

PRIVACY PRESERVING CHANNEL ACCESS USING  
BLINDFOLDED PACKET TRANSMISSIONS

By

Debasmit Banerjee

A DISSERTATION

Submitted to  
Michigan State University  
in partial fulfillment of the requirements  
for the degree of

Electrical Engineering – Doctor of Philosophy

2014

# ABSTRACT

## PRIVACY PRESERVING CHANNEL ACCESS USING BLINDFOLDED PACKET TRANSMISSIONS

By

Debasmit Banerjee

This thesis presents a novel wireless MAC-layer approach towards achieving channel access anonymity by preventing *linkability* and *traffic analysis*. Nodes autonomously select periodic TDMA-like time-slots for channel access by employing a channel sensing strategy, and they do so without explicitly sharing any identity information with other nodes in the network, thus preventing *linkability*. Without any message-based coordination, entire packet contents, including all addressing information, are hidden from other nodes and TDMA slot-allocation is achieved using blindfolded packet transmissions. The main idea behind this approach is to prevent identity exposure using encrypted packets, and thwart traffic analysis by taking advantage of the inherent periodic traffic pattern of TDMA protocols. The main contributions of this thesis are as follows. First, it presents ZEA-TDMA, a MAC protocol from the TDMA family, which is able to fulfill the aforementioned requirements. Additionally, an energy-aware approach is presented which can have applications in resource constrained sensor networks. Simulation-based experimental results have been presented to evaluate the functionality and performance of the proposed mechanisms. An analytical model has been developed to study the energy consumption of the proposed mechanism. In addition, a hardware module for wireless collision detection has also been developed. Finally, the thesis concludes with the system prototype implementation of the protocol and wireless network test-bed performance results, which demonstrate the functional feasibility of the developed concepts in this thesis.

Copyright by  
DEBASMIT BANERJEE  
2014

*To My Parents, Anindita and Sankar Banerjee,  
For their unconditional love and support*

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Dr. Subir Biswas. Without his immense support and mentoring through my whole PhD career, this would not have been possible. His patience, infinite energy, eagerness to help, and academic rigor in research guided me to continue improving myself in every aspect of my research. I would also like to thank members of my committee, Dr. Guoliang Xing, Dr. Joydeep Mitra, and Dr. Tongtong Li for their time and support.

I would also like to thank Mahmoud Taghizadeh and Bo Dong for their collaboration. Furthermore, I would like to extend my gratitude to the past and current members of our lab for the good times: Muhannad Quwaider, Anthony Plummer, Ali Aqel, Qiong Huo, Faeze Hajiaghajani, William Tomlinson, Stephan Lorenz, Yan Shi, Saptarshi Das, and Feng Dezhi.

Last but not least, I am greatly thankful to my parents, Anindita and Sankar Banerjee, for their unconditional love and encouragement. I would also like to extend my gratitude to the members of my family and my friends. A special thanks goes to my wife, Debanjana Mukherjee, for her love and support during the tough years of my PhD career.

# TABLE OF CONTENTS

LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
LIST OF ALGORITHMS .....	xiii
Chapter 1: Introduction .....	1
1.1 Background and Motivation .....	1
1.2 Current Solutions .....	2
1.3 Proposed Mechanism .....	2
1.4 Applications .....	3
1.5 Challenges .....	6
1.6 Dissertation Objectives .....	7
Chapter 2: Related Work .....	10
2.1 Privacy Preserving Systems .....	10
2.2 MAC-layer and Privacy Preservation .....	12
2.3 TDMA Protocols .....	15
2.4 Scope of Thesis .....	16
Chapter 3: Network and Threat Model .....	18
3.1 Introduction .....	18
3.2 Network Model .....	18
3.3 Trust Domain .....	19
3.4 Threat Model .....	20
3.5 Privacy Metric .....	21
3.6 Summary .....	23
Chapter 4: Zero Exposure Anonymous TDMA (ZEA-TDMA) .....	24
4.1 Time-Coded ( <i>Blindfolded</i> ) Packet Transmission .....	24
4.2 Slot Self-Allocation .....	25
4.3 <i>Interrupt packets</i> and Carrier Sensing .....	26
4.4 Third-party based Collision Detection and Resolution .....	27
4.5 Convergence .....	29
4.6 Zero-Exposure .....	30
4.7 Security Analysis .....	31
4.8 Evaluation .....	33
4.8.1 Functionality Validation .....	33
4.8.2 Convergence .....	37
4.9 Summary .....	41
Chapter 5: Zero Exposure Anonymous TDMA with <i>Shadow packets</i> (eZEA-TDMA) .....	43
5.1 Introduction .....	43

5.2	One-hop Information Dissemination.....	43
5.3	Slot Self-Allocation.....	44
5.4	<i>Shadow packets</i> for Two-Hop Information Dissemination.....	44
5.5	Pattern Based <i>Shadow packet</i> Scheduling.....	45
5.6	Collision Detection and Resolution.....	47
5.6.1	Collision Types.....	48
5.6.2	Detection of <i>regular-regular</i> collision .....	49
5.6.3	Collision Resolution .....	50
5.7	Allocation Convergence.....	51
5.8	Network Scalability and Network Dynamism .....	52
5.9	Evaluation.....	53
5.9.1	Functionality Validation .....	54
5.9.2	Performance Characterization .....	58
5.9.3	Power Consumption during Transience .....	67
5.10	Summary .....	75
Chapter 6:	Energy Aware ZEA-TDMA Design .....	76
6.1	Introduction .....	76
6.2	Sleep-Wake Scheduling .....	76
6.3	Protocol Description.....	77
6.4	Shared Wakeup Slots and Two-Way Handshake.....	80
6.5	Network Dynamics.....	82
6.6	Functional Limitations of the ZEA-TDMA-family .....	84
6.7	Evaluation.....	86
6.7.1	Without Wakeup Overhead .....	88
6.7.2	With Wakeup Overhead .....	91
6.8	Summary .....	93
Chapter 7:	Hardware System for Third-party Collision Detection.....	95
7.1	Introduction .....	95
7.2	Collision Detection.....	95
7.3	Collision Resolution.....	96
7.4	Collision Detection Sub-system .....	97
7.5	Summary .....	101
Chapter 8:	System Prototype for Privacy Preserving Channel Access.....	102
8.1	Introduction .....	102
8.2	Prototype Evaluation.....	102
8.2.1	Methodology.....	102
8.2.2	Functionality.....	103
8.2.3	Convergence characteristics .....	109
8.3	Summary .....	118
Chapter 9:	Conclusion and Future Work.....	119
9.1	Conclusion.....	119
9.2	Future Work .....	121

APPENDICES .....	123
APPENDIX A: Analysis of Collision Handling .....	124
APPENDIX B: Probability of collision between <i>regular packet</i> and <i>shadow packet</i> .....	130
APPENDIX C: Steady state power consumption model for eZEA-TDMA .....	132
APPENDIX D: List of Publications .....	137
REFERENCES .....	139

## LIST OF TABLES

Table 1: Baseline system parameters in simulation .....	33
Table 2: Network, Protocol and Simulation parameters .....	87
Table 3: List of parameters .....	124
Table 4: Symbols used for different parameters used in the analytical model .....	133

## LIST OF FIGURES

Figure 1. Two trust domains formed by two Body Area Networks .....	4
Figure 2. Pictorial summary of the investigated issues in this thesis .....	8
Figure 3. Scope of privacy preserving mechanisms with respect different layers of the network stack .....	17
Figure 4. Example network with multiple trust domains .....	20
Figure 5. (a) Local slot occupancy view to individual nodes (b) Slot occupancy view to a global observer .....	25
Figure 6. (a) Collided slot size (b) Random jitter and effective slot-size .....	27
Figure 7. Collision detection and resolution: (a) Neighbors of B, i.e. A and C are using overlapping slots for <i>regular packet</i> transmission. (b) When B detects the collision, it sends an <i>interrupt packet</i> immediately after the start of overlapping slots. (c) This causes C to delay its slot and settle on a non-overlapping slot within its 2-hop neighborhood.....	28
Figure 8. Functionality test for: (a) Static linear topology (b) State transition diagrams for topology in part (a).....	34
Figure 9. Functionality test for: (a) Dynamic network – two linear subnets joined through node 4 (b) State transition diagrams for topology in part (a) .....	36
Figure 10. Convergence characteristics for: (a) Linear (b) Grid, and (c) Fully-connected topology with varying F-ratio .....	37
Figure 11. Incremental node deployment convergence characteristics for: (a) 4x4 grid (b) 6x6 grid (c) 8x8 grid (d) 10x10 grid .....	41
Figure 12. Regular and <i>shadow packet</i> sending pattern .....	46
Figure 13. Types of collisions. (a) (i) and (ii) Shadow-shadow, (b) Shadow-regular, (c) Regular-regular .....	49
Figure 14. <i>Regular packet</i> collision pattern .....	50
Figure 15. Functionality tests for: (a) Static 4-node loop topology (b) Static 5-node fully-connected topology .....	55
Figure 16. Functionality test for dynamic 7-node linear topologies .....	57
Figure 17. Convergence characteristics for (a) Linear (b) Grid and (c) Fully-connected topology with F-ratio = 1 and 1.5 .....	60

Figure 18. Pattern termination time and Convergence time for (a) Linear (b) Grid and (c) Fully-connected topology with F-ratio = 1 and 1.5 .....	63
Figure 19. (a) Collision resolution time in frame counts (b) Collision resolution time in seconds .....	65
Figure 20. Network re-stabilization time required by a stable network when a new node is added .....	66
Figure 21. Optimal F-ratio for a fixed network size .....	67
Figure 22. Average transmitter and receiver power consumption for incremental node additions (a) 4x4 grid topology with F-Ratio = 1 (b) 4x4 grid topology with F-Ratio = 1.3 .....	69
Figure 23. Average transmitter and receiver power consumption for incremental node additions (a) 15 node fully connected topology with F-Ratio = 1 (b) 15 node fully connected topology with F-Ratio = 1.3 .....	70
Figure 24. Ratio of transient and steady state power consumption for (a) 4x4 grid topology with F-Ratio = 1 (b) 4x4 grid topology with F-Ratio = 1.3 .....	71
Figure 25. Ratio of transient and steady state power consumption for (a) 15 node fully connected topology with F-Ratio = 1 (b) 15 node fully connected topology with F-Ratio = 1.3 .....	72
Figure 26. Frequency distribution of transient and steady state power consumption ratio for (a) 4x4 grid topology with F-Ratio = 1 (b) 4x4 grid topology with F-Ratio = 1.3 .....	74
Figure 27. Frequency distribution of transient and steady state power consumption ratio for (a) 15 node fully connected topology with F-Ratio = 1 (b) 15 node fully connected topology with F-Ratio = 1.3 .....	75
Figure 28. Sleep-wake schedule followed by a node in a multi-domain network: node only wakes up to receive packets from neighbors belonging to the same domain. It goes to sleep when i) there is no transmission in the network ii) transmission slots in which it cannot decrypt or decipher the message .....	77
Figure 29. Two-way handshake for reliable wakeup .....	82
Figure 30. State machine for eZEA-TDMA with energy-efficiency module .....	84
Figure 31. Without wakeup overhead (a) Increasing traffic rate and constant burst size (b) Increasing burst size and constant traffic rate (c) Increasing burst size and decreasing traffic rate (d) Increasing burst size and increasing traffic rate .....	90
Figure 32. With wakeup overhead (a) Increasing traffic rate and constant burst size (b) Increasing burst size and constant traffic rate (c) Increasing burst size and decreasing traffic rate (d) Increasing burst size and increasing traffic rate .....	92

Figure 33. Hardware module using a simple threshold based mechanism to detect the duration of the received signal .....	96
Figure 34. Collision signal – only visible information: Oscilloscope screen-capture of RSSI ( $-50.0 \times V_{RSSI} - 45.5$ [dBm]) power at a receiver for (a) Transmission signal for 512 bit packet, (b) Collision between two 512 bit packets where the second transmitter has higher relative power at receiver (c) Unresolved collision across multiple frames (Slot: 30ms, TDMA Frame: 130ms) .....	98
Figure 35. Collision detection subsystem: hardware for third-party collision detection by observing transmission signal duration .....	100
Figure 36. Power spectral density (PSD) of the RSSI signal .....	101
Figure 37. Slot self-allocation in a single-hop network (Oscilloscope screen-capture) .....	104
Figure 38. Allocation dynamics observed by a passive listener in a multi-hop network for (a) Three-node linear topology with $T = 120$ ms (F-ratio = 1.33) (b) Three-node linear topology with $T = 90$ ms (F-ratio = 1) (c) Four-node linear topology with $T=150$ ms (F-ratio = 1.15) – overlapping transmissions from A and D is legal .....	105
Figure 39. Five-node (topology 1) F-ratio=1.15 (a) Timing diagram using RSSI traces from individual nodes (b) RSSI trace observed at a passive listener .....	106
Figure 40. Five-node (topology 1) F-ratio=1 (a) Pictorial representation of slot location of nodes (b) RSSI recorded at common sniffer using oscilloscope .....	107
Figure 41. Five-node (topology 2) F-ratio=1 (a) Pictorial rep. of slot location of nodes (b) RSSI recorded at common sniffer using oscilloscope .....	109
Figure 42. Experimental topologies for wireless network test-bed .....	109
Figure 43. Indoor laboratory setup of wireless network test-bed .....	110
Figure 44. Average convergence latency .....	111
Figure 45. Distribution of convergence latency depicting the effect of frame size on allocation convergence: (a) F-ratio= 1 (b) F-ratio = 1.15 (c) F-ratio = 1.3 (d) F-ratio = 1.5 (e) F-ratio = 2 .....	112
Figure 46. Variation in convergence latency and average convergence times: (a) Topology 1 (b) Topology 2 (c) Topology 3 (d) Topology 4 (e) Topology 5 (f) Topology 6 .....	115
Figure 47. Convergence latency for incremental node addition (a) Node addition order 1 (b) Node addition order 2 .....	117
Figure 48. Wake up and transmission-reception process for a node with mean traffic rate $\lambda=0.375$ , burst size $\beta=3$ , and $\delta=2$ .....	133

## LIST OF ALGORITHMS

Algorithm 1: Slot allocation algorithm for ZEA-TDMA .....	31
Algorithm 2: Sender and Receiver-side pseudo-code for message transmission .....	80

# Chapter 1: Introduction

## 1.1 Background and Motivation

Due to its shared nature, wireless networks are susceptible to passive eavesdropping. Adversaries can remain largely undetected and overhear messages ‘flying in the air’ to deduce information about the overall network, specific nodes in the network or the relationship between different network nodes. As a result, a critical issue in wireless ad-hoc networks is the threat to the privacy of participating network nodes.

There exist two major categories by which privacy of nodes in a wireless network can be exposed: (i) *Linkability by identity exposure*, and (ii) *Traffic analysis*. The first vulnerability stems from the fact that most of the traditional encryption schemes and their variants provide end-to-end encryption at a higher layer which protects the message content, but keeps identifier-based information exposed. For example, medium access control (MAC) addresses are sent in the clear for a majority of the current encryption mechanisms. As a result, it can be easily used by an adversary to identify users and track their locations, link source and destination pairs, classify relationships between nodes, and mark crucial nodes in the network.

The second vulnerability, which is traffic analysis, involves the statistical analysis and correlation of identifiable network traffic features to deduce communication characteristics and key relationships among nodes. This can be done even when packets are encrypted and can be used to indirectly identify individual network participants. Extractable traffic features include packet size, packet content, frequency of packets, packet inter-arrival times, packet arrival rate, and packet delay characteristics.

## 1.2 Current Solutions

A number of solutions have been proposed to address the privacy concerns in wireless networks. *Linkability* can be prevented to some extent by hiding low-level node-identifiers using encryption mechanisms [1], [2]. There has also been a few solutions proposed which involves dynamic modification of identifiers using pseudonym based mechanisms [3], which can also deter the node tracking capabilities of the attacker. Although such mechanisms prevent the direct exposure of node-identifiers, an attacker can still make use of the traffic pattern to perform statistical traffic analysis.

Solutions to prevent traffic analysis primarily involve in modifying the traffic pattern. This is done by introducing random delays at intermediate nodes, mixing the order of packets at forwarding nodes, injecting dummy packets into the normal traffic flow, or by altering the traffic pattern using network coding. However, each of these mechanisms require complete coordination and pre-formed trust between coexisting nodes in the network, which may not always be feasible.

Providing privacy at the MAC layer addresses concerns regarding lower-layer address visibility issues. Apart from hiding MAC layer related identity information, MAC anonymity techniques have another crucial advantage. They are transparent to higher layer protocols and do not interfere with them. Essentially, such anonymity protocols could be programmed in the network interface and can be independent and completely decoupled from any higher layer applications. Additionally, if such a mechanism can also prevent traffic analysis, it provides the entire privacy preservation package by preventing identity exposure as well as traffic analysis.

## 1.3 Proposed Mechanism

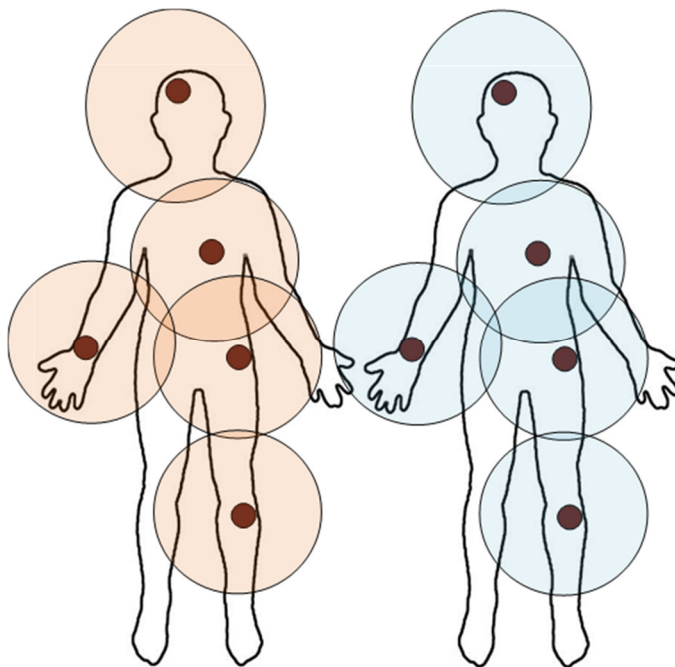
In this thesis we develop a distributed TDMA (Time Division Multiple Access) mechanism which provides privacy preservation in wireless multi-hop ad-hoc networks by protecting against

both *identity exposure* and *traffic analysis*. The developed protocol introduces a *zero exposure* paradigm in which explicit information interchange among nodes is prohibited in the network. Network nodes use “*blindfolded*” packet transmissions where no part of the messages transmitted by one node can be “*seen*” or deciphered by any other node in the network, i.e. a *zero exposure* environment. Nodes can only “*hear*” the presence of messages through sensing the channel. The protocol uses such “*blindfolded*” channel sensing to infer timing of transmitted packets and uses such timing information for TDMA slot self-scheduling. The advantage of such a *zero exposure* environment is that it automatically prohibits the *identity exposure* of nodes in the network. Additionally, since all nodes in a TDMA network send packets with the same periodicity, it is naturally more difficult [4], [5] for an adversary to perform *traffic analysis* as it does not provide a direct correlation between extractable traffic features. Furthermore, the concealment of identity of nodes also prevent the analysis of traffic features as it makes it difficult for traffic segregation in a broadcast one-hop neighborhood and adversaries need to rely on physical layer based mechanisms only to identify an individual traffic flow.

## 1.4 Applications

The proposed privacy preserving mechanism can have applications in a number of military-based, civilian-based and enterprise-based networks. For example, Body Area Networks (BANs) which consist of various health monitoring devices with wireless capabilities kept in close vicinity of the body, and, optionally, implanted in the body, can make use of such a privacy preserving MAC-layer protocol. In a BAN, wireless nodes form a personal wireless network exchanging sensitive patient information and often reporting these to a healthcare service provider using a secure connection through a device like the smartphone. While the wireless nodes from a large number of such BANs (i.e., one per patient) may co-exist from a wireless channel access

standpoint, the information privacy from each individual BAN, as well as the identity of the nodes may need to be protected, since it poses a direct privacy threat to the users. In that case, the nodes within each BAN may form a trust domain and follow a *zero exposure* standard towards nodes belonging to a different trust domain. This means that no information, including any identity related information, is shared across trust domains. As shown in Figure 1, when those two different BANs share the same wireless channel and come within communication range, then all the sensors in those BANs may need to settle on a set of collision-free TDMA slots for the MAC purposes without any form of message-content based coordination.



*Figure 1. Two trust domains formed by two Body Area Networks*

The proposed TDMA-based privacy preserving mechanism can also be useful in Internet of Things applications. Internet of Things (IoT) is envisioned as the future of the Internet with a world-wide network of connected objects embedded within the society and infrastructure [6]. Wireless IoTs in the form of heterogeneous devices, sensors, actuators, computers and smart

objects are expected to coexist and collaborate to develop applications towards the improvement of human life. Among many operational challenges [7], [8], privacy and security [9], [10] have been identified as a major issue towards developing successful IoT applications. In contrast to traditional networking settings, wireless IoT privacy is a particularly important one because of: a) their socially embedded nature, and b) the fact that many applications involve multi-party IoT deployment which is particularly vulnerable to various forms of privacy attacks.

As an IoT application example, consider a public place (e.g., a hospital, airport, large factory floor etc.) being monitored by a large set of IoT devices provided by different service providers. For instance, while the security company such as ADT may provide security relayed monitoring and actuation services using a set of IoT nodes, another service provider may provide air-quality and HVAC related monitoring and actuation services using another set of smart devices. Yet, another company may be responsible for fire detection and evacuation management using appropriate IoT devices. Since different service providers may want to maintain privacy of their own sensing and actuation data, each such set of nodes in this example will form a separate trust domain. The same set of privacy-aware wireless access issues as discussed in the BANs above are applicable in this case as well.

Similar issues also exist in multi-party crowd-sensing [11] applications, where inter-trust-domain privacy preservation is needed. In general, whenever multiple private wireless IoT networks, running TDMA MAC, attempt to share channel without exposing node identification for preventing traffic analysis attacks, the proposed solution becomes relevant. Also, since for a large number of proposed IoT based systems [12], [13] the wireless network is fairly static, TDMA provides channel access with very low capacity and energy loss due to collisions which are prevalent in CSMA family of MAC protocols.

## 1.5 Challenges

Despite the inherent advantages, there are a number of challenges involved in developing a privacy preserving TDMA protocol. Although TDMA slot allocation can be simply managed by a centralized access point, such centralized administration is not usually applicable for dynamic and ad hoc distributed networks. As a result, majority of the TDMA protocols for sensor and ad hoc networks in the literature have focused on distributed slot allocation solutions which rely on in-band [14], [15] or out-of-band [16]–[18] control mechanisms for slot-allocation. Such control mechanisms require network nodes to exchange their occupied slot information so that they can individually adjust their slot occupancy based on a distributed allocation algorithm. Such information interchange directly leads to *identity exposure*, and may not be desirable in privacy-sensitive networks where network nodes may be needed to remain anonymous to other untrusted nodes due to application requirements. This may imply that any identity-related information of a node, including MAC and other network-level addressing information remain unknown to its untrustworthy neighbors with whom it shares the same channel. We term such strict anonymity as a *zero exposure* environment.

Slot allocation in TDMA protocols for multi-hop wireless networks need to satisfy the constraint that at steady state, no two one-hop or two-hop neighbors can have partially or completely overlapping transmission slots. Overlap between slots of one-hop neighbors cause direct collisions and overlap between slots of two-hop neighbors can cause hidden collisions. In a *zero exposure* environment, exchanging slot occupancy information among one-hop or two-hop neighbors may be extremely complex or infeasible due to anonymity restrictions. Therefore, the primary challenge for *zero exposure* distributed slot allocation is the unavailability of any explicit means to disseminate the slot occupancy information within up to two-hop wireless

neighborhoods. Additional challenges stems from the fact that such a *zero exposure* distributed environment may lack any means of achieving a network-wide time synchronization, without which the task of cross-node slot alignment is non-trivial [19].

## 1.6 Dissertation Objectives

The main objective of this thesis is to design and develop a MAC-layer privacy preserving mechanism for wireless networks. *Identity exposure* is prevented by encrypting packets across all network layers including the MAC layer addresses. As a result, no identity revealing information, such as node IDs, network or MAC layer addresses are exposed. *Traffic analysis* is deterred by achieving TDMA slot-allocation without exchanging any explicit control information, message, or header contents. Network nodes use “*blindfolded*” packet transmissions where no part of the messages transmitted by one node can be “seen” or deciphered by any other node in the network, i.e. a *zero exposure* environment. Nodes can only “hear” the presence of messages through sensing the channel. The main idea of this approach is to use such “*blindfolded*” channel sensing to infer timing of transmitted packets and use such timing information for TDMA slot self-scheduling.

A pictorial summary of the investigated issues in this thesis is shown in Figure 2. Chapter 2 discusses the related work in this area. We first review privacy preserving solutions used in wired networks and explain why they are non-optimal for wireless environments. Next we discuss some wireless privacy preserving mechanisms proposed in the literature which are implemented at network layers above the MAC-layer. Then we point out some MAC-based strategies for achieving privacy and highlight the differences and advantages of our proposed scheme. Finally, we provide a brief survey on wireless TDMA protocols and discuss the challenges involved in the development of a distributed TDMA for wireless networks with the main goal of achieving privacy.

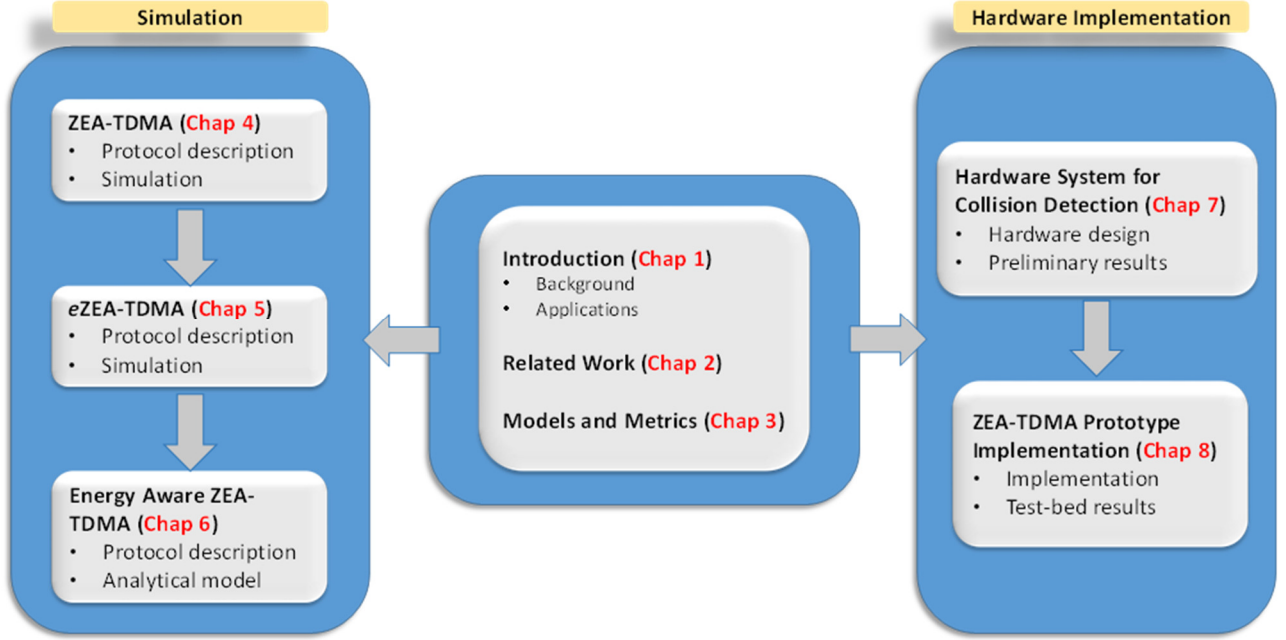


Figure 2. Pictorial summary of the investigated issues in this thesis

In Chapter 3, we provide the models and metrics that are used in the protocol development and evaluation. Here we introduce trust domains, outline the different models to be used in the protocol and state the limitations of the developed system.

Chapter 4 develops the *Zero Exposure Anonymous TDMA* (ZEA-TDMA) protocol. In this chapter, we introduce our MAC-based privacy preservation scheme and simulate the protocol using NS-2. In chapter 5, we develop a variant of ZEA-TDMA, *eZEA-TDMA*, which provides energy-efficiency at the cost of added complexity in implementation. NS-2 simulation results for *e-ZEA-TDMA* are also presented in this chapter. Chapter 6 presents an energy aware implementation which uses a sleep-wake schedule for idle energy savings. An analytical model for the energy module is also presented in this chapter.

In chapter 7, we develop the wireless collision detection hardware system and provide preliminary results on real test-bed scenarios. Next, in Chapter 8, we implement our scheme on a

wireless test-bed using hardware augmented with the developed wireless collision detection circuitry.

Finally, in chapter 9, we summarize the thesis and compile a list of future work on this topic.

## Chapter 2: Related Work

### 2.1 Privacy Preserving Systems

Multiple solutions have been proposed to counter traffic analysis. In [20], David Chaum proposed the idea of MIXes to hide correspondence between message senders and receivers to protect communication privacy and fight against traffic analysis attacks. The main idea behind MIXes is to cache incoming messages, divide them into fixed-sized chunks, perform cryptographic operations on those messages and then forward them to the next destination in an unpredictable order, making it difficult for statistical analysis. There have been multiple anonymous communication systems for wired networks proposed based on the idea of MIXes, like [21]–[23]. Onion routing [24] is a MIX-based mechanism which provides sender, receiver and communication anonymity using intermediate store-and-forward devices called onion routers. The main disadvantages of MIXes include caching of messages before forwarding them in a random order, which can cause unpredictable delays and no guarantees to quality of service; and decryption and re-encryption of messages, mostly using public key cryptographic operations, which can be computationally expensive.

Moreover, the above approaches mostly require static configuration as they are designed for wired networks and do not consider the broadcast and dynamic nature of wireless medium. Traffic padding and traffic morphing are some wired network traffic analysis prevention approaches which can be applied to wireless networks with lesser complexity. In [25], the authors propose a traffic padding scheme where nodes in a circuit inject dummy traffic between each other according to a probability distribution over packet inter-arrival times. One disadvantage of such an approach is that it introduces the overhead of additional dummy traffic. The authors in [26] propose a traffic morphing scheme which aims to thwart traffic analysis by modifying one class of traffic to look

like another class. As a result, it reduces the accuracy of traffic classification and incurs less overhead than traffic padding approaches.

Several onion routing based schemes have also been proposed for privacy in wireless networks. The main idea behind these routing layer anonymity schemes is to perform layer upon layer of encryption on the message such that the intermediate routing nodes remain unaware of the source and destination nodes. SDAR [27] and ARM [28] use onion based mechanism and a two-phase route discovery approach to achieve anonymous routing in wireless networks. In SDAR, the authors propose a secure distributed anonymous routing protocol for MANETs, which creates routes anonymously and dynamically without the sender being aware of either the intermediate nodes or the network topology. And, ARM provides resistance against powerful adversaries who can either have a global view of the network or can be part of the network. It achieves efficiency by using a combination of pseudonyms and symmetric keys, instead of using multiple public key encryptions and decryptions. It also provides resistance to traffic flow attacks by incorporating random padding and forwarding of packets even when nodes receive packets they are not required to forward. ODAR [29] is another anonymous routing protocol which achieves source and destination anonymity by using pseudonyms and achieves intermediate node anonymity by using Bloom filters. The usage of Bloom filters also provides an efficient storage mechanism for the source route.

Network coding is another approach to combat traffic analysis[30], [31]. In network coding, each message sent on a node's output link is some function or mixture of messages that arrived earlier on the node's input links. Messages routed through the network are encoded multiple times, until they reach the destination where they are decoded. This can prevent packet flow-correlation and message content correlation attacks, since incoming packets are uncorrelated to outgoing

packets. Priv-code [5] is a network coding based privacy preserving scheme for multi-hop wireless networks. In [5], the authors create a hypergraph-based network coding model for wireless networks and formalize an optimization problem whose objective function is to make each node have identical transmission rate. A decentralized algorithm for this optimization problem is provided which schedules an end-to-end unicast session over multiple paths. An information theoretic metric for privacy measurement using entropy is also developed to evaluate the performance of the approach. Due to the benefits of network coding, Priv-Code provides better network performance while achieving stronger privacy protection than the mix system.

## 2.2 MAC-layer and Privacy Preservation

The aforementioned wireless network anonymity techniques are implemented at higher layers. Privacy preserving schemes implemented at a lower layer, like the MAC layer, has several advantages. First, a MAC layer privacy scheme can prevent all forms of *identity exposure*, starting from the MAC layer and up. This will prevent identity-based *linkability* and reduces the chances of traffic correlation, flow tracing and user tracking. Second, a MAC layer privacy scheme will not interfere with higher layer protocols, which will not be needed to be modified to provide privacy. Essentially, such anonymity protocols could be programmed in the network interface and can be independent and completely decoupled from any higher layer applications.

Contention-free TDMA [16], [32] and contention based CSMA [33], [34] are the two broad categories of MAC protocols which are used for wireless networks. The contention-free nature and guaranteed fairness characteristics of TDMA protocols give them an advantage over CSMA protocols in terms of network performance in highly dense networks with real-time constraints on message latency. In primarily static networks, TDMA also provides additional energy benefits [35], [36] by allowing them to sleep and wake based on the duty cycle of the underlying data.

Unlike CSMA, however, the TDMA protocols suffer from the complexity and overhead of scheduling and generally need global topological information for setting the TDMA frame size a priori.

In terms of privacy, since CSMA-based access does not require any node-ID information to be exchanged among the participating nodes, they can access the channel even when their IDs are encrypted [1], [2] at all layers including at the MAC layer. This ensures full protection from identity exposure during channel access. Note that a link establishment between two nodes, however, requires the receiver to be able to read the MAC layer IDs of a packet to receive and process it. When the transmitter and the receiver belong to the same trust domain (i.e., in which mutual identity exposure is allowed), those nodes can share the MAC layer encryption key. This way, identity can be concealed across different trust domains for CSMA channel access, while links can be established within trust domains (see Figure 3). There are a few CSMA-based MAC layer anonymity schemes proposed in the literature. For example, in [37] a traffic reshaping algorithm is proposed to protect users' online privacy. It creates multiple virtual MAC interfaces, dynamically dispatches traffic flows among these interfaces, and reshapes different traffic features on each virtual interface to hide those of the original traffic. Since traffic reshaping does not use dummy packet padding, it thwarts traffic analysis without additional overhead for noise traffic. [2] provides a wireless link layer solution by replacing the interface identifiers with different hash values in a reverse one-way hash chain. By preventing the exposure of MAC addresses, it attempts to prevent attacks such as flow tracing, user tracking, and traffic pattern analysis. [38] and [39] both propose an anonymous 802.11 scheme for wireless ad-hoc networks. In [38] the authors claim that virtual carrier sensing in 802.11 undermines the anonymity of the wireless system as it makes the frame transmission sequence predictable, and propose a solution to hide the 802.11 point-to-

point communication relation. To break the predictability of the virtual carrier sensing, they use a probabilistic CTS and ACK sending scheme where all nodes receiving an RTS sends a CTS with a certain probability. The same probability applies to sending dummy positive ACK packets as well. [39] presents an anonymous MAC protocol which aims to provide receiver anonymity and reliability. Receiver anonymity is achieved with link encryption and broadcasting of data frames, whereas reliability is achieved by a selective repeat retransmission scheme, combined with a polling mechanism.

The access protocol IEEE 802.15.4 [40, p. 4] has been recently adopted for many IoT style peer-to-peer wireless networks. Since it belongs primarily to the CSMA family, it inherits the QoS and energy issues as pertaining to other CSMA protocols. Additionally, it does not assume a pure peer-to-peer distributed topology and requires full function devices (FFDs) and PAN coordinators to coordinate the operation among reduced function devices (RFDs). It also uses a super-frame structure consisting of a network beacon followed by contention access period and contention free periods for nodes requiring guaranteed bandwidth. Even when the MAC layer packets are encrypted, this sending pattern can be predictable and may divulge information via traffic analysis.

In contrast, a TDMA-based MAC protocol can be made to transmit with strict periodicity under high load, or with traffic padding under low-load conditions. Such inherent periodicity makes traffic features indistinguishable and provides defense against traffic analysis attacks [4], [5]. This eliminates the need for other forms of traffic analysis defenses like mixing [41], [42], traffic padding [25], and packet dropping [43]. This feature is not available in asynchronous CSMA based protocols. However, networks running TDMA MAC do not provide any protection from identity exposure. This is because during the TDMA slot scheduling phase, nodes require to exchange their identity, like the MAC layer ID information, as in-band [14], [15] or out-of-band [16], [17] control

information. Such information exchange is needed across all participating nodes (i.e., sharing the TDMA frame) even when they belong to different trust domains. In other words, nodes from different trust domains need to expose their identity as long as they want to be allocated slots within a shared TDMA frame. None of the existing distributed TDMA slot allocation strategies in the literature [14]–[17] is able to resolve this limitation.

## **2.3 TDMA Protocols**

Several TDMA protocols have been proposed in the literature. Classical centralized TDMA mechanisms face the roadblock of non-scalability and inefficiency for large ad hoc and infrastructure-less sensor networks. To address the scalability issue, Funneling-MAC [44] uses TDMA in high-traffic parts of a network, and CSMA in the low-traffic parts. This hybrid protocol also mitigates the funneling effect in high-data rate sensor networks where the nodes nearest to the sink often experience high congestion since all data to the sink is forwarded through them. However, this protocol relies on a central sink to coordinate the TDMA slot allocation. Any centralized protocol suffers from single point of failure weakness. Another hybrid design, Sensor-MAC (SMAC) [45] uses wake-sleep cycles in the presence of random channel access. The nodes contend for channel access during their wake periods, like IEEE 802.11, and they turn their radio off during sleep periods. The nodes form virtual clusters and all nodes within the same cluster maintain the same sleep-wake schedule. However, the protocol uses a static sleep interval with predetermined duty cycles. Moreover, the nodes are susceptible to collisions and the resulting energy inefficiency during their awake periods. TreeMAC [15] and TDMA-W [16] are examples of non-centralized energy-efficient TDMA MAC protocols. However they are dependent on time-synchronization which is a non-trivial task for distributed ad-hoc networks. Z-MAC [46] works in the presence of loose time-synchronization, but it still needs global time-synchronization at startup.

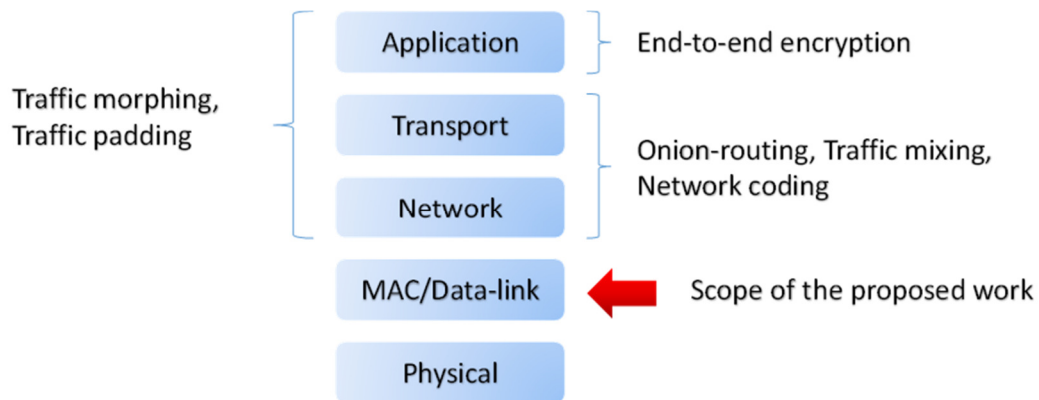
Another category of TDMA protocols are fully distributed and work in the absence of time synchronization. ISOMAC [14] and DRAND [17] are examples of such distributed TDMA protocols which work independent of network-wide time synchronization and are ideal for completely distributed ad hoc networks. These approaches rely on control packets to forward two-hop neighbors' slot allocation information in the network. Control packets usually identify the nodes sending the packet and contain slot allocation information of neighborhood nodes along with their identity information. For example, ISOMAC provides an in-band control mechanism for exchanging TDMA slot information with distributed MAC scheduling. A fixed-length bitmap vector is used in each packet header for exchanging relative slot timing information across immediate and up to two-hop neighbors. Existing distributed TDMA slot allocation mechanisms presented in the literature rely on information interchange among nodes which include identity information, thus violating anonymity requirements.

## 2.4 Scope of Thesis

The work done in this thesis aims to develop a distributed TDMA slot allocation mechanism that by-passes the requirement of any in-band or out-of-band information interchange for slot allocation across nodes both within and across trust domains. Unlike the existing TDMA-based solutions [14]–[17], [35], [44], [46], [47], the proposed scheme does not depend on any message-based coordination tactics to achieve TDMA slot assignments, which means that nodes do not need to explicitly share any information during the scheduling phase. To prevent inter-trust domain exposure of data or identity, we assume that entire packets, including the MAC header, are encrypted using an encryption key pre-shared among nodes belonging to the same trust domain. This will prevent any inter-trust domain information leakage during the entire network life-cycle. However, once a MAC schedule is established, nodes within a trust domain (e.g., BAN-1 in Figure

1) can exchange data by the way of using a trust-domain-specific shared key which is used for opening up the packet at the intended receiver. It uses the implicit information about time of arrival of packets and a novel wireless collision resolution mechanism for distributed TDMA with *zero identity exposure*. As a result, it provides (i) the positive aspects of a MAC-layer privacy preserving system, (ii) resistance to *traffic analysis* by virtue of its inherent TDMA-like periodic traffic pattern, and (iii) protection against *identity exposure*, which is a disadvantage of TDMA protocols.

The main scope of the work is to provide anonymity and privacy at and above the MAC-layer by preventing identity exposure and thwarting traffic analysis. The proposed mechanism does not defend against location-based attacks where an adversary tries to locate a transmitting a node based on physical layer characteristics such as signal strength [48], [49]. Figure 3 gives an overview of the scope of different privacy-preserving mechanisms discussed in the related work, and the network layer(s) at which they can implemented. It is inferential that a given mechanism provides privacy to all network layers at and above the layer to which it applies to.



*Figure 3. Scope of privacy preserving mechanisms with respect different layers of the network stack*

## Chapter 3: Network and Threat Model

### 3.1 Introduction

In this chapter, we present the models and metrics that have been used for the protocol development and subsequent evaluation. First, we present the connectivity model and provide a novel trust-domain based network structure to introduce the baseline concepts of privacy-aware channel access. Next, we discuss the threat model which analyzes the plausible threats to the system. Finally, we provide a privacy metric with a theoretical discussion on the effectiveness of a TDMA protocol in terms of privacy, compared to CSMA-based MAC protocols.

### 3.2 Network Model

A multi-hop mesh network can be modeled as a bidirectional graph  $\mathcal{G}(\mathcal{N}, E)$ , where  $\mathcal{N}$  represents the set of vertices (nodes)  $v_1, v_2, \dots, v_N$ , with  $N = |\mathcal{N}|$ , and  $E$  is the set of links between nodes.  $v_i$  and  $v_j$  are said to be one-hop neighbors if there exists a link  $e_{ij} \in E$ , i.e.,  $v_j$  is within transmission range of  $v_i$ . Since the graph is undirected,  $e_{ij} \in E$  implies  $e_{ji} \in E \forall i, j$ . It is also assumed that all nodes in  $\mathcal{N}$  transmit with a fixed transmission power. The maximum one-hop node degree of  $\mathcal{G}$  is represented by  $d$  such that for any given node  $v_i$ , the maximum number of links is less than or equal to  $d$ . For a node  $v_i$ , the two-hop node degree is  $|D^i|$ , where  $D^i$  is the set of all two-hop neighbors of  $v_i$  and is given by  $\sum e_{ij} + \sum e_{jk} \forall v_j, v_k \in \mathcal{N}$  such that  $e_{ij}, e_{jk} \in E$  and  $e_{ik} \notin E$ . The maximum two-hop node degree for  $\mathcal{G}$  is  $\max\{|D^i| \mid \forall v_i \in \mathcal{N}\}$ , and is represented as  $D = d^2$ .

We will use a TDMA-frame structure where time is divided into individual frames, and each frame is divided into  $n$  slots of size  $\tau$ , which is the time required to transmit a packet. The TDMA frame size  $T = n\tau$  is pre-decided for a given network and each node  $v_i$  gets one transmission

opportunity every TDMA frame in its assigned slot. We define frame ratio (i.e., F-ratio) as the ratio of the number of slots in a TDMA frame and the minimum number of slots required for all nodes in the network to have assigned slots. The F-ratio depends on the maximum two-hop node degree  $D$  of the network, i.e.  $\text{F-ratio} = \frac{n}{D+1}$ . The minimum number of slots required for the network sustainability is  $D+1$  since for the node with two-hop node degree  $= D$ , it would require  $D+1$  slots for all nodes in its two-hops neighborhood, including itself, to have non-overlapping slots.

All packets are assumed to be of fixed size. This is a reasonable assumption since larger packets can be fragmented and smaller packets can be padded to meet the specifications.

### 3.3 Trust Domain

Nodes are assumed to be partitioned into multiple co-existing trust domains. Scope of identity exposure is assumed to be only within a trust domain. We define the trust relationship between  $v_i$  and  $v_j$  as the binary operator  $\diamond$  such that  $v_i \diamond v_j = 1$  when nodes  $v_i$  and  $v_j$  trust each other and can reveal their identities to each other. A trust domain  $TD_p = \{v_1, \dots, v_m\}$  is defined as the set of all nodes such that if  $v_i, v_j \in TD_p$ , then  $v_i \diamond v_j = 1 \forall v_i, v_j$ . The presented protocol does not require higher layer cooperation from the nodes since it can be programmed in the network interface and can be independent of the higher layer applications. Protection against identity exposure can be provided by encrypting entire packets such that two nodes  $v_i \in TD_p$  and  $v_j \in TD_q$  can only decrypt each other's packets iff  $p = q$ . Any general purpose light-weight shared key encryption can suffice.

As for multi-hop routing, it is assumed that an end-to-end flow is contained within a trust domain. Meaning, the source, the destination, and the intermediate nodes of a flow are within the same trust domain, and therefore they can establish pair-wise links by using a pre-allocated shared

key for that particular trust domain. Nodes from different trust domains, however, do not share any key, and therefore cannot form data links although they are able to participate in the TDMA operation using our proposed method. This way, nodes from multiple trust domains can share the same channel using a TDMA frame, while an individual flow can only be contained within a trust domain.

The situation is explained using the network in Figure 4. Consider the 9-node example network in which nodes 1, 2, 3, 5 and 9 belong to trust domain 1, and nodes 4, 6, 7 and 8 belong to trust domain 2. According to the proposed network model, all 9 nodes share the same channel as shown by the example TDMA slot allocation arrangement in the figure. Observe that the flows in the network are always contained within the same trust domain. For TDMA slot-allocation purposes, we make the worst case assumption that all nodes belong to different trust domains and no information is available via packets.

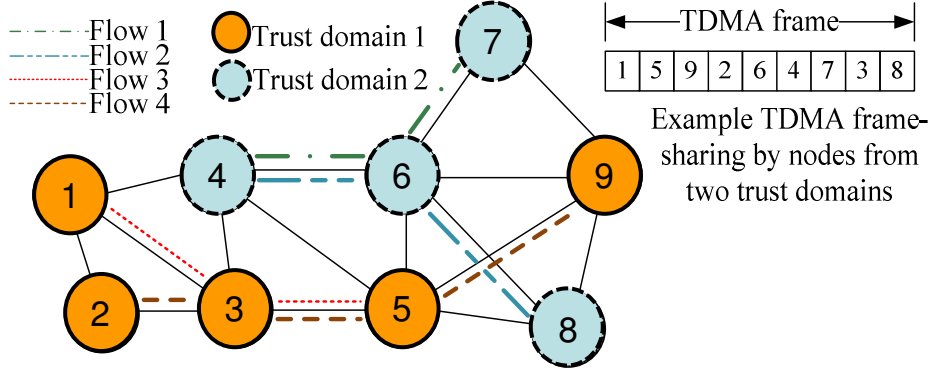


Figure 4. Example network with multiple trust domains

### 3.4 Threat Model

We observe that there can be passive adversaries in the network who are capable of monitoring network traffic and remain undetected. Passive identity snooping is the primary threat that the proposed system attempts to thwart. The main objective of an adversary is to discover

communication relationships in the network by tracing packets from its source to destination. The nodes in a trust domain can assume the role of an adversary, either individually or as a colluding set, and attempt to snoop identity of nodes in other trust domains. This means that an adversary can be capable of monitoring the incoming and outgoing packets of each node in the network. Finally, we assume that the adversary is incapable of deciphering packets within a finite amount of time to reveal the sender or receiver identities.

A packet sent out by a node  $x_i$  can be represented by the tuple  $\{x_i, t, E\}$ , where  $t$  is the transmission time, and  $E$  is the encrypted packet. The only useful information available to the adversary is  $\{t\}$ . Using this information, the adversary can measure the inter-packet intervals and try to find a correlation between the output links at different nodes. By correlating the flows at different nodes, the adversary can determine the routes of specific traffic flows, estimate the source and destination of packets and can discover relationships and functions of different nodes in the network. The traffic pattern can also act as upper-layer side channels and can be correlated to application layer protocols run by nodes [37]. The overall threat objective is to perform cross-trust-domain node-profiling, link layer topology estimation, node-tracking, and flow-tracking, which needs to be prevented.

### 3.5 Privacy Metric

We adopt the privacy metric used in [25], [43]. As pointed out by Levine et al [43], the inter-packet interval between packet  $i$  and  $i + 1$ ,  $\delta_i$ , is a random variable which is highly sensitive to dropped packets. Hence, we use random variable representing the count of the number of outgoing packets at a node over a non-overlapping time-window, as it is less sensitive to packet drops. The adversary correlates the packet counts on two outgoing links and decides that the two nodes (whose outgoing links are being monitored) carry the same flow if the correlation coefficient exceeds some

threshold. Privacy preservation Index (*PPI*) is the privacy metric that will be used to measure the quality of privacy provided by our protocol.

The adversary's observation time is the total time for which the attacker observes the two nodes  $x_i$  and  $x_j$  and is defined as  $\theta_{i,j}$ . The observation time is divided into fixed-sized non-overlapping windows  $W_k$  such that  $\theta_{i,j} = \sum_{k=1}^n W_k$ . For each node  $x_i$ , the adversary counts the number of packets  $p_k^{x_i}$  in a given window  $W_k$ . The correlation coefficient between two flows  $r(d)_{x_i x_j}$  is given by:

$$r(d)_{x_i x_j} = \frac{\sum_k ((p_k^{x_i} - \mu^{x_i})(p_{k+d}^{x_j} - \mu^{x_j}))}{\sqrt{\sum_k (p_k^{x_i} - \mu^{x_i})^2} \sqrt{\sum_k (p_{k+d}^{x_j} - \mu^{x_j})^2}} \quad (1)$$

where the delay  $d = 0$  and  $\mu^{x_i}$  and  $\mu^{x_j}$  are the means for the two sequences. The adversary determines that the two outgoing links carry the same flow if  $r(d)_{x_i x_j}$  exceeds some threshold  $\alpha$ .

We define this correlation as  $x_i \sim x_j$  if  $r(d)_{x_i x_j} \geq \alpha$ . Additionally,  $x_i \not\sim x_j$  if  $r(d)_{x_i x_j} < \alpha$ . Also,

if two outgoing links actually carry the same flow, we say  $x_i = x_j$ , otherwise  $x_i \neq x_j$ . Given  $x_i$ ,

let  $\mathcal{A} = \{x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_m\}$  be set of all nodes  $x_j$  for which the adversary calculates  $r(d)_{x_i x_j}$ . Essentially  $\mathcal{A}$  consists of all nodes which the adversary suspects to carry the same flow as  $x_i$ . We define  $\mathcal{C} = \{x_j | x_i \sim x_j \wedge x_i \neq x_j, x_j \in \mathcal{A}\}$  and  $\mathcal{U} = \{x_j | x_i \not\sim x_j \wedge x_i = x_j, x_j \in \mathcal{A}\}$ . The

*false positive* rate, defined as the erroneous detection that two unrelated nodes carry the same flow,

is given by  $\frac{|\mathcal{C}|}{|\mathcal{A}|}$ . Similarly the *false negative* rate, defined as the erroneous detection that two

nodes are unrelated even though they carry the same flow, is given by  $\frac{|\mathcal{U}|}{|\mathcal{A}|}$ . A high value of  $\alpha$

increases *fpr* and decreases *fnr*, while a low  $\alpha$  has the opposite effect. The adversary chooses  $\alpha$

such that  $fpr = fnr$ . The privacy preservation index is given by  $PPI = 2fpr = 2fnr$ . A low

$PPI$  would indicate that the  $fpr$  and  $fnr$  for the adversary is low, which indicates that the defense against traffic analysis is ineffective. Similarly, high  $PPI$  indicates that the defense is effective. Theoretically, the worst value for  $fpr$  and  $fnr$  from the perspective of an adversary is 0.5 which is the same as random guess.

Since the proposed TDMA-based privacy preserving solution makes all nodes transmit packets at the same rate, for any given  $x_i$  we have  $p_k^{x_i} = p_k^{x_j} \forall (x_i, x_j), x_j \in \mathcal{A}, k \in \{1, 2, \dots, n\}$ . Consequently,  $r(d)_{x_i x_j} = 1 \forall x_j \in \mathcal{A}$ . This implies that theoretically for any value of  $\alpha$ ,  $PPI \geq 1$ . So, the TDMA-based solution presented here can provide better privacy than random guessing, which was assumed to be the best in [25], [43].

### 3.6 Summary

The trust-domain based network model provides an overview of the interaction between different nodes in the network and information leakage across trust-domains is prevented using encrypted packets. The primary threat to the system is from an individual or colluding adversaries whose goal is to analyze traffic patterns by correlating multiple flows. The work done in this thesis aims to prevent such traffic analysis by achieving TDMA slot allocation without the use of any information in packets. Such TDMA slot usage creates a strictly periodic traffic pattern, which makes it difficult to analyze separate flows as qualitatively discussed using the privacy metric. In the next chapter, we introduce our privacy preserving TDMA protocol, ZEA-TDMA.

## Chapter 4: Zero Exposure Anonymous TDMA (ZEA-TDMA)

In this chapter, we propose a distributed TDMA slot allocation protocol that relies on absolutely no information exchange among the participating nodes. This novel property allows the protocol to work in restricted anonymous and privacy-sensitive environments in which nodes may need to cooperate in distributed TDMA but are not allowed to explicitly exchange any information such as node-IDs in order to preserve their anonymity.

### 4.1 Time-Coded (*Blindfolded*) Packet Transmission

Once a slot ( $\tau$ ) is self-selected, a node periodically sends packets in that slot once in every TDMA frame (of duration  $T$ ). The packet a node periodically sends during its own TDMA slot is known as a *regular packet*. All the one-hop neighbors of a node listen to the channel and implicitly detect the node's slot timing by noting its *regular packet* transmission time. Since the neighbors cannot decipher anything from the packets, they do not know the identity of the transmitting node, but merely the presence of it. Using such implicit information, each node in the network can learn the slot timing of all its one-hop neighbors without actually knowing the identification of the neighbor nodes.

In the absence of time synchronization, each node maintains its own TDMA frame, starting at its own slot time. Figure 5(a) shows how the nodes in a three-node network maintain information about the slot occupancy of other nodes with respect to their own individual TDMA frames. We use phase ( $\phi$ ), which represents the location (i.e. timing) of a node's slot from the perspective of a hypothetical global observer maintaining a global frame, to represent slot locations of all nodes in the network. For example, to a global observer, the phase of node A is  $\phi_A$  as shown in Figure 5(b). Without inter-node time synchronization, the absolute phase does not have any meaning from an

individual node's (i.e., that of a local observer) standpoint. However, a node can detect its phase difference with another node. Phase difference between two nodes is the time difference between their respective slots. For example, the quantity  $\delta_{AB}$  in Figure 5(a) represents the phase difference between nodes B and A as perceived by B.

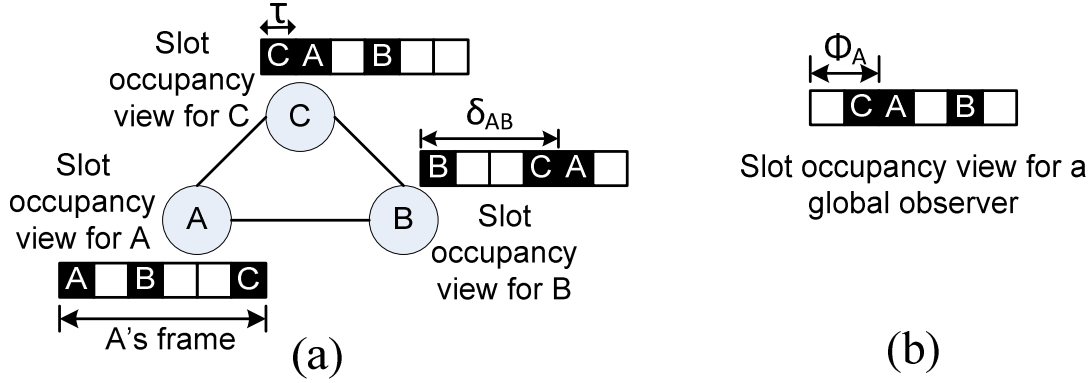


Figure 5. (a) Local slot occupancy view to individual nodes (b) Slot occupancy view to a global observer

## 4.2 Slot Self-Allocation

Upon joining the network, a node chooses an arbitrary time as its frame starting point and gathers the slot occupancy information about its one-hop neighbors by simply listening to their *regular packet* transmissions. After one frame-duration, a node receives *regular packet* transmissions from all nodes within its one-hop neighborhood and creates a one-hop slot occupancy list from the time of arrival of the *regular packets*. Then the node self-allocates the next available slot and defines it to be the new starting point of its own frame. At this point, the node starts sending *regular packets* in that slot for every consecutive frame.

It is necessary for a node to send a *regular packet* in its slot every frame to maintain the ownership of its slot. In case the traffic pattern is not periodic and a node does not have any packets to send in a frame, it can send a 'dummy' *regular packet*. This is required because the only way a

node's neighbors become aware of a node's existence is when they receive a *regular packet* and they implicitly know that the particular slot is used by another node in their one-hop neighborhood. If, due to energy savings reasons, a node cannot afford to send 'dummy' *regular packets*, the node can choose to not send anything during its slot. This may essentially result in a node giving up its slot and any other node with no assigned slot is free to claim the slot. When the node has data to send again, it first checks for transmission in its previous slot and if it is free, it starts reusing that slot. However, if the slot is currently being used by another node, it randomly selects a different slot which is available.

### 4.3 *Interrupt packets and Carrier Sensing*

Slot allocation in TDMA protocols for multi-hop wireless networks need to satisfy the constraint that at steady state, no two one-hop or two-hop neighbors can have partially or completely overlapping transmission slots. Overlaps between slots of one-hop neighbors cause direct collisions and overlaps between slots of two-hop neighbors can cause hidden collisions. The slot self-allocation process makes sure that nodes are aware of their one-hop neighbors' slot locations before they self-allocate a slot. However, since nodes are not aware of their two-hop neighbors' slots, this does not guarantee that nodes select slots non-overlapping with their two-hop neighbors, which might result in hidden collisions. We use *interrupt packets*, which is a novel and reactive approach, to resolve such hidden-collisions between two-hop neighbors.

Every node senses the channel before sending its *regular packet*, and if the channel is found to be busy, the node defers its transmission. Once the channel becomes free, it transmits the *regular packet* with a random backoff, and uses the new transmission time as its slot time. When a node selects a slot which is overlapping with one of its two-hop neighbors, a common neighbor of these two nodes detects this illegal slot assignment by detecting a collision between their *regular*

*packets*. Once detected, it sends an *interrupt packet* in the next frame, immediately after the start of the collided slot. Without time and slot-synchronization, the overlapping slots are almost always partially overlapping, which, in turn, causes the node with the latter slot to defer its transmission when it senses that the channel is busy due to the *interrupt packet*. This mechanism of third-party collision detection and resolution is discussed in further detail in the next section.

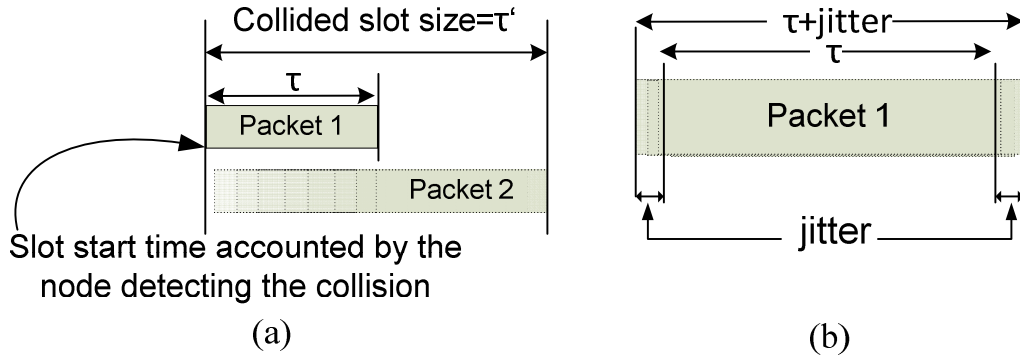


Figure 6. (a) Collided slot size (b) Random jitter and effective slot-size

#### 4.4 Third-party based Collision Detection and Resolution

Collision detection is done by checking the duration of the received signal. Since the size of the *regular packets* is fixed and other delay components including queuing and processing delays are assumed to be constant, a received signal that lasts longer than *regular packet* duration can be inferred as an overlapping transmission or collision at the receiver, as seen in Figure 6 (a). In this mechanism, a collision can only be detected if two nodes' slots have slots that are partially overlapping. This is ensured by a random jitter added by the nodes before self-allocation or before selecting a slot after its original slot gets deferred. The added random jitter results in increasing the effective slot size  $\tau$  by 3.5 %. All nodes also add a random jitter to their slot after a fixed  $c_1$  number of frames to randomize the exact location of their slots. This functionality prevents two

nodes from owning exactly overlapping slots. Figure 6(b) shows the random-jitter and the effective slot size.

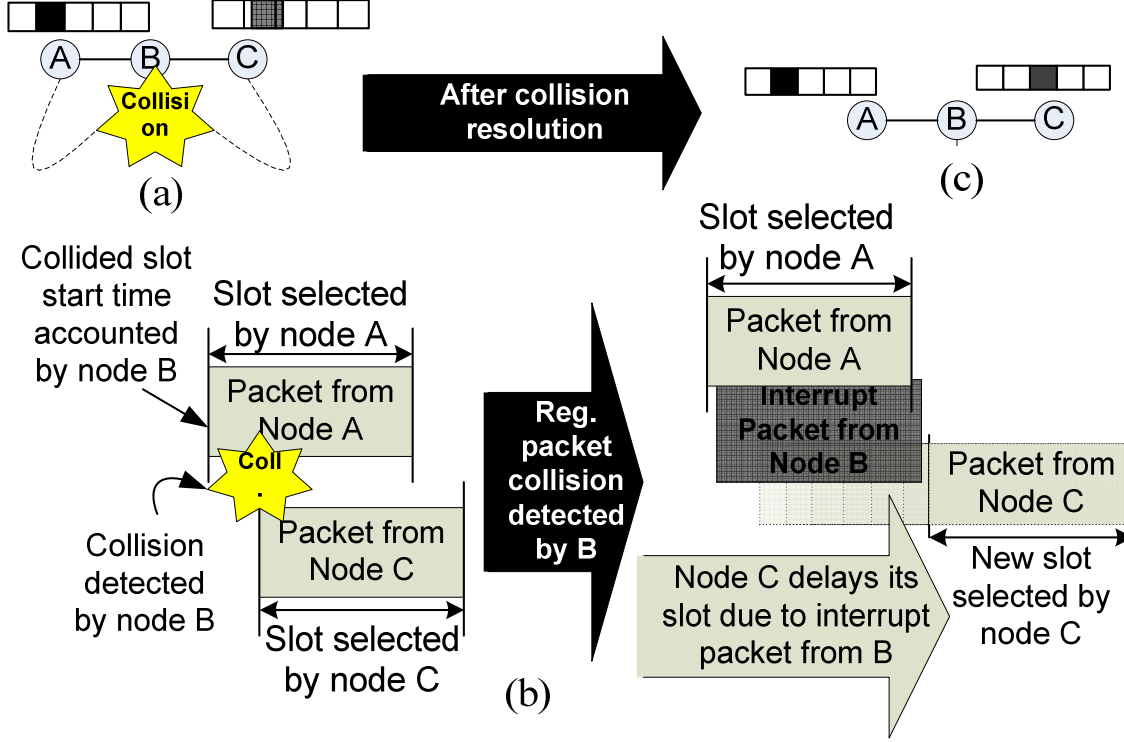


Figure 7. Collision detection and resolution: (a) Neighbors of B, i.e. A and C are using overlapping slots for regular packet transmission. (b) When B detects the collision, it sends an interrupt packet immediately after the start of overlapping slots. (c) This causes C to delay its slot and settle on a non-overlapping slot within its 2-hop neighborhood

When a node senses *regular packet* collisions between two of its neighbors, it resolves the collision by sending an *interrupt packet*. However, instead of sending an *interrupt packet* immediately, the node waits for  $c_2 \times T$  duration to ensure that it's a collision due to illegal slot usage and not a transient collision. We explain the two-hop collision resolution mechanism with *interrupt packets* using the example in Figure 7. In the example, nodes A and C, which are two-hops away from each other, share overlapping slots. Node B detects the collision from the length of the signal and marks the collided slot start time, as pointed out in Figure 7. To resolve the collision, node B sends an *interrupt packet* immediately after the start of node A's slot. The

*interrupt packet* keeps the channel busy for node C which causes it to defer its slot until the end of the slot occupied by node A and hence resolves the collision between nodes A and C. Collisions among more than two nodes are resolved by multiple *interrupt packets*. For *interrupt packets* to successfully resolve collisions, two criterions must be fulfilled: (i) the slots occupied by the colliding nodes have to be partially overlapping, and (ii) the *interrupt packet* must keep the channel busy until the end of the first node's slot (node A in the Figure 7). The first criterion is fulfilled by the random jitter added by the nodes and the second criterion is taken care of by making the *interrupt packet* size same as the *regular packet* size.

The *interrupt packet* sent by a node  $v_i$  ( $pkt_i^{int}$ ) and the *regular packet* sent by a node  $v_j$  ( $pkt_j^{reg}$ ) appear similar in all aspects to a node  $v_k$  which cannot decipher any information from either of the packets. Hence an overlap between  $pkt_i^{int}$  and  $pkt_j^{reg}$  can be interpreted as an illegal slot use between  $v_i$  and  $v_j$  by  $v_k$ . This will cause  $v_k$  to send an *interrupt packet* ( $pkt_k^{int}$ ) and this process can continue indefinitely. To avoid such a situation, a node only sends an *interrupt packet* when it detects an illegal slot use for a  $c_2$  number of consecutive frames, as mentioned earlier.

A detailed analysis of the collision handling process is provided in Appendix A.

## 4.5 Convergence

Once a node self-allocates a slot, it might defer its selected slot to resolve temporary collisions due to conflicting slots with any of its two-hop neighbors. Steady state for a node is defined as the time instant when the node finalizes its slot location. A given network is said to have reached convergence when all nodes in the network have reached steady state and there are no further changes in network dynamics. This also indicates a network-wide collision-free state.

## 4.6 Zero-Exposure

Note that all protocol syntaxes of the presented slot allocation algorithm rely on implicit local information about the time of packet arrival and collision detection at the receiver using the received signal duration. Hence the TDMA slots are allocated by a simple channel sensing based mechanism and without any message-based coordination or explicit control-information exchange and thus satisfy the original objective of *zero-exposure*. Since the algorithm does not use any information from the packets, it is ideal for use in environments where encryption is implemented at all protocol layers, or when MAC-layer frames or entire packets are encrypted. In a network with multiple trust-domains, nodes belonging to different trust-domains may not be able to decipher information from each other's packets, yet they will be able to use the algorithm for a TDMA slot allocation for channel access.

The basic slot-allocation algorithm is summarized in Algorithm 1. The four major phases of the algorithm are described below:

1. Once alive, a node goes through the initialization phase during which the node listens to *regular packet* transmissions in its neighborhood and creates a one-hop slot occupancy list.
2. After a TDMA frame duration  $T$ , the node starts its slot-selection phase when it randomly selects a slot non-overlapping with any of its one-hop neighbors and sets the current local time as its relative slot time. After slot selection, the node performs different operations based on the current local time.
3. It sends a *regular packet* during its own slot time and sends an *interrupt packet* if there is a scheduled *interrupt packet* pending. If the channel is found to be busy during its slot, a node defers its current slot and selects a slot time once the channel becomes free.

4. The reception phase runs in parallel with its regular operations and is triggered when the node receives a packet. The duration of the received signal is measured and if it is same as the *regular packet* duration, the node receives the packet and processes it. If the packet is corrupt or the duration of the received signal is longer than the *regular packet* duration, the node detects an illegal slot usage, i.e. collision, and schedules an *interrupt packet* to resolve the collision.

---

**Algorithm 1**

---

**Initial: Node A**

```

while(  $t < T$  )
| listen to regular packet transmissions
| create one-hop slot-occupancy list

```

**Slot selection: Node A**

```

select non-overlapping slot randomly
set my_slot_time =  $CURRENT\_TIME \% T$ 

```

**Start: Node A**

```

while( ALIVE )
| if(  $CURRENT\_TIME \% T = my\_slot\_time$  )
|   while( channel = BUSY )
|     //do nothing
|     send regular packet
|     set my_slot_time =  $(CURRENT\_TIME - \tau) \% T$ 
|   else if(  $CURRENT\_TIME = interrupt\_packet\_time$  )
|     send interrupt packet

```

**Reception: Node A**

```

if(  $\tau' > \tau$  )
| if( collision previously detected )
|   set current_collision_count++
| else
|   set current_collision_count = 0
|   if( current_collision_count >  $c_2$  )
|     interrupt_packet_time =  $T + \varepsilon + (CURRENT\_TIME - \tau')$ 

```

---

**Slot allocation algorithm for ZEA-TDMA**

---

## 4.7 Security Analysis

In this section, we analyze the security aspects of ZEA-TDMA. As mentioned in the threat model, the main objective of an adversary is to perform cross-trust-domain node-profiling, link

layer topology estimation, node-tracking, and flow-tracking. To perform node-profiling and tracking, the adversary needs to reveal the identity of the sender or receiver from the broadcast packet contents. For link-layer topology estimation and flow-tracking, the adversary needs to trace the flow of packets from the source to destination through intermediate nodes by either correlating messages using their contents or by performing timing and statistical correlation of flows at different nodes.

As discussed in the trust model, entire packets are encrypted using trust-domain specific pre-shared keys. We assume that the adversary is not capable of breaking the encryption using cryptanalysis. Since the content of packets cannot be deciphered by any node without the possession of the encryption key, cross-trust-domain identity snooping is prohibited. This, in turn, prevents node-profiling and node-tracking that may have been performed if packet contents were visible. Furthermore, encrypting entire packets can also prevent message correlation attacks [50]. This is because an attacker may not be able to correlate incoming and outgoing packets using a bitwise comparison. Packet size attack using packet size correlation [5], [37] is only effective against protocols using variable packet lengths and is avoided in our protocol as it uses fixed length packets.

In timing analysis attacks, an adversary searches for temporal dependencies between transmissions in multiple flows and performs link-layer topology estimation or flow-tracking if a sufficient correlation between different flows is found. For example, in a wireless network, if an adversary can correlate the traffic patterns (using features such as inter-packet interval, packet-count, etc) at two different parts of the network, with a deterministic or predictable delay, then he/she can conclude that the flows are highly correlated. However, when a periodic traffic pattern is used, the temporal dependencies can be minimized. If all nodes transmit at the same rate, all

flow rates that the adversary can observe will be highly correlated, leading to high false positives. The periodic traffic pattern is inherently provided by using TDMA-based channel access, which thwarts common traffic analysis techniques such as analyzing packet frequency, packet inter-arrival times, arrival rate, and packet delay characteristics.

## 4.8 Evaluation

The proposed ZEA-TDMA slot allocation algorithm has been implemented within the ns2 MAC simulation module. The baseline simulation parameters are shown in Table 1. The protocol has been evaluated for linear, fully-connected and grid topologies with network size ranging from 5 to 100 nodes. Network dynamics have been tested by joining two converged sub-networks. The protocol has also been tested for two different node deployment policies – one where all network nodes are introduced at the same time (static), and the other in which nodes are introduced incrementally once the remaining network has converged (incremental).

Table 1: Baseline system parameters in simulation

Propagation model	Two-ray ground
Channel bandwidth	2Mbps
Transmission range	240m
Network size	5-100 nodes
Regular & <i>Interrupt packet</i>	1024 Bytes
Slot duration	4.146 ms

### 4.8.1 Functionality Validation

The functionality of the protocol has been tested by running simulations for static as well as dynamic networks. Figure 8 and Figure 9 shows functionality results for static and dynamic linear topologies. The y-axis represents the phase of the nodes with respect to a hypothetical global observer (see Sec. 4.1), and the x-axis represents the time in terms of the number of rounds or

frames. For all experiments, the frame size is set to 6 slots (25 ms) which is one slot more than the maximum number of up to two-hop neighbors for the corresponding topology. An inter-node phase difference of less than 4.146 ms, which is the slot size  $\tau$ , indicates that two nodes have overlapping slots. Specific events of the protocol functionality have been pointed out using alphabetical markings and the corresponding state transition diagrams have been shown alongside the results.

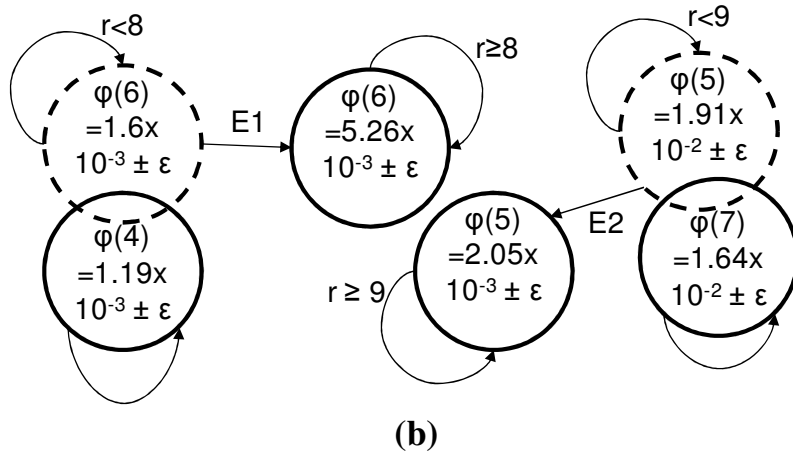
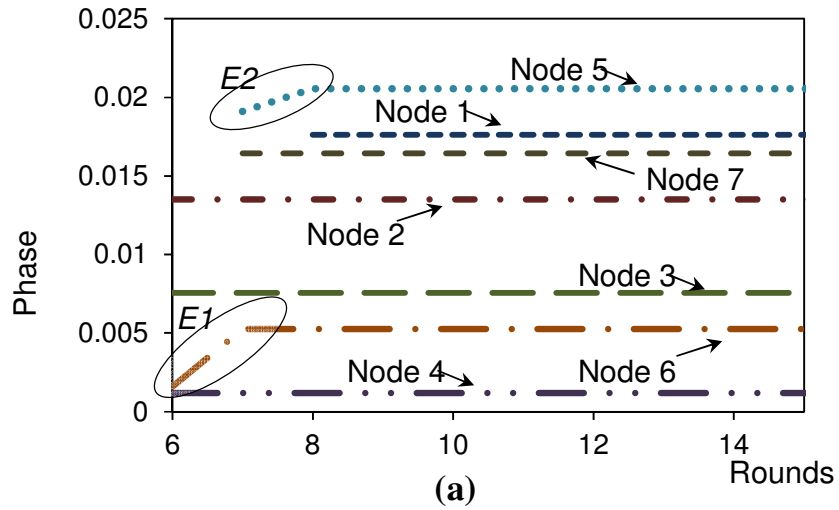


Figure 8. Functionality test for: (a) Static linear topology (b) State transition diagrams for topology in part (a)

### 1. Static Network

Functionality tests for static networks have been performed by introducing all nodes in the network at the same time. Figure 8(a) shows the phase of nodes in a linear network with 7 nodes and demonstrates how hidden collisions are resolved by a common neighbor. Initially, node 4 and node 6 have overlapping slots as seen from the figure. This is detected by node 5 which sends an *interrupt packet* to resolve the collision. This effect is marked by *E1* in the figure when node 6 defers its slot due to the *interrupt packet* sent by node 5, and selects a slot non-overlapping with node 4. *E2* in the figure depicts how overlapping slots between nodes 5 and 7 are resolved by *interrupt packet* sent by node 6. The state transition diagram of nodes 4, 5, 6 and 7 are shown in Figure 8(b). Transient states are shown with dashed lines and the round number ( $r$ ) represents the time. The phase of a node  $i$  :  $\varphi(i)$  represents the state of the node, and overlapping slots are shown in the figure by intersecting states. It can be seen in the diagram that event *E1* and *E2* causes node 6 and node 5 to change their states, respectively.

### 2. Dynamic Network

Figure 9(a) demonstrates the functionality of the protocol when two isolated converged sub-networks with linear topology are joined due to the insertion of a new node. After the two subnets (each with three nodes connected in a linear fashion) have converged, node 4 enters the network at round = 2003, and joins these two subnets to form a network of seven nodes connected linearly. In the figure, *E1* marks the slot selection and start of *regular packet* transmission for node 4. At *E2*, node 5 detects overlap between slots of node 4 and node 6 and sends an *interrupt packet* to

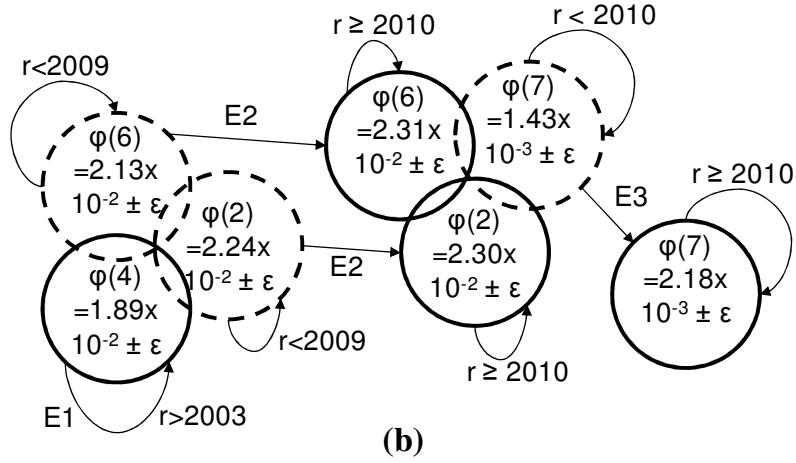
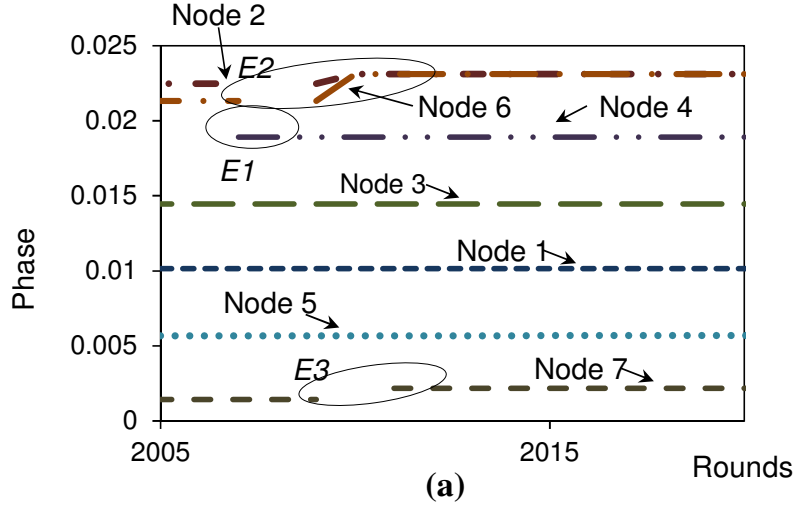


Figure 9. Functionality test for: (a) Dynamic network – two linear subnets joined through node 4 (b) State transition diagrams for topology in part (a)

resolve this overlap. As a result, node 6 delays its slot and selects a non-overlapping slot with node 4. Node 3 also detects an overlap between node 2 and node 4 at  $E2$  and sends an *interrupt packet* to resolve this. The *interrupt packet* shifts node 2's slot as can be seen in the figure. The new slot selected by node 6 overlaps with node 7's slot, and, as a result, at  $E3$ , node 7 defers its slot and selects a new slot non-overlapping with node 6. It can be seen from the state transition diagram in Figure 9(b) that event  $E2$  causes node 6 and node 2 to change their states. Here, event  $E2$  represents the transmission of *interrupt packets* from both node 3 and node 5.

The above scenarios demonstrate functional characteristics of the protocol using two specific examples. We have done extensive validation testing with linear, fully connected and grid network topologies using the two node-deployment policies – static and incremental and the protocol functionality have been validated through such testing.

#### 4.8.2 Convergence

Convergence time is defined as the time it takes for all nodes in the network to get a steady slot assigned to it. We also define F-ratio as  $\frac{n}{D}$ , i.e. the ratio of the chosen TDMA frame size (i.e. number of slots  $n$ ) and the minimum frame size needed for a network to reach convergence, which is determined by the maximum two-hop node degree  $D$ . For example, F-ratio = 1 corresponds to the smallest possible frame size and demonstrates the situation in which the frame size is exactly equal to the number of required slots. Note that due to slot sharing among nodes farther than two-hops, the minimum frame size is usually much lesser than the actual number of nodes in the network. Figure 10 and Figure 11 show the convergence results for the two different node deployment policies and different F-ratio values.

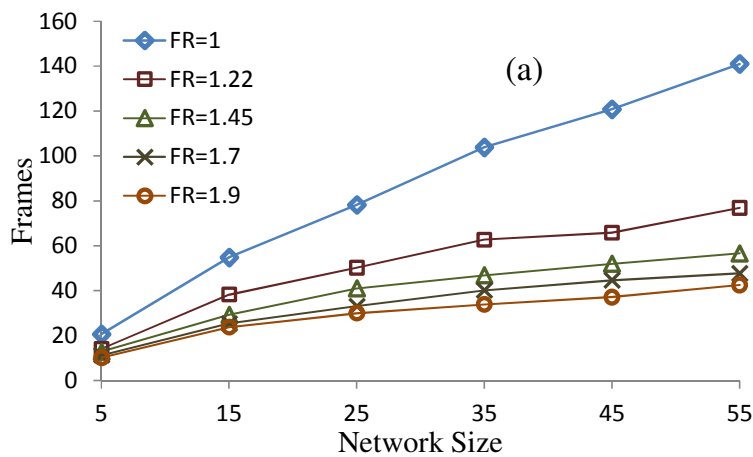


Figure 10. Convergence characteristics for: (a) Linear (b) Grid, and (c) Fully-connected topology with varying F-ratio

Figure 10 (cont'd)

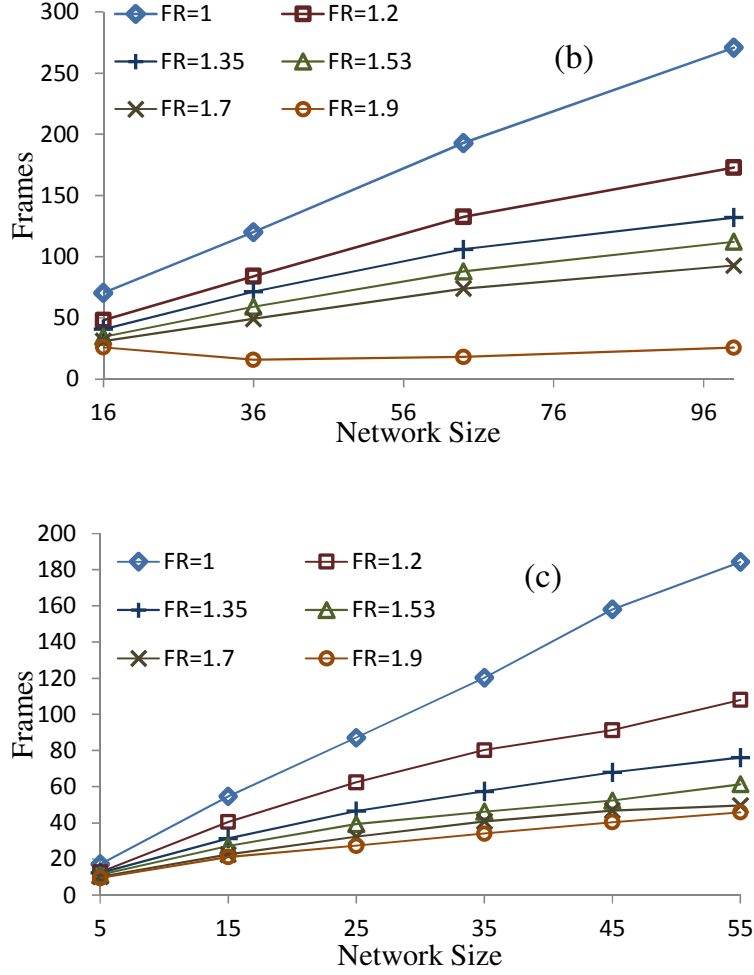


Figure 10. Convergence characteristics for: (a) Linear (b) Grid, and (c) Fully-connected topology with varying F-ratio

### 1. Static Node Deployment

This subsection presents the time it requires for a network to converge when all nodes are introduced in the network at the same time (i.e. static deployment). Figure 10(a) shows the convergence time of a static linear network with network size varying from 5 to 55 nodes. The x-axis represents the network size and the y-axis shows the convergence time in terms of the number of frames. Results have been shown for different F-ratio (FR) values between 1 and 2. It is seen that when the F-ratio is 1, the convergence time is maximum. Also, with an increase in the network

size, the network takes more time to converge. This is because when the F-ratio is 1, nodes are very tightly spaced in a frame and change in a single node's slot location may cause other nodes to defer their slots. Also, with larger number of nodes in the network, this effect is propagated throughout the network as can be seen from the graph. With increase in F-ratio, the convergence time decreases even as the number of nodes in the network increases. This is because there is enough space in the frame to accommodate all nodes in the network. Due to the larger frame size, a shift in one node's slot location does not necessary cause other nodes to defer their slots. For  $F\text{-ratio} \geq 1.45$ , the disparity in convergence time for different network sizes is much lesser when compared to lower F-ratios. This is due to the fact that for higher F-ratios, nodes have sufficient space in the frame to converge and hence the reduction in convergence time when F-ratio is further increased is not significant.

Figure 10(b) shows the convergence time of a grid network with static node deployment. The observations that can be made for this network are similar to linear network results. It can be seen, that with an increase in F-ratio, the convergence time decreases as nodes quickly self-allocate slots from a larger frame space. Although the results have a pattern similar to the linear network results, the convergence times are almost double in the case for grid topology. This is due to the topological difference between the two networks. A grid topology is more complex in nature with the presence of loops which result in circular dependencies among the nodes which results in the delays in convergence. However, for high F-ratios (e.g.  $F\text{-ratio} = 1.9$ ), the convergence time is almost same as linear networks due to the ample space in the frame which reduces the effect of slot changes among nodes in the network.

For fully-connected networks shown in Figure 10(c), the pattern of results is similar to linear topology. However, the time taken to converge is higher than linear networks and lower than grid

network. This is because of the following two reasons. Firstly, a change in one node's slot affects all nodes in this topology, which is unlike linear topologies, where the effect is limited up to two hop neighbors. However, this effect is diminished for higher F-ratios as can be seen from the graph. Secondly, the circular allocation dependency of grid network is not encountered in a fully-connected topology hence keeping the convergence time lower.

## ***2. Incremental Node Deployment***

This subsection presents the time it requires for a network to converge when nodes are added one at a time. A new node is added only after the network is converged after the previous node addition. Although experiments have been done for linear, fully-connected and grid network topologies, only results for grid network is shown as the other two networks mostly have instant convergence after a node is added.

Figure 11 show the convergence time for incremental node deployment for grid networks with varying sizes. The x-axis represents the node number being added and the y-axis shows the time required (in terms of number of frames) for the entire network to converge. This node addition results in incremental formation of the topology as the nodes are added individually. This means that as nodes are added to form a 10x10 grid network, the network topology goes from a 4x4 grid to 6x6 grid and so on, finally forming a 10x10 grid network. This is reflected in the results where we see that the convergence time for nodes in a larger network (e.g. 10x10) have similar initial pattern as that of smaller networks (e.g. 6x6, 8x8).

Another observation from the results is the difference in convergence time for different nodes in the network. Few nodes have instant (zero) convergence, while others have non-zero convergence. This was influenced by the initial neighborhood of a node when it joined the network. For our deployment policy in a grid topology, a node can be initially joined to one neighbor or two

neighbors. In situations where a node joins two neighbors, it has more information about its neighborhood and hence selects the next available slot which is collision free. However, in cases where a node joins one neighbor, it selects a slot based on information from only one node and ends up selecting a slot overlapping with a two-hop neighbor, which is then resolved, and hence results in non-zero convergence. Another observation that can be made from the results is that for higher F-ratios, the network tends to converge faster than instances with lower F-ratios. This is due to the fact that the probability of a new node finding a free slot is higher when the frame size is larger.

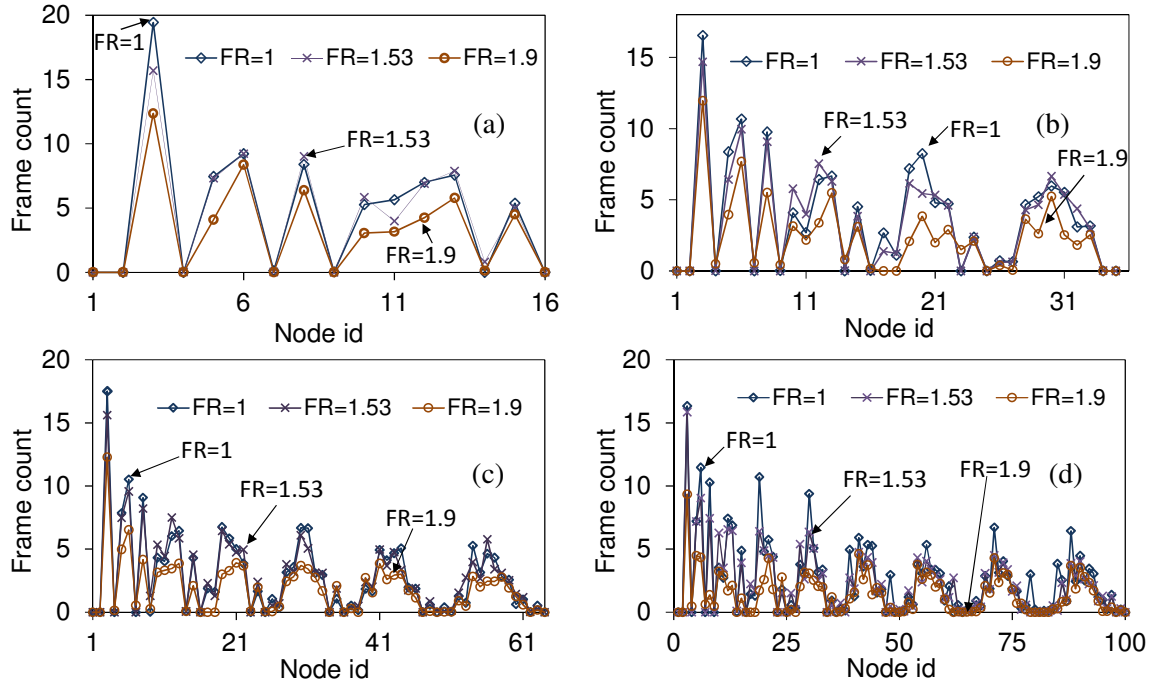


Figure 11. Incremental node deployment convergence characteristics for: (a) 4x4 grid (b) 6x6 grid (c) 8x8 grid (d) 10x10 grid

## 4.9 Summary

This chapter presents the ZEA-TDMA protocol in which slot occupancy information is disseminated by time-coded packet transmissions, and that is without using any explicit information in packets. Packet arrival times are used as the only source through which slot

occupancy information is implicitly transmitted. The functionality and evaluation of the protocol has been demonstrated using NS2 simulations. This chapter lays the groundwork for the development of protocols which use similar underlying principles and provide a richer set of functionality, but at the cost of added complexity.

## Chapter 5: Zero Exposure Anonymous TDMA with *Shadow packets* (eZEA-TDMA)

### 5.1 Introduction

Although ZEA-TDMA achieves anonymity, it is highly energy inefficient since in order to maintain slot occupancy, nodes are required to send a packet in every TDMA frame, even when there is no traffic. This causes unnecessary energy expenditure on resource-constrained wireless nodes. In this chapter, we present *eZEA-TDMA*, a variant of the ZEA-TDMA protocol, which provides the syntax for the development of a sleep-wake scheduling mechanism for idle-energy savings. The protocol uses a novel *pattern-based shadow packet mechanism* to disseminate slot occupancy information to up to two-hop wireless neighborhood. This added feature enables the development of an energy-efficiency sleep-wake scheduling mechanism presented in Chapter 6.

### 5.2 One-hop Information Dissemination

For one-hop neighborhood information dissemination, *eZEA-TDMA* uses the same principle as ZEA-TDMA. Time-coded packets are used for implicit detection of slot locations of one-hop neighbors, and a relative time-keeping enables an asynchronous slot-allocation process. However, in *eZEA-TDMA*, a non-overlapping slot usage up to two-hop neighborhood is ensured by using a proactive approach, unlike ZEA-TDMA which resolves two-hop slot overlaps by detecting collisions first and then reacting to them by using *interrupt packets*. Although it reduces the complexity of the protocol, the main disadvantage of such an approach is increased convergence time. Since there are no efforts to prevent collisions, they are a natural part of the convergence process and add to the convergence time of the network. In contrast, *eZEA-TDMA* disseminates two-hop information using a novel concept of *shadow packets* and tries to prevent collision in a

proactive manner. Equipped with two-hop neighbors' slot locations, nodes are less prone to select slots overlapping within the two-hop neighborhood.

### 5.3 Slot Self-Allocation

As mentioned, the slot self-allocation process for *e*ZEA-TDMA follows the same principle as ZEA-TDMA. Upon joining the network, a node listens to the channel for a predefined amount of time and gathers the slot occupancy information using the transmission times in its neighborhood. Next, the node self-allocates an available (free) slot and defines it to be the starting point of its own frame. At this point, the node starts sending *regular packets* periodically in its selected slot.

### 5.4 *Shadow packets* for Two-Hop Information Dissemination

Achieving collision-free slot assignment in TDMA requires the slot timing information of a node to be disseminated up to its two-hop neighbor nodes. In the absence of any explicit message-based coordination, the following innovative mechanism of time-coded two-hop information dissemination is introduced.

In this mechanism, a node sends a packet called *shadow packet* during its one-hop neighbors' transmission slots. More formally, node  $v_i$  sends a packet called *shadow packet* during  $v_j$ 's transmission slot  $\forall e_{ij} \in E$ . This *shadow packet* disseminates  $v_j$ 's slot timing information to  $v_k \forall k$  such that  $e_{ik} \in E$ . By noting the start timing of a *shadow packet*, a node  $v_k$  can infer the slot timing of one of its two-hop neighbors. The end result of this *shadow packet* transmission process is that all nodes are informed about the slot timing of their two-hop neighbors. This *shadow packet* based two-hop information dissemination, coupled with the time-coded one-hop information dissemination, makes a node aware of the slot timing of all nodes up to its two-hops, which is the basic requirement for any distributed TDMA slot selection algorithm.

The functional difference between a *shadow packet* and a *regular packet* is that the latter is sent by a node during its own TDMA slot, whereas the former is sent by a node during the slots of its one-hop neighbors. Since we assume that by default, nodes cannot decipher information from packets, *shadow packets* and *regular packets* are differentiated by making *shadow packets* slightly shorter in length than *regular packets*, such that the two can be distinguished from the duration of the received signal at the receiver node. The slot size  $\tau$  is assumed to be the same as the *regular packet* size to accommodate both packet types. Both the *shadow* and *regular packet* sizes are pre-fixed which can be trivially achieved by padding packets with smaller sizes or fragmenting larger packets.

If transmissions are not managed carefully, the *shadow* and *regular packets* can cause collisions in the following manner. Consider a node  $v_i$  that sends *shadow packets* during the slots of one of its one-hop neighbors  $v_j$ . Now, the *shadow packets* sent by  $v_i$  will collide with the *regular packets* sent by  $v_j$  (i.e. at the same time slots) at  $v_k \forall k$  such that  $e_{ik} \in E$  and  $e_{jk} \in E$ . We introduce the following pattern based shadow scheduling to avoid such collisions.

## 5.5 Pattern Based *Shadow packet* Scheduling

A node  $v_i$  sends *regular packets* based on a globally pre-determined frame-level bitmap pattern. Each bit in the pattern represents a TDMA frame.  $v_i$  either transmits a *regular packet* or skips the transmission based on the value of the bit for the corresponding frame. For example, when the pattern is set to '1110',  $v_i$  sends *regular packets* in its allocated slot in three consecutive TDMA frames (corresponding to the first three 1's) and then it skips the fourth frame (corresponding to the 0 in the pattern); the cycle continues for a super-frame of four TDMA frames which is the length of the pattern when the pattern is 1110. The fourth frame in the super-frame is skipped so that all of  $v_i$ 's one-hop neighbors can send *shadow packets* to disseminate  $v_i$ 's transmission slot

information. In other words, the one-hop neighbors of a node send *shadow packets* in an inverse pattern (i.e., 0001 in this case) that is 1's complement of the globally pre-set bitmap pattern. This is how the collisions between shadow and *regular packets* are avoided. For example, in Figure 12, the global pattern is set to '101'. Observe that node B sends *shadow packets* corresponding to its one-hop neighbor A with a 1's complement pattern '010' to avoid collisions with node A's *regular packets*. Although the pattern is globally pre-determined, the nodes maintain their individual pattern sequences asynchronously. As a result, a node first needs to learn about the pattern phase of each of its one-hop neighbors by listening to their transmissions over multiple super-frames. Once the pattern phase for a one-hop neighbor is detected, the node then starts sending the corresponding *shadow packets* following the 1's complement of the detected pattern phase.

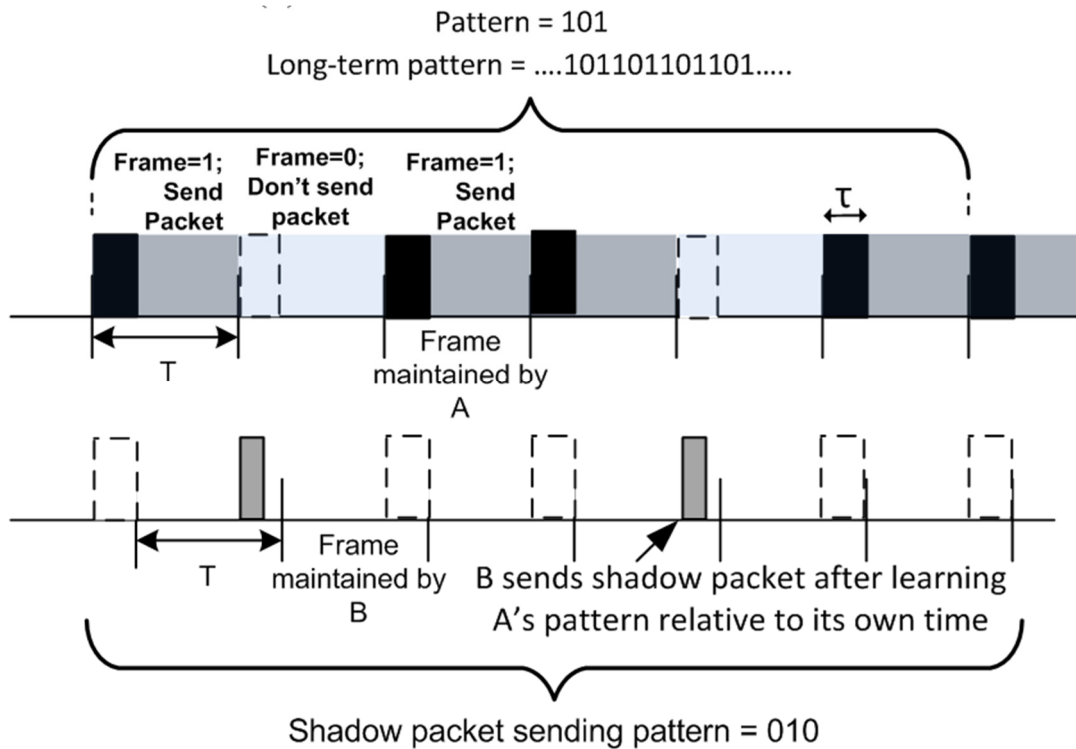


Figure 12. Regular and shadow packet sending pattern

Every node in the network maintains a list of slots currently occupied by its one-hop and two-hop neighbors using regular and *shadow packet* timing information. If there are no changes to this slot occupancy list for a predefined amount of time  $\Omega$ , the node determines that the network within its two-hop locality has stabilized and it (i) saves the current slot occupancy list, (ii) stops sending *shadow packets*, (iii) stops following the bitmap pattern, and (iv) starts sending *regular packet* in every consecutive TDMA frame. The node is said to have reached steady state and it starts to constantly monitor the transmissions in its neighborhood. After reaching steady state, if it observes any transmission that is not present in its saved slot occupancy list, it determines that there has been an event of network topology change. Such events can occur from addition of a new node to the network and joining or merging of two or more networks. On the event of network topology changes, the node (i) starts following the bitmap pattern, and (ii) starts sending *shadow packets* using the current slot occupancy list. The chosen value of  $\Omega$  will depend on the expected frequency of network topology changes.

## 5.6 Collision Detection and Resolution

Similar to ZEA-TDMA, a unique feature of *eZEA-TDMA* is that since it relies only on the packet transmission times and has no dependency on the packet content, it works even when a received packet is corrupted due to channel error or packet collision. The only requirement is that when a node receives a corrupt packet in a slot, it should be able to mark the corresponding slot as occupied. In fact, collisions are used as a valuable source of information as it can indicate overlapping slot ownership within a two-hop neighborhood. Specific collision types are identified and resolved as described below.

### 5.6.1 Collision Types

Unlike ZEA-TDMA, a collision in *eZEA-TDMA* can be among *shadow packets*, *shadow* and *regular packets*, or *regular packets*. All the three types of collisions are shown in Figure 13. A gray slot shown in the figure represents a node sending a *shadow packet* for one of its one-hop neighbors, whereas a black slot represent a node sending a *regular packet* in its own TDMA slot. In Figure 13(a)(i), when nodes B and D send a *shadow packet* in node A's slot, both the *shadow packets* collide at node C. In Figure 13(a)(ii), nodes A and E, which are more than two hops away share the same slot. The *shadow packet* that node D sends in E's slot and the *shadow packet* node B sends in A's slot collide at C. From Figure 13(a), it can be seen that the collision between two *shadow packets* can be due to: (i) the propagation of a single node's slot occupancy information, or (ii) propagation of slot occupancy information of two different nodes more than two hops away. In either case, the collision between two *shadow packets* does not violate the requirement for slot occupancy that nodes within two hops should not share the same slot.

In Figure 13(b), nodes A and D share the same slot and the *shadow packet* sent by node B in A's slot collides at C with D's *regular packet*. But collision between *regular* and *shadow packets* does not result in violation of the two-hop slot occupancy requirement, either. Although this can result in a reduced throughput for node D since its *regular packet* collides with the *shadow packet* from another node, there is no explicit syntax in the protocol to resolve such collisions because of two reasons. First, collisions between *regular* and *shadow packets* are transient collisions. Once the node sending *shadow packets* converges and reaches steady-state, such transient collisions cease to occur. Second, such instances can occur with a probability of  $\frac{4}{3n}$ , where  $n$  is the number of slots in a TDMA frame. This probability is fairly low for larger instances of  $n$ . The derivation for this probability is shown in Appendix B.

The third scenario, depicted in Figure 13(c), happens when there is collision between *regular packets*. This situation needs to be avoided as it causes direct violation of the two-hop slot occupancy requirement because a collision between two *regular packets* means that two nodes (within two-hops) have overlapping slots.

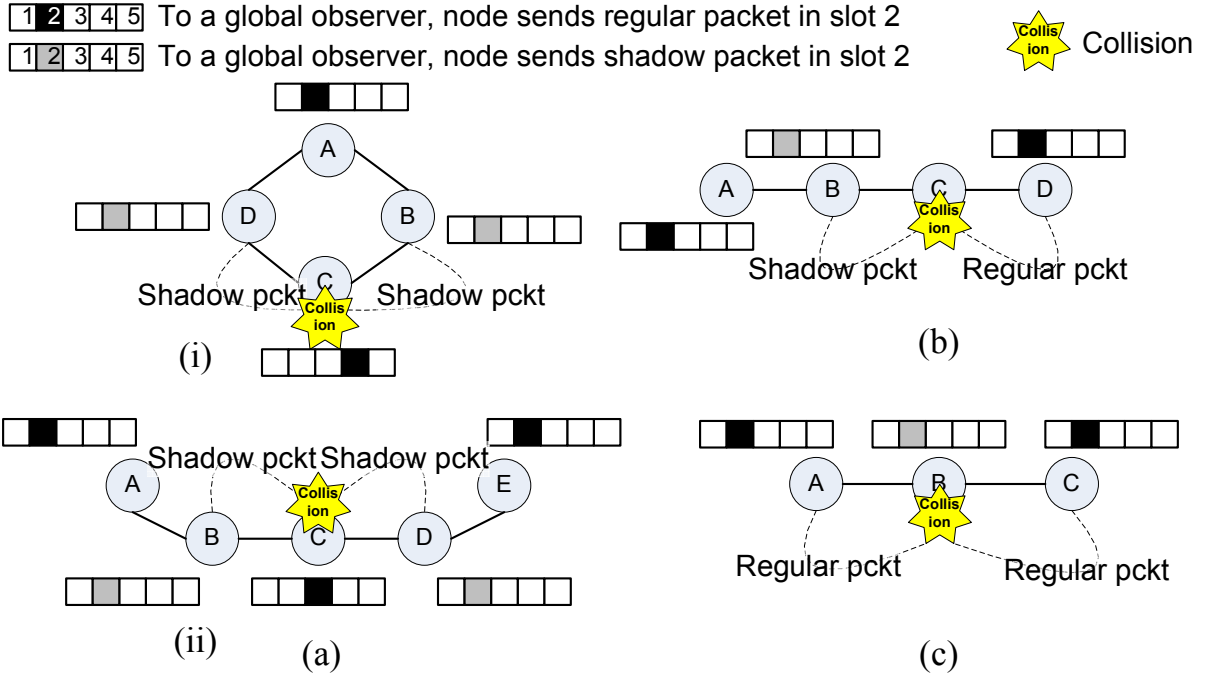


Figure 13. Types of collisions. (a) (i) and (ii) Shadow-shadow, (b) Shadow-regular, (c) Regular-regular

### 5.6.2 Detection of *regular-regular* collision

Detection of a *regular-regular* collision is a two-step procedure as follows. First a node needs to detect that there is a collision among its neighbors, and then it needs to distinguish this specific type of collision from the two other types. As in ZEA-TDMA, collision detection is done by checking the duration of the received signal. Any received signal that lasts longer than *regular packet* duration can be inferred as an overlapping transmission or collision at the receiver. Identifying the type of collision is done by monitoring the pattern of collisions over multiple frames. We explain this using the following example.

Node A pattern	...1 0 1 1 0 1 1 0 1...	...1 0 1 1 0 1 1 0 1...
Node B pattern	...0 1 1 0 1 1 0 1 0...	...1 0 1 1 0 1 1 0 1...
Collision pattern	...N N C N N C N N C...	...C E C C E C C E C..













Frame #	Regular-Regular collision pattern 1	Regular-Regular collision pattern 2
x		
x+1		
x+2		
x+3		
x+4		
x+5		

Figure 14. Regular packet collision pattern

Let  $C$  indicate a collision or corrupt packet reception,  $N$  indicate a successful packet reception (i.e. no collision), and  $E$  indicate a silent or empty slot, i.e. no packet reception. Then for ‘101’ as the pattern, a *regular-regular* collision follows the sequence of  $..CNNCNNC..$  or  $..CECCEC..$  over multiple frames as shown in Figure 14. These are the only two possible patterns exclusive to *regular-regular* collisions. Although the collided slot size can range from one-slot size (i.e. fully overlapped transmissions) to two-slot sizes (i.e. slightly overlapped transmissions), the node detecting the collision considers only the starting time of the first slot to mark the slot as collided.

### 5.6.3 Collision Resolution

Collisions in *eZEA*-TDMA are resolved in a manner similar to *ZEA*-TDMA. To prevent one-hop collisions, every node senses the channel before sending a *regular packet* and if the channel is found to be busy, the node defers its slot and transmits the *regular packet* only after the channel becomes free, after applying a random jitter. This basic channel sensing mechanism prevents one-

hop collisions. Two-hop collisions are resolved using *interrupt packets* which are sent by a node which detects a *regular-regular* collision between its one-hop neighbors. As discussed earlier, the *interrupt packet* has the same packet length as *regular packets*, and keeps the channel busy for the node using the latter part of collided slot thus resolving the collision. During the collision resolution process, if colliding nodes come within transmission range of each other due to mobility or fluctuations in transmission power, their collision will automatically be resolved by the one-hop collision resolution mechanism of simple channel sensing.

*Interrupt packets* work successfully only if the conflicting slots are not exactly overlapped. Since nodes randomly select an available slot, the probability of an exact overlap between two allocated slots belonging to different nodes is quite low. To further reduce the probability, a random jitter is introduced. When a node selects a slot during self-allocation, or it selects a slot after its original slot gets deferred due to transient collisions, it adds a random jitter to the newly selected slot time. To avoid exact slot overlapping over time due to clock drift or due to merging two disconnected networks, jitter is also used after  $c_1$  frames, which is a predefined number. The range of the random jitter in *eZEA-TDMA* has been chosen to be between  $0.006\tau$  and  $0.06\tau$ . The minimum value of the jitter needs to be large enough to account for channel propagation delay (i.e. the time needed for the *interrupt packets* to reach one-hop distance). Note that in its present form, the protocol does not explicitly support clock drift. However, if due to clock drift, there is a change in slot location of a node, the node in question undergoes slot self-allocation to select a new slot.

## 5.7 Allocation Convergence

If a node's self-allocated slot collides with that of any of its up to two-hop neighbors, the node is required to select a new slot to resolve such transient collisions. *Allocation convergence* for a node is defined as the time instant when the node finalizes its slot location after resolving any

transient collisions as stated above. After *allocation convergence*, a node constantly monitors for any change in slot-occupancy within its one-hop neighborhood and if there are no changes in occupancy for a predefined time period  $\Omega$ , the node stops sending the *shadow packets* for its one-hop neighbors. It's sufficient for a node to monitor its one-hop neighborhood since the effect of addition of a new node should be contained within its two-hops. For e.g. if  $v_j$  joins the network as a one-hop neighbor of  $v_i$ ,  $v_j$ 's effect should be felt only by the one-hop neighbors of  $v_i$ . *Allocation convergence* happens in a distributed manner since decisions can be made by each node autonomously. Hence, we consider the average *allocation convergence* for the entire network in our performance analysis, which is the average of the *allocation convergence* times of all nodes in the network.

## 5.8 Network Scalability and Network Dynamism

When a node joins a network and self-allocates a slot, existing nodes in the network detect the new node simply by sensing the new slot usage. The protocol inherently scales with network size as long as the maximum two-hop node degree (i.e.,  $D$ ) is less than the chosen TDMA frame size  $T$ , i.e.  $D\tau \leq T$ . Variable frame size at different parts of the networks based on the local  $D$  values is not supported by the protocol for avoiding complexity. This is in line with the distributed TDMA protocols in the literature [14], [16], [17], [46] majority of which do not support variable TDMA frame-sizes.

In theory, the protocol is capable of handling network dynamism such as frequent node additions, mobility and network merging. However, an optimal performance is achieved when the network is relatively static like wireless sensor networks and smart-home applications on Internet of Things (IoTs) [51]. Mobile networks such as VANETs [52] are not ideally suited for this protocol. This is a reasonable assumption since majority of networks running TDMA-based MAC

layer have a relatively static setup. Mobile networks with frequent topological changes can still run *e*ZEA-TDMA. However, it would provide relatively poor performance because of frequent occurrences of transient collisions, and nodes staying in steady-state for very short period of time compared to the convergence time. This is the cost incurred for having an information-less, time-coded distributed TDMA protocol. Additionally, node failures and nodes randomly leaving the network are handled in the same way. When a node does not detect a transmission during a neighbor's slot it assumes that the node has failed or left the network and simply removes the node from its current slot occupancy list. If the failed node revives, it first tries to use its previously owned slot. However, if the slot is occupied, it undergoes the slot self-allocation process afresh.

## 5.9 Evaluation

We have implemented the proposed *e*ZEA-TDMA using the ns2 MAC simulation module. The baseline simulation parameters were the same as ZEA-TDMA and is shown in Table 1. The protocol has been evaluated for different network topologies including linear, loop, fully-connected, grid and random mesh with network size ranging from 3 to 100 nodes.

Network dynamics have been tested by joining two converged sub-networks. The *shadow packet* pattern used in the evaluation is 101. The simulations have been run for varying F-ratio values as it indicates the constraints of the TDMA frame size on the overall performance of the network. As defined in Chapter 3, F-ratio is given by  $\frac{n}{D+1}$ , which is the ratio of the number of slots in a TDMA frame and the minimum number of slots required for entire network sustainability. F-ratio = 1 is the smallest possible frame size and demonstrates the situation in which the frame size is exactly equal to the number of required slots. Note that there is no direct relation between the network size and F-ratio because due to slot sharing among nodes farther than two-hops, the minimum frame size is usually much less than the actual number of nodes in the network. So, if

the network size can be increased without affecting  $D$ , then it is not necessary to change the TDMA frame size  $T$ , and F-ratio will remain constant since  $\frac{n}{D+1}$  remains constant. We also show results discussing the cost and benefits of using different F-ratio values.

### 5.9.1 Functionality Validation

The functionality of the protocol has been tested by running simulations for static as well as dynamic networks with linear, loop and fully-connected topologies. Apart from the bit-map pattern being 101, the other parameters used were same as ZEA-TDMA. Figure 15 and Figure 16 shows the functionality graphs for loop, fully-connected and linear networks. Similar to the functionality results for ZEA-TDMA, the y-axis represents the phase (normalized by frame size) of the nodes with respect to a global observer, and the x-axis represents the time in terms of the number of frames. For all functionality validation experiments, the frame size is set to 5 slots which represent the maximum number of up to two-hop neighbors for all the topologies that were experimented with, i.e. F-ratio was set to 1 which means the protocol functionality was evaluated in the most stringent situation possible. An inter-node phase difference of less than 0.2, which is the normalized phase equivalent of the slot size  $\tau$  (4.096 ms), indicates that two nodes have overlapping slots. Specific events of the protocol functionality have been pointed out in the figures using the alphabetical markings.

#### 1. Static Network

Like ZEA-TDMA, tests for static networks have been performed by introducing all nodes in the network at the same time. Figure 15(a) shows the phases of nodes in a 4-node loop topology. The convergence is pretty straight-forward in this case. Since nodes are introduced at the same time, the initial slot-selection causes nodes 1 and 2 to select overlapping slots in  $E1$ . However, this is quickly resolved once they start sending *regular packets* and node 1 senses the channel to be

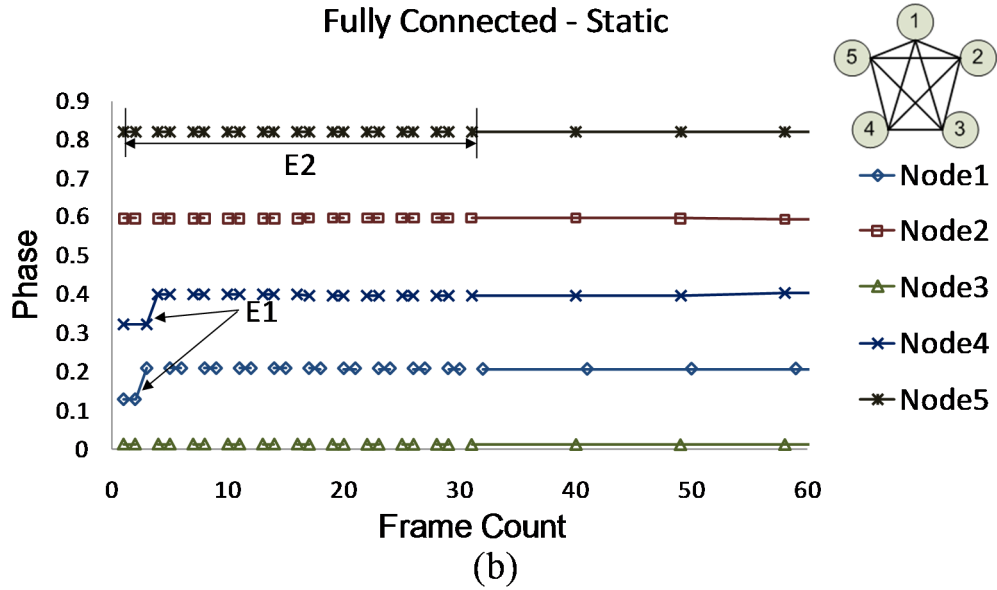
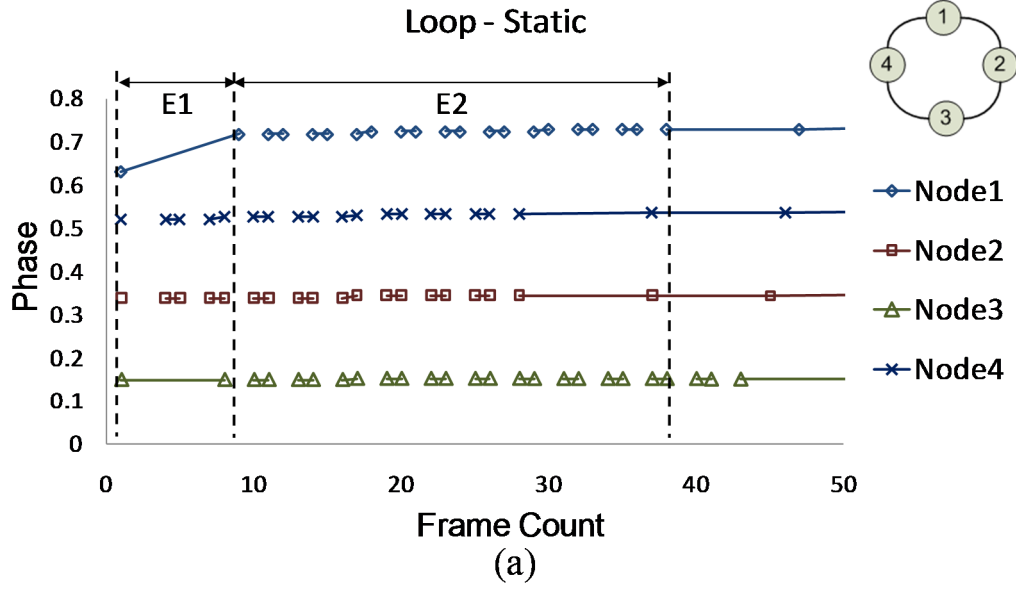


Figure 15. Functionality tests for: (a) Static 4-node loop topology (b) Static 5-node fully-connected topology

busy and defers its slot location, which is represented as  $E2$  in the figure. The other nodes in the network have correct initial slot selection and hence no changes were noticed in the slot locations. The pattern ‘101’ can be seen from the gaps in the line representing a node’s phase, since they correspond to the ‘0’ in the pattern. Since there were no changes in network dynamics detected in the network after convergence, all nodes stop sending *shadow packets* in a distributed fashion.

Figure 15(b) depicts the phase of the nodes in a fully-connected network with 5 nodes. Initially nodes 1 and 3 have overlapping slots until frame count = 2. Node 1 detects the channel to be occupied (by node 3) while sending its *regular packet* and defers its slot to 0.2074, which causes its slot to be overlapped with that of node 4, as shown by the marking *E1* in the figure. This results in node 4 deferring its slot and selecting a non-overlapping slot at frame count = 4. After these changes in slot occupancy, all the nodes have converged with fixed allocated slots and they continue to transmit *regular* and *shadow packets* according to the chosen pattern ‘101’ as marked in the figure for node 5 as *E2*. After there has been no event of network topology change for a predefined time period, the nodes stop following the pattern which can be seen from the solid lines after 30 frames.

## 2. Dynamic Network

Figure 16 demonstrates the functionality of the protocol when two isolated converged sub-networks with linear topology are joined due to the insertion of a new node. In Figure 16(a), after the two subnets (each with three nodes connected in a linear fashion) have converged and stopped following the pattern, node 4 enters the network at frame count = 57, and joins these two subnets to form a network of seven nodes connected linearly. Different events shortly before and after the joining are described below using various markers in the figure. ***E1***: From the collision pattern, node 4 detects that two of its neighbors (nodes 3 and 5) have overlapping slots and sends an *interrupt packet* to resolve the collision. ***E2***: The *interrupt packet* sent by node 4 causes node 3 to delay its slot and to select a non-overlapping slot at frame count = 72. ***E3***: The slot initially selected by node 4 has a slight overlap with node 6’s slot. When node 6 senses the channel before sending its *regular packet*, it finds the channel to be busy due to the *shadow packet* sent by node 5 for its one-hop neighbor node 4. This causes node 6 to adjust its slot and to select a slot that is non-

overlapping with node 4's slot. The change in node 6's slot also necessitates node 7 to make adjustments to its own slot as seen in the figure. It can be also observed from the figure that all network nodes eventually stop sending shadow packets in a completely distributed fashion at different times.

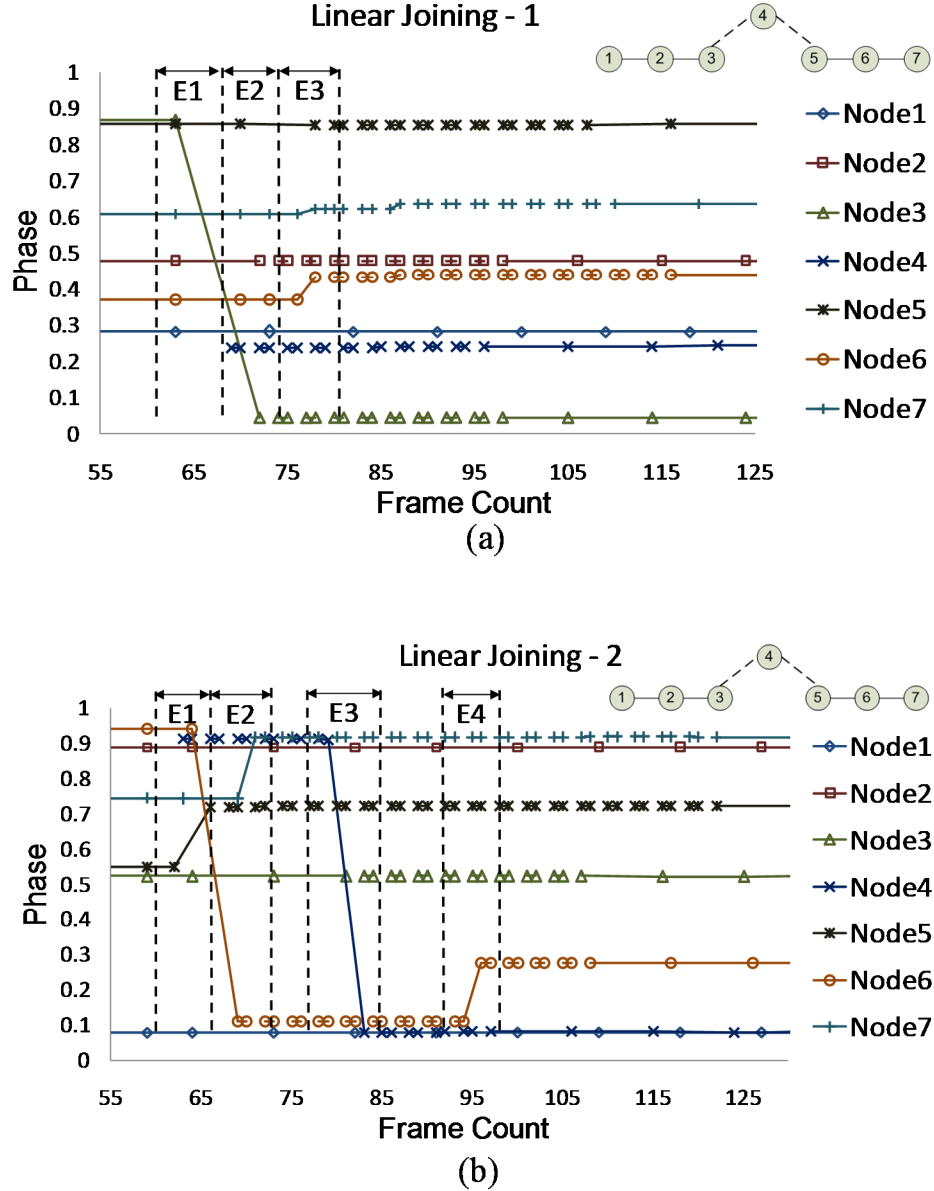


Figure 16. Functionality test for dynamic 7-node linear topologies

Figure 16(b) shows a similar network joining scenario but with more slot re-allocations. **E1:** The overlap between node 4 and 6 is resolved by node 5 which causes node 6 to change its slot location. Also, after node 4 joins the network, nodes 3 and 5 become two-hop neighbors and hence their prior slot assignments become illegal. This is resolved by *interrupt packet* sent by node 4 and a change in node 5's slot location can be observed. **E2:** Node 5's change in slot location causes an overlap with node 7's slot, which is then resolved by node 6. Although, node 7's newly selected slot overlaps with node 4's slot, this is a valid overlap since the two nodes are more than two-hops away. **E3:** As a result of the slot selected by node 4, packets sent by node 2 collided with node 4's packets. This, after being detected, is resolved by node 3, causing node 4 to select a slot directly overlapping with node 1. However, this is a valid slot assignment, and is not resolved. **E4:** The new slot selected by node 4, however, resulted in illegal slot assignment due to an overlap with node 6's slot. This is resolved by an *interrupt packet* from node 5, after which all nodes have valid slot assignments.

### 5.9.2 Performance Characterization

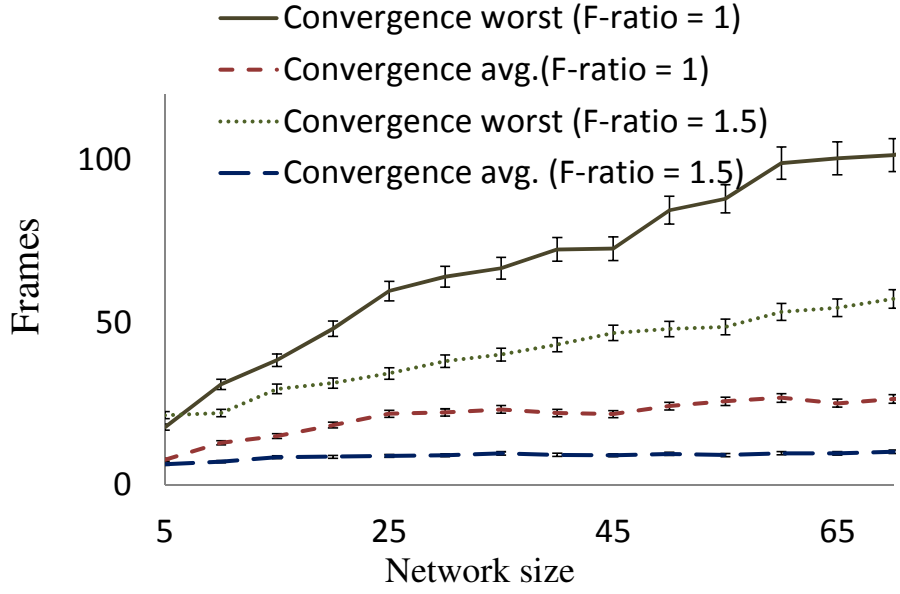
To characterize the performance of *eZEA-TDMA* we measure convergence time, which is the interval from the time a node is added to the network to the time the node obtains a steady slot assigned to it. Furthermore, we also report the *pattern-termination* (PT) time which is the interval from node addition to when the node stops sending *shadow packets* (i.e. stops following the desired *regular packet* transmission bitmap pattern). The PT-time is dependent on the allocation convergence time of a node, as well as the chosen value of  $\Omega$ , which is the duration for which a node monitors the network before stopping to follow the bitmap pattern. We have used  $\Omega = 35$  in our simulations.  $\Omega$  should be selected based on the dynamism in the network. If nodes are constantly joining or leaving the network, then a higher value of  $\Omega$  will ensure a faster convergence

for the newly added nodes. However, higher  $\Omega$  also implies impaired throughput due to *shadow packets*. We also evaluate the performance of the protocol based on the collision resolution and network re-stabilization time. Collision resolution time is defined as the time taken by a node to resolve collision among its neighbors, and network re-stabilization time is defined as the time it takes for a stable network to re-stabilize once a node is added to the network. Note that the protocol has not been compared with other distributed TDMA protocols in the literature, since none of them offers strict anonymity as done by *eZEA-TDMA*.

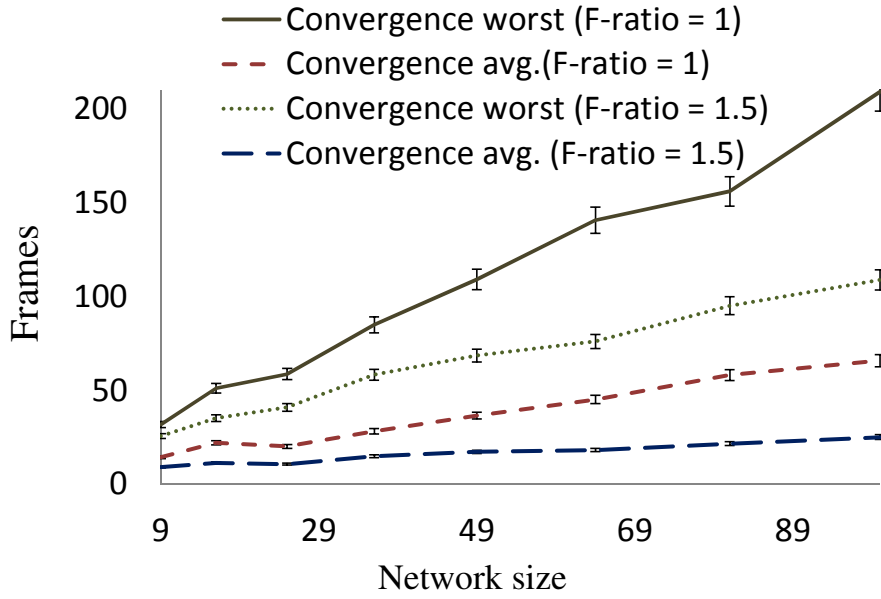
### 1. Allocation Convergence

Since the allocation process is distributed, the convergence time can be different for different nodes. The graphs in Figure 17 show the average and worst case convergence time with 95% confidence range based on the variance of 300 simulation runs of linear, grid and fully-connected topologies where all nodes are simultaneously introduced in the network. The x-axis represents the network size and the y-axis represents the time to converge in terms of the number of frames. The average case convergence shows the network-wide average of the convergence time and the worst case convergence time is plotted for the node which takes the longest to converge.

From the graphs, it can be observed that convergence times for F-ratio = 1 is always higher than that for F-ratio = 1.5. Also for linear and grid topologies with higher network sizes, the worst case convergence is considerably higher than the average case convergence. This was caused because with an increase in network size, even though all other nodes in the network have self-allocated slots, there are always one or two nodes which take more time to self-allocate a slot. This in turn increases the worst-case convergence time. The effect is more severe when F-ratio = 1, as the frame size with this F-ratio is at its maximum capacity (i.e. the absolute minimum frame size for the given network). Also, with this constrained F-ratio, the jitter from one node sometimes causes



(a)



(b)

Figure 17. Convergence characteristics for (a) Linear (b) Grid and (c) Fully-connected topology with  $F\text{-ratio} = 1$  and  $1.5$

Figure 17 (cont'd)

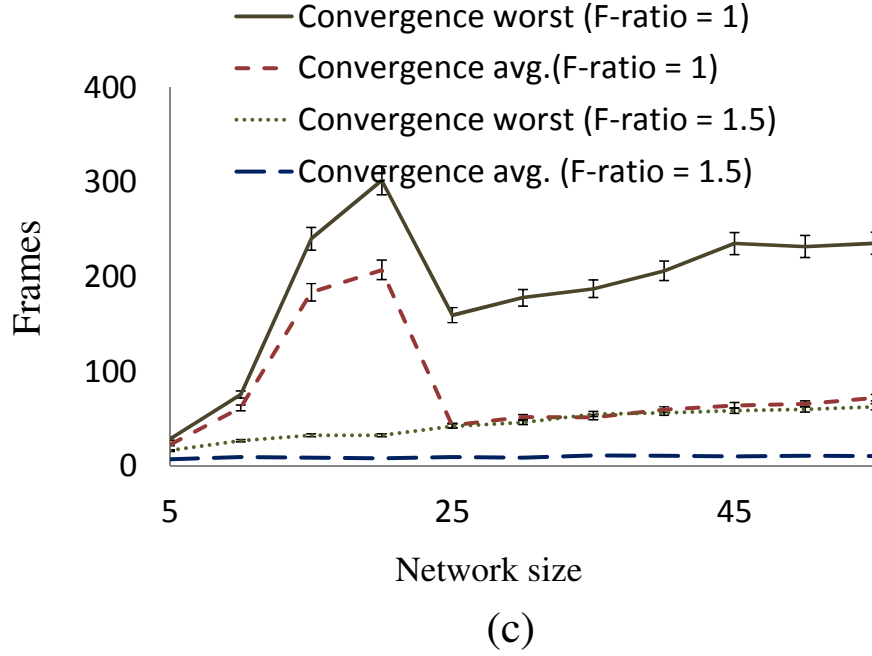


Figure 17. Convergence characteristics for (a) Linear (b) Grid and (c) Fully-connected topology with F-ratio = 1 and 1.5

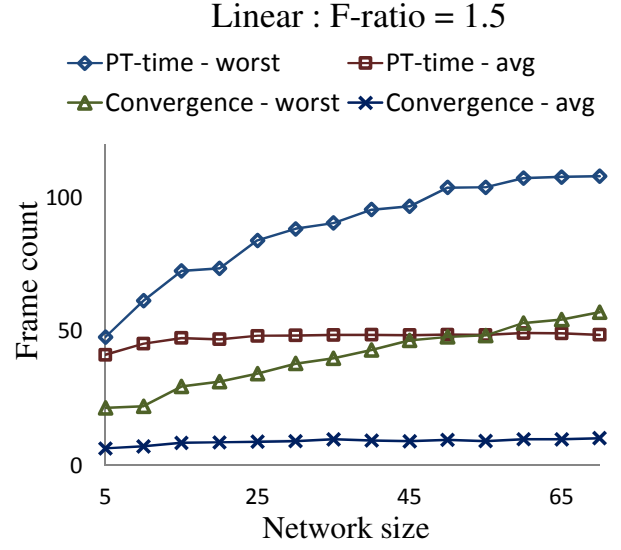
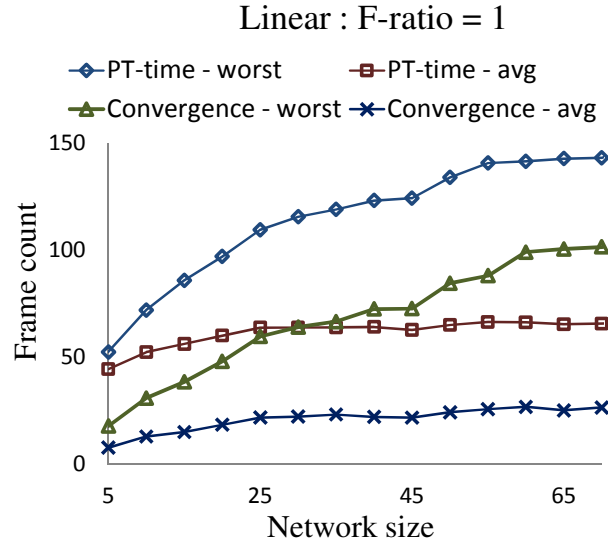
its neighbor to defer its slot. This effect, however, is not observed network-wide as it vanishes with the distance from the affected node due to the jitter from other nodes. Another observation from Figure 17(a) and (b) is that the average case convergence for linear networks is almost constant, unlike grid networks, which show a slight increase in the average convergence time. The reason behind this is the topological difference between the two networks. A grid topology is more complex in nature with the presence of loops which results in circular dependencies among the nodes which results in a minor delay to find a self-allocated slot.

Figure 17(c) shows the convergence graphs for a fully-connected network for F-ratio=1 and 1.5. It can be observed that for F-ratio = 1, there is a rapid increase in the convergence time until network size = 20. Then a sudden drop of convergence time is seen after which the time increases with a less steep slope. The reason behind this is as follows. The value of  $D$  of a fully connected

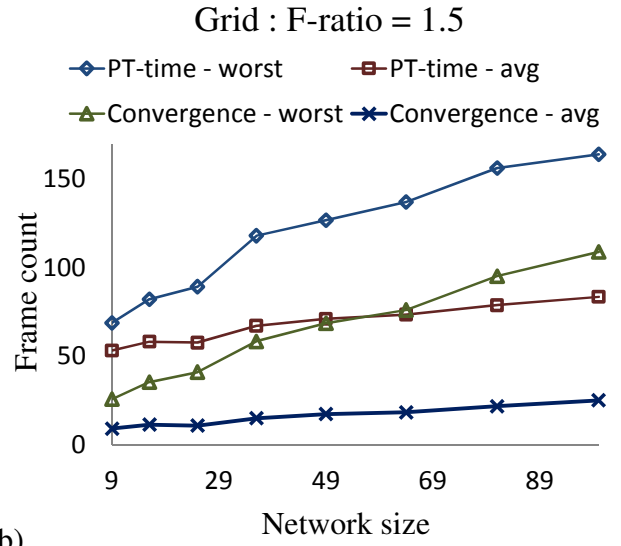
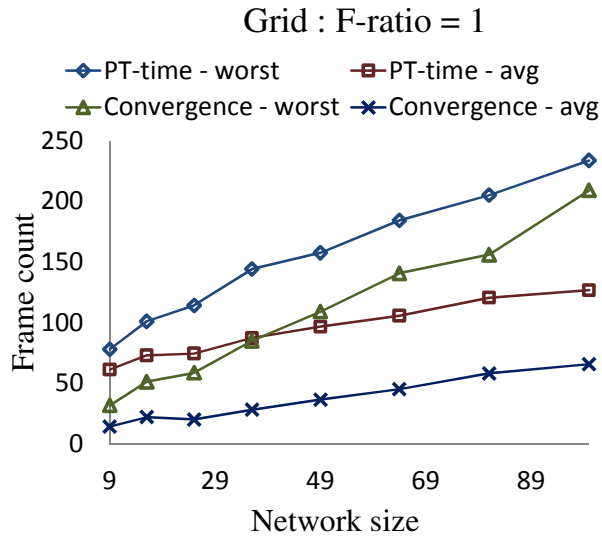
network is equivalent to the network size and with an increase in network size; there is a sharp increase in convergence time. But when the network size becomes greater than 20 nodes, the accumulative jitter added by each node results in the effective F-ratio to be larger than the actual F-ratio (i.e. 1). The increase in the effective F-ratio results in the decrease in convergence time as can be seen by the steep drop in the graph. As the network size is further increased, the convergence time also increases, but the accumulative jitter plays an additional part which results in a less steep increase than seen when the network consisted of 5 to 20 nodes. This phenomenon is not observed for F-ratio = 1.5 as the frame-size is already large enough for fast slot-allocation convergence.

## 2. *PT- Time*

In Figure 18 (a), (b), and (c) we plot the PT- time along with the convergence times for comparison. We show the PT-time for average case and worst case, similar to the previous results. The common observation from these results is that the PT-time follows a very similar trend as the convergence time, which is an expected behavior since the PT-time is dependent on the convergence time. However, there are minor variations since the *pattern termination* of a node is also dependent on the convergence of its neighbors.



(a)



(b)

Figure 18. Pattern termination time and Convergence time for (a) Linear (b) Grid with F-ratio = 1 and 1.5

Figure 18 (cont'd)

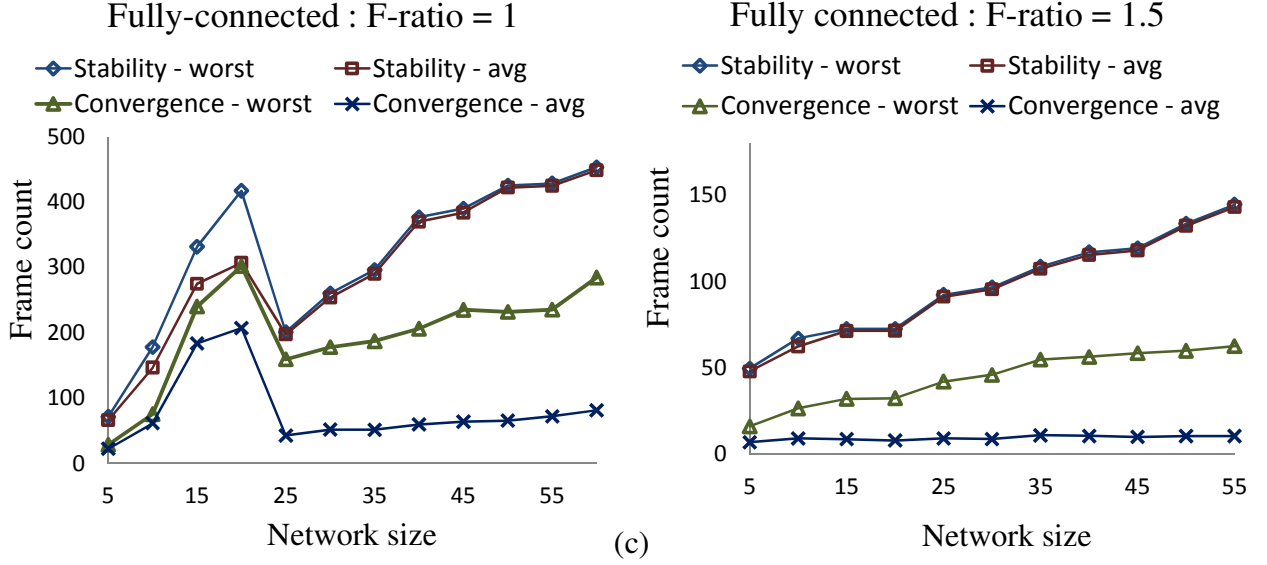


Figure 18. Pattern termination time and Convergence time for (c) Fully-connected topology with F-ratio = 1 and 1.5

From Figure 18 (c) it can be seen that for fully-connected topologies, the worst case convergence time is lower than the average case PT-time, which is a variation from the other topologies presented. This can be attributed to the fact that since all nodes are connected, the PT-time of every node is dependent on the convergence of every other node in the network. Also, the PT-time for average case is almost same as that for the worst case because of the same reason. Once the last node converges to a steady state, all nodes stop sending *shadow packets* roughly at the same time.

### 3. Collision Resolution

The collision resolution time depends on the number of nodes involved in the collision. Figure 19(a) and (b) demonstrate the collision resolution time required by one node to resolve collision between  $n$ -nodes. As the number of nodes involved in collision increases, the time required to resolve the collisions also increase. Figure 19(a) shows the number of frames or rounds required

to resolve all collisions and Figure 19(b) plots the same data in terms of the actual time required to resolve the collisions. The topology corresponding to the results shown is a star-topology where all nodes involved in collision are two-hops away from each other and only one-node, in the center, is responsible to resolve all the collisions. An example of a five-node star topology used in the experiment is shown in the graphs.

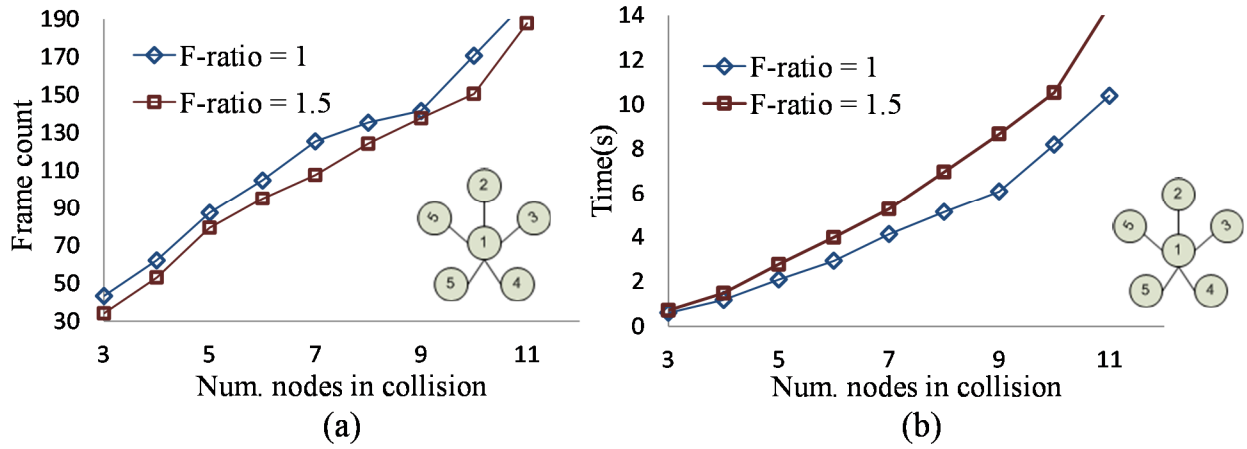


Figure 19. (a) Collision resolution time in frame counts (b) Collision resolution time in seconds

The graphs have been plotted for F-ratio = 1 and 1.5. When frame size is large (F=1.5), a node can find an empty slot without disrupting other nodes in the network. This leads to smaller number of frames required to resolve all collisions. However, the actual time required for collision resolution is less when F = 1. This is because for F=1.5, there is an additional overhead involved in terms of time, due to the larger frame size.

#### 4. Network Re-stabilization

Depending on the network topology, a part of the network or the whole network may be destabilized when a new node is added to a stable network. The value of  $\Omega$  has been kept fixed for all experiment runs. Figure 20 shows the time required for all nodes in a fully-connected network

to reach PT-time as a function of network size after a new node joins the network. The new node is introduced when all current nodes have stopped sending *shadow packets*. For F-ratio = 1, the number of frames required for the network to stabilize (i.e. reach PT-time) increases with the network size. But after the network size reaches 22, the PT-time is random and unpredictable. The reason behind this is that as the network size increases, the cumulative space provided by the random jitter may add up to accommodate the new node. The randomness of the jitter results in the PT-time being unpredictable. There are situations when the nodes are ‘equally spread’ across the whole frame, spatially occupying the frame resulting in no immediate space for the new node. In such scenarios, nodes need to move and delay their slots according to the protocol and require more time to converge. With a higher F-ratio, the number of frames required to accommodate the new node is lesser, as there is more space in the frame for the new node to squeeze in. A similar randomness in stabilization time is observed here after the network size reaches 13. This is because for a higher F-ratio, the number of nodes which need to move their slots, to occupy the new node, becomes lesser.

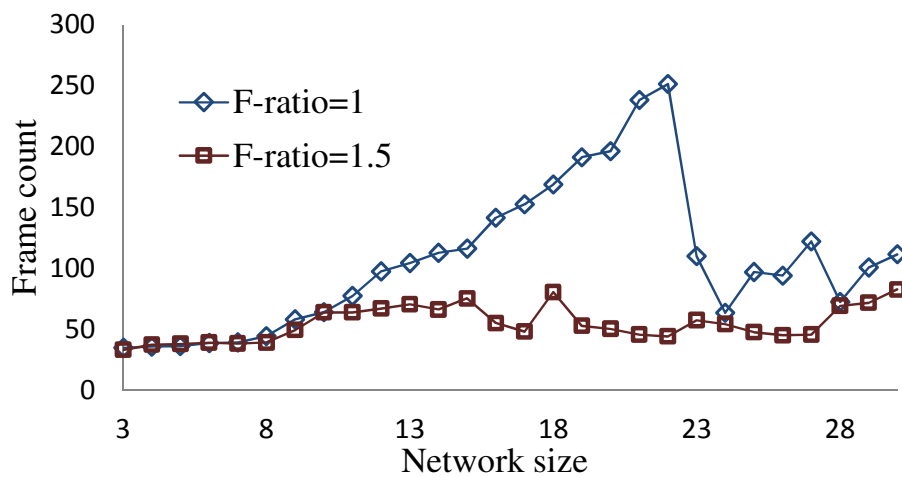


Figure 20. Network re-stabilization time required by a stable network when a new node is added

### 5. Optimal F-ratio

Although higher F-ratio results in lower number of frames required for network convergence, it also adds extra overhead in terms of the actual time due to the larger frame size. Figure 21 shows the results for convergence-time against different F-ratio values for a fully-connected network with seven nodes. From the graph, it can be seen that the time required for fixing a slot is fairly constant for both average and worst case from F-ratio = 1.27 to 1.64. But both the average and worst case stability time is the lowest at F-ratio = 1.27. From this, it can be concluded that although F-ratio = 1 is the minimum frame-size that can accommodate all nodes in the network; it is not the optimal frame-size to use in terms of convergence and PT-time. Also, higher F-ratio values do not provide better results.

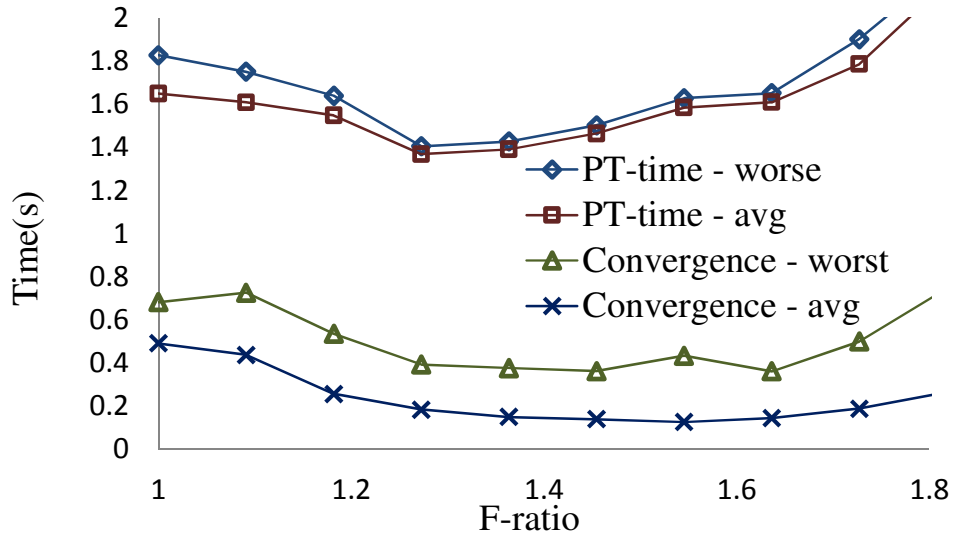


Figure 21. Optimal F-ratio for a fixed network size

#### 5.9.3 Power Consumption during Transience

Since different packets like the *shadow* and *interrupt packets* are transmitted during the convergence itself, it is interesting to study the power consumption during the allocation convergence process, i.e. the **transient state**. Two different network topologies have been tested

in which the network was formed incrementally by adding nodes every 30 seconds to form a 4x4 grid and a 15 node fully connected topology, respectively. During the simulation, nodes transmitted packets at full transmission power and the transmission and reception power consumption values of 81mW and 30mW [53] were used, based on Mica2 power consumption data. Figure 22 and Figure 23 plots the average power expenditure per node per TDMA frame for F-Ratio = 1 and 1.3. Both the power expended to transmit packets ( $P_{wTx}$ ) and the power expended to receive packets ( $P_{wRx}$ ) have been plotted separately for better observation.

It can be observed that for all topologies and F-ratios,  $P_{wTx}$  has spikes when nodes are added to the network. However, it can be seen that when the first few nodes were added, the spikes were smaller sized than when later nodes were added. The reason is that as the network reaches its maximum capacity, the convergence takes more time. This can be very well observed in Figure 23(a) when the 15th node is added to a fully connected network with F-ratio = 1, the network reaches its maximum capacity and takes much longer to converge. Once converged, the  $P_{wTx}$  has a steady consumption. This is the steady state power consumption and is different from when the network is in a transient state. When the 14th node was added, the ratio between the convergence time and node addition interval (30s) is 0.076, i.e. the convergence time is very short even when nodes are added at short intervals of 30s. However, as the network reaches its maximum capacity, i.e. when the 15th node was added, the ratio goes up to 0.66. However, for relatively static networks like wireless sensor networks and smart-home applications on Internet of Things (IoTs), the momentary increase in energy consumption due to *shadow packet* transmissions will be insignificant when compared to the up-time of the network. When spikes are observed at  $P_{wTx}$ , it is seen that  $P_{wRx}$  dips below the steady state power consumption. This is because of three reasons: (i) although multiple *shadow packets* are being sent (increasing  $P_{wTx}$ ), many of them are being

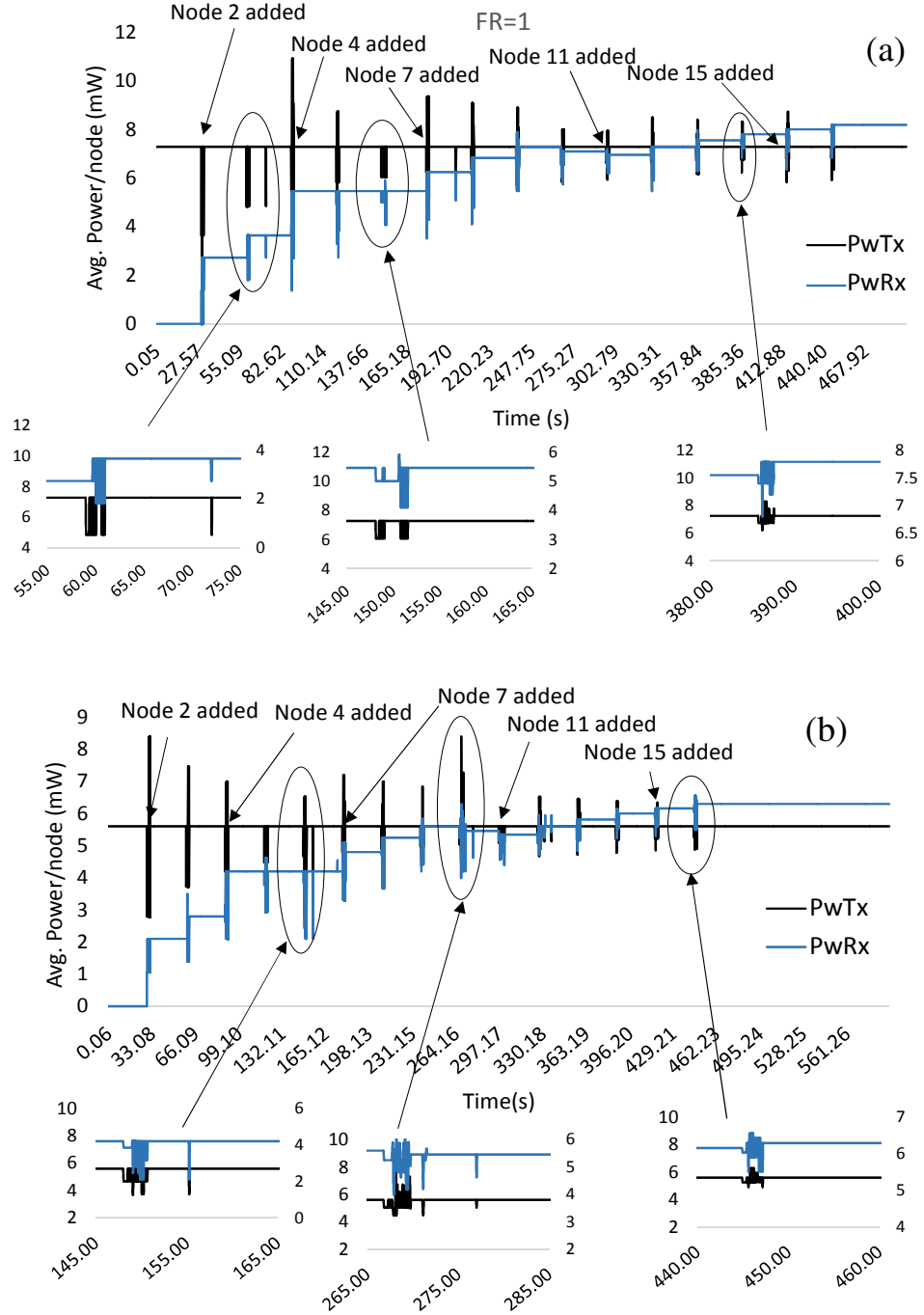


Figure 22. Average transmitter and receiver power consumption for incremental node additions (a) 4x4 grid topology with F-Ratio = 1 (b) 4x4 grid topology with F-Ratio = 1.3

sent to represent the transmission slot of a single node, i.e. at the same time, and hence the total registered reception power is low irrespective of the number of transmission (*shadow*) packets (ii)

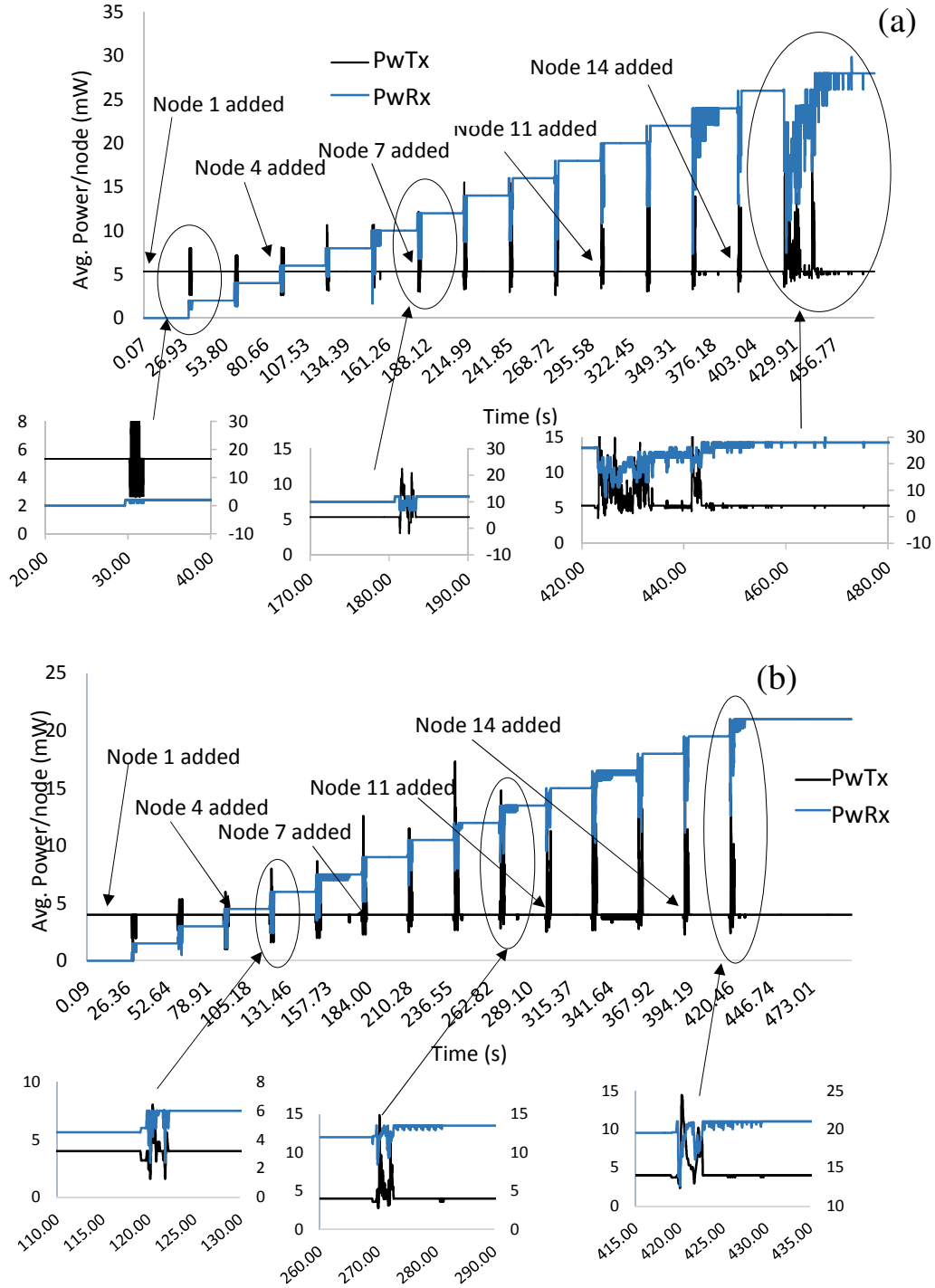


Figure 23. Average transmitter and receiver power consumption for incremental node additions (a) 15 node fully connected topology with F-Ratio = 1 (b) 15 node fully connected topology with F-Ratio = 1.3

when changes in network dynamics are observed, nodes start sending *shadow packets* and start following the bitmap pattern. A gap in the pattern means lesser packets transmitted and received,

and due to asynchronous patterns which may not coincide, an overall lower number of packets are transmitted (iii) to optimize the network and reduce the total number of *shadow packets*, nodes transmit *shadow packets* with a 60% probability, resulting in an overall lower number of packets received. It can also be observed that steady state PwRx ( $PwRx_{steady}$ ) steadily increases with time, but steady state PwTx ( $PwTx_{steady}$ ) remains constant. The reason is that the effective number of receivers increases at a rate that is greater than or equal to the number of transmitters. To illustrate this, when the network size is 5 for a fully connected network, the number of transmitters is 5, and the number of receivers is 4x5. However, when the network size is 15, the number of transmitters is 15, but the number of receivers is 14x15.

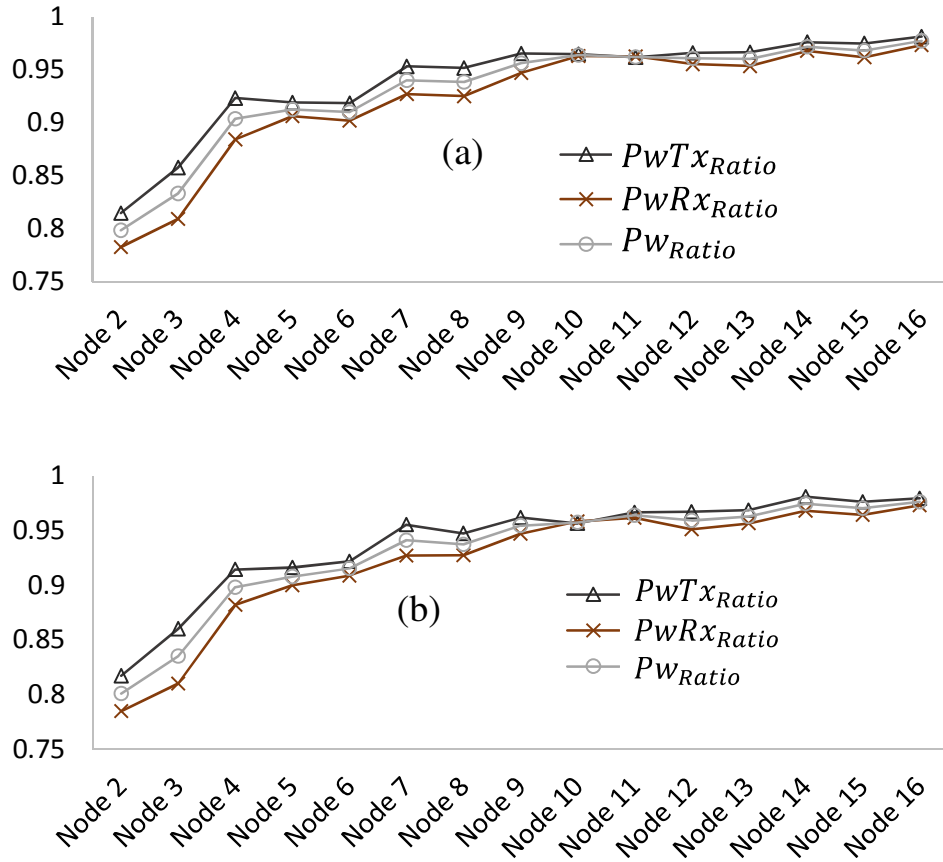


Figure 24. Ratio of transient and steady state power consumption for (a) 4x4 grid topology with  $F\text{-Ratio} = 1$  (b) 4x4 grid topology with  $F\text{-Ratio} = 1.3$

Figure 24 and Figure 25 plots  $PwTx_{ratio}$ ,  $PwRx_{ratio}$ , and  $Pw_{ratio}$  for both the networks from an average of 20 simulation runs for F-ratio = 1 and 1.3. Each data point in the graph denotes the addition of a node in the network and the corresponding power consumption ratio.  $PwTx_{ratio}$  is the ratio of average PwTx during transient phase ( $PwTx_{transient}$ ) and average PwTx during steady state ( $PwTx_{steady}$ ). Similarly,  $PwRx_{ratio} = \frac{PwRx_{transient}}{PwRx_{steady}}$ . Finally,  $Pw_{ratio}$  is given by  $\frac{PwTx_{transient} + PwRx_{transient}}{PwTx_{steady} + PwRx_{steady}}$ . It can be observed that for grid topology, all three parameters plotted are less than 1. This means that the transient state power consumption is always less than the steady state power consumption, which indicates that there is no energy overhead incurred due to the protocol characteristics. However, for a fully connected topology, the  $PwTx_{transient}$  is almost 1.5 times  $PwTx_{steady}$ . This extra expenditure due to additional transmission packets is compensated by the reception power expenditure which is lower during the transient phase due to the reasons stated earlier. The resultant  $Pw_{ratio}$  can be observed to be roughly around 1.1, which indicates a reduced overall overhead in fully connected topology.

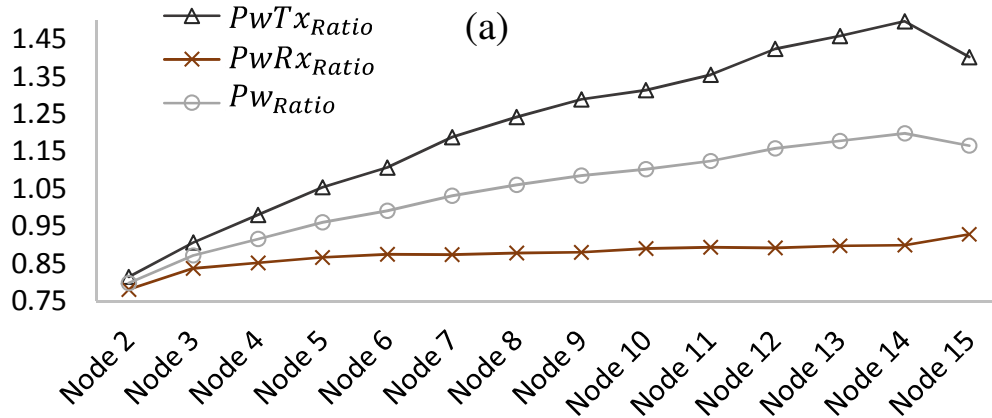


Figure 25. Ratio of transient and steady state power consumption for (a) 15 node fully connected topology with F-Ratio = 1 (b) 15 node fully connected topology with F-Ratio = 1.3

Figure 25 (cont'd)

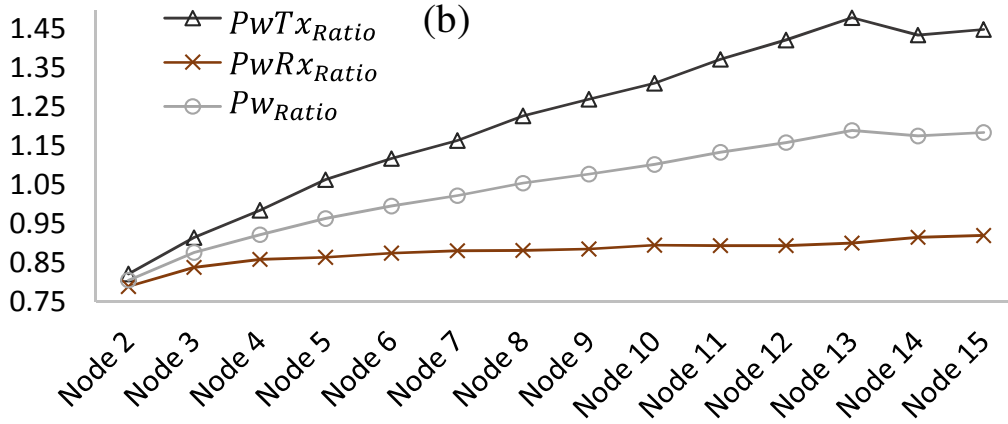


Figure 25. Ratio of transient and steady state power consumption for (a) 15 node fully connected topology with F-Ratio = 1 (b) 15 node fully connected topology with F-Ratio = 1.3

Figure 26 and Figure 27 plots the frequency distribution of  $Pw_{ratio}$  for grid and fully connected topologies with F-ratio = 1 and 1.3. It can be seen that for the tested topologies,  $Pw_{ratio}$  gradually increases as more nodes are added to the network. As more nodes were added, the  $PwTx_{transient}$  increased due to the increase in the *shadow packet* transmissions, as well as the number of collisions needed to be resolved. This can be clearly observed in the figure from the fact that the distributions shift right as more nodes are added to the network. The escalated network perturbation as higher number of nodes were added causes a steady rise in  $Pw_{ratio}$ .

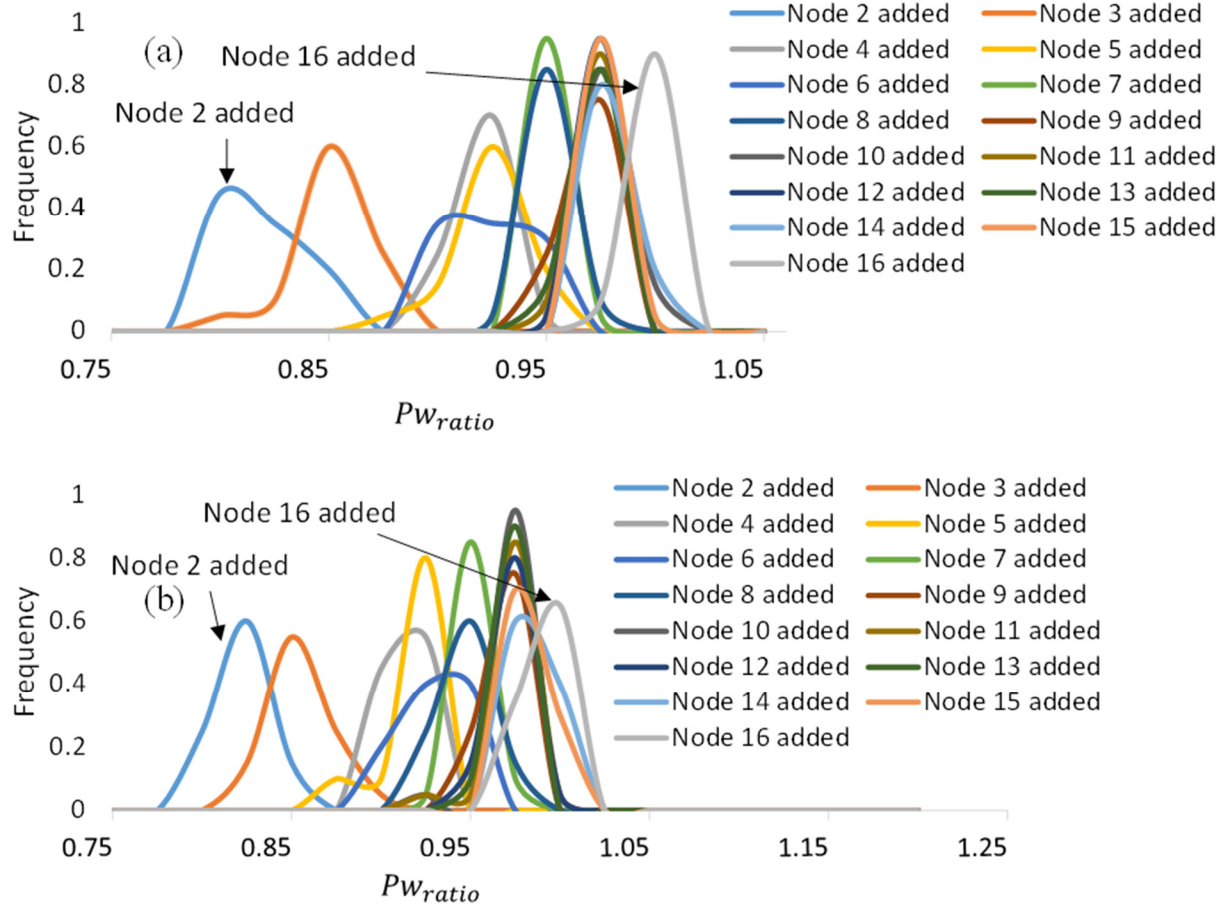


Figure 26. Frequency distribution of transient and steady state power consumption ratio for (a) 4x4 grid topology with  $F\text{-Ratio} = 1$  (b) 4x4 grid topology with  $F\text{-Ratio} = 1.3$

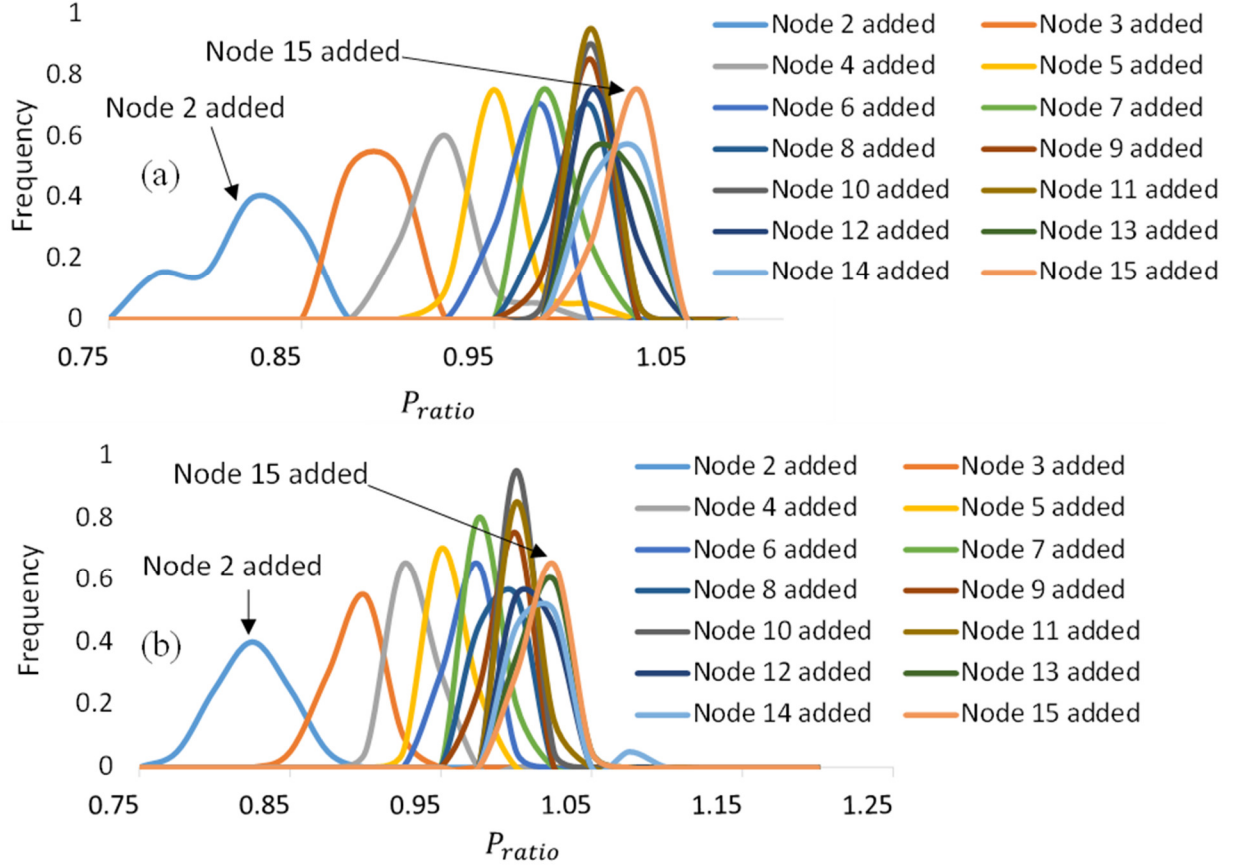


Figure 27. Frequency distribution of transient and steady state power consumption ratio for (a) 15 node fully connected topology with  $F\text{-Ratio} = 1$  (b) 15 node fully connected topology with  $F\text{-Ratio} = 1.3$

## 5.10 Summary

*eZEA*-TDMA paves the pathway for the development of an energy-efficiency algorithm. The developed energy-efficiency module employs a sleep-wake scheduling process for which nodes need to be aware of the slot occupancy of up to their two-hop neighborhood. *eZEA*-TDMA enables this by means of *shadow packet* usage and thus facilitates the energy-efficiency module which operates as an idle-energy saving supplement to the MAC protocol.

## Chapter 6: Energy Aware ZEA-TDMA Design

### 6.1 Introduction

*e*ZEA-TDMA, which is a variant of the ZEA-TDMA protocol using *shadow packets*, is not energy efficient by itself since each node needs to permanently listen and monitor the state of slot allocations in its neighborhood. Therefore, a significant amount of energy is wasted by the wireless interface due to overhearing and idle-listening of the channel [45], [54], [55]. However, unlike ZEA-TDMA, nodes running *e*ZEA-TDMA are aware of their two-hop neighbors' slot locations by virtue of *shadow packets*. This auxiliary information can be utilized to develop an energy efficiency module which can be supplemented to the *e*ZEA-TDMA protocol to implement a sleep-wake scheduling for idle energy saving after the nodes reach steady-state.

The energy efficiency module is not tied to the MAC slot-allocation process and is activated in a node after it has reached steady state. The energy model is developed assuming negligible transmission errors, as well as fixed transmission power, as mentioned in the network model. Additionally, since nodes use fixed transmission power, the energy expended for packet transmission is constant for all nodes. Fixed transmission power also ensures no topological variability due to transmission range fluctuations.

### 6.2 Sleep-Wake Scheduling

The basic idea of an energy management protocol is to devise a duty cycling technique to save energy by switching nodes between sleep and awake states. Since nodes can only decipher packets from neighbors which belong to its trust domain, it can sleep during the slots of neighbors belonging to a different trust domain. Figure 28 shows an example sleep-wake scenario in a network with multiple trust domains, where a node sleeps when: (i) there is no channel activity,

and (ii) during unreadable message transmissions; and remains awake during (i) its own slot time, and (ii) the slot times of its neighbors belonging to the same trust domain. The following subsection presents an energy efficiency protocol that minimizes the idle listening and overhearing problem among nodes within the same trust domain.

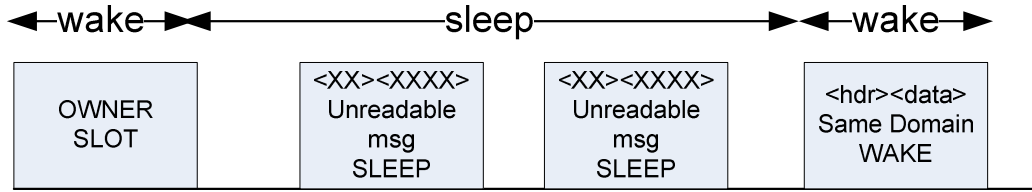


Figure 28. Sleep-wake schedule followed by a node in a multi-domain network: node only wakes up to receive packets from neighbors belonging to the same domain. It goes to sleep when i) there is no transmission in the network ii) transmission slots in which it cannot decrypt or decipher the message

### 6.3 Protocol Description

After a node reaches steady-state, it remains awake for a predefined amount of ‘buffer-time’,  $P$ , before entering the energy management stage. During  $P$ , it does the following. It continues to send *regular packets* in its designated slot and monitors the network for any changes in dynamics. Changes in network dynamics are determined by comparing the steady state slot occupancy list with the current network state. In addition, the node also creates a list of neighbors which belong to the same trust domain as itself. This can be done by attempting to decrypt packets that are received during the steady state and checking for correct header information, or some other similar criterion. When a packet can be correctly decrypted using the pre-shared trust domain key, the slot in which the packet was sent can be labeled as a slot belonging to a node in the same trust domain.

To eliminate the overhearing problem, each node needs to have a wakeup slot, where the node listens for *wakeup packets*. As the name suggests, the wakeup slot is a slot used to wake a node

from its sleep state. Each node checks its current slot occupancy list and randomly selects a slot unused by any of its two-hop neighbors as its wakeup slot. It then includes the relative location of its wakeup slot time in its *regular packet* header. One-hop neighbors of the node which belong to the same trust-domain calculates the actual wakeup slot location of the node from the relative time in the header and the *regular packet* transmission time and stores it with the corresponding node-identifier. If the node does not detect any changes in the network for  $\mathcal{P}$ , it goes into the energy-management stage. In the energy-management stage, a node is already aware of the slots belonging to nodes in its trust domain using the list assimilated during  $\mathcal{P}$ , and hence does not need to decrypt all received packets.

Once in the energy-management stage, the node follows a sleep-wake scheduling to eliminate energy wasted due to idle listening or overhearing. If a node does not have any packets to send during a frame, or if the node is not receiving packets from any other neighbors belonging to the same trust domain, it only wakes up during its wakeup slot in the frame.

The pseudocode for message transmission is shown in Algorithm 2 and the procedure is described below:

1. If node  $A$  has a packet to send to node  $B$ , it first checks a local-variable  $lastMsgSent_B$ , which stores the time when a message was last sent to  $B$ . If the time difference between the current time and  $lastMsgSent_B$  is longer than a given threshold ( $\delta$ ), it sends a *wakeup packet* during node  $B$ 's *wakeup slot*.
2. Node  $A$  then sends the *regular packet* during its own slot in the next frame and also updates  $lastMsgSent_B$  with the current time.

3. When node  $B$  receives a *wakeup packet* from  $A$ , it first checks for its own node ID in the destination field. If  $B$  finds that the packet was destined for it, it retrieves  $A$ 's slot time from the steady state slot occupancy list and stays awake during  $A$ 's slot in the next frame.
4. Once  $B$  receives a packet from  $A$ , it updates a local-variable  $lastMsgReceived_A$  with the current time.
5. Node  $B$  stays awake in  $A$ 's slot in every consecutive frame until the difference between the current time and  $lastMsgReceived_A$  reaches a given threshold value ( $\delta$ ).
6. Apart from the *wakeup slot* and slots used to send or receive packets, the nodes remain asleep in all other slots in a frame.

The parameter  $\delta$  should be chosen carefully since it directly impacts the energy consumption of the sender and receiver nodes. Higher values of  $\delta$  would increase the idle listening time for the receiver node. Consequently, smaller values of  $\delta$  means that the sender node has to send *wakeup packets* more frequently for traffic patterns with short inter-burst time.

The proposed energy management framework is similar to that proposed in TDMA-W [16]. The notable advantages of our proposed framework however are addressing: a) *wakeup packet* loss due to wakeup slot sharing, and b) network dynamism to support node-additions even when nodes are in energy management stage, both of which are discussed next.

---

### Algorithm 2

---

#### Initial: Node A

##### Sender: Node A

*/\* packet for dest<sub>B</sub> arrives from upper layer \*/*

**if**(*lastMsgSent<sub>B</sub> > Th<sub>send</sub>*)

    | wait for B's wakeup slot  
    | send wakeup packet to B  
    | wait for end of current TDMA frame

**if**( *CURRENT\_TIME % T = my\_slot\_time* )

    | send **regularPacket**  
    | *lastMsgSent<sub>B</sub> = CURRENT\_TIME*

##### Receiver: Node B

*/\* receive wakeupPacket from source<sub>A</sub> in wakeup slot \*/*

**if**( *wakeupPacket[dest\_field] = my\_address* )

    | wait for end of current TDMA frame  
    | **if**( *CURRENT\_TIME % T = source<sub>A</sub>\_slot\_time* )  
        | wakeup  
        | receive message from source<sub>A</sub>  
        | *lastMsgReceived<sub>A</sub> = CURRENT\_TIME*

*/\* check source<sub>i</sub>\_slot\_time in every frame,  $\forall$  neighbors source<sub>i</sub> \*/*

**if**( *CURRENT\_TIME % T = source<sub>i</sub>\_slot\_time* &&

*CURRENT\_TIME - lastMsgReceived<sub>i</sub> < Th<sub>receive</sub>* )

    | wakeup  
    | else  
    | sleep

---

**Sender and Receiver-side pseudo-code for message transmission**

---

## 6.4 Shared Wakeup Slots and Two-Way Handshake

Since nodes select wakeup slots independently, it can be possible for nodes to select partially or completely overlapping slots, resulting in sharing of wakeup slots. A shared wakeup slot is

allowed since a node always knows from the destination field of a packet if it is the intended receiver. However, a node may receive a corrupt or non-decipherable packet in the wakeup slot. This can happen when nodes in different trust domains share wakeup slots or when multiple senders send wakeup packets to the same node in the same frame. In this case, a node remains awake for the next frame and if it receives packets, it stays awake in the corresponding senders' slots from the following frames. Conversely, if the node does not receive any message, it means that the wakeup packet was intended for a receiver belonging to a different trust-domain. In that case, the node goes to sleep again from the next frame onwards.

The main issue of using shared wake up slot arises when the intended recipient node also tries to send data. Let us consider a situation where  $e_{ij} \in E \wedge e_{jk} \in E \wedge w_i = w_j$ , where  $w_i$  represents the wakeup slot of node  $v_i$ . If  $v_j$  tries to wake  $v_i$  at the same frame  $v_k$  tries to wake  $v_j$ , then  $v_j$  will 'miss' the wakeup packet sent by  $v_k$ . This leads to unreliable packet delivery since  $v_k$  assumes  $v_j$  is awake and sends data packets for  $v_j$  in its own slot. We address this issue by introducing acknowledgement ACK-W for wakeup packets. When  $v_i$  sends a wakeup packet for  $v_j$  in  $w_j$  of frame  $f$ ,  $v_j$  sends an ACK-W packet in  $w_j$  in frame  $f+1$ , indicating the receipt of the wakeup packet. To receive the ACK-W successfully,  $v_i$  stays awake in  $w_j$  in frame  $f+1$  if it had already sent a wakeup packet in  $w_j$  of frame  $f$ . This combination of wakeup packet and ACK-W (two-way handshake) ensures a sender node that the destination node is awake. In case node  $v_i$  does not receive an ACK-W at  $w_j$ , it tries to initiate the two-way handshake again at frame  $f+2$ . This process is repeated  $\sigma$ -times after which a node gives up. The value of  $\sigma$  can be decided based on the application and network requirements. The two-way handshake provides resilience towards

wakeup packet losses or unnecessary transmissions when the destination node has left the network.

The two-way handshake process is demonstrated in Figure 29.

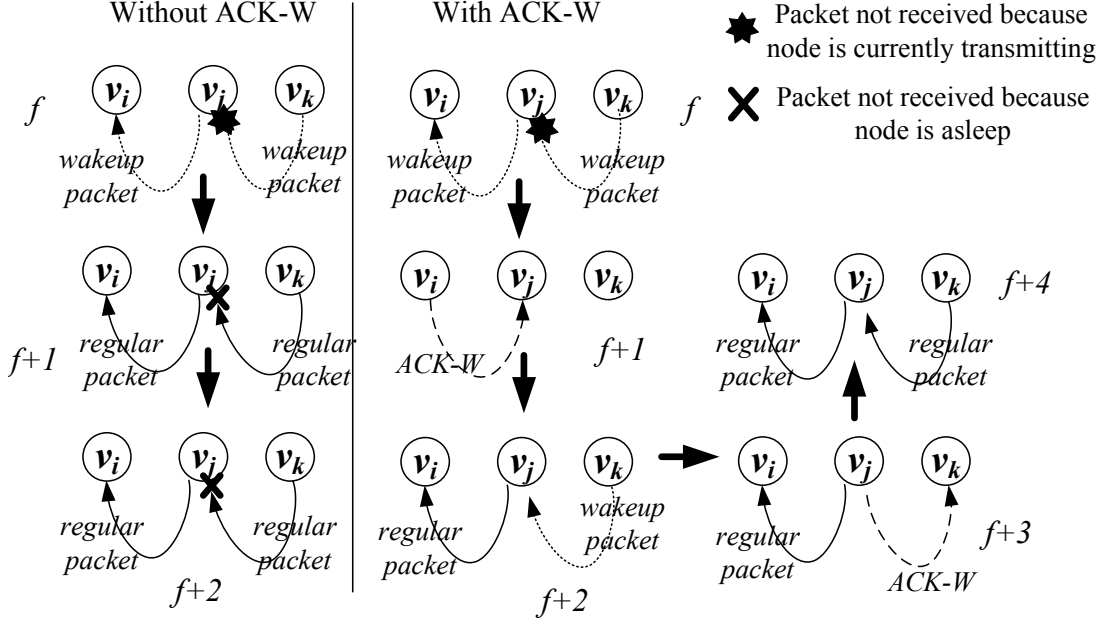


Figure 29. Two-way handshake for reliable wakeup

## 6.5 Network Dynamics

To support network dynamics, a node has to perform a rapid switching from energy-management mode to slot-allocation mode when it notices any transmissions from a new node in the network. When in energy-management mode, the node maintains a regular sleep-wake scheduling cycle as mentioned earlier. It also periodically remains awake for  $q$  consecutive frames after every  $\omega$  frames. During these  $q$  frames, the node remains awake and listens to the network and checks for any events of network topology changes. It also sends a *regular packet* during its own slot in each of those  $q$  frames. The requirements for  $q$  and  $\omega$  are as follows:

1.  $q$  should be long enough so that any new node sends at least one transmission in  $q$  frames, i.e.  $q$  should be greater than the number of 0's in the bitmap pattern.

2.  $\omega$  should be short enough so that a new node does not change from *slot-allocation mode* to the *energy-management mode* within  $\omega$  frames, i.e.  $\omega$  should be greater than  $\Omega + \mathcal{P}$ .

Ideally,  $\omega \gg q$

If a node receives a new transmission during the  $q$  frames, it immediately changes from energy-management mode to slot-allocation mode. It starts sending *regular packets* and *shadow packets* to inform the new node about its own slot location and the slot locations of its one-hop neighbors. The node also cross-verifies the new node's slot location with its steady state slot allocation list to see if the new node has selected an overlapping slot, and if an overlapping slot selection is detected, it sends an *interrupt packet* to resolve the collision. Once steady-state is reached, it adds the new node's relative slot location in its steady-state slot-allocation list. The new node also creates its own steady state slot allocation list after it reaches the steady state. After this, the nodes in slot-allocation mode remain awake for  $\mathcal{P}$  before switching back to energy-management mode.

The state machine of the ZEA-TDMA protocol with energy efficiency module is presented in Figure 30. After a node reaches steady state, it switches to energy-management mode once the buffer-time has elapsed. In energy-management mode, it monitors the network every  $\omega$  frames for  $q$  consecutive frames. If there are any new transmissions detected in the one-hop neighborhood, it starts the slot-allocation algorithm. Otherwise, it goes back to the energy-management mode.

We develop an analytical model in Appendix C, which estimates the power consumption by the transmitter and the receiver when energy management is used at steady state.

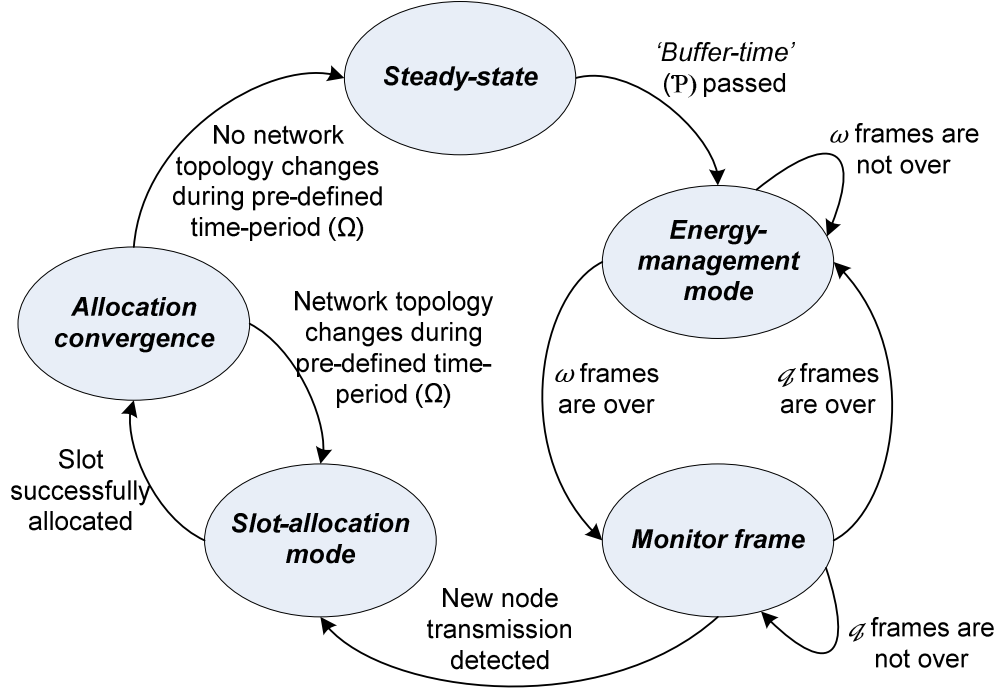


Figure 30. State machine for eZEA-TDMA with energy-efficiency module

## 6.6 Functional Limitations of the ZEA-TDMA-family

eZEA-TDMA suffers from the following drawbacks which are the price for information-free MAC slot scheduling. First, using a bitmap pattern results in loss of throughput during the transient phase, since nodes skip transmitting *regular packets* when there is a ‘0’ in the pattern. A longer pattern length (i.e. one ‘0’ and a large number of ‘1’s) can greatly reduce the bandwidth loss; however, this also means longer duration needed for a node to learn the bitmap pattern of other nodes. Therefore, the pattern should be chosen carefully in order to strike a balance between the above two factors. The shortest possible pattern ‘10’ is not feasible since in this case, from an observer node’s standpoint, both *shadow* and *regular packets* appear in the same sequence. This prevents the observer node from detecting (and resolving) a *regular-regular* collision. It turns out that the length of the shortest functionally feasible pattern is 3-bits. A 3-bit pattern of 101 has been used in the implementation. With ‘101’, the bandwidth loss is 33% between when a node starts

sending *shadow packets* due to a network perturbation, and when it stops sending them. Using an 11-bit pattern (e.g., ‘11111011111’), for example, would reduce the bandwidth loss to 9%, but would increase the time to learn the pattern by almost four times. Note that there is no relationship between the length of the pattern and the maximum two-hop node degree  $D$ . For example, the length of the pattern can be as short as 101 and  $D$  can be as high as 100, as long as the TDMA frame is large enough to accommodate slots of all the nodes, i.e.  $T \geq D\tau$ . The incurred bandwidth loss due to the bitmap pattern can also be counteracted by using *shadow packets* for sending user application data since their content are not actually used for the protocol to work.

Second, since nodes maintain their individual slot occupancy lists based on received transmissions, when a node  $v_i$  does not have packets to transmit, its neighbors may assume that the given node has failed or has left the network. As mentioned in [Section 5.8](#), if a new node occupies  $v_i$ ’s slot in such a situation,  $v_i$  has to undergo the slot allocation process afresh. This can be inefficient for highly intermittent traffic pattern along with frequent node additions. However, there are two different ways to address this issue, which had been omitted in the protocol description to avoid further complexity. In the first mechanism, nodes can send dummy *regular packets* when there is no traffic. This fills the gap in traffic, maintains a regular and periodic traffic, and guarantees transmission slots whenever there is traffic. However, this method can be energy inefficient for resource constrained wireless networks. In the second solution, the inactivity of a node due to a gap in the traffic is probabilistically determined using a threshold value chosen based on the traffic pattern. To elaborate, if for a traffic pattern, the average burst size is  $\beta$  and the average inter-burst interval is  $\alpha$ , then a node needs to undergo the slot allocation process if the gap in its traffic exceeds  $\alpha - \beta$ . Following the same, when the one-hop neighbors of a node do not receive any transmission from it for the duration  $\alpha - \beta$ , they delete its entry from their slot occupancy list.

This can also address frequent node failures since the node can reuse its slot if it revives within  $\alpha - \beta$  duration. This solution, however, assumes prior knowledge of the traffic pattern. Handling inefficiencies due to intermittent traffic in non-deterministic traffic patterns is beyond the scope of this work.

Third, a byproduct of TDMA slot allocation without any message-based coordination is the extended convergence time. However, since *e*ZEA-TDMA and ZEA-TDMA do not have an explicit control phase, nodes can start transmitting data right from when it is introduced in the network. Although packet delivery is unreliable due to collisions before a slot has been assigned, this issue can be addressed by using mechanisms for guaranteed packet delivery available to upper layer protocols. Additionally, it was observed during evaluation that convergence times were typically longer when all nodes were introduced in the network at the same time. Nodes converged faster in a more dynamic scenario, i.e. when they were introduced one at a time. Such incremental node addition is also a more realistic assumption in wireless environments where nodes join or leave the network at different times. These results have been illustrated in the following simulation experiments.

Finally, an assumption made for the energy management model was that of an ideal channel. If a channel with negligible bit error rate is unavailable, the energy model can be adapted to withstand unreliable packet delivery by using an acknowledgement based mechanism. However, for simplicity sake, the details of the mechanism have not been discussed in this thesis.

## 6.7 Evaluation

A steady state power consumption model has been developed in the Appendix C and provides an insight on the power consumption of the transmitter and receiver nodes at steady state with the developed energy efficiency module enabled. In this section, we compare the results obtained from

our model to simulation results. The energy efficiency protocol was simulated using a Java-based time-driven simulator. For simplicity, the bit-error rate was considered to be negligible. The traffic was modeled as a Poisson process where the traffic rate and inter-burst interval were exponentially distributed with means  $\lambda$  and  $\frac{\beta}{\lambda}$ , respectively. The values used for different parameters in the model and simulation are listed in Table 2.  $p^t$ ,  $p^r$  and  $p^s$  values are based on a Mica2 mote transmitting at full transmission power. The TDMA frame size was 1.23 s (i.e. 300 slots), and the maximum one-hop node degree  $d$  was fixed at 6.

Table 2: Network, Protocol and Simulation parameters

Network parameters	
$n$	300 slots
$d$	6
Protocol parameters	
$\delta$	10
Node transceiver parameters	
$p^t$	81mW
$p^r$	30mW
$p^s$	0.003mW
Simulator parameters	
$\tau$	4 ms
$N ( \mathcal{N} )$	20-100 nodes
Simulation time	30000 s
Iterations	500

For the simulation, the network size and the node degree for each node was randomly selected for each run and was between 20 to 100 nodes, and 0 to 6 edges, respectively. The total number of iterations for each combination of  $\lambda$  and  $\beta$  was 500, and the reported value is the mean value of all the iterations. The results have been divided into two parts, first, where the extra power consumption due to reception of corrupt wakeup packet has not been considered, i.e. without wakeup overhead, and second, where the extra power consumption has been taken into account,

i.e. with wakeup overhead. The results demonstrate how the power consumption at the transmitter ( $P^{tx}$ ) and receiver ( $P^{rx}$ ) change when the size of each burst ( $\beta$ ) and rate at which packets are sent ( $\lambda$ ) are varied.  $P^{tx}$  and  $P^{rx}$  for the model have been obtained using equations (29) and (32) respectively, in Appendix C.

### 6.7.1 Without Wakeup Overhead

When a node receives a corrupt wakeup packet in its wakeup slot, it remains awake for all slots in the next TDMA frame to determine if it was the intended receiver for the wakeup packet. This energy wasted due to idle listening is termed as wakeup overhead and was ignored in the model development. In this section, to make a fair comparison between the model and the simulation results, the wakeup overhead is ignored in the simulation. As it can be observed from Figure 31, the simulation results closely match the results obtained from the developed model.

**Increasing traffic rate ( $\lambda$ ) and constant burst size ( $\beta$ ).** The traffic rate is directly proportional to almost all the power consumption components except  $P_w^{tx}$  (28) and  $P_w^{rx}$  (31), which are constant irrespective of  $\beta$  or  $\lambda$ . As the rate at which packets are sent increases, the sender consumes more power to send those packets. Also,  $P^{rx}$  is higher than  $P^{tx}$  since the receiver consumes more power to remain awake in the transmission slots of all its transmitters, unlike the transmitters which only wake up once during its own transmission slot.  $\lambda$  has a lower effect on  $P_{w_t}^{tx}$  (26) because the sender has to wake up its receivers once every inter-burst interval, which in turn depends on the burst size, and a constant burst-size dampens the effect of  $\lambda$ . The plotted curves have a ‘knee’ effect due to the exponentially increasing values of  $\lambda$ , i.e.  $\lambda$  increases exponentially as it is increased from  $\frac{1}{200}$  to  $\frac{1}{100}$  and so on.

**Increasing burst size ( $\beta$ ) and constant traffic rate ( $\lambda$ ).** The burst size has an inverse effect on the power consumption of the receiver and the transmitter. The effect of burst-size on the receiver is much higher than that on the transmitter. This is because for lower burst-sizes, the receiver needs to wake up more often. After each burst, the receiver remains awake for  $\delta$  additional frames and as the burst-size increases, the effect of  $\delta$ , which is a constant, is reduced.  $P^{tx}$  is affected by  $P_{w_t}^{tx}$ , which is higher for extremely low burst sizes, but diminishes as the burst-size is increased. This is because the transmitter needs to wake up its receivers only once before each burst, and with higher  $\beta$ , the power consumed to do this reduces as the transmitter needs to send wakeup packets less frequently. Also, the burst-size has no effect on the transmission of packets, and hence  $P_{t_\beta}^{tx}$  remains constant, affecting  $P^{tx}$  as well.

**Increasing burst size ( $\beta$ ) and decreasing traffic rate ( $\lambda$ ).** As the burst size increases and traffic rate decreases, the power consumed at both the transmitter and receiver decreases. This is because reduced traffic rate means reduced power consumption due to transmission and reception of packets. And, increased burst size means reduced overhead due to less number of wakeup packet transmission at the transmitter. Additionally, since the receiver needs to remain awake for a constant period  $\delta$ , as the burst-size increases, the overhead due to  $\delta$  decreases in comparison.

**Increasing burst size ( $\beta$ ) and increasing traffic rate ( $\lambda$ ).** Due to the increased rate at which packets are sent, increasing  $\lambda$  increases the power consumption at both the transmitter and receiver. In contrast, with an increase in burst-size, the power consumption decreases due to the reduced overhead at the transmitter due to reduced wakeup packet sending cost, and at the receiver in the form of reduced additional wakeup cost ( $\delta$ ). Both these factors have a competing effect on each other and the optimum values for traffic rate and burst size can be observed when the power consumption is minimum.

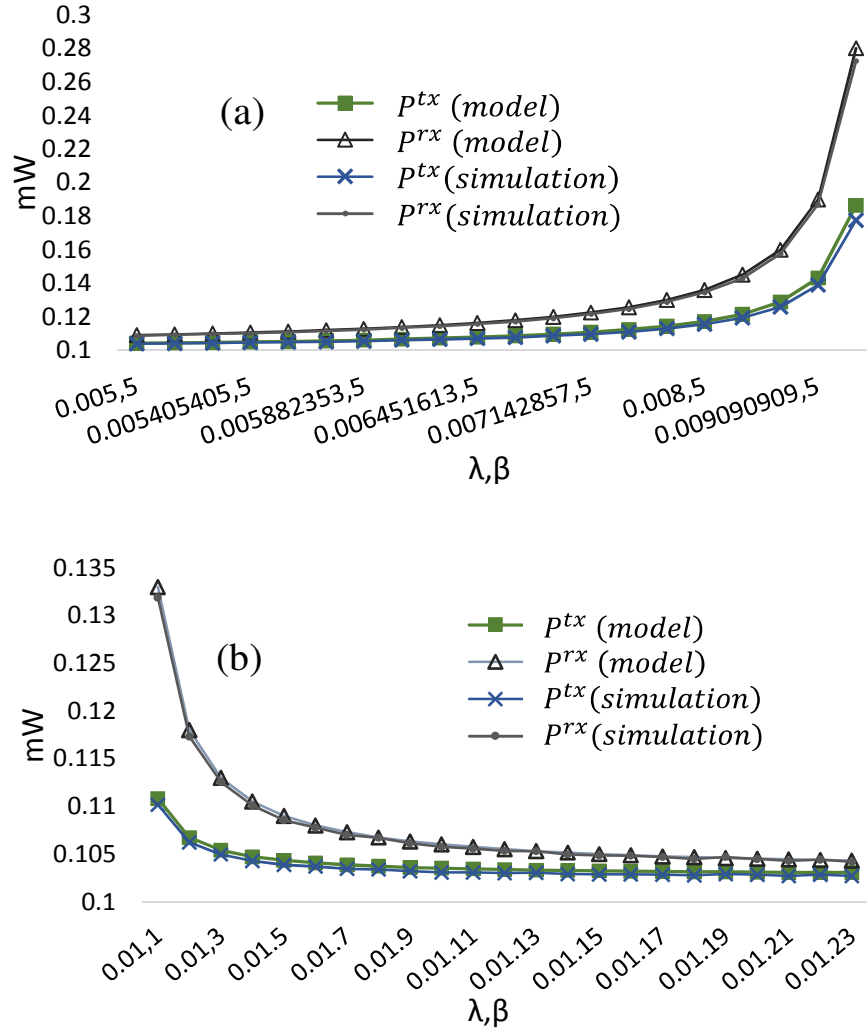


Figure 31. Without wakeup overhead (a) Increasing traffic rate and constant burst size (b) Increasing burst size and constant traffic rate

Figure 31 (cont'd)

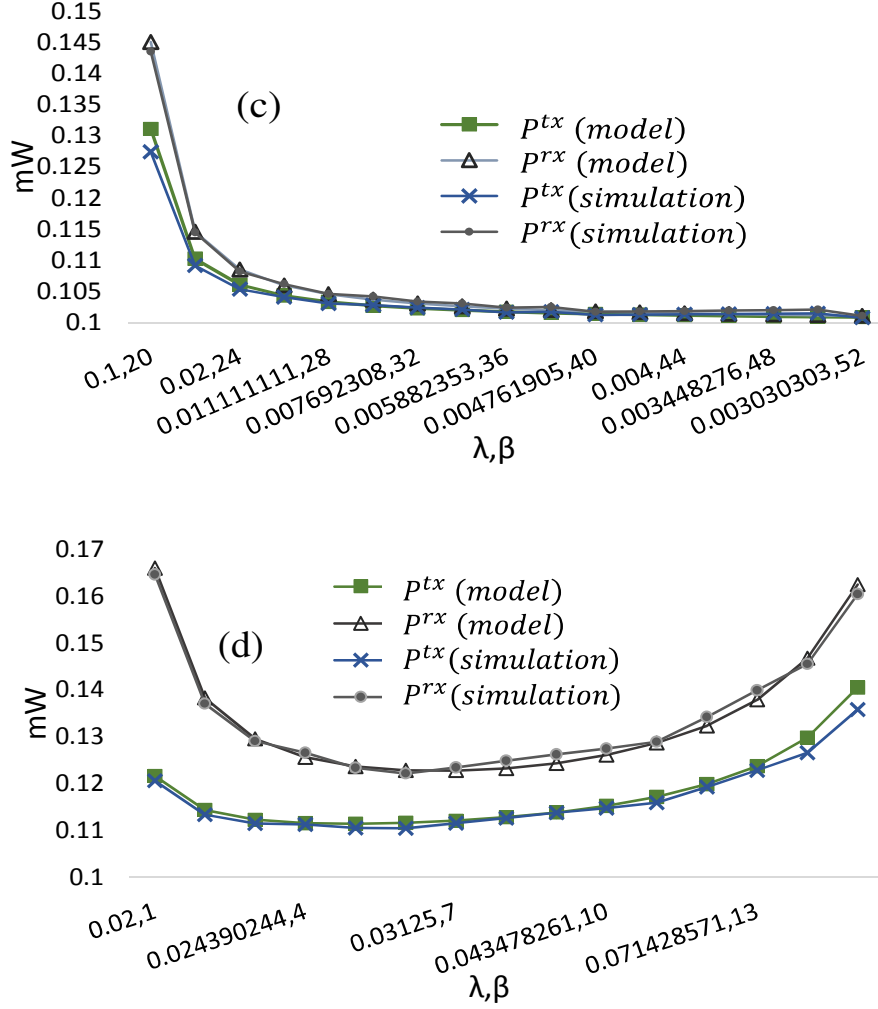


Figure 31. Without wakeup overhead (c) Increasing burst size and decreasing traffic rate (d) Increasing burst size and increasing traffic rate

### 6.7.2 With Wakeup Overhead

In this section we discuss the simulation results where the effect of wakeup overhead has been taken into consideration, shown in Figure 32. It is intuitive that the wakeup overhead only affects the receiver power consumption. In the figures it is observed that the transmitter power consumption remains the same, while the receiver power obtained from the simulation is higher.

Another observation that can be made is that as the ratio  $\frac{\beta}{\lambda}$  increases, the overhead due to corrupt

wakeup packet reduces. The reason behind this is that with the increase in the ratio, the inter-burst interval also increases. This causes (i) the frequency of occurrence of a corrupt wakeup packet to reduce, and (ii) the probability of multiple transmitter trying to communicate with the same receiver to decrease, in turn dropping the overhead. This was the conjecture based on which the wakeup overhead was ignored in the model development.

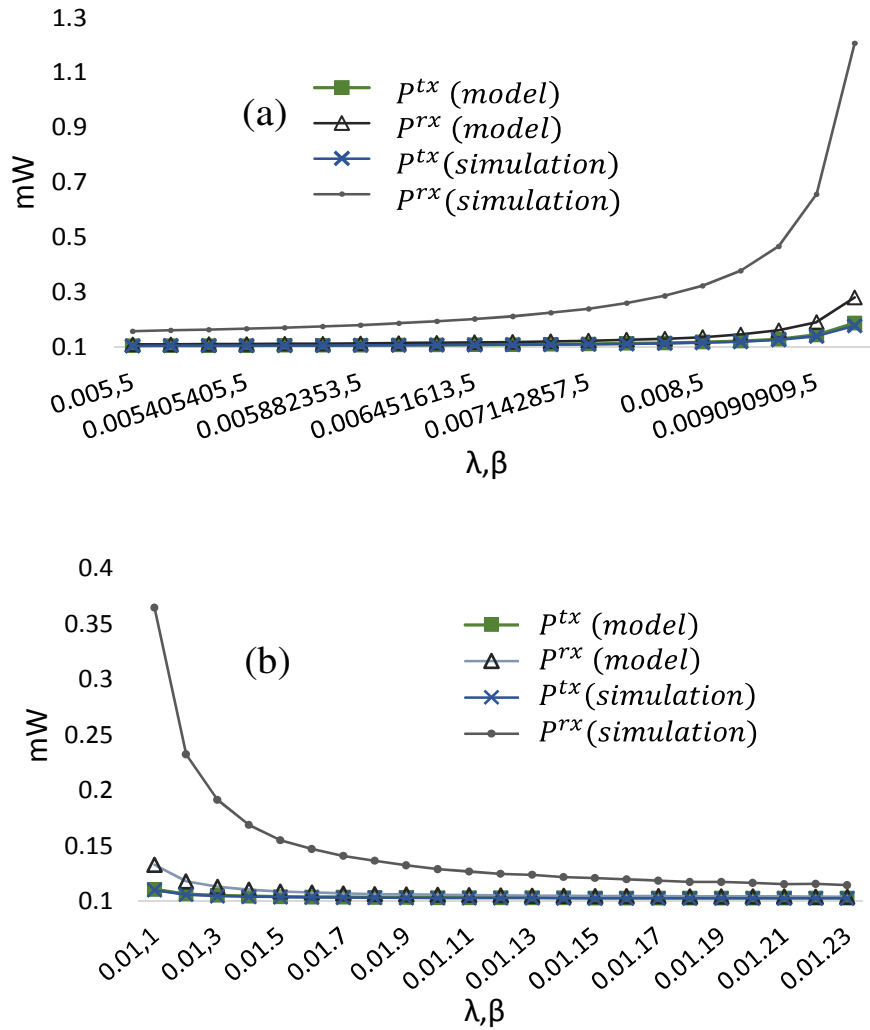


Figure 32. With wakeup overhead (a) Increasing traffic rate and constant burst size (b) Increasing burst size and constant traffic rate

Figure 32 (cont'd)

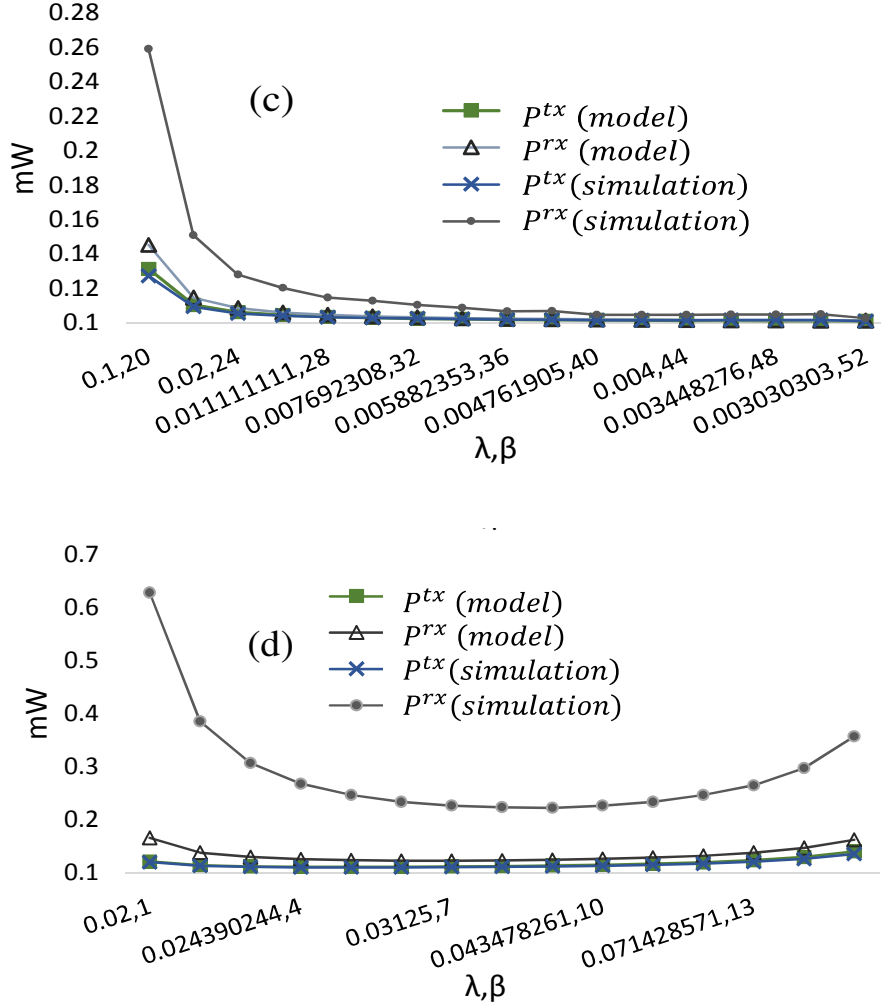


Figure 32. With wakeup overhead (c) Increasing burst size and decreasing traffic rate (d) Increasing burst size and increasing traffic rate

## 6.8 Summary

This chapter presents an energy-efficiency and management module which can be supplemented to the *e*ZEA-TDMA protocol to implement a sleep-wake scheduling for idle energy saving after the nodes reach steady-state. Additionally, a model has also been developed which analyzes the energy consumption when energy management is used. A Java-based simulator was developed to simulate the energy efficiency protocol and the results were found to be very close

to those obtained by the analytical model. In the following chapters, we implement the ZEA-TDMA on a real-life wireless environment along with the collision detection module which was assumed to be available during the protocol simulation.

## Chapter 7: Hardware System for Third-party Collision Detection

### 7.1 Introduction

As discussed in the previous chapters, in wireless networks, the major cause of collisions is due to the hidden terminal problem. The hidden terminal problem occurs when two nodes, which are two-hop neighbors of each other, try to send a message simultaneously. Both the messages collide at the common neighbor resulting in a hidden collision and, consequently, the message being dropped by the common neighbor. ZEA-TDMA and eZEA-TDMA were developed under the assumption that the common neighbor can detect the collision and inform the two transmitters, thereby resolving the collision. In this chapter, we develop a hardware module which enables the third-party collision detection and resolution mechanism proposed in our protocol.

### 7.2 Collision Detection

As discussed in the previous chapters, the main principle behind collision detection process in the proposed protocol is to ascertain the duration of the received radio signal. Since all packets are assumed to be of fixed length, the duration of the radio signal due to packet reception is same as the slot size  $\tau$ . Now, a collision occurs when two packets overlap with each other, corrupting the data at the common neighbor. When two packets are partially overlapping, the duration of the colliding slot is  $\tau'$ , where  $\tau' > \tau$ , and can be determined from the received signal duration. The common neighbor, which is the third-party here, can detect that there has been a collision, by measuring the duration of each received signal. As shown in Figure 33, a wireless node equipped with a hardware module capable of detecting the start and end of a received signal, using some threshold based mechanism, would be able to measure the duration of the signal. A collision is detected if the duration of received signal is greater than the known packet duration  $\tau$ , after which the collision resolution mechanism is triggered. The collision, however, cannot be detected if the

packets are completely overlapping, i.e.  $\tau' = \tau$ . Due to the lack of slot or time-synchronization and the randomness of the slot selection process, such situations are extremely rare as nodes rarely select slots that are exactly overlapping with each other. However, to make such situations highly unlikely, all nodes add a random jitter after a constant duration:  $c_1 \times T$ , to further reduce the probability of complete overlap.

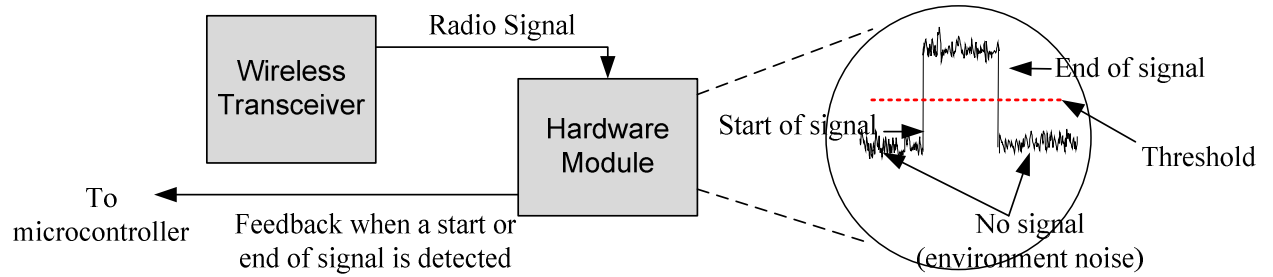


Figure 33. Hardware module using a simple threshold based mechanism to detect the duration of the received signal

### 7.3 Collision Resolution

The collision resolution process has also been discussed in the previous chapters. To summarize, once a collision is detected, the node which detects the collision waits for  $c_2$  frames to make sure that the collision is not a temporary occurrence and is actually due to the usage of two overlapping TDMA slots by its neighbors. Since the two transmitting nodes are unaware of the collision, they continue using their selected slots, which in turn results in a collided slot of the same duration  $\tau'$ , and at the exact same time in the TDMA frame of the common neighbor. If the collision occurs for a predefined number of frames  $c_2$ , the node sends an *interrupt packet* to resolve the collision. The *interrupt packet* is of the same length  $\tau$  as the *regular packet* and is sent exactly after the beginning of the collided slot. As a result, when the node, using the latter part of the collided slot, tries to send its *regular packet*, it senses the channel to be busy due to the *interrupt packet* and defers its transmission to send the *regular packet* only after the channel becomes free. It then uses

this deferred transmission time as its new slot time for future frames. Since the *interrupt packet* is the same size as the *regular packet*, the new slot time no longer overlaps with the slot of the node which is using the former part of the collided slot.

#### 7.4 Collision Detection Sub-system

The following collision detection hardware module was developed for 900MHz Mica2 motes which are used for the wireless test-bed evaluation of ZEA-TDMA. The CC1000 radio chip in Mica2 has a Received Signal Strength Indicator (RSSI) which gives an analogue output signal at the RSSI/IF pin of the chip. The output current of this pin, which is inversely proportional to the input signal level, is converted to RSSI voltage ( $V_{RSSI}$ ) which ranges between 0 – 1.2V. This voltage is then measured by the ADC channel of the Atmega128L micro-controller of the Mica2 mote. The RSSI voltage can be converted to the received signal power using the equation:

$$P = -50.0 \times V_{RSSI} - 45.5 [dBm] \quad (2)$$

The protocol has been developed on top of the default CSMA-based MAC protocol for the CC1000 in Tinyos 2.x. We intentionally disabled the ACK, random backoff, and carrier sense in the default MAC implementation, and provide similar services in our implementation.

Figure 34(a) shows the oscilloscope screen-capture of the RSSI power and the corresponding  $V_{RSSI}$  measured at the ADC channel of a receiver node receiving a 64 byte packet. This is the only visible information that is available to the neighbors of the transmitting node. If the ambient noise is disregarded by a threshold mechanism, the slot size  $\tau$  is given by the duration between the rising edge of the signal, and the falling edge of the signal, which gives  $\tau = 27$  ms. Figure 34(b) shows the received power at node B when two nodes, A and C, are transmitting simultaneously. In the figure,  $t_1$  indicates the start of node A's slot,  $t_2$  indicates the start of node C's slot, and  $t_3 - t_2 = 38$

ms is the duration of the collided slot  $\tau'$ . The second spike at  $t_2$  is because the received power due to node C's transmission is higher than that of node A's transmission. Although the transmission powers were same for the two nodes, the received power can be affected due to multiple reasons, including distance from the transmitter, environmental conditions, etc. Figure 34(c) shows unresolved collisions occurring after  $T$  duration, since the transmitting nodes continue using their original slots.

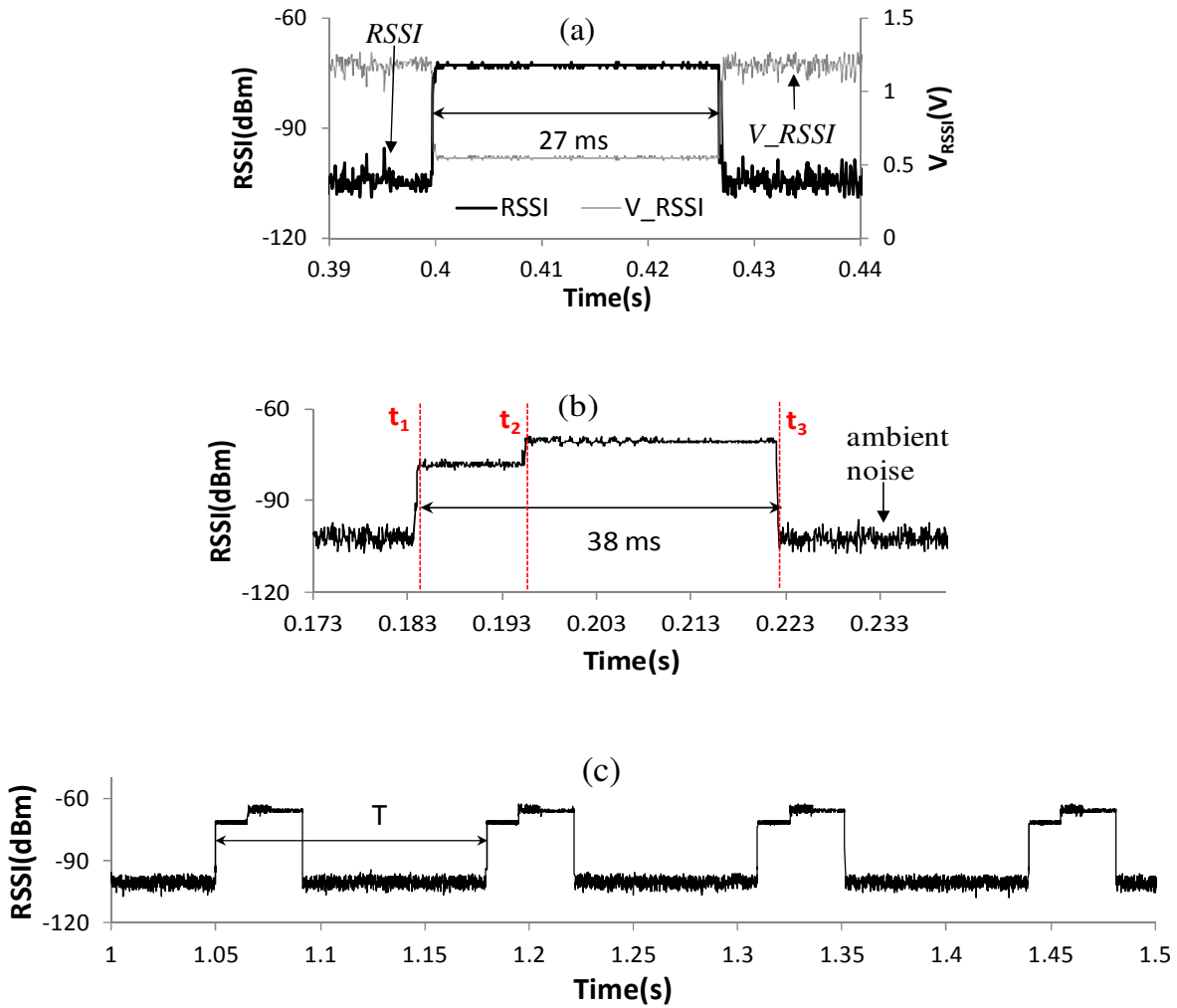


Figure 34. Collision signal – only visible information: Oscilloscope screen-capture of RSSI ( $-50.0 \times V_{RSSI} - 45.5$  [dBm]) power at a receiver for (a) Transmission signal for 512 bit packet, (b) Collision between two 512 bit packets where the second transmitter has higher relative power at receiver (c) Unresolved collision across multiple frames (Slot: 30ms, TDMA Frame: 130ms)

For collisions to be detected accurately, the start and the end of a transmission need to be identified precisely and instantly. The current carrier sensing solution present in Mica2 and other similar low-cost wireless hardware is insufficient for the collision detection module for the following reasons. First, the TinyOS driver on Mica2 hardware does not provide a direct mechanism for instantaneous detection of the start or end of a transmission. For example, the RSSI of a received packet can be read using a down function call, which usually suffers from execution and scheduling latencies, leading to imprecise reception time detection. Additionally, if the packet gets corrupted, it may be discarded at a lower layer, which means that the RSSI data may not be available at all. Second, there is no direct way to adjust the carrier sense threshold and to set the reception range to be exactly equal to the carrier sense range. For these reasons, it was necessary to develop a separate hardware module, which is discussed next.

To sense the channel occupancy, a channel sensing module was developed as shown in Figure 35. The channel sensing module takes  $V_{RSSI}$  from the Mica2 and filters it through a low pass filter. In our settings, the receiving signal can be as low as -90 dBm, which can almost be embedded into the ambient noise. Figure 36 shows the power spectral density of the RSSI signal, and it can be clearly seen that there are interferences at around 2 kHz, 2.4kHz, 7 kHz and their harmonics. The low pass filter has a cutoff frequency at 600Hz, and a latency of 0.27 ms. After the low pass filter, the weak signal has a 6 dBm separation from ambient noise, which provides enough room for the channel occupancy detection module. Once the channel occupancy detection module detects the start or the end of a transmission in its neighborhood, it sends an interrupt to Mica2, and the slot duration is measured from the time difference between the two interrupts. When a node detects that  $\tau' > \tau$ , as required by the protocol syntax in [Section 4.4](#), it sends an *interrupt packet* immediately after the beginning of the collided slot to resolve the collision. Note that this channel

sensing module is an extremely low cost solution and can be built from off the shelf components. It acts as an add-on to the Mica2 mote in our experiments, but can also be integrated with other systems either during manufacturing or as an add-on.

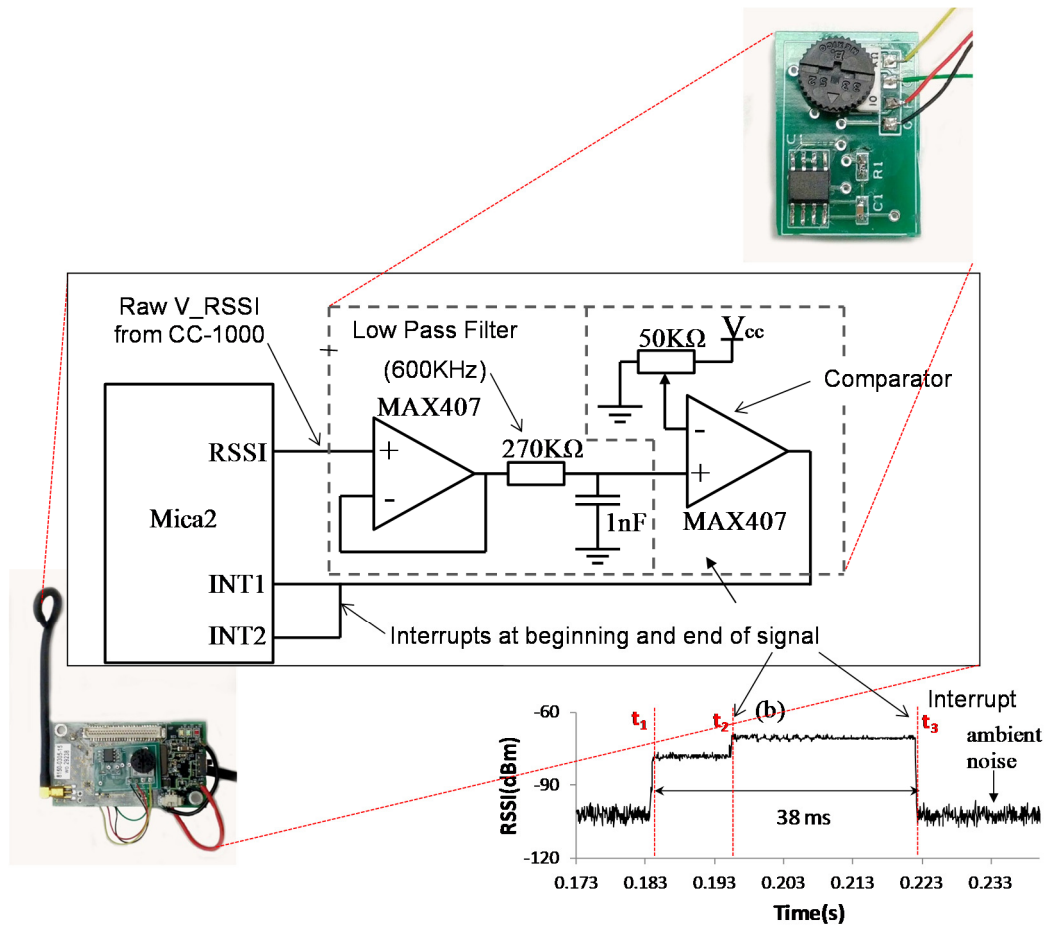
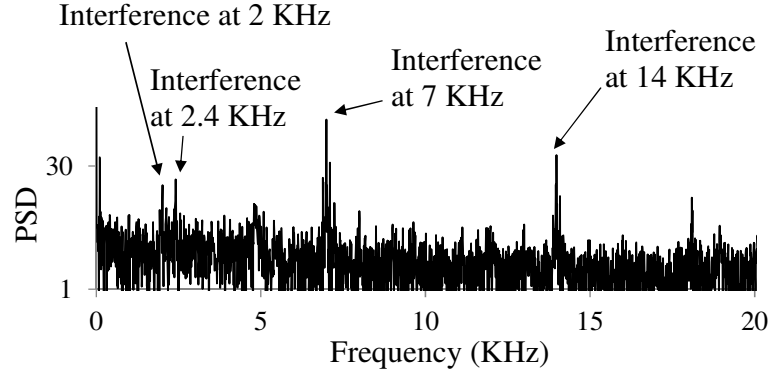


Figure 35. Collision detection subsystem: hardware for third-party collision detection by observing transmission signal duration



*Figure 36. Power spectral density (PSD) of the RSSI signal*

## 7.5 Summary

In this chapter we present the third-party wireless collision detection system that can be used for ZEA-TDMA. We first describe the collision detection and resolution mechanism and then provide a hardware realization of the mechanism by developing an add-on hardware module capable of detecting channel occupancy. In the next chapter we implement ZEA-TDMA in Tinyos 2.x and use Mica2 motes to show the slot self-allocation, and collision detection and resolution process. We also present the evaluation of our protocol functionality in a real wireless network test-bed.

## **Chapter 8: System Prototype for Privacy Preserving Channel Access**

### **8.1 Introduction**

In this chapter, we perform prototyping and system development of the proposed ZEA-TDMA protocol using Mica2 wireless nodes and Tinyos-2.x. We also present the functionality and performance validation in a real-life wireless test-bed consisting of hardware-augmented Mica2 nodes. We make similar worst-case assumptions that each node belongs to different operators or trust domains and do not exchange any form of information between them. The main purpose of prototyping and implementation on a real network is to demonstrate the functional feasibility of the developed concepts in this thesis.

### **8.2 Prototype Evaluation**

#### **8.2.1 Methodology**

We implemented the ZEA-TDMA MAC channel access protocol in Tinyos 2.x and evaluated it on a test-bed containing up to ten Mica2 sensor nodes. The test-bed is deployed in an indoor laboratory environment. The main purpose of our evaluation is to (i) validate the protocol functionality for different network topologies, and (ii) analyze the convergence characteristics of the nodes during the experiments. Note that the protocol has not been compared with other privacy preserving solutions since no functionally equivalent protocol was found in the literature. Each node in the test-bed was equipped with the developed channel sensing module and was capable of detecting when the channel was busy during transmissions. The evaluation has been done for both fully-connected (single-hop) network, as well as multi-hop networks. The third-party collision detection and resolution has been evaluated for only multi-hop networks, since there will be no collisions in a single-hop network. In our multi-hop experiments, instances where two-hop

neighbor nodes select non-overlapping slots are very similar to single-hop slot-selection. Hence, for multi-hop experiments we have only shown the results where collisions occur.

In all our experiments, we have used 65 byte packets and a maximum random jitter of 3ms, which makes the effective slot size  $\tau=30\text{ms}$ . For our results, we have plotted the RSSI values received at a common channel sniffer due to the transmissions from nodes in the network. The RSSI values in dBm have been plotted over time, and may not be exactly same for every node although every node transmits using the same transmission power. We have evaluated the protocol using varying F-ratio. A higher F-ratio will decrease the probability of collision between two-hop neighbors as they have a higher chance of randomly selecting non-colliding slots.

### 8.2.2 Functionality

#### 1. Fully-connected network

Figure 37 shows the *regular packet* transmissions and the slot allocation process for a fully connected network with three nodes – A, B and C. All nodes join the network at the same time and they listen for one-hop neighbor transmissions for  $T$  duration before selecting their slot location. Since no other node is transmitting, all the nodes select their slots randomly and start transmitting *regular packets* at the same time. In this example, node C selects a slot non-overlapping with nodes A and B. However, node A selects a slot overlapping with node B. This is not evident from the figure because node A senses the channel to be busy due to node B's *regular packet* and defers its transmission until after the channel becomes free. Hence, node A's slot location is directly after node B's slot. After their slots have been selected, all nodes continue transmitting *regular packets* at their selected slot locations.

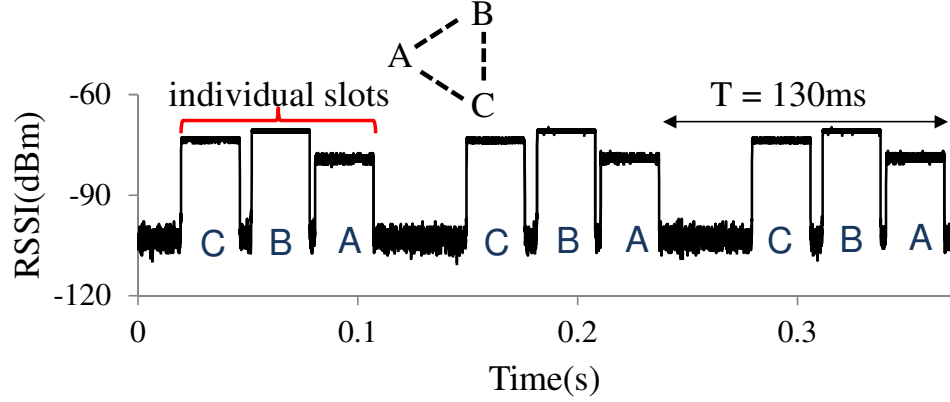


Figure 37. Slot self-allocation in a single-hop network (Oscilloscope screen-capture)

## 2. Multi-hop network

### a. Three node linear topology

The multi-hop network is created by placing the nodes 1m apart in a linear network topology and adjusting the transmission powers such that their transmission range is not more than 1.2m. Figure 38(a) shows the result for a multi-hop network with F-ratio = 1.33, and connectivity as shown at the top of the figure. In this example, node B joins the network first and selects a slot. Then nodes A and C join the network together and select overlapping slots, where node A occupies the former part of the collided slot, and node C occupies the latter part. Node B detects the collision from the slot duration  $\tau' > \tau$ , and waits for  $c_2 \times T$  seconds (not shown in figure) to ensure that the prolonged channel occupancy is due to illegal slot-usage. It then sends an *interrupt packet* immediately after the beginning of the collided slot. The part where the *interrupt packet* is being sent is enlarged beside the plot for better clarity. The *interrupt packet* sent by node B has a slightly lower RSSI at the sniffer than the *regular packet* sent by node A, as indicated in the figure by the difference  $\Delta$ . The *interrupt packet* keeps the channel busy for node C, which then defers its *regular packet* transmission, and selects a slot immediately after node A's slot. This new slot selected by

node C overlaps with node B's slot, causing B to defer its transmission as indicated by the dashed arrow in the figure.

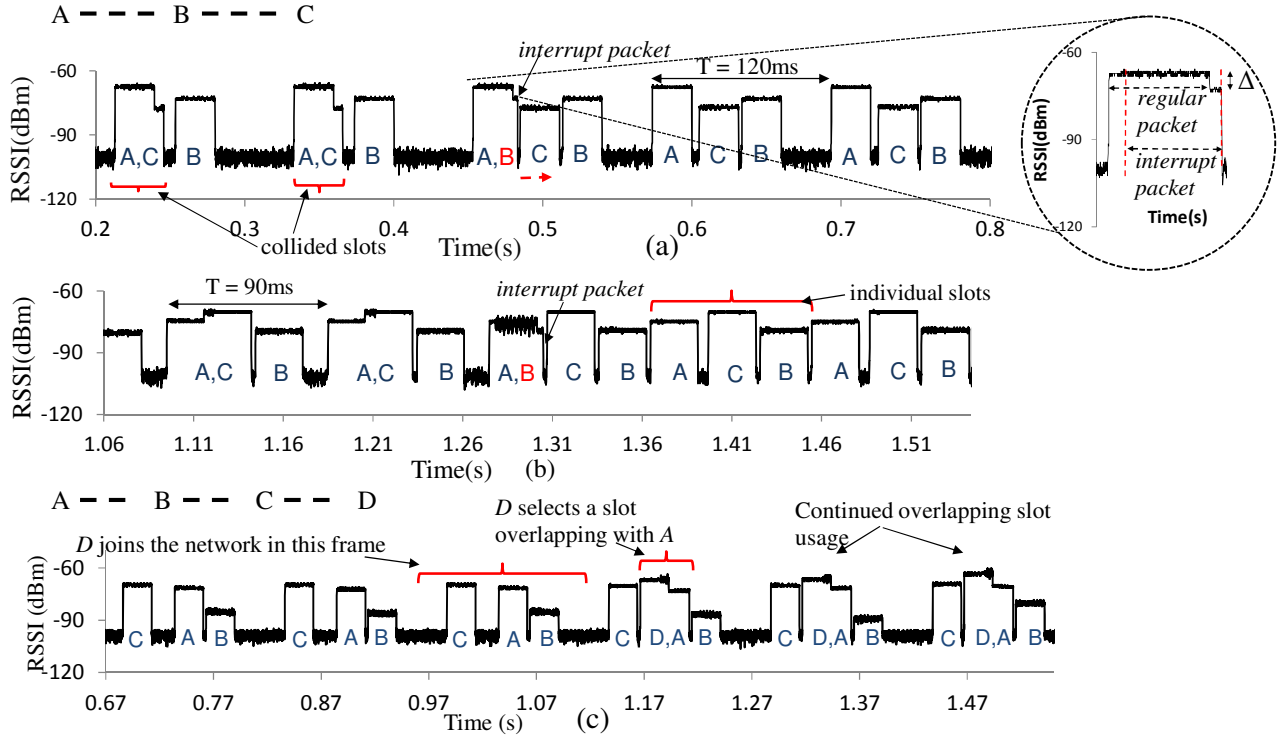


Figure 38. Allocation dynamics observed by a passive listener in a multi-hop network for (a) Three-node linear topology with  $T = 120$  ms ( $F$ -ratio = 1.33) (b) Three-node linear topology with  $T = 90$  ms ( $F$ -ratio = 1) (c) Four-node linear topology with  $T = 150$  ms ( $F$ -ratio = 1.15) – overlapping transmissions from A and D is legal

For Figure 38(b),  $F$ -ratio = 1. Here, all the nodes join the network like in the previous scenario, and nodes A and C select overlapping slots. The collision is detected and resolved by node B when it sends an *interrupt packet* right after 1.277 s. This defers node C's transmission, which in turn affects node B's slot. Once all the nodes in the network settle on a TDMA slot by themselves, the network stabilizes as seen around 1.36 s. The small  $F$ -ratio creates a 'tight' situation in the TDMA frame as nodes occupy tightly spaced slots. Individual slots occupied by the nodes are pointed out in the figure.

*b. Four-node linear topology*

The slot allocation functionality for a four node linear topology is shown in Figure 38(c). To demonstrate network dynamism, nodes A, B and C (in the topology shown in the figure) are introduced to the network initially. After these nodes settle on collision free slots, the fourth node D is introduced in the network. Since node D can only listen to node C's transmission, it randomly selects a slot non-overlapping with node C. The slot selected by D is overlapping with A's slot, which is more than two-hops away from D, and hence is a legal slot assignment.

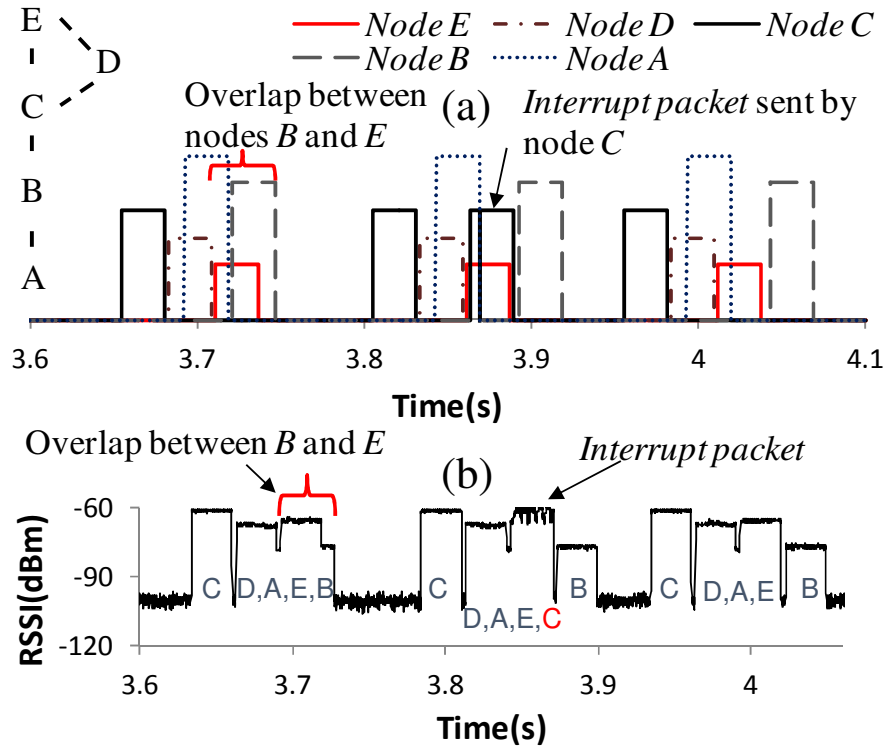


Figure 39. Five-node (topology 1)  $F\text{-ratio}=1.15$  (a) Timing diagram using RSSI traces from individual nodes (b) RSSI trace observed at a passive listener

*c. Five-node random topology*

A five-node random topology (topology 1), as shown in the top-left of Figure 39, was created to test more complex topologies. The slot allocation for  $F\text{-ratio} = 1.15$  is shown in the figure. Figure 39(a) demonstrates the exact slot location and slot duration of the individual nodes over

time and demonstrates how the slot locations change due to different network dynamics. Only the relevant events have been shown in the figure due to space constraints. The y-axis is omitted as it doesn't have any real meaning. The slot locations of the different nodes are distinguished from their different y-axis amplitude. For example, all slots occupied by node A will have the same y-axis value. Figure 39(b) plots the corresponding RSSI values recorded at a common channel sniffer. From the figures, two-hop neighbor nodes B and E occupy overlapping slots which is detected by C after  $c_2 \times T$  duration (not shown). Node C then sends an *interrupt packet*, which causes node B to defer its transmission and select a collision-free slot.

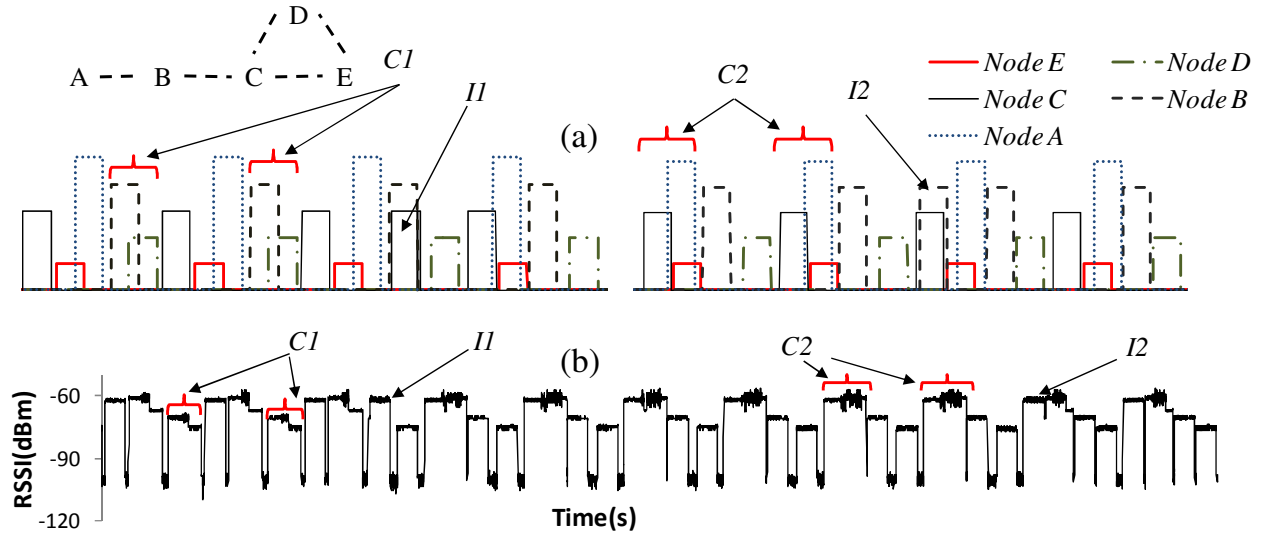


Figure 40. Five-node (topology 1)  $F\text{-ratio}=1$  (a) Pictorial representation of slot location of nodes (b) RSSI recorded at common sniffer using oscilloscope

Figure 40 plots the slot allocation for the same topology, but for  $F\text{-ratio} = 1$ , which creates a much 'tighter' situation. C1 in the figure indicates the illegal slot usage by nodes B and D. This illegal usage is detected by node C, a common neighbor of both, which in turn resolves the collision by sending an *interrupt packet* pointed in figure as I1. The *interrupt packet* causes node D, which occupied the latter part of the collided slot, to defer its transmission and select a slot non-

overlapping with node B. However, the deferring of node D's transmission causes node C to defer its transmission as it finds the channel to be busy due to D's *regular packet* transmission. The movement of node C's slot leads to (i) deferring of node E's transmission slot, and (ii) node C's slot overlap with node A as marked by C2 in figure. This overlap is then resolved by node B sending an *interrupt packet* at I2, which results in movement of node A's slot, as well as movement in node B's slot. However, the current slot-allocation of all nodes in the network does not have any illegal slot usage, and hence leads to convergence.

Figure 41 plots the same data as Figure 40, but for a different five-node topology (topology 2) as shown in the figure, and for F-ratio = 1. This topology is more complex in terms of slot allocation because here, there are four possible combinations of slot-overlaps, and only one node (C) to resolve all of them, unlike the previous topology with three possible overlap combinations and two nodes (B and C) to resolve them. The first collision C1 is between nodes B and D, and is resolved by *interrupt packet* I1 in the figure. I1 cause node B to move, which in turn causes node A to move and select a slot overlapping with node E, as indicated by C2 in figure. Node C then detects C2 and sends *interrupt packet* I2 to resolve it. I2 creates a domino effect causing node E, and then subsequently nodes C and D to move their slots. This results in node D selecting a new slot which overlaps with B, shown as C3. *Interrupt packet* I3 resolves C3 by moving B's slot, but causes A's slot to move and create C4, which is a new overlap between nodes E and A. C4 gets resolved by *interrupt packet* I4, but in turn causes movement in E's, C's and D's slots respectively. The new slot selected by node D overlaps with B's slot and cause C5, which is finally resolved by *interrupt packet* I5. The new slot allocation resulting after I5 is legal and does not cause any other collisions. The example here shows that when F-ratio is lowest, the network can sometimes take long to converge due to unavoidable collisions.

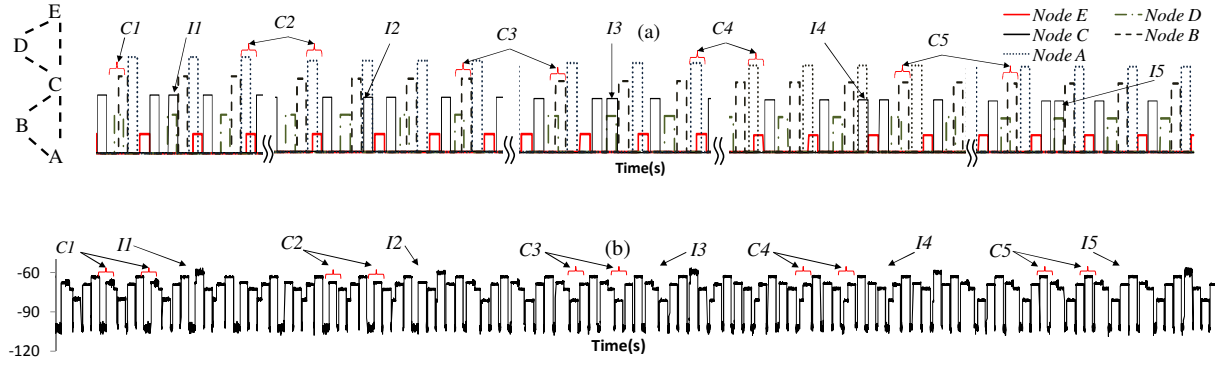


Figure 41. Five-node (topology 2)  $F\text{-ratio}=1$  (a) Pictorial rep. of slot location of nodes (b) RSSI recorded at common sniffer using oscilloscope

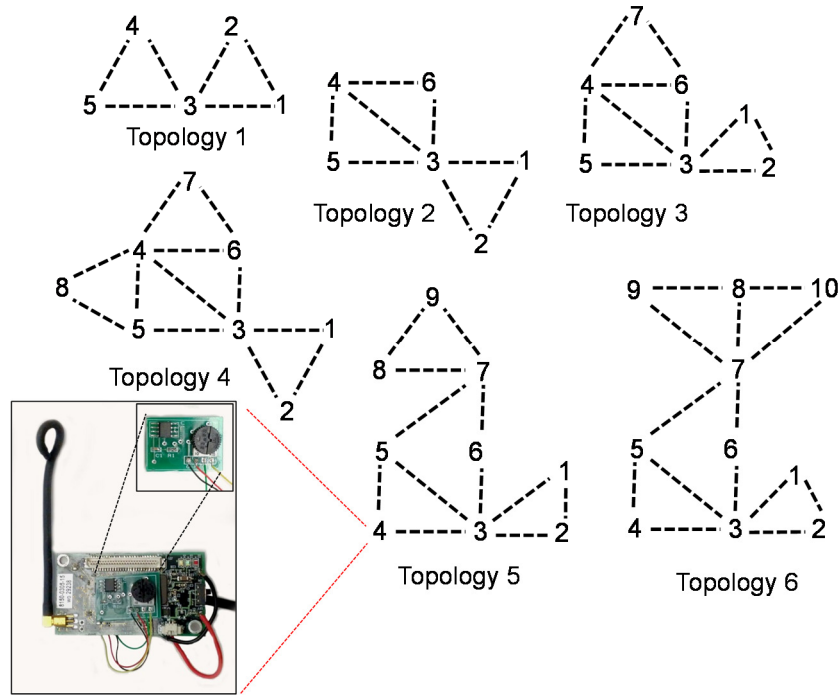
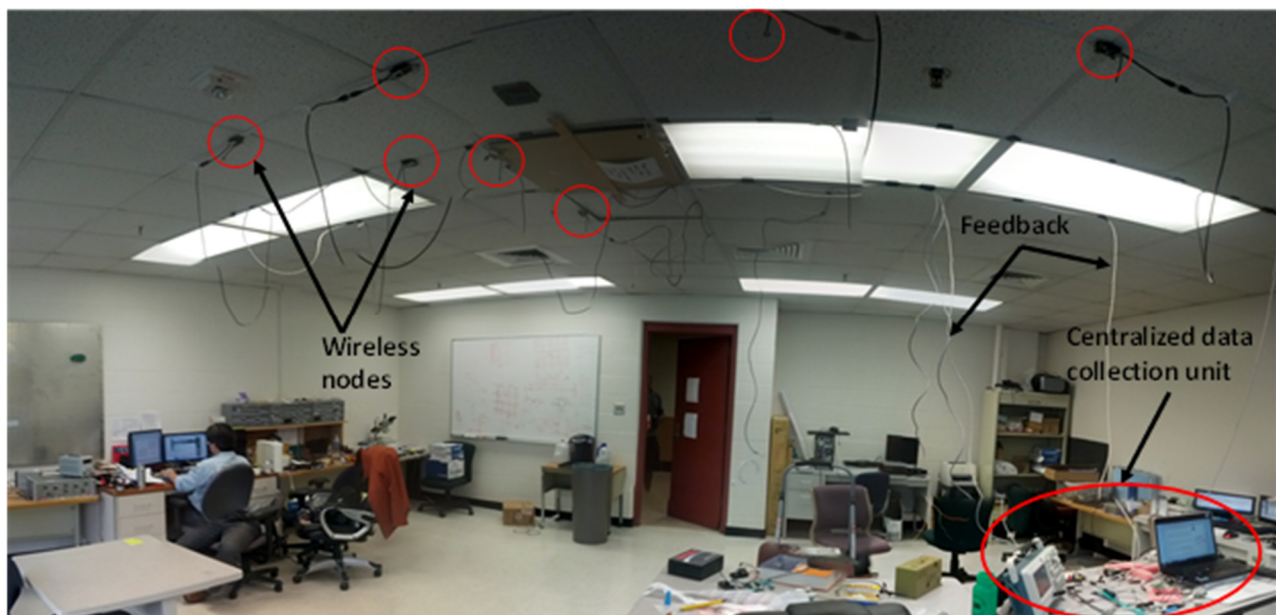


Figure 42. Experimental topologies for wireless network test-bed

### 8.2.3 Convergence characteristics

In this section, we discuss the allocation convergence of nodes in the six different network topologies as shown in Figure 42. In order to depict a real-life multi-hop wireless network, the topologies shown in the figure were grown organically without maintaining any progressive structure. Figure 43 shows the indoor laboratory setup of the wireless network test-bed. The

wireless nodes labelled in the figure form the topologies depicted in Figure 42. A constant source of power was provided to each node through a ‘hacked’ USB port. Although a wireless sniffer node was used to detect an incoming transmission, there were no direct means to determine the exact transmission times of each of the individual nodes in the network, since no identity information was used in the individual packets. For data collection and to demonstrate the results, a feedback system connected to a centralized data collection unit was built. This data collection unit consisted of a 16-channel oscilloscope and a laptop computer, and successfully captured the exact transmission times of each of the nodes through the feedback system.



*Figure 43. Indoor laboratory setup of wireless network test-bed*

The convergence latency indicates the time it takes for the entire network to reach a stabilized state where all nodes occupy collision free slots. In our experiments, we used two forms of node additions, one in which all nodes were introduced in the network at the same time, and another in which nodes were incrementally added to the network once the previous network stabilized. For both scenarios, a collision resolution wait period of  $c_2 = 6$  was used. In Figure 44 and Figure 45,

we show the average convergence latency and the corresponding distributions for 30 experiment runs for each F-ratio value.

Figure 44 shows the average convergence time of the six different topologies for all F-ratios tested. The topological dependence on convergence can be observed from the figure, however, the effect is not prominent. This is because although the topology is organically incremental, there is no direct linear relationship between topology size =  $N$  and topology size =  $N+1$ . Additionally, the topology 4 displays erratic behavior, especially for F-ratio = 1 and F-ratio = 1.15. This is because during the experiment sets for these two F-ratios, there was a topological rearrangement due to instability in wireless connectivity. A decrease in the average convergence time ( $N = 9$  to  $N = 10$ ) can also be observed for F-ratio = 1.3, 1.5, 2. This is attributed to an added node which can send *interrupt packet* to help in quicker collision resolutions. However, the same is not observed for F-ratio = 1, 1.15 because lower F-ratio means tighter TDMA-frame spacing for higher number of nodes.

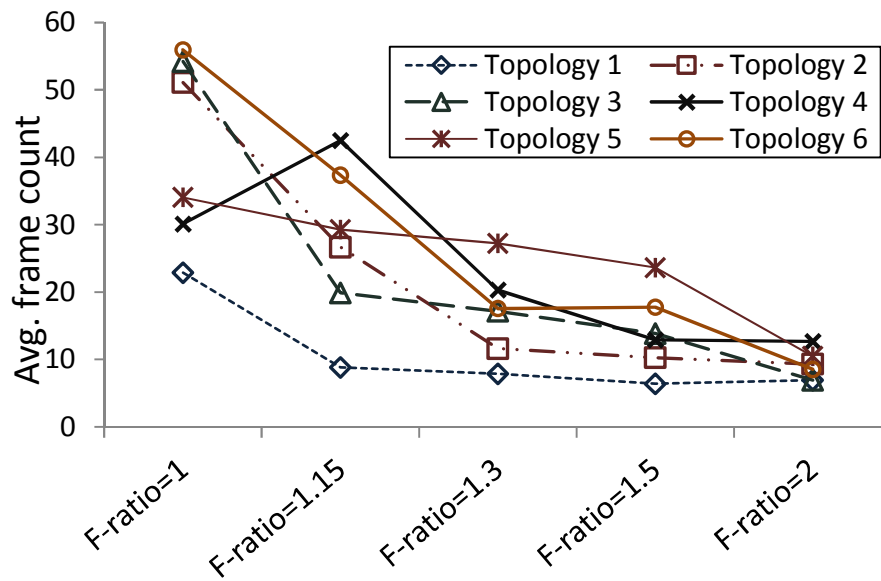


Figure 44. Average convergence latency

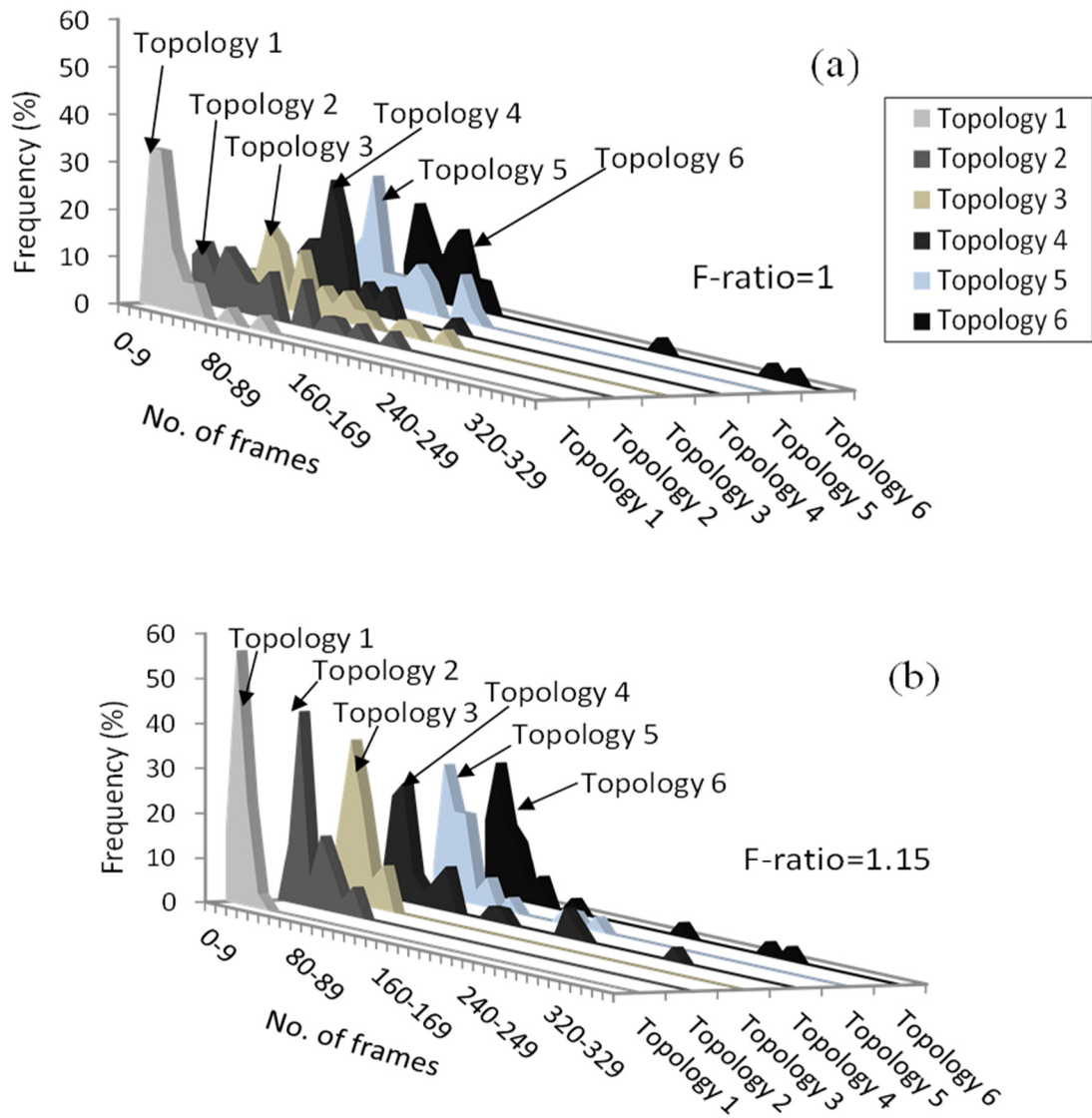


Figure 45. Distribution of convergence latency depicting the effect of frame size on allocation convergence: (a)  $F\text{-ratio}=1$  (b)  $F\text{-ratio}=1.15$

Figure 45 (cont'd)

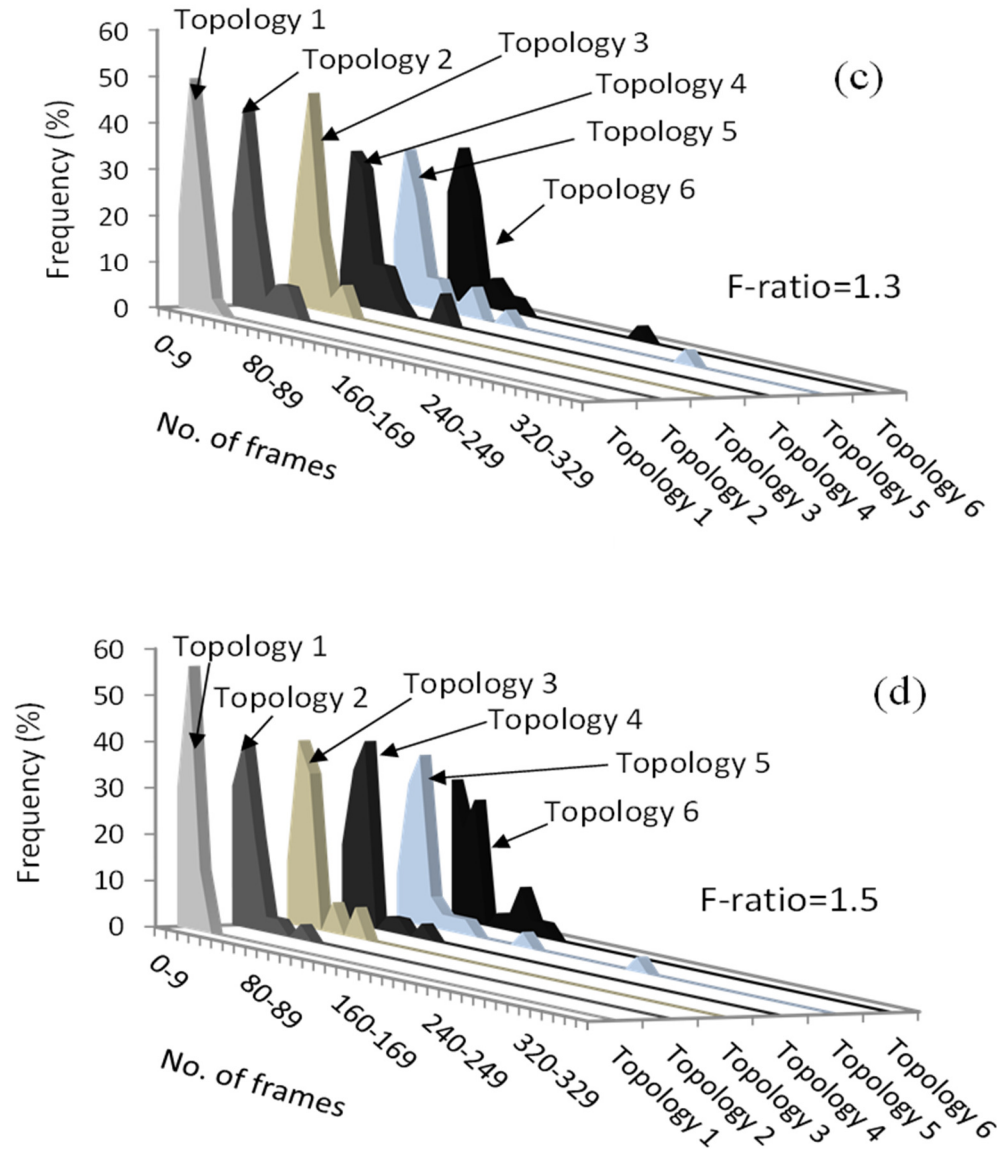


Figure 45. Distribution of convergence latency depicting the effect of frame size on allocation convergence: (c) F-ratio = 1.3 (d) F-ratio = 1.5

Figure 45 (cont'd)

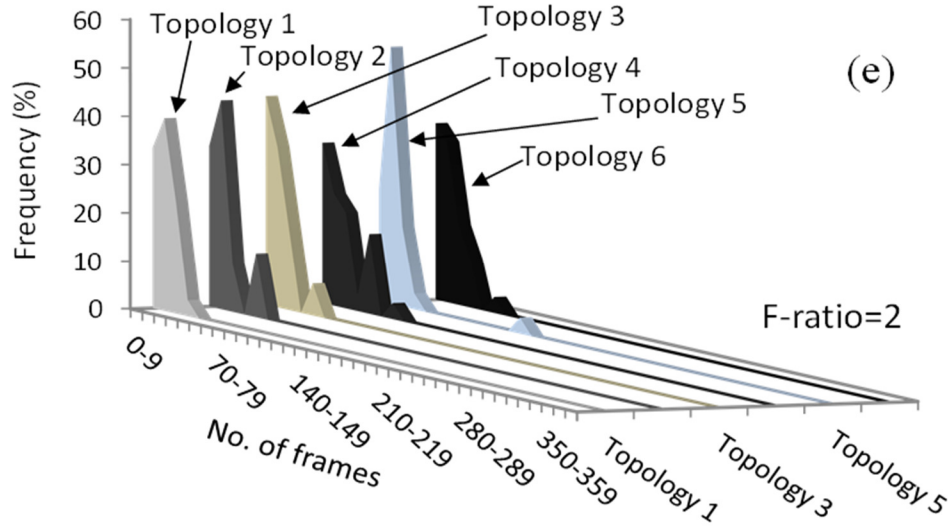


Figure 45. Distribution of convergence latency depicting the effect of frame size on allocation convergence: (e)  $F\text{-ratio} = 2$

Figure 45 plots the distribution of the convergence latency for 30 different experiment runs for each F-ratio value. The primary observation here is a clear left-shift in the distributions for larger F-ratios (i.e. TDMA frame sizes) for all topologies. This demonstrates that larger TDMA frame sizes allow faster convergence, with the added cost of lower overall data rates. It is also observed that for larger topologies, the distributions shift right. The primary contributing factor to this is chain collisions, which are collisions resulting from the resolution of an existing collision as was observed during the functional evaluation. The occurrence of such chain collisions increase with increasing network size.

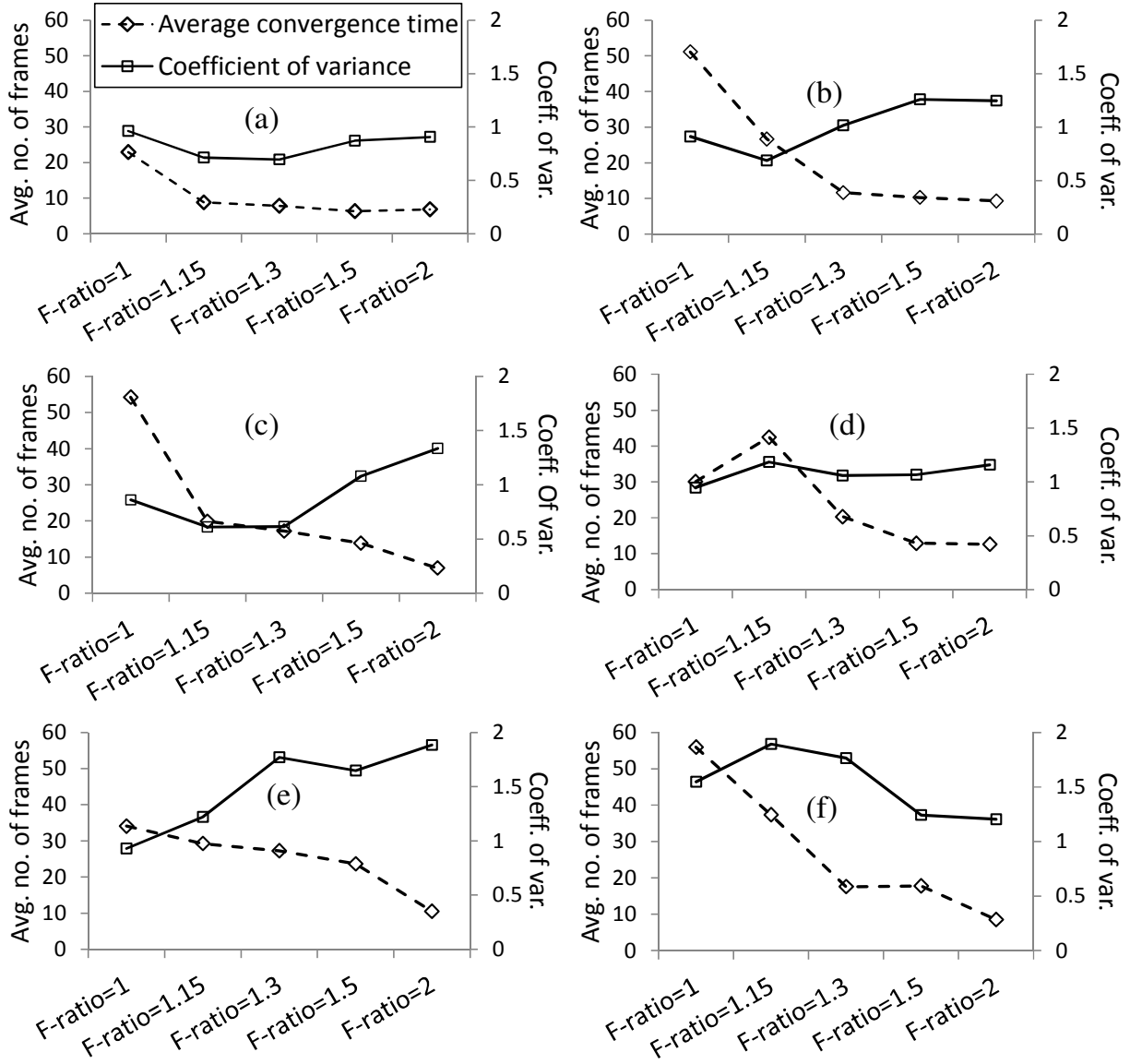


Figure 46. Variation in convergence latency and average convergence times: (a) Topology 1 (b) Topology 2 (c) Topology 3 (d) Topology 4 (e) Topology 5 (f) Topology 6

Next, in Figure 46, we plot the coefficient of variance alongside the average convergence time to depict the behavior of the convergence latency over multiple experimental runs. The most prominent observation here is that the coefficient of variance increases with increasing F-ratios. The main reason behind that is at higher F-ratios there are more possible steady state allocation states, which increases the coefficient of variance. Additionally, the variance is lowest for F-ratio

= 1.15, and F-ratio = 1.3, indicating these F-ratios provide the most consistent performance. The average convergence times shrinks with higher F-ratio because the domino effect of collision resolution and slot movements is reduced with higher F-ratios.

Figure 47 plots the convergence latency for the incremental node addition scheme. In this scheme, each node was added to the network once the previous network was stabilized. The resultant network topology after all nodes were added was the same as topology 6. The nodes were added incrementally in two different orders (Order 1 and Order 2), and the convergence latency of the entire network was observed. The average convergence latency over 10 different experiment runs is plotted in Figure 47. The order of node additions is noted in the x-axis of the graphs. It was observed that some nodes settle on a finalized slot faster than other nodes. This is primarily due to the fact that some nodes have lower visibility of the network, i.e. higher number of two-hop neighbors, than other nodes. This causes the node to get involved in hidden collisions more than other nodes, which delays the network convergence. For example, in topology 6, node 3 has 1 two-hop neighbor (i.e. node 7) and 5 one-hop neighbors (i.e. nodes 1, 2, 4, 5, 6). On the other hand, node 6 has 2 one-hop neighbors (nodes 3, 7) and 7 two-hop neighbors (8, 9, 10, 1, 2, 4, 5). Hence node 6 is six times more likely to be involved in a hidden collision. In Order 2, the network takes maximum time to converge when nodes 4, 5, and 8 are added since their addition disrupts the previously added nodes' settled slots.

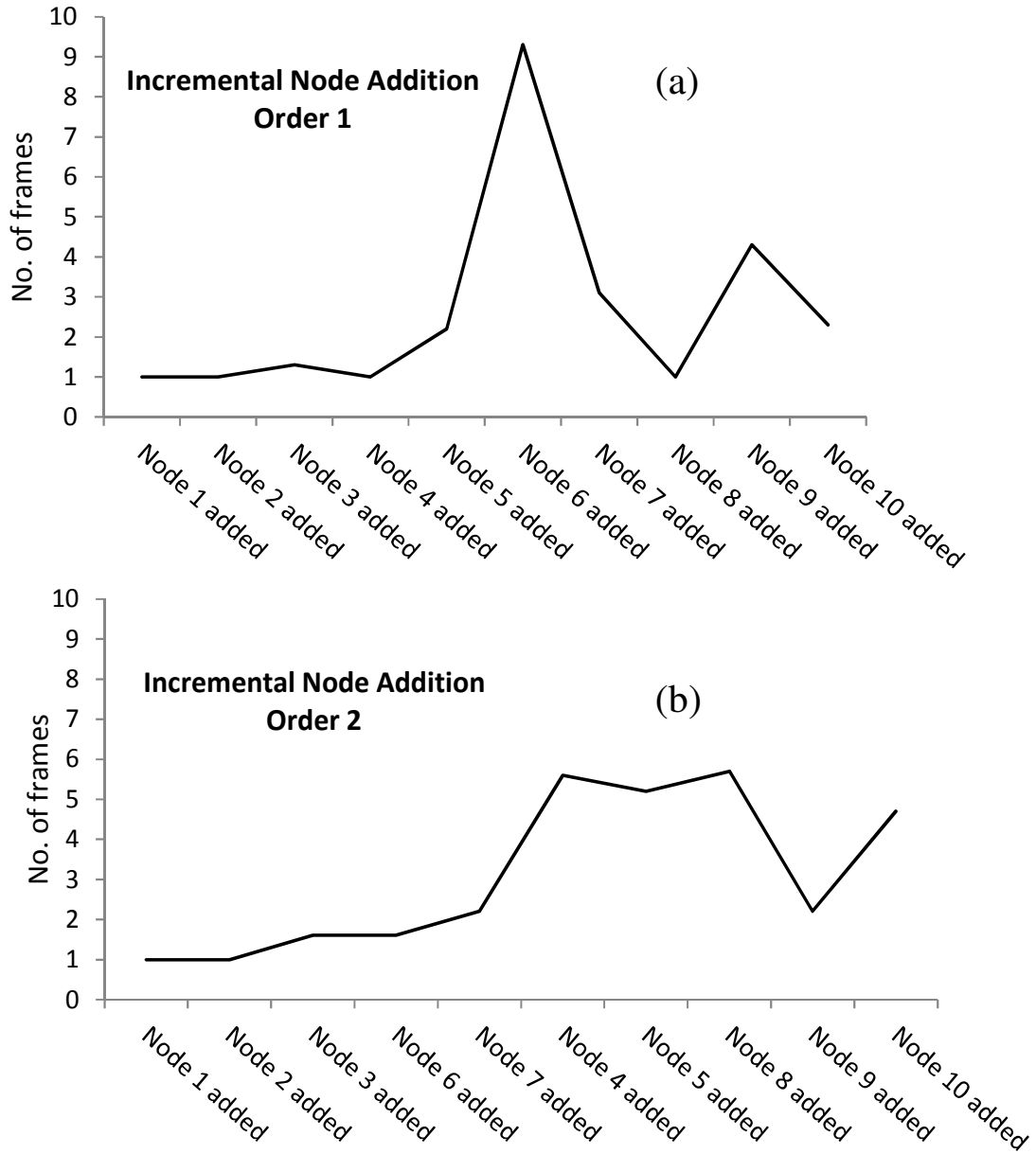


Figure 47. Convergence latency for incremental node addition (a) Node addition order 1 (b) Node addition order 2

It can be also seen that the convergence is fairly quick when nodes are added incrementally than when all nodes are introduced at the same time. This characteristic is useful since in most real-life wireless networks, nodes are almost always incrementally introduced to a network, i.e. there are rarely cases where all nodes join a network at the exact same instance.

### 8.3 Summary

In this chapter we demonstrate how the developed ZEA-TDMA achieves TDMA slot allocation in a real-life multi-trust-domain wireless network. As shown in the previous chapters, the mechanism achieves distributed slot allocation using blind-folded packet transmissions. As a result, the protocol prevents inter-trust-domain privacy exposure by not relying on any explicit in-packet information including MAC layer source and destination addresses. The protocol has been prototyped in Tinyos-2.x along with add-on hardware on Mica2 motes for collision detection and resolution. Extensive experimentation for evaluating functional validity and protocol convergence is reported in this chapter. Multiple network topologies have been used for evaluating functional and performance consistency.

## Chapter 9: Conclusion and Future Work

### 9.1 Conclusion

In this thesis we showed that the presented privacy preserving mechanism using *blind-folded* channel access is an effective solution to prevent *traffic analysis* and *identity exposure*. We developed a TDMA protocol which can successfully allocate slots to wireless nodes simply by sensing the wireless channel. The proposed protocol, ZEA-TDMA, has two major advantages over regular distributed TDMA protocols. First, it does not require any information from packets for slot allocation. This annuls the necessity of exchanging identities among nodes or running a complex slot allocation algorithm. Second, the protocol does not require packet decoding during the slot allocation process. Such a feature adds a number of useful advantages. It enables seamless addition of new nodes without revealing any information about the existing nodes, as well as reduced impact of high-error channels on the slot allocation process. Since the transmission scheduling is independent of any message-based information, a low SNR would have lesser impact than mechanisms which depend on explicit control messages. Additionally, nodes that are within the carrier sensing range but not within direct communication range can also coordinate to allocate collision free slots, resulting in a sooner onset of the convergence process than possible with control message-based schemes. These features of ZEA-TDMA makes it ideal for a privacy preserving protocol since it can prevent *linkability* and the inherent TDMA traffic pattern can thwart *traffic analysis*.

In Chapter 4 we presented the proposed ZEA-TDMA protocol and introduced the main principles used in the slot allocation scheme. Nodes in a wireless environment self-allocate slots and use a reactive mechanism to rectify incorrectly selected slots. Before converging to a legal slot usage, nodes transition through incorrect slot locations and use a third-party collision detection

and resolution mechanism to rectify illegal slot-usage. The protocol was implemented in NS2 and showed evaluation results for functionality and convergence characteristics.

In Chapter 5 we developed *e*ZEA-TDMA, a variant of ZEA-TDMA, which at the cost of some added complexity, provides faster convergence time for the network and enables the development of an energy efficiency module. The faster convergence is achieved using a proactive approach instead of a reactive one for slot-allocation. Individual nodes propagate their one-hop slot occupancy list by sending *shadow packets* in their neighbors' slot times, and use a pattern-based sending pattern to avoid collisions. Unavoidable collisions are resolved using the collision detection and resolution mechanism developed for ZEA-TDMA. Experimental evaluation of *e*ZEA-TDMA was performed using NS2 and the functionality, convergence, and the transient power consumption characteristics were shown in the results. An analytical model was also developed to characterize the performance of the protocol in certain scenarios.

In Chapter 6 we presented an energy aware ZEA-TDMA design to address the energy issues pertaining to the protocol features. The developed energy efficiency module functions as an addition to *e*ZEA-TDMA and uses a sleep-wake scheduling process to achieve reduced energy loss due to idle-listening. A model for the energy efficiency protocol is developed and results from the developed model are compared to simulation results.

The feasibility of the proposed third-party based wireless collision detection mechanism is demonstrated in Chapter 7. The collision detection system is implemented by developing a custom hardware module, which can be augmented to a Mica2 wireless node, or any other similar device, to successfully detect a wireless collision from the duration of the received signal.

In Chapter 8, we present a practical use of the developed privacy preserving TDMA protocol in an application scenario by developing a system prototype of the proposed protocol. The protocol is implemented in a wireless network test-bed to emulate an IoT environment and experimental evaluation results are presented to demonstrate the functional feasibility of the developed concepts in this thesis.

## 9.2 Future Work

One common drawback of TDMA protocols, including ZEA-TDMA, is that the TDMA frame size needs to be pre-determined. Such a requirement creates constraints on the network size and traffic pattern, and can be very inefficient in dynamic scenarios. When the number of nodes in the network can significantly vary, the frame size needs to be fixed such that it can support when the network size is maximum. This means that (i) there needs to be an accurate estimate or prior knowledge on the maximum network size, and (ii) when the network size is very small, a majority of the bandwidth is wasted. This can result in very inefficient network-wide performance. A solution to such a problem is the modification of ZEA-TDMA to support multiple slot usage by nodes for heterogeneous traffic support. Such a multi-slot ZEA-TDMA will enable nodes to use multiple available slots in a TDMA frame depending of network traffic requirements. Such a solution is challenging to implement in ZEA-TDMA because of the unavailability of any form of information. A multi-slot ZEA-TDMA protocol has been developed where nodes create virtual *clones* of themselves, and run the ZEA-TDMA state-machine on each of those clones. Ongoing work on this topic include the implementation of multi-slot ZEA-TDMA and the evaluation of the network-wide performance like throughput and latency in varying traffic conditions.

Although, current literature provides the apparatus for the development of wireless privacy-preserving CSMA-based approaches, limited work has been done on the actual implementation of

such a protocol. Unlike TDMA protocols, CSMA does not require node identity sharing for MAC channel access. Hence, identity can be encrypted for protection against *identity exposure* and *linkability*. And, *traffic analysis* prevention techniques used in wired networks, like traffic morphing, traffic padding and defensive dropping, which can be tailored for wireless networks and used alongside a CSMA-based MAC protocol. Currently, there lacks a proper evaluation of the privacy preserving performance of ZEA-TDMA with a similar MAC-based privacy preserving protocol. Another avenue of extension of the current work can be geared towards the development of a privacy preserving CSMA-based MAC protocol, and compare the developed privacy preserving metric. This will give us a clear idea where ZEA-TDMA stands in terms of providing privacy. The metric, Privacy Preservation Index (*PPI*), is calculated using the correlation coefficient between different traffic flows to detect if two nodes in the network are carrying the same flow. A high correlation between the traffic flows of two nodes which are actually carrying the same traffic would mean that the privacy preserving mechanism is ineffective. The network model proposed in Chapter 3 will be used for routing. Since we focus on the privacy provided by the MAC-layer, the route can be fixed for different destinations. Moreover, it can be assumed that packets are only routed by nodes belonging to the same trust domain as the source node. Encrypting entire packets will prevent packet and message correlation attacks, and using fixed sized packets will prevent packet sized attacks. A TCP or FTP session can create real traffic scenarios and the measured *PPI* can be compared for both MAC-based privacy schemes. Network level performance metrics like throughput and latency can also be compared and the effect of varied traffic rate can be evaluated for the developed protocols. This study will demonstrate the relevance of TDMA-based privacy preserving systems and eventually create possibilities for the development of more such protocols.

## APPENDICES

## APPENDIX A: Analysis of Collision Handling

In this section, we first analyze the probability of unresolved collisions in ZEA-TDMA, which is an indirect measure of loss of bandwidth during a convergence process. This analysis will help us infer the limitations of the protocol based on the system-level implementation details. Next, we model the cost of the collision resolution process as a function of the *regular packet* loss-rate. Since *interrupt packets* result in collisions with *regular packets*, this analysis can help deciding the protocol parameter  $c_2$  (see [Section 4.4](#)), which decides the collision resolution speed.

Table 3: List of parameters

$\tau$	Slot duration
$\tau'$	Collided slot duration
$t_A$	Start time of A's slot
$\varepsilon_c$	Clock error
$\vartheta$	Maximum clock resolution
$\delta_p$	Lower layer delays
$\delta_s$	OS delays
$int_{AB}$	<i>Interrupt packet</i> transmitted by A received by B
$t_A^{int_{AB}}$	Initiation time of transmission of $int_{AB}$
$t_B^{int_{AB}}$	Reception time of $int_{AB}$ at B
$t_A^{reg}$	Initiation time of transmission of <i>regular packet</i>

Theoretically, a collision is detected if  $\tau' > \tau$ , where  $\tau$  is the regular slot duration and  $\tau'$  is the collided slot duration. However, in practical situations, this may not be true due to hardware limitations and several other factors. From Figure 7, let the start time of slots of node A and C are  $t_A$  and  $t_C$  respectively. In reality, a collision is detected if  $\tau' \geq \tau + \ell$  where  $\ell > 0$ . Now, the collided slot duration  $\tau'$  can be expressed as:

$$\tau' = t_C + \tau - t_A \quad (3)$$

Since  $\tau' \geq \tau + \ell$ , using (3) we have

$$t_C - t_A \geq \mathcal{K} \quad (4)$$

The above expression is the condition which needs to be satisfied to resolve a collision between A and C. The value of  $\mathcal{K}$  is affected by the following factors. (i) Clock error ( $\varepsilon_c$ ). Clocks, especially in low-cost hardware, suffer from jitter (ii) Clock resolution ( $\vartheta$ ). Since the common neighbor (third-party) sends the *interrupt packet* exactly after the beginning of the collided slot, the soonest it can schedule the *interrupt packet* is dependent on the clock resolution provided by the operating system (OS) clock (iii) Lower-layer delays ( $\delta_P$ ). This is attributed to the combination of propagation delay, transmission delay, and processing delay (iv) OS delays ( $\delta_S$ ). This is dependent on the task scheduling procedure and the processing lag and is a factor of the OS running on the network node. Typically,  $\mathcal{K}$  can be a few hundred microseconds to a few milliseconds, depending on the above factors.

**Lemma 1.** *The value of  $\mathcal{K}$  is given by the following expression:  $\varepsilon_c + \vartheta + \delta_P + \delta_S$*

*Proof:* Let  $t_B^{intBC}$  be the scheduled time when the sending process of *interrupt packet* ( $int_{BC}$ ) will be initiated at node B, and  $t_C^{intBC}$  be the time when the  $int_{BC}$  from B reaches C. Since  $t_B^{intBC}$  is scheduled to be immediately after node A's slot, we have  $t_B^{intBC} = t_A + \vartheta + \delta_S$ . Similarly, let  $t_C^{reg}$  be the time when the sending process of *regular packet* was initiated by node C. Following previous assumptions,  $t_C = t_C^{reg}$ . Now, collision will be resolved when  $t_C^{intBC} \leq t_C^{reg}$ . After the sending process of  $int_{BC}$  is scheduled, it encounters clock error, as well as lower-layer delays before it reaches C. So,

$$t_B^{intBC} + \varepsilon_c + \delta_P = t_C^{intBC}$$

Using  $t_B^{intBC} = t_A + \vartheta + \delta_S$ ,

$$t_A + \vartheta + \delta_S + \varepsilon_c + \delta_P = t_C^{int_{BC}} \quad (5)$$

Now, since  $t_C = t_C^{reg}$  and  $t_C^{int_{BC}} \leq t_C^{reg}$ ,

$$\begin{aligned} t_A + \vartheta + \delta_S + \varepsilon_c + \delta_P &\leq t_C \\ \Rightarrow t_C - t_A &\geq \vartheta + \delta_S + \varepsilon_c + \delta_P \end{aligned} \quad (6)$$

Since there are no other delay components, we can use (4) and (6) to derive the equality:

$$\ell = \vartheta + \delta_S + \varepsilon_c + \delta_P \quad (7)$$

**Theorem 1.** *The probability ( $\wp$ ) that a collision between two nodes  $v_i$  and  $v_j$  will not be resolved is inversely proportional to  $T^2$ , where  $T$  is the TDMA frame size*

*Proof:* Let the start time for slots of nodes  $v_i$  and  $v_j$  be  $t_i$  and  $t_j$ , respectively. Since the slots are chosen randomly,  $t_i$  and  $t_j$  are uniform random variables. So,

$$t_i \sim U(0, T) \text{ and } t_j \sim U(0, T)$$

We use another random variable  $K$  to denote the time difference between the slots of  $v_i$  and  $v_j$ ,

$$K = t_i - t_j \quad (8)$$

The joint density function of  $t_i$  and  $t_j$  can be expressed as

$$f_{t_i, t_j}(x, y) = \frac{1}{T^2}; \quad 0 < x < T, 0 < y < T \quad (9)$$

The cumulative density function of  $K$  is given by

$$F_K(\ell) = P(K \leq \ell) \quad (10)$$

From (8) and (10),

$$F_K(\ell) = P(t_i - t_j \leq \ell) \quad (11)$$

Using (9) and (11),

$$\begin{aligned}
F_K(\kappa) &= \begin{cases} \int_0^{T+\kappa} \int_{x-\kappa}^T \frac{1}{T^2} dy dx, & -T < \kappa < 0 \\ 1 - \int_{\kappa}^T \int_0^{x-\kappa} \frac{1}{T^2} dy dx, & 0 < \kappa < T \end{cases} \\
&= \begin{cases} \frac{1}{T^2} \left[ \frac{T^2}{2} + \kappa T + \frac{\kappa^2}{2} \right], & -T < \kappa < 0 \\ \frac{1}{T^2} \left[ \frac{T^2}{2} + \kappa T - \frac{\kappa^2}{2} \right], & 0 < \kappa < T \end{cases} \tag{12}
\end{aligned}$$

And, the probability density function can be derived as,

$$\begin{aligned}
f_K(\kappa) &= \frac{dF_K(\kappa)}{d\kappa} \\
&= \begin{cases} \frac{1}{T^2} (T + \kappa), & -T < \kappa < 0 \\ \frac{1}{T^2} (T - \kappa), & 0 < \kappa < T \end{cases} \tag{13}
\end{aligned}$$

Therefore, probability that  $|t_i - t_j| \leq \kappa$ , i.e. the probability that a collision between two nodes  $v_i$  and  $v_j$  will not be resolved is

$$\begin{aligned}
\wp &= \frac{1}{T^2} \int_{-\kappa}^0 (T + z) dz + \frac{1}{T^2} \int_0^{\kappa} (T - z) dz \\
&= \frac{1}{T^2} [2\kappa T - \kappa^2] \tag{14}
\end{aligned}$$

From (14) it can be observed that  $\wp \propto \frac{1}{T^2}$ .

During collision resolution, one *regular packet* is lost for every *interrupt packet* that is sent since an *interrupt packet* collides with a *regular packet* and corrupts it. Let  $\eta$  be the cost associated with using *interrupt packets*, where  $\eta$  is expressed as the loss rate of *regular packets* as a percentage of number of TDMA frames. Before deriving  $\eta$ , we need to find the probability of collision,  $p$ , between two nodes  $v_i$  and  $v_j$ , where  $t_i < t_j$  and there exists a  $v_k$  such that  $e_{ik}, e_{jk} \in$

$E$  and  $e_{ij} \notin E$ , i.e.  $v_i$  and  $v_j$  are two-hop neighbors. Let the set of all nodes within two-hops neighborhood of  $v_j$  be  $D^j$  ([Section 3.2](#)). The vulnerable zone for collision between  $v_i$  and  $v_j$  is  $2\tau$ . Assuming that all nodes in  $D^j$  occupy consecutive slots without any fragmentation, the total possibilities from which  $v_i$  can choose a slot is given by  $T - (|D^j| \times \tau - \tau)$ , where  $T = n\tau$  is the TDMA frame size with  $n$  slots in a TDMA frame. So, the probability of collision between  $v_i$  and  $v_j$  is:

$$p = \frac{2}{n - |D^j| + 1} \quad (15)$$

where  $n > |D^j|$ . However, the above equation does not consider fragmentation between slots which can occur when there is a gap between the consecutive slots of nodes  $v_i$  and  $v_{i+1}$ , i.e.  $t_{v_{i+1}} - (t_{v_i} + \tau) < \tau$ , given  $t_{v_{i+1}} \geq t_{v_i} + \tau$ . If the sum of the fragmented intervals for all  $v_i \in D^j$  is expressed as:

$$s_{D^j} = \sum (t_{v_{i+1}} - (t_{v_i} + \tau)), \forall (t_{v_{i+1}} - t_{v_i}) < 2\tau \quad (16)$$

Then  $p$  becomes,

$$p = \frac{2}{n - d_Y + 1 - \frac{s_{D^j}}{\tau}} \quad (17)$$

As mentioned in [Section 4.4](#), an *interrupt packet* is sent every  $c_2$  frames. Given  $p$ , we can say that one *regular packet* is corrupted by an *interrupt packet* every  $\frac{c_2}{p}$  frames, which is the loss-rate for *regular packets*. Now, since the probability that a collision will not be resolved is  $\wp$ , the loss-rate can be modified to  $\frac{c_2}{p} (1 - \wp)$ , and the loss-rate expressed as a percentage is:

$$\eta = \frac{p}{c_2(1 - \wp)} \quad (18)$$

From the derived equations, it can be seen that both  $p$  and  $\eta$  decreases as  $T$  is increased. This is obvious because as the TDMA frame size is increased with the number of nodes in the network being kept constant, the probability of collision decreases, which in turn affects the loss-rate of *regular packets* due to *interrupt packets*.

From the above analysis, it can be concluded that apart from increasing  $T$ , the probability of collision resolution can be increased by decreasing the value of the delay parameters contributing to  $k$ . This means that superior hardware capabilities can greatly reduce the chances of occurrence of collision non-resolution. Additionally, it is also observed that if factors like network density and number of nodes are kept unchanged, the *regular packet* loss-rate can be reduced by increasing the probability of collision resolution  $(1 - \wp)$ . However, if  $\wp$  cannot be reduced, the speed of resolution  $c_2$  should be adjusted to achieve the acceptable loss-rate, which will cost the system a delay in collision resolution.

## APPENDIX B: Probability of collision between *regular packet* and *shadow packet*

Continued collision between *regular packet* and *shadow packet* can cause a temporary loss of throughput of the node whose *regular packet* is involved in the collision. As shown in Figure 13(b), such a collision can occur when  $v_i$  and  $v_k$  have overlapping slots, and  $v_j$  is a common neighbor of both. This is not explicitly resolved by the protocol since this is considered to be a transient collision and is resolved as soon as  $v_j$  reaches steady state.

Here we calculate the probability of such a collision between  $v_i$ 's *regular packet* and  $v_j$ 's *shadow packet*. For a given bitmap pattern of 101, there are three possible combinations in which  $v_i$  and  $v_k$  can select their patterns. (i)  $v_i = \dots 101101\dots$  and  $v_k = \dots 101101\dots$ , (ii)  $v_i = \dots 101101\dots$  and  $v_k = \dots 011011\dots$ , and (iii)  $v_i = \dots 101101\dots$  and  $v_k = \dots 110110\dots$ , out of which the last two combinations can cause a *regular-shadow* collision between  $v_i$  and  $v_j$ . Hence the probability of selecting collision causing patterns is

$$p_{\text{bitmap}} = \frac{2}{3} \quad (19)$$

Let the start time for slots of  $v_i$  and  $v_k$  be  $p$  and  $q$ , respectively. Since the slots are chosen randomly,  $p$  and  $q$  are uniform random variables  $U(0, T)$ . Without slot synchronization, the temporal placement of  $p$  and  $q$  can be anywhere. So, given a time  $t_i$ , the range of  $p$  is

$$[t_i, t_i + T] \quad (20)$$

Now, given that the start time of  $v_i$ 's slot =  $t_i$ , the probability of an overlap between  $v_i$ 's and  $v_k$ 's slots is the probability that the start time of  $v_k$ 's slot is within the range  $[t_i - \tau, t_i + \tau]$ , More formally,

$$\wp = P(\text{overlap}) = \int_0^T P(t_i - \tau < q < t_i + \tau \mid p = t_i) \frac{1}{T} dp \quad (21)$$

The conditional probability  $P(t_i - \tau < q < t_i + \tau \mid p = t_i)$  can be evaluated as  $\frac{2\tau}{T}$ , irrespective of  $t_i$ . Hence, the above equation can be simplified to,

$$\wp = \frac{2\tau}{T} \int_0^T \frac{1}{T} dp = \frac{2\tau}{T} \quad (22)$$

Further simplifying (22) using  $T = n\tau$  we get,

$$\wp = \frac{2}{n} \quad (23)$$

Therefore, using Eq. (19) and Eq. (23), the probability of collision between  $v_i$ 's *regular packet* and  $v_j$ 's *shadow packet* is given by

$$\begin{aligned} \rho &= p_{\text{bitmap}} \times \wp \\ &= \frac{4}{3n} \end{aligned} \quad (24)$$

$\rho$  gives the probability of collision between a regular and a *shadow packet* due to slot overlap between nodes more than two-hops. For larger network sizes or TDMA-frame sizes (i.e. higher values of  $n$ ), this probability will be quite low and hence such a collision type is not explicitly handled by the protocol, as discussed in [Section 5.6](#).

## APPENDIX C: Steady state power consumption model for eZEA-TDMA

We derive an analytical model for the steady state power consumption when energy-management is used with ZEA-TDMA. Table 4 below includes a list of all notations used in the following analysis. The average traffic rate  $\lambda$  is defined as the average rate at which a sender sends packets and can be expressed in average number of packets per TDMA frame. Since the sender can only send a maximum of one packet in a TDMA frame,  $\lambda \leq 1$ . Additionally, the burst size  $\beta$  is defined as the average size of each burst of traffic and is expressed in number of packets. In other words, burst size is the number of consecutive frames a sender sends packets. The average inter-burst interval in terms of number of frames can be calculated as  $\frac{\beta}{\lambda}$ . The set of senders is given by  $S = \{s_1, s_2, s_3, \dots\}$ , where  $S \subset \mathcal{N}$ . We model the power consumption for a link-layer multicast scenario, where each sender  $s_i$  sends messages to a given set of receivers  $R_i$ , where  $R_i \subset \mathcal{N}$ , where  $\mathcal{N}$  is the set of all nodes in the network. For simplicity, we assume that the set of receivers for any given sender is randomly chosen with a uniform distribution from its set of one-hop neighbors and  $0 \leq |R_i| \leq d$ , where  $d$  is the maximum one-hop node degree in the network. We also ignore the extra power consumption due to the two-way handshake process, as well as the excess energy wastage when receiving corrupt wakeup packets in the model development as these will be negligible with high burst sizes.

We express  $p^t$  as the power consumed during transmission,  $p^r$  as the power consumed during reception of a packet or while idle listening, and  $p^s$  as the power consumed while sleeping. It is assumed that  $\delta < \beta(\frac{1}{\lambda} - 1)$ , i.e. the transmitter needs to wake its receivers before each burst. The wake up and transmission-reception process is illustrated in Figure 48.

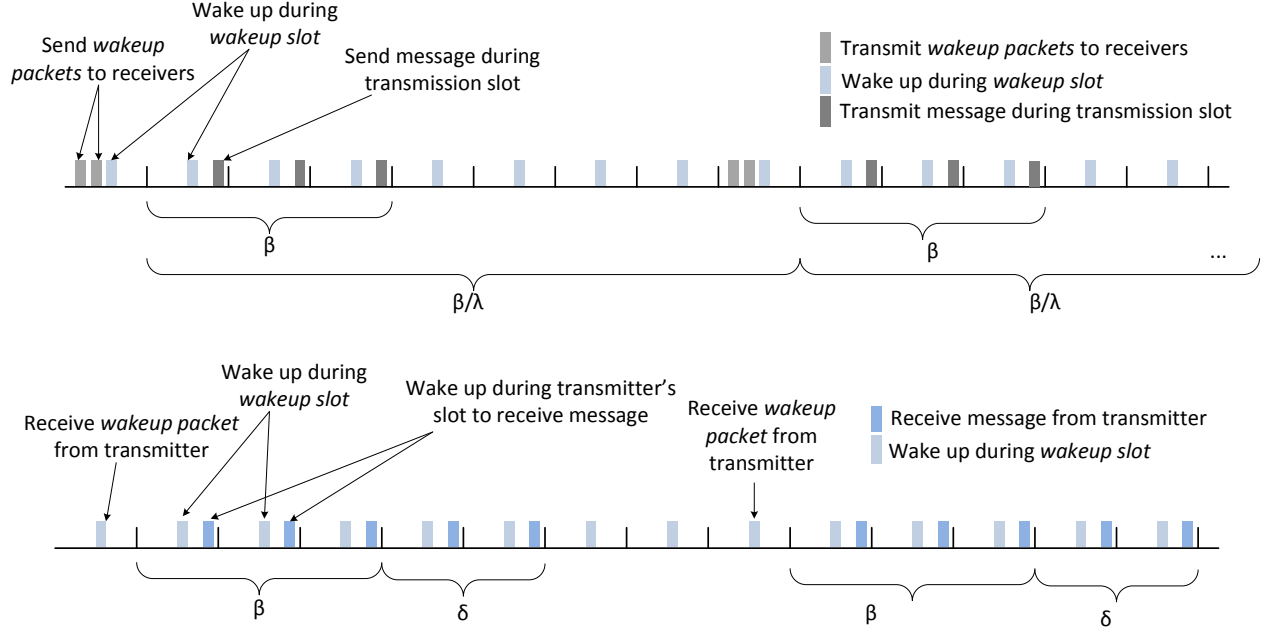


Figure 48. Wake up and transmission-reception process for a node with mean traffic rate  $\lambda=0.375$ , burst size  $\beta=3$ , and  $\delta=2$

Table 4: Symbols used for different parameters used in the analytical model

Symbol	Definition
$\mathcal{N}$	Set of all nodes in the network
$\delta$	Wake threshold
$\tau$	Slot size
$\lambda$	Average sending rate
$\beta$	Average burst size
$p^t$	Power consumed during transmission
$p^r$	Power consumed during receiving/idle listening
$p^s$	Power consumed during sleep
$n$	Number of slots in a TDMA frame

The power consumed by a transmitter can be divided into three primary components: (i) to send wakeup packets to all its receivers (ii) to transmit the packets, and (iii) to wake up in its wakeup slot. Since the maximum one-hop node degree is  $d$ , the average number of receivers for each

transmitter is  $\frac{d}{2}$ . A transmitter sends  $\frac{d}{2}$  wakeup packets once before each burst, i.e. in the frame preceding each burst. The energy consumed for the transmission of these wakeup packets is  $\frac{d}{2} \times p^t \times \tau$ . The transmitter uses one slot in the same frame as its wakeup slot, which will be considered separately. The remainder of the frame, i.e.  $(n - \frac{d}{2} - 1)$  slots, are spent sleeping, for which the energy consumption is  $(n - \frac{d}{2} - 1) \times p^s \times \tau$ . Since the inter-burst interval is  $\frac{\beta}{\lambda}$  TDMA-frames, the total power consumed by a transmitter to send wakeup packets before each burst is given by:

$$P_{w_t}^{tx} = \frac{\frac{d}{2} \times p^t \times \tau + (n - \frac{d}{2} - 1) \times p^s \times \tau}{n \times \tau \times \frac{\beta}{\lambda}} \quad (25)$$

In typical wireless systems,  $p^s \ll p^r < p^t$ . Hence, to simplify the equations, we ignore the power consumed during sleep. Setting  $p^s = 0$  the above equation can be rewritten as:

$$P_{w_t}^{tx} = \frac{\frac{d}{2} \times p^t \times \lambda}{n \times \beta} \quad (26)$$

where  $tx$  in  $P_{w_t}^{tx}$  denotes the transmitter and  $w_t$  denotes wakeup packet transmissions.

With an average burst size of  $\beta$ , the transmitter sends one message in its own transmission slot per TDMA frame over  $\beta$  frames. It uses one slot per frame for its wakeup slot which will be taken into account separately. The remainder of the  $n-2$  slots per TDMA frame are spent using sleeping. The energy consumed to send  $\beta$  messages is  $\beta \times p^t \times \tau$ . Since the transmitter sends a burst every inter-burst interval, the power consumed to transmit messages is given by  $P_{t_\beta}^{tx}$  below, with  $t_\beta$  denoting the transmission of  $\beta$  packets. The sleep energy has been ignored deliberately due reasons stated earlier.

$$P_{t_\beta}^{tx} = \frac{\beta \times p^t \times \tau}{n \times \tau \times \frac{\beta}{\lambda}} = \frac{\lambda \times p^t}{n} \quad (27)$$

The transmitter wakes up during its own wakeup slot in every frame. The consumed power attributed to wakeup slot is constant and is given by:

$$P_w^{tx} = \frac{p^r \times \tau}{n \times \tau} = \frac{p^r}{n} \quad (28)$$

Combining (26), (27) and (28), we get the total power consumed by a sender:

$$P^{tx} = P_{w_t}^{tx} + P_{t_\beta}^{tx} + P_w^{tx} \quad (29)$$

The main components of the receiver's power consumption can be divided into three parts: (i) to receive all packets from its corresponding transmitters, (ii) to wake up in its wakeup slot, and (iii) to remain awake for an entire TDMA frame when it receives a corrupt wakeup packet. For the sake of simplicity, the extra power consumption when receiving a corrupt wakeup packet has been omitted because of its infrequent occurrence with the increase in inter-burst interval. During reception, the receiver wakes up during the transmission slots of each of its corresponding transmitters to receive the packets for the burst duration resulting in an energy consumption of  $\beta \times \frac{d}{2} \times p^r \times \tau$ . Following this, the receiver remains awake for  $\delta$  additional frames as explained in [Section 6.3](#). The additional consumption is given by  $\delta \times \frac{d}{2} \times p^r \times \tau$ . Since each burst occurs once every inter-burst duration ( $\frac{\beta}{\lambda}$ ), ignoring the power consumed during sleep, the power consumed to receive messages can be expressed as:

$$P_{r_\beta}^{rx} = \frac{(\beta + \delta) \times \frac{d}{2} \times p^r \times \tau}{n \times \tau \times \frac{\beta}{\lambda}} = \frac{\lambda \times (\beta + \delta) \times \frac{d}{2} \times p^r}{n \times \beta} \quad (30)$$

where  $rx$  in  $P_{r_\beta}^{rx}$  denotes the receiver and  $r_\beta$  denotes reception of a burst size of  $\beta$ .

The power consumed to wake up during its wakeup slot will be same for all nodes and is given by the expression:

$$P_w^{rx} = \frac{p^r \times \tau}{n \times \tau} = \frac{p^r}{n} \quad (31)$$

Using (30) and (31), the total power consumed by a receiver is:

$$P_{rx} = P_{r_\beta}^{rx} + P_w^{rx} \quad (32)$$

Energy performance trends and insights obtained from these models are presented in results in [Section 6.7](#).

## APPENDIX D: List of Publications

### Peer Reviewed Journals

1. **D. Banerjee**, M. Taghizadeh, S. Biswas, *Anonymous network coexistence with slotted wireless channel access*, Elsevier Computer Communications Journal, October 2014
2. **D. Banerjee**, B. Dong, S. Biswas, *Privacy-preserving Channel Access for Internet of Things*, IEEE Internet of Things Journal, August 2014
3. **D. Banerjee**, C. Daigle, B. Dong, K. Wurtz, R. Newberry, J. Siegford, S. Biswas, *Detection of jumping and landing force in laying hens using wireless wearable sensors*, Journal of Poultry Science, August 2014
4. C. Daigle, **D. Banerjee**, S. Biswas, R. Montgomery, J. Siegford, *Moving GIS indoors: spatiotemporal analysis of agricultural animals*, PLoS ONE 9(8): e104002., August 2014
5. C. Daigle, **D. Banerjee**, S. Biswas, J. Siegford, *Non-caged laying hens remain unflappable while wearing body-mounted sensors: levels of agonistic behaviors remain unchanged and resource use is not reduced after habituation*, Journal of Poultry Science, October 2012

### Peer Reviewed Conferences

6. C. Daigle, **D. Banerjee**, R. Montgomery, S. Biswas, and J. Siegford, *Moving Geographic Information Systems research indoors: spatiotemporal analysis of agricultural animals*, In Proceedings of the 48th International Congress of the ISAE, Vitoria-Gastiez, Spain, July 2014
7. **D. Banerjee**, B. Dong, S. Biswas, M. Taghizadeh, *Privacy-preserving Channel Access using Blindfolded Packet Transmissions*, In proceedings of IEEE COMSNETS 2014, Bangalore, India, January 2014
8. **D. Banerjee**, B. Dong, S. Biswas, *ZEA-TDMA: design and system level implementation of a TDMA protocol for anonymous wireless networks*, In proceedings of SPIE Defense and Security Symposium, Mobile Multimedia/Image Processing, Security, and Applications 2013: Multimedia Algorithms and Systems, Baltimore, Maryland, April 2013
9. **D. Banerjee**, M. Taghizadeh, S. Biswas, *Distributed TDMA for Privacy Sensitive Anonymous Networks*, In proceedings of IEEE GLOBECOM 2012, Anaheim, California, December 2012
10. **D. Banerjee**, S. Biswas, C. Daigle, J. Siegford, *Remote Activity Classification of Hens Using Wireless Body Mounted Sensors*, In proceedings of the 9<sup>th</sup> International Conference on Wearable and Implantable Body Sensor Networks (BSN 2012), London, United Kingdom, May 2012

11. **D. Banerjee**, M. Taghizadeh, S. Biswas, *Zero-Exposure Distributed TDMA Using Time-coded Packet Transmissions*, In proceedings of IEEE GLOBECOM 2011, Houston, Texas, December 2011

## **Workshops**

12. C. Daigle, **D. Banerjee**, R. Montgomery, P. Thompson, J. Swanson, S. Biswas, and J. Siegford, *Integrating Technology and Animal Welfare: Space and Resource Use of Individual Non-cage Laying Hens*, presented in Midwest ASAS-ADSA Meeting, Des Moines, IA, March 2014
13. C. Daigle, **D. Banerjee**, J. Siegford, S. Biswas, J. Swanson, *Developments and directions of monitoring non-cage laying hens using a wireless body mounted sensor*, Annual Meeting of the Netherlands Society for Behavioural Biology, November 2011

## REFERENCES

## REFERENCES

- [1] B. Greenstein, D. McCoy, J. Pang, T. Kohno, S. Seshan, and D. Wetherall, "Improving Wireless Privacy with an Identifier-free Link Layer Protocol," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, New York, NY, USA, 2008, pp. 40–53.
- [2] Y. Fan, B. Lin, Y. Jiang, and X. Shen, "An Efficient Privacy-Preserving Scheme for Wireless Link Layer Security," in *IEEE Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008*, 2008, pp. 1–5.
- [3] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "MASK: anonymous on-demand routing in mobile ad hoc networks," *IEEE Trans. Wirel. Commun.*, vol. 5, no. 9, pp. 2376–2385, Sep. 2006.
- [4] B. Radosavljevic and B. Hajek, "Hiding traffic flow in communication networks," in *IEEE Military Communications Conference, 1992. MILCOM '92, Conference Record. Communications - Fusing Command, Control and Intelligence*, 1992, pp. 1096–1100 vol.3.
- [5] Z. Wan, K. Xing, and Y. Liu, "Priv-Code: Preserving privacy against traffic analysis through network coding for multihop wireless networks," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 73–81.
- [6] "Internet of Things in 2020: Roadmap for the Future." [Online]. Available: <http://www.caba.org/resources/Documents/IS-2008-93.pdf>.
- [7] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
- [8] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.
- [9] C. P. Mayer, "Security and Privacy Challenges in the Internet of Things," *Electron. Commun. EASST*, vol. 17, no. 0, Feb. 2009.
- [10] V. Srinivasan, J. Stankovic, and K. Whitehouse, "Protecting Your Daily In-home Activity Information from a Wireless Snooping Attack," in *Proceedings of the 10th International Conference on Ubiquitous Computing*, New York, NY, USA, 2008, pp. 202–211.
- [11] B. L. 0004, Y. Jiang, F. Sha, and R. Govindan, "Cloud-enabled privacy-preserving collaborative learning for mobile sensing,," in *SenSys*, 2012, pp. 57–70.
- [12] O. Vermesan and P. Friess, *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. Rivers Publishers.

- [13] "Lockitron - Keyless entry using your phone." [Online]. Available: <https://lockitron.com/>. [Accessed: 28-Feb-2014].
- [14] F. Yu, T. Wu, and S. Biswas, "Toward In-Band Self-Organization in Energy-Efficient MAC Protocols for Sensor Networks," *IEEE Trans. Mob. Comput.*, vol. 7, no. 2, pp. 156–170, 2008.
- [15] W.-Z. Song, R. Huang, B. Shirazi, and R. LaHusen, "TreeMAC: Localized TDMA MAC Protocol for Real-time High-data-rate Sensor Networks," *Pervasive Mob Comput*, vol. 5, no. 6, pp. 750–765, Dec. 2009.
- [16] Z. Chen and A. Khokhar, "Self organization and energy efficient TDMA MAC protocol by wake up for wireless sensor networks," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004*, 2004, pp. 335–341.
- [17] I. Rhee, A. Warriar, J. Min, and L. Xu, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad Hoc Networks," *IEEE Trans. Mob. Comput.*, vol. 8, no. 10, pp. 1384–1396, 2009.
- [18] J. Chen and S. Jiang, "Improvement of Slots Utilization with a Stealing-TDMA Protocol for Ad Hoc Network," in *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, 2006, pp. 1–5.
- [19] 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, Dec. 2002.
- [20] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Commun ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.
- [21] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: design of a type III anonymous remailer protocol," in *2003 Symposium on Security and Privacy, 2003. Proceedings*, 2003, pp. 2–15.
- [22] M. Rennhard and B. Plattner, "Introducing MorphMix: Peer-to-peer Based Anonymous Internet Usage with Collusion Detection," in *Proceedings of the 2002 ACM Workshop on Privacy in the Electronic Society*, New York, NY, USA, 2002, pp. 91–102.
- [23] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 482–494, 1998.
- [24] D. Goldschlag, M. Reed, and P. Syverson, "Onion Routing for Anonymous and Private Internet Connections," *Commun. ACM*, vol. 42, pp. 39–41, 1999.
- [25] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: attacks and defenses," in *IN: PROCEEDINGS OF ESORICS*, 2006, pp. 18–33.

- [26] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis," in *In Proceedings of the 16th Network and Distributed Security Symposium*, 2009, pp. 237–250.
- [27] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba, "SDAR: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks," in *29th Annual IEEE International Conference on Local Computer Networks*, 2004, 2004, pp. 618–624.
- [28] S. Seys and B. Preneel, "ARM: Anonymous Routing Protocol for Mobile Ad hoc Networks," in *20th International Conference on Advanced Information Networking and Applications*, 2006. *AINA 2006*, 2006, vol. 2, pp. 133–137.
- [29] D. Sy, R. Chen, and L. Bao, "ODAR: On-Demand Anonymous Routing in Ad Hoc Networks," in *In Proceedings of The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2006, pp. 267–276.
- [30] Y. Fan, Y. Jiang, H. Zhu, J. Chen, and X. S. Shen, "Network Coding Based Privacy Preservation Against Traffic Analysis in Multi-Hop Wireless Networks," *Trans Wirel. Comm*, vol. 10, no. 3, pp. 834–843, Mar. 2011.
- [31] P. A. Chou and Y. Wu, "Network Coding for the Internet and Wireless Networks," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 77–85, 2007.
- [32] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "DESYNC: Self-Organizing Desynchronization and TDMA on Wireless Sensor Networks," in *6th International Symposium on Information Processing in Sensor Networks*, 2007. *IPSN 2007*, 2007, pp. 11–20.
- [33] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers, and IEEE-SA Standards Board, *IEEE Standard 802.11-1999: "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications."* New York, N.Y., USA: Institute of Electrical and Electronics Engineers, 1999.
- [34] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2004, pp. 95–107.
- [35] J. Ma, W. Lou, Y. Wu, M. Li, and G. Chen, "Energy Efficient TDMA Sleep Scheduling in Wireless Sensor Networks," in *IEEE INFOCOM 2009*, 2009, pp. 630–638.
- [36] Y. Wu, S. Fahmy, and N. B. Shroff, "Energy Efficient Sleep/Wake Scheduling for Multi-Hop Sensor Networks: Non-Convexity and Approximation Algorithm," in *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, 2007, pp. 1568–1576.
- [37] F. Zhang, W. He, and X. Liu, "Defending Against Traffic Analysis in Wireless Networks through Traffic Reshaping," in *2011 31st International Conference on Distributed Computing Systems (ICDCS)*, 2011, pp. 593–602.

- [38] Y. Qin, Y. Yin, D. Huang, and N. Shah, "A comparative study on anonymous 802.11n protocols," in *IEEE Military Communications Conference, 2008. MILCOM 2008*, 2008, pp. 1–7.
- [39] S. Jiang, "An Anonymous MAC Protocol for Wireless Ad-hoc Networks," in *Mobile and Wireless Network Security and Privacy*, S. K. Makki, P. Reiher, K. Makki, N. Pissinou, and S. Makki, Eds. Springer US, 2007, pp. 191–204.
- [40] IEEE Computer Society, LAN/MAN Standards Committee, and Institute of Electrical and Electronics Engineers, "IEEE Std 802.15.4-2011, IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs)." .
- [41] S. Jiang and N. H. Vaidya, "A mix route algorithm for mix-net in wireless mobile ad hoc networks," in *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, 2004, pp. 406–415.
- [42] S. Jiang, N. F. Vaidya, and W. Zhao, "A dynamic mix method for wireless ad hoc networks," in *IEEE Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force*, 2001, vol. 2, pp. 873–877 vol.2.
- [43] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing Attacks in Low-Latency Mix Systems," in *Financial Cryptography*, A. Juels, Ed. Springer Berlin Heidelberg, 2004, pp. 251–265.
- [44] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: A Localized, Sink-oriented MAC for Boosting Fidelity in Sensor Networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2006, pp. 293–306.
- [45] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *IEEE INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, 2002, vol. 3, pp. 1567–1576 vol.3.
- [46] I. Rhee, A. Warrier, M. Aia, J. Min, and M. L. Sichitiu, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," *IEEEACM Trans. Netw.*, vol. 16, no. 3, pp. 511–524, 2008.
- [47] Y. Zhang, S. Zheng, and S. Xiong, "A Scheduling Algorithm for TDMA-Based MAC Protocol in Wireless Sensor Networks," in *First International Workshop on Education Technology and Computer Science, 2009. ETCS '09*, 2009, vol. 3, pp. 148–151.
- [48] A. Smailagic and D. Kogan, "Location sensing and privacy in a context-aware computing environment," *IEEE Wirel. Commun.*, vol. 9, no. 5, pp. 10–17, Oct. 2002.
- [49] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *IEEE INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, 2000, vol. 2, pp. 775–784 vol.2.

- [50] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An Efficient Privacy-Preserving Scheme against Traffic Analysis Attacks in Network Coding," in *IEEE INFOCOM 2009*, 2009, pp. 2213–2221.
- [51] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, "Smart objects as building blocks for the Internet of things," *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan. 2010.
- [52] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives," in *2006 6th International Conference on ITS Telecommunications Proceedings*, 2006, pp. 761–766.
- [53] B. Awerbuch, D. Holmer, H. Rubens, K. Chang, and I.-J. Wang, "The pulse protocol: sensor network routing and power saving," in *2004 IEEE Military Communications Conference, 2004. MILCOM 2004*, 2004, vol. 2, pp. 662–667 Vol. 2.
- [54] J. M. Reason and J. M. Rabaey, "A Study of Energy Consumption and Reliability in a Multi-Hop Sensor Network," *ACM Mob. Comput. Commun. Rev.*, vol. 8, pp. 84–97, 2004.
- [55] T. van Dam and K. Langendoen, "An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2003, pp. 171–180.
- [56] L. M. Feeney, "Exploring semantic interference in heterogeneous sensor networks," 2008, p. 45.
- [57] D. Domenicali, L. De Nardis, and M. Di Benedetto, "UWB body area network coexistence by interference mitigation," in *IEEE International Conference on Ultra-Wideband, 2009. ICUWB 2009*, 2009, pp. 713–717.
- [58] N. Nordin and F. Dressler, "Effects and Implications of Beacon Collisions in Co-Located IEEE 802.15.4 Networks," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, 2012, pp. 1–5.
- [59] S. Drude, "Requirements and Application Scenarios for Body Area Networks," in *Mobile and Wireless Communications Summit, 2007. 16th IST*, 2007, pp. 1–5.
- [60] O. C. Omeni, "A Perspective of the BAN MAC." .
- [61] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical identification of encrypted Web browsing traffic," in *2002 IEEE Symposium on Security and Privacy, 2002. Proceedings*, 2002, pp. 19–30.
- [62] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, and M. Welsh, "Simulating the Power Consumption of Large-scale Sensor Network Applications," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2004, pp. 188–200.

- [63] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta, "ADHOC MAC: New MAC Architecture for Ad Hoc Networks Providing Efficient and Reliable Point-to-Point and Broadcast Services," *Wirel. Netw.*, vol. 10, no. 4, pp. 359–366, Jul. 2004.
- [64] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta, "RR-ALOHA, a Reliable R-ALOHA broadcast channel for ad-hoc inter-vehicle communication networks," in *in: Proceedings of Med-Hoc-Net 2002*, 2002.
- [65] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, unobservability, pseudonymity, and identity management – a consolidated proposal for terminology," 2005.
- [66] Y. Zhang, Y. Zhang, and K. Ren, "DP #x000B2AC: Distributed Privacy-Preserving Access Control in Sensor Networks," in *IEEE INFOCOM 2009*, 2009, pp. 1251–1259.
- [67] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET," in *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*, New York, NY, USA, 2007, pp. 19–28.
- [68] D. He, J. Bu, S. Zhu, M. Yin, Y. Gao, H. Wang, S. Chan, and C. Chen, "Distributed privacy-preserving access control in a single-owner multi-user sensor network," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 331–335.
- [69] K. Ren, W. Lou, K. Kim, and R. Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1373–1384, Jul. 2006.
- [70] D. Huang, "Traffic Analysis-based Unlinkability Measure for IEEE 802.11B-based Communication Systems," in *Proceedings of the 5th ACM Workshop on Wireless Security*, New York, NY, USA, 2006, pp. 65–74.
- [71] F. Armknecht, J. Girao, A. Matos, and R. L. Aguiar, "Who Said That? Privacy at Link Layer," in *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, 2007, pp. 2521–2525.
- [72] Y. Qin, D. Huang, and B. Li, "STARS: A Statistical Traffic Pattern Discovery System for MANETs," *IEEE Trans. Dependable Secure Comput.*, vol. 11, no. 2, pp. 181–192, Mar. 2014.