks,” in *Proc. IEEE INFOCOM*, June 2002.

[21] ——, “Power-Saving Protocols for IEEE 802.11-based Multi-Hop Ad Hoc Networks,” *Computer Networks*, vol. 43, no. 3, pp. 317–337, 2003.

[22] R. Zheng, J. C. Hou, and L. Sha, “Asynchronous Wakeup for Ad Hoc Networks,” in *Proc. ACM MobiHoc*, June 2003.

[23] I.-C. Chou, C.-M. Chao, and J.-P. Sheu, “An Adaptive Quorum-Based Energy Conserving Protocol for IEEE 802.11 Ad Hoc Networks,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 560–570, 2006.

[24] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, “Quorum-based Asynchronous Power-Saving Protocols for IEEE 802.11 Ad Hoc Networks,” *Mobile Networks and Applications*, vol. 10, no. 1-2, pp. 169–181, 2005.

[25] Z.-T. Chou, “Optimal Adaptive Power Management Protocols for Asynchronous Wireless Ad Hoc Networks,” in *Proc. IEEE WCNC*, Mar. 2007.

[26] S.-H. Wu, C.-M. Chen, and M.-S. Chen, “An Asymmetric Quorum-based Power Saving Protocol for Clustered Ad Hoc Networks,” in *Proc. IEEE ICDCS*, June 2007.

[27] ——, “An Asymmetric and Asynchronous Energy Conservation Protocol for Vehicular Networks,” *IEEE Trans. Mobile Comput.*, vol. 9, pp. 98–111, 2010.

[28] S.-H. Wu, M.-S. Chen, and C.-M. Chen, “Fully Adaptive Power Saving Protocols for Ad Hoc Networks Using the Hyper Quorum System,” in *Proc. IEEE ICDCS*, June 2008.

[29] A. Keränen and J. Ott, “Increasing Reality for DTN Protocol Simulations,” Helsinki University of Technology, Tech. Rep., July 2007.

[30] L. M. Feeney and M. Nilsson, “Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment,” in *Proc. IEEE INFOCOM*, Mar.–April 2001.

[31] M. Stemm and R. H. Katz, “Measuring and Reducing Energy Consumption of Network Interfaces in Hand-held Devices,” *IEICE Trans. on Communications*, vol. E80-B, no. 8, pp. 1125–1131, 1997.

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

[33] B. J. Choi and X. Shen, “Adaptive Exponential Beacon Period Protocol for Power Saving in Delay Tolerant Networks,” in *Proc. IEEE ICC*, June 2009.

[34] ——, “Adaptive Asynchronous Clock based Power Saving Protocols for Delay Tolerant Networks,” in *Proc. IEEE GLOBECOM*, Nov.–Dec. 2009.

[35] ——, “Adaptive Asynchronous Sleep Scheduling Protocols for Delay Tolerant Networks,” *IEEE Trans. Mobile Comput.*, to appear.

[36] ——, “DSA: Distributed Semi-Asynchronous Sleep Scheduling Protocol for Mobile Wireless Networks,” in *Proc. IEEE ICC*, June 2011.

[37] ——, “Distributed Clock Synchronization in Delay Tolerant Networks,” in *Proc. IEEE ICC*, May. 2010.

[38] 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.

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

[40] K. Römer, “Time Synchronization in Ad Hoc Networks,” in *Proc. ACM MobiHoc*, Oct. 2001.

[41] 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.

[42] R. Zheng, J. C. Hou, and L. Sha, "Optimal Block Design for Asynchronous Wake-Up Schedules and Its Applications in Multihop Wireless Networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 9, pp. 1228–1241, 2006.

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

[44] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic Properties of the Random Waypoint Mobility Model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[45] A. K. Salkintzis and C. Chamzas, "An In-Band Power-Saving Protocol for Mobile Data Networks," *IEEE Trans. Commun.*, vol. 46, pp. 1194–1205, 1998.

[46] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, June 2002.

[47] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. ACM SenSys*, Nov. 2003.

[48] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-Efficient Collision-free Medium Access Control for Wireless Sensor Networks," in *Proc. ACM SenSys*, Nov. 2003.

[49] 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.

[50] H. Jun, M. H. Ammar, M. D. Corner, and E. W. Zegura, "Hierarchical Power Management in Disruption Tolerant Networks with Traffic-Aware Optimization," in *Proc. ACM CHANTS*, Sept. 2006.

[51] C.-L. Tsao, Y.-T. Wu, W. Liao, and J.-C. Kuo, "Link Duration of the Random Way Point Model in Mobile Ad Hoc Networks," in *Proc. IEEE WCNC*, April 2006.

[52] M. Abdulla and R. Simon, "Characteristics of Common Mobility Models for Opportunistic Networks," in *Proc. ACM PM2HW2N*, Oct. 2007.

[53] "The Network Simulator (NS-2)," http://www.isi.edu/nsnam/ns/.

[54] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," in *Proc. ACM WDTN*, Aug. 2005.

[55] J. A. John, *Cyclic Designs*. Chapman and Hall Ltd., 1987.

[56] S.-L. Wu, P.-C. Tseng, and Z.-T. Chou, "Distributed Power Management Protocols for Multi-hop Mobile Ad Hoc Networks," *Computer Networks*, vol. 47, no. 1, pp. 63–85, 2005.

[57] C.-M. Lin, G.-M. Chiu, and C.-H. Cho, "A New Quorum-Based Scheme for Managing Replicated Data in Distributed Systems," *IEEE Trans. Computers*, vol. 51, no. 12, pp. 1442–1447, 2002.

[58] S. Lang and L. Mao, "A Torus Quorum Protocol for Distributed Mutual Exclusion," in *Proc. ICPDCS*, Oct. 1998.

[59] J. Marshall Hall, *Combinatorial Theory*, 2nd ed. John Wiley and Sons, Inc., 1983.

[60] L. D. Baumert, *Cyclic Difference Sets*. Springer-Verlag, 1971.

[61] J. Singer, "A Theorem in Finite Projective Geometry and Some Applications to Number Theory," *Trans. of the American Mathematical Society*, vol. 43, no. 3, pp. 377–385, May 1938.

[62] W.-S. Luk and T.-T. Wong, "Two New Quorum Based Algorithms for Distributed Mutual Exclusion," in *Proc. IEEE ICDCS*, May 1997.

[63] W. Wang, V. Srinivasan, and M. Motani, "Adaptive Contact Probing Mechanisms for Delay Tolerant Applications," in *Proc. ACM/IEEE MOBICOM*, Sept. 2007.

[64] 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.

[65] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket Switched Networks: Real-World Mobility and Its Consequences for Opportunistic Forwarding," University of Cambridge, Technical Report UCAM-CL-TR-617, Feb. 2005.

[66] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," in *Proc. ACM WDTN*, Aug. 2005.

[67] "TinyOS," http://www.tinyos.net/.

[68] Y. Wu, S. Fahmy, and N. Shroff, "Optimal Sleep/Wake Scheduling for Time-Synchronized Sensor Networks With QoS Guarantees," *Networking, IEEE/ACM Transactions on*, vol. 17, no. 5, pp. 1508 –1521, Oct. 2009.

[69] 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.

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

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

[72] L. Jiang, J.-H. Huang, A. Kamthe, T. Liu, I. Freeman, J. Ledbetter, S. Mishra, R. Han, and A. Cerpa, "SenSearch: GPS and Witness Assisted Tracking for Delay Tolerant Sensor Networks," in *Proc. ADHOC-NOW*, Sept. 2009.

[73] D. Raskovic and D. Giessel, "Battery-Aware Embedded GPS Receiver Node," in *Proc. MobiQuitos*, Aug. 2007.

[74] 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.

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

[76] 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.

[77] 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.

[78] 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.

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

[80] J.-Y. Chen and J. Hu, "On The Convergence of Distributed Random Grouping for Average Consensus on Sensor Networks with Time-Varying Graphs," in *Proc. IEEE CDC*, June 2007.

[81] R. L. Cigno, M. Nardelli, and M. Welzl, "SESAM: A Semi-Synchronous, Energy Savvy, Application-Aware MAC," in *Proc. IEEE/IFIP WONS*, Feb. 2009.

[82] S. Lai, B. Ravindran, and H. Cho, "Heterogenous Quorum-Based Wake-Up Scheduling in Wireless Sensor Networks," *IEEE Trans. Computers*, vol. 59, pp. 1562–1575, 2010.

[83] P. Dutta and D. Culler, "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications," in *Proc. ACM SenSys*, Nov. 2008.

[84] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A Unifying Link Abstraction for Wireless Sensor Networks," in *Proc. ACM SenSys*, Nov. 2005.

[85] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-Connect: A Low-Latency Energy-Efficient Asynchronous Neighbor Discovery Protocol," in *Proc. ACM/IEEE IPSN*, April 2010.

[86] C.-S. Hsu and Y.-C. Tseng, "Cluster-Based Semi-Asynchronous Power-Saving Protocols for Multi-hop Ad Hoc Networks," in *Proc. IEEE ICC*, May 2005.

[87] X. Lin, M. Moh, and T.-S. Moh, "On Energy-Efficient Semi-Asynchronous MAC Protocols for Multi-hop Ad-hoc Networks," in *13th IEEE Asia-Pacific Computer System Architecture Conference (ACSAC)*, Aug. 2008.

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

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

[90] 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.

[91] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," *Commun. ACM*, vol. 21, no. 7, 1978.

[92] L. Schenato and G. Gamba, "A Distributed Consensus Protocol for Clock Synchronization in Wireless Sensor Network," in *Proc. IEEE CDC*, Dec. 2007.

[93] L. Xiao, S. Boyd, and S. Lall, "A Scheme for Robust Distributed Sensor Fusion Based on Average Consensus," in *Proc. ACM/IEEE IPSN*, April 2005.

[94] L. Fang and P. Antsaklis, "Information Consensus of Asynchronous Discrete-Time Multi-Agent Systems," in *Proc. American Control Conference*, June 2005.

[95] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research Challenges and Applications for Underwater Sensor Networking," in *Proc. IEEE WCNC*, April 2006.

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

[97] V. Rodoplu and M. K. Park, "An Energy-Efficient MAC Protocol for Underwater Wireless Acoustic Networks," in *Proc. of MTS/IEEE*, Sept. 2005.

[98] A. F. Harris, III, M. Stojanovic, and M. Zorzi, "When Underwater Acoustic Nodes Should Sleep with One Eye Open: Idle-Time Power Management in Underwater Sensor Networks," in *Proc. ACM WUWNet*, Sept. 2006.

[99] N. Chirdchoo, W.-S. Soh, and K. C. Chua, "MU-Sync: A Time Synchronization Protocol for Underwater Mobile Networks," in *Proc. ACM WUWNet*, Sept. 2008.

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

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

[102] 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.

[103] 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

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

[105] 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.

[106] 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.

[107] 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.

[108] 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.

[109] 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.

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

[111] 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.

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

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

[114] 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.

[115] 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.

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