WIRELESS MESH PERSONAL AREA NETWORKS

– System Design and Analysis

by

JIANLIANG ZHENG

A dissertation submitted to the Graduate Faculty in Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2006

UMI Number: 3204977

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This manuscript has been read and accepted for the
Graduate Faculty in Engineering in satisfaction of the
dissertation requirement for the degree of Doctor of Philosophy.

12/29/2005
Date

Professor **Myung J. Lee**
Chair of Examining Committee

12/29/2005
Date

Dean **Mumtaz Kassir**
Executive Officer

Professor **Michael Anshel**
The City University of New York

Professor **Tarek N. Saadawi**
The City University of New York

Professor **Yi Sun**
The City University of New York

Professor **Kaliappa Ravindran**
The City University of New York

Professor **Jizhong Xiao**
The City University of New York
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

WIRELESS MESH PERSONAL AREA NETWORKS

– System Design and Analysis

by

Jianliang Zheng

Advisers: Professor Myung J. Lee

Professor Michael Anshel

A wireless mesh network is an emerging paradigm in the field of wireless networking technologies ranging from personal area networks (PANs) to wide area networks (WANs). This dissertation focuses on wireless mesh personal area networks (WMPANs) and addresses some key issues of WMPANs, including medium access control (MAC), scheduling and midware, addressing, routing, and security.

The IEEE standard, 802.15.4, defines the physical (PHY) layer and MAC sublayer specifications for low rate WMPANs. An in-depth performance study is conducted to help IEEE to verify and/or improve the design and facilitate researchers and manufacturers to develop products based upon this new standard. Some problems are identified and remedies are suggested. New approaches such as transmission scheduling and medium access midware are also studied. While most research work concentrates on low rate

WMPANs, a parallel and accumulative reception scheme using code division multiple access (CDMA) is proposed for high rate WMPANs and other high rate wireless mesh networks.

At the core of WMPANs lies the design problem for efficient and scalable routing algorithms. Given the constraints on storage and power supply, most existing routing protocols are too cumbersome for WMPANs. An adaptive block addressing (ABA) scheme is proposed for logic address assignment and network autoconfiguration purpose. By binding logic addresses to the network topology, routing can be done without going through route discovery. Several routing protocols, all based on ABA, are designed and evaluated, including adaptive robust tree (ART) routing, meshed ART (MART) routing, dual routings, and topology-guided distributed link state (TDLS) routing.

Security is another problem that should be solved before WMPANs can be pervasively deployed. A systematic analysis of the threats faced by WMPANs is performed. Problems with the security functions and service functions of the current WMPAN architecture are covered and solutions are provided. A new security architecture based on a lightweight public key scheme, called Derivable Public Key (DPK), is also proposed. In DPK, each device only needs to obtain the public key of the trust center and its own secret key. These two keys enable a device to communicate securely with any other device.

To my parents, my wife and my sons

Acknowledgements

I would like to thank all the people who have led me to this particular pinnacle of achievement. I am indebted to a great many individuals for their support in bringing this dissertation to completion.

Foremost, I would like to express my full gratitude to my mentor, Dr. Myung J. Lee. As an adviser, Dr. Lee has guided me to the wireless world using his broad knowledge, wisdom, and patience. He has always been an inspiring source of help and support for me. Dr. Lee also shares with me his vision for life. His sincerity, dedication, and gentleness has helped me to be the person I can be.

I am also full of gratitude to my co-mentor, Dr. Michael Anshel. Dr. Anshel's rich knowledge in security and cryptography has been the strong backing of my research in the area of communication security. His understanding, support, and superior personality has been an important stimulus that keeps me moving in this challenging research direction.

I would also like to thank Dr. Tarek N. Saadawi and Dr. Yi Sun for all the instruction and support during my doctoral study. Thank Dr. Tarek N. Saadawi, Dr. Yi Sun, Dr. Kaliappa Ravindran, and Dr. Jizhong Xiao for taking time to review this dissertation and serving on my examining committee.

I would like to thank Dr. Umit Uyar for bringing me into the Communications and Networks research group and for all his instruction. I also want to thank Dr. Frederick E. Thau for his support, which has made my doctoral study easier and more fruitful.

Thanks also go to Chunhui Zhu, Yong Liu, Xuhui Hu, Taekyoun Kwon, Hsin-Hui Juan, June-Seung Yoon, Baozhi Chen, Junjun Li, and all other people at the City University of New York (CUNY) that I worked with during these years.

I am much obliged to my wife, Jie, a wonderful research colleague and a perfect life partner. Without her love and support, I could not have been where I am now standing. Also thank my two lovely sons, Zhi and Evan, for all the happiness they have brought me. I am forever indebted to my family. Thank my parents, my sister and brothers, my parents-in-law, and brother-in-law for so many years of support, encouragement, and understanding.

Last, but not least, I want to thank the financial supporters of my doctoral research: the U.S. Army Research Laboratory and the Samsung Advanced Institute of Technology.

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

ABA	Adaptive Block Addressing
ACK	ACKnowledgement
ACL	Access Control List
AD	Acceptance Degree
AES	Advanced Encryption Standard
AODV	Ad hoc On-demand Distance Vector
AODVjr	AODV junior
AP	Access Point
ART	Adaptive Robust Tree
ARTT	ART Table
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLS	Basic Link State
BO	Beacon Order
BPSK	Binary Phase Shift Keying
BS	Base Station

BSS	Basic Service Set
BTR	Broadcast Transaction Record
BTT	Broadcast Transaction Table
BWA	Broadband Wireless Access
CAP	Contention Access Period
CBC-MAC	Cipher Block Chaining Message Authentication Code
CBR	Constant Bit Rate
CCA	Clear Channel Assessment
CCK	Complementary Code Keying
CCM	Counter with CBC-MAC
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CID	Context ID
CRC	Cyclic Redundancy Check
CSMA-CA	Carrier Sense Multiple access with Collision Avoidance
CTS	Clear-To-Send
DLS	Distributed Link State
DOS	Denial Of Service
DPK	Derivable Public Key
DR	Dual Routings
DS-CDMA	Direct Sequence CDMA
DSDV	Destination Sequenced Distance Vector

DSL	Digital Subscriber Line
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
ECC	Elliptic Curves Cryptography
ED	Energy Detection
ELS	Extended Link State
ESS	Extended Service Set
ET	Exposed Terminal
FDMA	Frequency Division Multiple Access
FFD	Full Function Device
FHSS	Frequency-Hopping Spread Spectrum
GPS	Global Positioning System
GTD	Guard Time Duration
GTS	Guaranteed Time Slot
HR-WPAN	High Rate Wireless Personal Area Network
HT	Hidden Terminal
IC	Integrity Code; Interference Cancellation
ID	IDentifier
IFS	Inter-Frame Space
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6

ISM	Industrial Scientific Medical
LLC	Link Layer Control
LOS	Line Of Sight
LQ	Link Quality
LQI	Link Quality Indication
LR-WPAN	Low Rate Wireless Personal Area Network
LST	Link State Table
MAC	Medium Access Control; Message Authentication Code
MAI	Multiple Access Interference
MANET	Mobile Ad hoc NETwork
MART	Meshed ART
MASM	Medium Access Scheduling Midware
MEMS	Micro-ElectroMechanical System
MIMO	Multiple-Input Multiple-Output
MPIB	MAC PAN Information Base
MTU	Maximum Transmission Unit
NACK	Negative ACK
NAT	Network Address Translation
NTT	Non-Tree Table
NPDU	Network Protocol Data Unit
NWK	NetWorK
OFDM	Orthogonal Frequency Division Multiplexing

OLSR	Optimized Link State Routing
O-QPSK	Offset Quadrature Phase Shift Keying
OSPF	Open Shortest Path First
PER	Packet Error Rate
PGP	Pretty Good Privacy
PHY	PHYsical
PKI	Public Key Infrastructure
PMP	Point-to-MultiPoint
PN	Pseudo-random Noise
POS	Personal Operating Space
PPDU	PHY Protocol Data Unit
PRN	Packet Radio Network
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RC	Radius Counter
RCFM	Route ConFirMation
RD	Reuse Distance
RERR	Route ERRor
RF	Radio Frequency
RFD	Reduced Function Device
RIP	Routing Information Protocol
ROT	Receiver Oriented TDMA

RREP	Route REPly
RREQ	Route REQuest
RTS	Request-To-Send
SAP	Service Access Point
SD	Superframe Duration
SIR	Signal-to-Interference Ratio
SNMP	Simple Network Management Protocol
SO	Superframe Order
SOHO	Small Office Home Office
SPF	Single Point of Failure
SS	Spread Spectrum; Subscriber Station
SSCS	Service Specific Convergence Sublayer
STA	STAtion
TBRPF	Topology Broadcast based on Reverse-Path Forwarding
TC	Trust Center
TDLS	Topology-guided Distributed Link State
TDMA	Time Division Multiple Access
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TM	Train Multicast
TSC	Time Slot Cycle
TST	Time Slot Table
TTL	Time To Live

UWB	Ultra WideBand
VoD	Video on Demand
VoIP	Voice on IP
WDS	Wireless Distribution System
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WMLAN	Wireless Mesh Local Area Network
WMPAN	Wireless Mesh Personal Area Network
WMSN	Wireless Mesh Sensor Network
WSN	Wireless Sensor Network
ZDO	ZigBee Device Object

Part I

Introduction and Overview

Chapter 1

Introduction

1.1 Background and Motivation

1.2 Contributions of the Dissertation

1.3 Outlines of the Dissertation

Chapter 1

Introduction

1.1 Background and Motivation

A wireless mesh network is an emerging paradigm in the field of wireless networking technologies ranging from personal area networks (PANs) to wide area networks (WANs). The commercial success of Wi-Fi has in part stimulated the development of wireless mesh networks. The persistent driving force, however, comes from the envisioned advantages of wireless mesh networks themselves. First of all, mesh connectivity significantly enhances the network performance, such as fault tolerance, load balancing, and throughput; in addition, the self-configuring and self-healing feature of wireless mesh networks not only enables them to be deployed on the fly and on the cheap, but also enhances system resilience and reliability; moreover, the ability of wireless mesh networks to provide flexible coverage areas, particularly where other wireless technologies tumble due to the lack of line of sight (LOS), makes them unique for many applications. With all these, plus the minimal up-front investment and the simple off-the-shelf devices

that are outfitted with radio communications gear, a consensus that wireless mesh networks will play a pivotal role on the next wireless frontier may be on the cards.

The focus of this dissertation is wireless mesh personal area networks (WMPANs). Many features of WMPANs, for instance, using globally available industrial scientific medical (ISM) frequency bands, low cost, power conservation, self-configuration, secure data transfer and so on, make them a technology having the promise to unify all those simple devices from different manufacturers and bring networks to the level of each person. WMPANs, as an enabling technology, not only open the door to an enormous number of new applications, but also add value to many existing applications. By extending networks to cover all the simple devices and with the emerging of many interesting and wonderful applications, we are ushering in a ubiquitous networking era.

Nevertheless, before the promise of WMPANs can be realized, much research remains to be done. One essential challenge is capacity and range enhancement. It is well known that multi-hop wireless networks suffer from scalability issues [108–110]. This behooves researchers to revisit the whole network protocol stack, especially the physical (PHY) layer, the medium access control (MAC) sublayer, and the network (NWK) layer. And cross-layer optimization is likely to be a commonplace. Another wrinkle is how to provision efficient and scalable routing in mesh environments. Yet an

additional kink is privacy and security [8, 9], which has stymied the deployment of wireless networks. Moreover, energy consumption is of great concern [10] and power saving is an indispensable feature if those networks are to survive, let alone thrive. And low cost is another critical feature required for the pervasive deployment of those networks. Finally, self-configuration, zero or minimal maintenance, and coexistence with other wireless networks make designing a WMPAN even more challenging.

1.2 Contributions of the Dissertation

This dissertation addresses some key issues of WMPANs, including medium access control, scheduling and midware, addressing, routing, and security. The IEEE standard, 802.15.4, defines the PHY layer and MAC sublayer specifications for low rate WMPANs. The primary design goal of IEEE 802.15.4 is low cost, low power consumption, and support of simple devices such as sensors and actuators. As a result, 802.15.4 MAC has some distinct design features, for instance, relatively short backoff period, no request-to-send (RTS) and clear-to-send (CTS) control frames, beacon mode and superframe structure, and orphaning and coordinator relocation. An in-depth performance study is conducted to help IEEE to verify and/or improve the design and facilitate researchers and manufacturers to develop products based upon this new standard. Some problems are identified, in-

cluding repeated collisions, high collision probability at the beginnings of superframes, non-atomic transactions, insufficient support of multi-hop beacon enabled networks, and improper default status of transceiver. Remedies are suggested. Two new approaches, receiver oriented time division multiple access (TDMA) and medium access scheduling midware (MASM), are also studied. The simple receiver oriented TDMA (ROT) can be used to remove most collisions, which result from hidden terminal problems, and the MASM scheme can be employed to eliminate all collisions.

While most research work concentrates on low rate WMPANs, a parallel and accumulative reception scheme is also proposed for high rate WMPANs and other high rate wireless mesh networks. The new scheme exploits the code division multiple access (CDMA) technology to provision many-to-one simultaneous wireless communication service without utilizing any time division multiple access (TDMA) scheme. It also supports accumulative receptions, namely, a receiver can buffer partially damaged (re-)transmitted frames and add those frames efficiently to form an error-free frame. Our simulation results show that the new scheme outperforms IEEE 802.11 in both non-mobile and mobile environments.

At the core of WMPANs lies the design problem for efficient and scalable routing algorithms. Given the constraints on storage and power supply, most existing routing protocols including ad-hoc on-demand distance vector (AODV) routing [11], dynamic source routing (DSR) [12], optimized

link state routing (OLSR) [13], and topology broadcast based reverse-path forwarding (TBRPF) routing [14], prove to be too cumbersome for WM-PANs. An adaptive block addressing (ABA) scheme is proposed in this dissertation for logic address assignment as well as network autoconfiguration purpose. The scheme takes into account the actual network topology and thus is fully topology-adaptive. By binding logic addresses to the network topology, routing can be done without going through route discovery. This eliminates the initial route discovery latency, saves storage space otherwise needed for routing table, and reduces the communication overhead and energy consumption.

While ABA itself is a self-contained routing protocol, several other routing protocols, all based on ABA, are further designed and evaluated. The adaptive robust tree (ART) routing combines the ABA scheme with a tree link repair scheme, thus eliminating the single point of failure (SPF) problem, which is faced by most tree routings. By meshing neighbors together, meshed ART (MART) further improves the robustness and quality of routes. The dual routing approach, which merges MART and another on-demand wireless routing, takes advantage of both proactive and reactive routing protocols to provide optimal or near-optimal routes and, at the same time, maintain low initial latency, low control overhead, and low memory consumption. Finally, a distributed link state (DLS) scheme is proposed and put on top of the ABA scheme to improve the quality of routes, robustness, and

load balancing. The network topology reflected in logic addresses is used as a guideline to tell towards which direction (rather than next hop) a packet should be relayed. The next hop is derived from each relaying node's local link state table (LST). The routing scheme, named as topology-guided DLS (TDLS) as a whole, scales well with regard to various performance metrics. The ability of TDLS to provide multiple paths also precludes the need for explicit route repair, which is the most complicated part in many wireless routing protocols.

Security is another problem that should be solved before WMPANs can be pervasively deployed. Pursuing security in WMPANs is a challenging task. On one hand, wireless communications are inherently susceptible to interception and interference. On the other hand, most devices in WMPANs are resource-constrained and lack physical safeguards. A systematic analysis of the threats faced by WMPANs is performed. Problems with both the security functions and other service functions of the current WMPAN architecture are covered and remedies are suggested. A new security architecture based on a lightweight public key scheme, called Derivable Public Key (DPK), is also proposed. In DPK, each device only needs to obtain the public key of the trust center (TC) and its own secret key. These two keys enable a device to communicate securely with any other device. By contrast, in other public key schemes, a device has to get the public key of each communication peer. For a network with large number of tiny de-

vices, those old-good public key schemes consume too much network resources. As a result, public key schemes have so far been excluded from resource-constrained wireless networks, and authentications for multicast and broadcast communications are not supported. The lack of authentications for multicast and broadcast communications in resource-constrained wireless networks accounts for many attacks. The DPK scheme holds the promise to change this situation and resource-constrained wireless networks can expect to be better secured.

1.3 Outlines of the Dissertation

The dissertation comprises twelve chapters, which fall into five parts. The first part, *Introduction and Overview*, consists of chapters 1–3. The second part, *Lower Layer Issues – MAC, Scheduling, and Midware*, includes chapters 4–6. The third part, *Network Layer Issues – Addressing and Mesh Routing*, contains chapters 7–9. The fourth part, *Security Issues – Threats and Countermeasures*, has two chapters, that is, chapters 10 and 11. The last part, *Conclusions and Future Work*, contains the last chapter, chapter 12. Following outlines the topics covered in this dissertation.

- Chapter 1 sets forth the background, motivation, and contributions of the dissertation.
- Chapter 2 gives an overview of wireless mesh networking.

- Chapter 3 presents an introduction to wireless mesh personal area networks.
- Chapter 4 evaluates the performance of the IEEE 802.15.4 standard.
- Chapter 5 presents two transmission scheduling schemes, i.e., receiver oriented TDMA (ROT) and medium access scheduling midware (MASM).
- Chapter 6 proposes a parallel and accumulative reception MAC scheme, called parallel and accumulative reception using time division multiple access (PAR-CDMA).
- Chapter 7 introduces an adaptive block addressing (ABA) scheme, a memory-efficient self-routing protocol, called adaptive robust tree (ART) routing, and its meshed form, meshed ART (MART) routing.
- Chapter 8 describes a dual routing approach.
- Chapter 9 proposes an efficient scalable wireless mesh routing protocol, called topology-guided distributed link state (TDLS).
- Chapter 10 focuses on the security problems in WMPANs and provides a detailed analysis of the threats faced by WMPANs.
- Chapter 11 proposes a novel public key scheme, called derivable public key (DPK).
- Chapter 12 summarizes the dissertation and discusses the open issues and future research directions.

Chapter 2

A Contemporary Overview of Wireless Mesh Networks

Chapter 2

A Contemporary Overview of Wireless Mesh Networks

Wireless mesh networking is a promising technology for numerous applications, and especially appeals to those applications that can not be directly supported by other wireless technologies. The commercial success of Wi-Fi and advances in many wireless technologies have in part stimulated the development of wireless mesh networks. The persistent driving force, however, comes from the envisioned advantages of wireless mesh networks themselves, including flexible coverage, robustness, self-configuration, easy maintenance, low cost, and so on. Nevertheless, before the technology is ripe for major commercial development, there still exist some hurdles that researchers as well as enterprises need to overcome. This chapter presents an overview of this emerging technology, including its advantages, opportunities, challenges, and related standards activities around IEEE and ZigBee.

2.1 Introduction

A disruptive technology, wireless mesh networking, is knocking on the doors of communication industry. While conceptually the technology is not new, not until recently have researchers, enterprises, as well as consumers thought this may be something over the next hill rather than a pie in the sky. Wireless networks have been making inroads into private residences, office

buildings, universities, and other industrial and commercial venues around the globe in the past several years. This commercial success together with some emerging wireless technologies, especially radio technologies such as multiple-input multiple-output (MIMO) systems [1, 2] and directional and smart antennas [3, 4], suggests it is now the time for wireless mesh networking to come into play. Seeing its potential to reshape the landscape of communications, major electronic consumer companies as well as small startups are staking out this emerging technology and preparing for the market to take off.

What is distinctive about wireless mesh technology is neither *wireless* nor *mesh*, but the binding of these two. Both the wired Internet and the public switched telephone network (PSTN) are essentially mesh networks and have long been there; and Wi-Fi, the to date dominant wireless local area network (WLAN) technology, is also five years old. In stark contrast to the wired Internet and PSTN, wireless mesh technology allows designers to build electronic networks without ripping apart buildings or tearing up streets to wire miles of copper or fiber cables. Yet it has a flexible coverage and can seep where is beyond the reach of other wireless technologies such as Wi-Fi, which has been mainly used as a last-hop technology so far.

Although the above mentioned advantages of wireless mesh technology already put it in a superior position, one can enumerate more. For example, mesh connectivity significantly enhances the network performance,

Table 2.1: Wireless Mesh Standards

Wireless mesh	Standard
LR-WPAN	IEEE 802.15.5, ZigBee
WLAN	IEEE 802.11s
WiMAX	IEEE 802.16
Mobile-Fi	IEEE 802.20

such as fault tolerance, load balancing, and throughput; in addition, the self-configuring and self-healing feature of wireless mesh networks not only enables them to be deployed on the fly and on the cheap, but also enhances system resilience and reliability; moreover, with the minimal up-front investment and being easy to adjust and expand, wireless mesh networks cater for the requirements of different consumers, large or small. With all these, wireless mesh technology is going to open a world of possibilities and develop a burgeoning market in the foreseeable future, though considerable research efforts are still needed.

While a few companies have been rolling out proprietary wireless mesh products for some time, it is the involvement of international standard groups, the major driving force behind various technologies, that signals the arrival of wireless mesh era. As can be seen from Table 2.1, IEEE has been playing a key role in the development of wireless mesh standards for low rate wireless personal area networks (LR-WPANs), wireless local area networks (WLANs), and wireless metropolitan area networks (WMANs) including WiMAX (Worldwide Interoperability for Microwave Access) and

Mobile-Fi.

The remainder of this chapter is organized as follows. Opportunities and challenges faced by wireless mesh technology are first discussed in section 2.2. A brief overview of wireless mesh standards activities around IEEE and ZigBee is presented in section 2.3. And conclusions are given in section 2.4.

2.2 Opportunities and Challenges

The world wide web (WWW) has revolutionized the way people acquire information and created a huge market. Now engineers are trying to weave another web, but without thread this time. Albeit wireless mesh technology is still in its infancy, the potential with which it will transform our world is enormous. Featured flexible coverage, quick deployment, easy maintenance, high reliability, high scalability, resilience, and low-cost, wireless mesh technology will be a strong candidate and sometimes even the only player for many existing and new applications.

Scenarios where wireless mesh technology is likely to provide a more versatile or affordable solution than other wired or wireless technologies include:

- extensive coverage areas, for example, within an office building, throughout a stadium, or spanning a sprawling facility.

- areas that are unwired, under-wired, or hard-to-wire, such as highways, conduits, golf courses, or farmlands.
- emergency situations, for instance, fire fighting or military operations.

And following are a few examples of the applications that wireless mesh technology could notch up.

- *Campus and municipal networking*: Given its advantages, wireless mesh technology is a natural choice for open air networks that cover areas ranging from college campus to part or all of a metropolitan area. Without wiring, wireless mesh networks can provide a wide spectrum of networking functions, including Internet access, off-the-scene lecturing, public service enquiry, communications for first responders, and urban habitat monitoring. And being able to provide in-vehicle access to building blueprints, traffic monitoring and controlling systems, geographical information database, and weather forecast information, wireless mesh networks have promised to bring about a substantial improvement in public health and security.
- *Far-flung rural region Internet service provisioning*: In most remote rural areas, wiring for Internet access is impractical. In this case, a wireless mesh can be employed to span the network coverage from a wired backbone, thus sidestepping the problem of the most expensive and complicated “last mile” connections.

- *Transportation and shipping*: Wireless mesh networks are well suited to transportation and shipping industries. From airports to seaports to subways, wireless mesh networks can be deployed to meet various needs, for example, information inquiring, security and video surveillance, inventory tracking or logistics, to name a few. A wireless mesh can also be used to establish communications among moving vehicles, providing the ability to share information on traffic conditions.
- *Warehousing and manufacturing*: In facilities such as large crowded warehouses and factories, pulling cables proves to be a difficult job; one either can not get to a location or may run into Ethernet's 100-meter cabling limitation. And maintenance could be an even harder task. Cables can be eroded or inadvertently damaged in such harsh environments, and locating and replacing damaged cables is a headache. Wireless mesh technology, on the other hand, can greatly simplify the installation and ease the maintenance of a network, thereby providing an affordable and unobtrusive solution.
- *Environment surveillance*: By combining different wireless mesh networks such as WiMAXs, WLANs, and LR-WPANs, human being will have a closer look at the globe they have been living, from densely populated metropolitan areas, to placid suburbs, to remote rural or maintain regions. With the ability to monitor air, water, soil, and many more, human being will be sitting in a better position for protecting

environments and preventing disasters, both natural and man-made.

With the beautiful picture unfolded above, we now take a look at the challenges faced by researchers and enterprises as well. Routing is a key function of wireless mesh networks. In some cases, routing protocols designed for wireless mobile ad hoc networks (MANETs) such as ad-hoc on-demand distance vector (AODV) routing [11], dynamic source routing (DSR) [12], optimized link state routing (OLSR) [13], and topology broadcast based reverse-path forwarding (TBRPF) routing [14], can be directly applied to mesh networks. And standard Internet routing protocols such as open shortest path first (OSPF) [15] protocol and routing information protocol (RIP) [16] can also be used if the network topology is relatively stable. Nevertheless, for a large scale wireless mesh network, those layer 3 routing protocols do not scale well in terms of frequency bandwidth, memory storage, and end-to-end latency. As such, researchers begin to seek other approaches such as cross-layer design and layer 2 or 2.5 routing. New routing protocols based on multi-radio, multi-channel, and directional and smart antennas [3,4] are also being studied. Cost metrics such as link quality, congestion, node energy level, degree of connectivity, as well as the old-age hop count are being considered. And compromises among various performance metrics are being ascertained. In short, much research needs to be done to increase the network capacity, support resource-constrained devices, and provide quality of service (QoS), which is a must-have aspect if video on

demand (VoD), voice on IP (VoIP), and other real-time applications are to sit happily on meshes.

Security is a major concern for a wireless mesh network and should be one of the first problems to be solved. Unauthorized users should be prevented from joining the network and prohibited from accessing services provided by the mesh. Data should be secured during transmissions to prevent from being read by either eavesdroppers or intermediate relaying nodes. Although the advanced encryption standard (AES) [17] has been widely adopted for encryption in wireless mesh networks and work is already afoot to incorporate the IEEE 802.11i security standards into wireless mesh products to counter various security threats, it is still a far from trivial task to provide security services efficiently in wireless mesh networks, particularly in LR-WPANs and WSNs where resources such as frequency bandwidth, power supply, memory storage, and computational capacity are highly stringent. And the situation becomes even more severe in an unarchitected ad hoc mesh network that is set up to accommodate different users equipped with wireless products from different vendors and of different functions and capacities. In such a heterogeneous network, other issues such as fairness and cooperation may also arise.

Wireless mesh networks that comprise large number of battery-powered devices would not be expected to go too far without the feature of low power consumption. For instance, for sensing and control devices in LR-WPANs

and WSNs, the replacement of batteries may cost a user more than the devices themselves. It is not only very cumbersome but also practically impossible to replace the batteries in some applications that call for a pervasive deployment of sensors and controllers or demand a deployment in remote or inaccessible locations. Low power consumption is normally achieved by allowing devices to go to sleep for most of the time. This in general requires devices be synchronized so that data transmissions can be scheduled. However, synchronization and scheduling in a multi-hop wireless network is not easy, though not impossible. Low cost is another primary objective for large scale LR-WPANs and WSNs. While low cost generally translates into reduced processing resources and simplified software, the network performance should still be maintained at an acceptable level.

Wireless mesh networks can be deployed by anybody, at any time, at various locations, and with diverse coverage areas. With this versatility, another issue, coexistence, surfaces. It would be a disaster if wireless mesh networks can not coexist with other wireless networks or among themselves. Wireless devices need to share frequency bandwidths, much like cars need to share lanes. If nobody follows rules, one goes nowhere. Another related issue is interoperability. Interoperability problem may result in fragmented market and hinder the fast uptake of a technology. When the market is filled with a jumble of proprietary wireless devices that can not talk to each other whereas they should be able to, users are at a loss for which to choose or

whether to choose at all. All the above urges the establishment of open standards and shows the need for market education.

History has seen the power of open standards and so will the future. Open standards make it possible to manufacture compatible products in large volume, which allows a significant cut in price. Low price boosts the market, which in turn increases the demand for the products and thus further drives the price down. This cycle repeats and eventually forces most proprietary products to drop out of the marketplace. While the importance of open standards is obvious, standards activities also face some challenges. Since challenging a well accepted open standard is practically an up-hill battle, some enterprises having developed proprietary products simply will not applaud the idea of establishing a new standard that is against their benefits. And some others may even try to make a new standard nothing but a wrapper of their proprietary technologies. It is often expensive to modify or improve an open standard due to backward compatibility problems. Therefore, it is essential that a new standard incorporates state-of-the-art technologies.

2.3 An Overview of Wireless Mesh Standards Activities

In this section, we present an overview of wireless mesh standards activities around IEEE and ZigBee, including activities in low rate wireless mesh PANs (IEEE 802.15.5 and ZigBee), wireless mesh LANs (IEEE 802.11s),

and wireless mesh MANs (IEEE 802.16 and 802.20).

2.3.1 Low Rate Wireless Mesh PAN – IEEE 802.15.5 and ZigBee

IEEE 802.15.4 [5] specifies the physical (PHY) layer and medium access control (MAC) sublayer functions of LR-WPANs, and the ZigBee Alliance, a so far 160-member strong industrial alliance, has been working on the specifications for network layer, application layer, and security of LR-WPANs since 2002. The IEEE 802.15.5 task group, kicked off in May 2004, is currently working to provide an architectural framework for interoperable, stable and scalable wireless mesh topologies for both low rate and high rate WPAN devices.

Both IEEE 802.15.5 and ZigBee are working on wireless mesh PANs. However, while IEEE 802.15.5 is still in the initial state of call for proposals, ZigBee ratified its 1.0 specifications on December 14, 2004. More details of IEEE 802.15.5 and ZigBee will be given in chapter 3.

2.3.2 Wireless Mesh LAN – IEEE 802.11s

A WLAN or a Wi-Fi hotspot is usually a single Wi-Fi coverage area – a single house, a store, an office, a building, a campus or a park. Typically hotspots cover areas no larger than a football field. Although their large numbers are certainly a measure of popularity and growth (the market research firm In-Stat/MDR projects that by 2007 there will be over 40,000

hotspots nationwide), there are some significant limitations to this technology, including a line of sight (LOS) requirement, the need for a connection to a wired network, difficulties in roaming, and limited range. Although the IEEE 802.11-1999 (2003 edition) standard provides a four-address frame format for exchanging data packets between access points (APs) for the purpose of creating a Wireless Distribution System (WDS), it does not define how to configure or use a WDS. It was not designed with mesh as a primary use. It suffers from unfairness problems when both single-hop and multi-hop communications exist. That is, one-hop communications capture the channel, leaving no chance to multi-hop communications. As a result, IEEE 802.11 has been mainly used as a last-hop technology so far. The need to pull cables to create connectivity has negated its low cost promise, offset its strength when competing with other technologies, and limited its applications. To address above issues, a new task group was formed in May 2004 under IEEE 802.11, named "IEEE 802.11s Extended Service Set (ESS) mesh."

According to the Project Authorization Request (PAR) [6], 802.11s ESS mesh, as an extension to the IEEE 802.11 MAC, is a collection of APs interconnected with wireless links that enable automatic topology learning and dynamic path configuration. It will create an 802.11 WDS that supports both broadcast/multicast and unicast delivery at the MAC layer using radio-aware metrics over self-configuring multi-hop topologies. An ESS Mesh is

functionally equivalent to a wired ESS, with respect to the stations (STAs) relationship with the Basic Service Set (BSS) and ESS.

A target configuration of 802.11s is up to 32 devices participating as AP forwarders in the ESS Mesh. This is based on a proposal from the U.S. Navy Research Laboratory (US-NRL). The US-NRL proposal defined an algorithm that runs a mesh of 32 APs from multiple vendors. The proposal would extend the 802.11f work on communications between two APs to include communications among multiple APs by adding some over-the-air messages to the 802.11 standard at so-called 2.5 layer. Large configurations with more than 32 APs may also be contemplated by 802.11s.

802.11s ESS mesh shall seamlessly co-exist with existing WLAN technologies. It shall also be backward compatible with other LAN standards and networking protocols so that the existing applications are well supported. It shall operate without any change on the current 802.11 physical (PHY) layer that operates in the unlicensed 2.4 and 5 GHz frequency bands. It shall utilize IEEE 802.11i security mechanisms, or an extension thereof, for the purpose of securing an ESS Mesh in which all of the APs are controlled by a single logical administrative entity for security. It shall allow the use of one or more IEEE 802.11 radios on each AP in the ESS Mesh.

802.11s ESS provides a wide range of applications [7]. The usage categories include:

- Residential

- Office
- Campus/Community/Public access
- Public safety
- Military

At the time of this writing, a dozen of proposals have been submitted to 802.11s. However, two camps have gotten the most attention in pushing their proposals. One is the Wi-Mesh Alliance led by Nortel Networks, Philips Electronics, and InterDigital Communications. The other is the Simple Efficient Extensible Mesh (SEEMesh), whose members include Intel, Motorola, Nokia, NTT DoCoMo, Texas Instruments, and Firetide. Wi-Mesh and SEEMesh are geared towards different-sized networks. Wi-Mesh is looking at the whole range of deployments, while SEEMesh may have had its focus on the smaller consumer-electronic and small office home office (SOHO) area. The further process shall include letter ballot expected to be held in June 2006, sponsor ballot in May 2007, and finally become a part of working group standard by March 2008.

2.3.3 Wireless Mesh MAN – IEEE 802.16 and 802.20

2.3.3.1 IEEE 802.16

In the future, Mesh networks may extend outdoors. Two new standard efforts, 802.16 and 802.20, are concentrating on broadband wireless network-

ing. Initially created to address point-to-multipoint (PMP) communications in the 28-32 GHz range, the 802.16 group is now working on systems operating in frequency bands as low as 2 GHz. The first standard, covering licensed frequencies from 10 to 66 GHz, was completed in October 2001 and published in April 2002. Operation in the tens of GHz range requires LOS communications, which is the biggest disadvantage of the basic standard. The main advantage is that single-carrier modulation can be used. The basic standard supports for PMP communication with high data rate up to 75 Mbps to individual subscriber stations (SSs) within the range of 5 km.

Several extensions have been considered by the commercial alliance group called WiMAX (Worldwide Interoperability for Microwave Access) for addressing fixed and portable/mobile broadband wireless access (BWA) in metropolitan areas.

- 802.16a, published in April 2003, aims to support mesh deployment and non-LOS transmissions up to 50 km range in both licensed and license free frequency band of 2-11 GHz;
- 802.16b operates in the 5-6 GHz frequency band and provides QoS;
- 802.16d and 802.16e are for adding mobility. Mobility is addressed with multicarrier transmission standards, orthogonal frequency division multiplexing (OFDM), that are capable of reliable communications with moving users;
- 802.16f is for improving multi-hop functionality;

- 802.16g is intended to deal with efficient handover and improve QoS.

WiMAX most appeals to the following applications: as lower-cost alternatives to Digital Subscriber Line (DSL) or cable modem access, and as an urban wireless access network providing BWA. The latter is usually intended to work in conjunction with Wi-Fi hotspots and with third-generation (3G) cellular systems.

2.3.3.2 IEEE 802.20

IEEE 802.20, sometimes called Mobile-Fi, also specifies the mobile air interface for wireless broadband. But it uses licensed frequency bands up to 3.5 GHz and seeks to boost real-time data transmission rates in wireless MANs, with as low a latency as 10 milliseconds. It can deliver data to mobile users traveling at a speed up to 250 kilometers per hour and thus is suitable for deployment in high-speed vehicles. It supports for asymmetric link, with 16 and 32 Mbps for downlink and uplink respectively. As a direct competitor to 3G wireless cellular technologies, it also offers global mobility, hand-off, and roaming support.

2.4 Conclusions

Many forces are drawing researchers as well as manufacturers to wireless technologies. The explosive growth of wireless communications has

driven the cost of radio devices down and the quality up. With the ability to reduce installation costs, add flexibility, and ease deployment and maintenance hassles, the attractiveness of wireless technologies needs little reinforcement. Recently there has been a whole slew of research dedicated to wireless mesh networking. Behind this are some unique features of wireless mesh networks such as cost-effectiveness, flexible coverage, fault tolerance, load balancing, self-configuring, self-healing, and minimal up-front investment. Although the technology is still in its early stage, a consensus that wireless mesh technology will enter the mainstream and play a pivotal role on the next wireless frontier may be on the cards.

However, to make wireless mesh networks be all they can be, much research remains to be done. Among the issues that merit special attention are capacity and range enhancement, scalable multi-hop routing, privacy and security, power efficiency, self-configuration, zero or minimal maintenance, coexistence, and standards establishment. Encouragingly, large companies and industry alliances are now actively involved in the research on wireless mesh networks, and several IEEE standards task groups have also been established to work on new standards for wireless mesh networks. With all those efforts and other advances in wireless technologies, we are ushering in a new era – the wireless mesh era.

Chapter 3

Wireless Mesh Personal Area Networks

Chapter 3

Wireless Mesh Personal Area Networks

The new IEEE standard, 802.15.4, defines the physical (PHY) layer and medium access control (MAC) sublayer specifications for low rate wireless personal area networks (LR-WPANs). It shows promise to bring ubiquitous networking into our lives, at least technically. Unlike other standards targeting high or moderate data rate applications, IEEE 802.15.4 is a global standard designed for low data rate, low power consumption, and low cost applications. This so called enabling standard will bring many simple, originally standalone devices into networks, and thus not only opens the door to enormous number of new applications, but also adds value to many other existing applications. The ZigBee Alliance, a 160-member strong industrial alliance, has been working on the Network and upper layers of LR-WPANs. And the IEEE 802.15.5 task group, kicked off in May 2004, is currently working to provide an architectural framework for interoperable, stable and scalable wireless mesh topologies for both low rate and high rate WPAN devices. In this chapter, we first present a few application scenarios to show the potential extent to which wireless mesh PANs[a] can affect our lives. Then we give an overview of the IEEE 802.15.4 standard, focusing on its feasibility and functions in establishing ubiquitous networks. We also summarize the ongoing mesh standards activities of IEEE 802.15.5 and ZigBee.

[a]Since high rate wireless personal area network (HR-WPAN) is more often referred to as ultra wideband (UWB), low rate wireless personal area network (LR-WPAN) is simply referred to as wireless personal area network (WPAN), or wireless mesh personal area network (WMPAN) when mesh capability is emphasized. So the terms LR-WPAN, WPAN, and WMPAN are used interchangeably in this dissertation.

3.1 Why Another Standard?

The past several years have seen the rapid growth of wireless networking. As more functions are incorporated into networks, the cost of installation and maintenance of wired networks continues to escalate. Cable installation is often a costly and time-consuming activity. Such installation often requires installers to pull cables above the ceiling and drop cables through walls to network outlets that they must install into the walls. Network maintenance is not a trivial task either. It may take quite some time to locate and replace a damaged cable, and the downtime caused by this could cost far more than the cable itself. Furthermore, recabling the network because of building reorganization, office partition, or other renovation has always been a frustrating thing. Compared with wired networks, wireless networks provide advantages in deployment, cost, size, and distributed intelligence. Wireless technology not only enables users to set up a network quickly, but also enables them to set up a network where it is inconvenient or impossible to wire cables. The "care free" feature and convenience of (re-)deployment make a wireless network more cost-efficient than a wired network in general. The portability and mobility provided by wireless networks also offer users better connectivities, which could significantly change their daily life behaviors. Two main forces have driven the fast development of wireless networking. One is the desire to eliminate the wires for applications in

802.2 LLC			
802.11 MAC			
802.11 2.4 GHz FH, DS 1, 2 Mbps	**802.11b** 2.4 GHz HR/DS 5.5, 11 Mbps	**802.11a** 5 GHz OFDM 54 Mbps	**802.11g** 2.4 GHz OFDM 54 Mbps

Figure 3.1: IEEE 802.11 Family

places due to various reasons such as convenience, low cost, or visual effects. Another is to set up a network where it is impossible or difficult for a wired network such as disaster salvage, law enforcement, and battle field communications.

Wireless local area network (WLAN) technology has been among the rapidest developing wireless technologies, and it is reshaping the local area networking landscape. Most WLANs are based on a set of IEEE specifications, 802.11 and its extensions 802.11b/a/g (Figure 3.1). The initial 802.11 [20] operates in the 2.4 GHz industrial scientific medical (ISM) frequency band. It employs the frequency-hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS), and supports a mandatory data rate of 1 Mbps and an optional data rate of 2 Mbps. 802.11b (a.k.a. Wi-Fi) defines the high speed extension for the DSSS, also known as high rate direct sequence spread spectrum (HR/DSSS). It supports two additional higher data rates, 11 Mbps and 5.5 Mbps. To date, the most widely available and implemented wireless LANs are based on 802.11b, which is often referred to as Wi-Fi, its synonymous trademarked name. To support band-

width hungry applications, IEEE has proposed another two specifications, 802.11a and 802.11g. Both 802.11a and 802.11g support as high a data rate as 54 Mbps. However, 802.11a radios transmit at 5 GHz and use OFDM (Orthogonal Frequency Division Multiplexing) instead of DSSS. The higher operating frequency equates to relatively shorter range, which means, when compared to 802.11b, more access points may be needed to cover a facility. The different radio frequency and modulation types of 802.11a and 802.11b also cause interoperability problems between these two specifications. On the other hand, 802.11g tries to be backwards-compatible with 802.11b while raising the data rate. 802.11g adopts 802.11a's OFDM to achieve the 54Mbps data rate, but in the 2.4 GHz band instead. For 802.11b compatibility, 802.11g incorporates 802.11b's Complementary Code Keying (CCK) to achieve data rates of 5.5 and 11Mbps in the 2.4GHz band.

With WLANs deployed in offices, homes, as well as other public sites such as hotels, airports, train stations, supermarkets, and restaurants, you can roam around while maintaining a wireless connection to your organization's network and the Internet. Such unencumbered connectivity provides seamless access to the same content no matter when or where you need it.

Bluetooth [21] (IEEE 802.15.1) is the first well known standard facing low data rate applications (Figure 3.2). However, it is still struggling for success in marketing. One of reasons is the boom of Wi-Fi. The sharp drop in price of Wi-Fi has circumvented the envisioned price advantage in

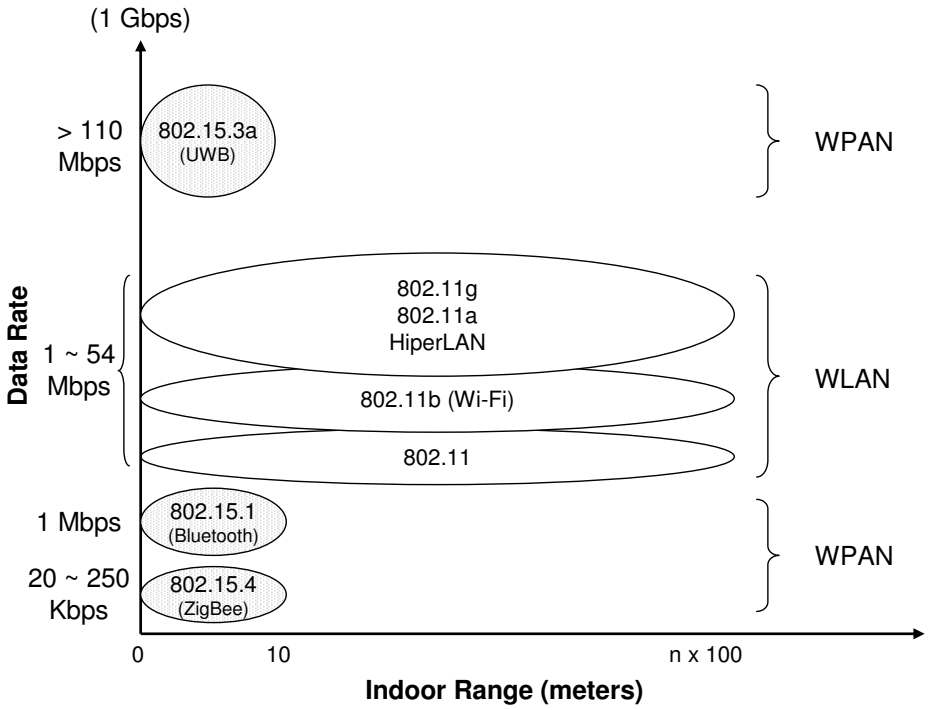


Figure 3.2: Wireless Networking: WLANs and WPANs

Bluetooth in some applications. Another factor is that the effort of Bluetooth to cover more applications and provide quality of service (QoS) has led to its deviation from the design goal of simplicity. The complexity of Bluetooth makes it expensive and unappealing for some simple applications requiring low cost and low power consumption.

As more and more low-cost high-quality devices appear in the market and new applications emerge everyday, short-range wireless personal area networks (WPANs), both low data rate and high data rate, are on the horizon. Two major efforts of IEEE are underway to boost the development of WPANs (Figure 3.2). One is the specifications of IEEE 802.15.3a, also known as ultra wideband (UWB) for high rate WPANs. The other is

the specifications of IEEE 802.15.4 (referred to as 802.15.4 here) for low rate WPANs (LR-WPANs). In this chapter, we will not discuss high rate WPANs, as high rate has long been the focus of research and applications for well understood reasons. Rather we will concentrate on low rate WPANs, specifically the ones based upon IEEE 802.15.4.

So far, low data rate applications have relatively been left in oblivion. Nonetheless, this is no reason to conclude that those applications are less important. In fact, low data rate applications (see section 3.2) are closer to our daily lives than high data rate applications. If high data rate applications can help us save time, low data rate applications have the potential to save our lives. The main obstacle to low data rate applications has been the lack of feasible techniques, especially those based upon global standards. However, with the release of IEEE 802.15.4 as well as advances in other related fields such as embedded processors, micro-electromechanical systems (MEMS), and radio technologies, low data rate applications are expected to thrive and play an increasingly important role in our lives.

The rest of the chapter is structured as follows. First, we present some application scenarios. Next, we give an overview of 802.15.4. Then, we summarize the ongoing mesh standards activities of IEEE 802.15.5 and ZigBee. Finally, we present our conclusions.

3.2 Application Scenarios

802.15.4 is an enabling standard in the sense that it complements wireless standards such as Wi-Fi and UWB. It distinguishes itself from other wireless standards by various features, like low data rate, low power consumption, low cost, self-organization, and flexible topologies (discussed later). It supports applications for which other standards are inappropriate. It not only opens the door to an enormous number of new applications, but also adds value to many existing applications. With various simple devices able to connect to networks, ubiquitous networking is closer than ever to us.

A host of applications can benefit from the new standard, including:

- *Automation and control:* home, factory, warehouse.
- *Monitoring:* safety, health, environments
- *Situational awareness and precision asset location (PAL):* military actions, firefighter operations, autonomous manifesting and real-time tracking of inventory
- *Entertainment:* learning games, interactive toys.

To elaborate on the potential extent to which 802.15.4 can affect our lives, we present a few application scenarios in the following paragraphs. However, we are not claiming that these applications will eventually come true. It may take some time to prove the technology and develop a viable market.

Neither are we implying that these are the most typical applications. Some of them are selected simply for undeniable good causes.

Life saving: No matter how careful a person you are, it is still likely that someday you forget to turn off the gas stove or go to sleep without locking the door to your home. This oversight may cost you a lot, or even endanger the lives in your home. While you are so confident in your driving skill, you may not know that the brake of your car is going to malfunction or that the tire will blow up before you notice any abnormality. You may also involve in a car accident in which you have no wrong doings except that you do not have enough time to react to the faults of others. Science and technology have greatly improved our quality of life, but also increased the dangers we face. The water we drink everyday and the air we breathe all the time may be contaminated or even poisoned due to some inadvertent or hostile behaviors. There are many other life threatening scenarios such as fires, floods and earthquakes. Many life saving systems proved to be ineffective. Some are too complicated and too expensive for pervasive deployment, some stop working too quickly due to the rapid depletion of their batteries, and some others lack the ability to network, which is one of key features in a life saving system. As a global standard designed for low-data-rate, low-power-consumption and low-cost applications, 802.15.4 has promise as a life saver. With the ability to connect all the simple devices such as sensors and actuators, an 802.15.4 system can monitor various events and automatically take

appropriate actions when needed.

Beyond home automation: To free people from daily trivialities, home automation is among the most attractive applications of 802.15.4. Notwithstanding the fact that the devices in your home come from different manufacturers, they will be able to talk to each other and cooperate with each other. You can configure your home network in a way that the light intensity will be lowered automatically when you turn on the TV, and the TV will mute itself when the phone rings or when you pick up the phone to make a call. You can even do more with 802.15.4. For example, everybody in your home can have a private electronic profile, which could be a tiny 802.15.4 compliant device, and the other devices will react accordingly by detecting this profile. Now the lights, the temperature, the music, the TV programs and the websites will all be automatically set according to your personal taste, provided there are no other profiles nearby or your profile has the highest priority. By keeping the private profile in your pocket, you can further take the automation outdoors or into your office, as shown in the following scenarios. Your car will open the door for you as you approach it, and automatically adjust the seat and tune the radio into the program you like. You may not deem this a great achievement, but you will really be grateful to the new technologies if your hands are full of baggage, or even worse, it is raining at that moment. Now let us switch the scenario from home to office. The attendance system will automatically record the time

when you arrive. You will be able to go through the hall door and enter your office without fumbling for keys. The lights as well as the air conditioner will be turned on. The computer will log you into your account. All these are done without your intervention. When it is the time to leave for home, you do not bother to turn off all those devices or lock the doors. They know it is also the time for them to take a break, just like you. You do not need to perform a complicated configuration for your profile to achieve the above, since your profile can be automatically built through some self-learning procedures. However, one critical point in this type of applications is the security. You have to secure the data in your profile, find a way to prevent it from being misused or otherwise compromised, and have a recovery mechanism in case it is lost.

Moving around with ease and safety: With the large number of signposts or other simple devices distributed along the streets, highways and other places, you no longer worry about getting lost. The devices installed in your car will tell you where you are and which way to go. You may get similar services from a global positioning system (GPS), but the new distributed system will be able to give you more accurate and more specific information, and you can continue to use it even if you are within a building or a tunnel where GPS can not cover. In fact, you can get much more useful information from the new system, such as what is the speed limit, whether the street ahead is one way or two way, and even the traffic status or accident

information of each street ahead. You get all the information without looking around as you usually do. Some accidents are caused by the poor sight or lack of line of sight (LOS). For example, you may not notice that there is a pedestrian rushing across the street because your sight is blocked. But now you have a better sight which does not depend on the LOS or your eyesight. With all the information, you can even put your car into an auto-driving state, which not only gives you a break but also is safer than manual driving in some cases such as when you are tired. Your car will slow down when approaching a crossroad and stop if the traffic light is red. By detecting the cars and pedestrians around, it also knows how to cooperate with others, much likely better than you. And your car will definitely refuse to turn into the wrong direction of a one way or run at a speed exceeding the limit. The networking ability of the system will also enable your family members to track your position, provided you give such permission. By informing your accurate position, you can quickly get help from outside when you are in an emergency or involved in an accident. Unlike using an emergency call, you do not need to talk and describe where you are, as you may lack the strength to talk or you do not know where you are. Instead, what you need to do maybe is just to press an emergency button and all the necessary information will be sent out. Using such a system, it is also possible to track the public transportation. If you ever waited for tens of minutes at a bus station or train station in the chilly wind or under the hot sunlight, you can

appreciate being able to catch the next bus or train just in time. There are many other functions can be exploited from such a system, like adjusting the traffic lights dynamically according to the different traffic loads in different streets and tracking speeding cars or stolen cars. Some key issues for implementing such a system are cost, power consumption and security. The cost of devices and their installation and maintenance is an important factor that determines the feasibility of pervasive deployment. Although some of the devices could be mains powered, some others will be battery powered either due to the requirement of portability or for easy deployment. Short battery lifetime is obviously a trouble for such a large system. Another key issue is the security. For example, a mischievous person could insert fake signposts to mislead drivers. 802.15.4 includes various mechanisms to tackle these issues (see section 3.3).

3.3 An Overview of IEEE 802.15.4

The new IEEE standard, 802.15.4[1], defines the physical (PHY) layer and medium access control (MAC) sublayer specifications (Figure 3.3) for low rate wireless personal area networks (LR-WPANs). Those LR-WPANs support simple devices that consume minimal power and typically operate in the Personal Operating Space (POS) of 10 meters or less. Two types of topologies are supported in 802.15.4: (1) a one-hop star; (2) a multi-hop peer-

[1] All results in this chapter apply to the IEEE 802.15.4 draft D18 [5]

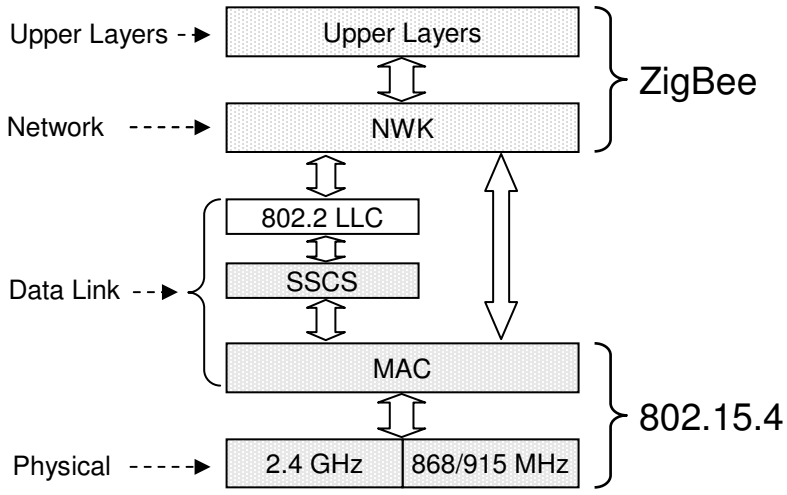


Figure 3.3: LR-WPAN Architecture

to-peer topology when lines of communication exceed 10 meters. However, the logical structure of the peer-to-peer topology is defined by network layer. Currently the ZigBee Alliance[2] is working on the network and upper layers. A device in an LR-WPAN can use either a 64-bit IEEE address or a 16-bit short address assigned during the association procedure, and a single 802.15.4 network can accommodate up to 64k (2^{16}) devices.

In the following subsections, we give a brief description of some important design features in 802.15.4. Complete specifications can be found in [5].

[2]Whose name is derived from the communication method used by honeybees for communicating. Bees dance zigzag to share the information of the position, distance and direction of the food they just found.

3.3.1 Operating in the ISM Bands and at Various Data Rates

802.15.4 defines two PHY layers, the 2.4 GHz band PHY and the 868/915 MHz band PHY. The license free industrial scientific medical (ISM) 2.4 GHz band is available worldwide, while the ISM 868 MHz and 915 MHz bands are available in Europe and North America respectively. Total 27 channels with three different over air data rates are allocated in 802.15.4, i.e., 16 channels with a data rate of 250 Kbps (or expressed in symbols, 62.5 Ksym/sec) in the 2.4 GHz band, 10 channels with a data rate of 40 Kbps (40 Ksym/sec) in the 915 MHz band, and 1 channel with a data rate of 20 Kbps (20 Ksym/sec) in the 868 MHz band. The global availability of ISM bands not only precludes the need of frequency license for any 802.15.4 compliant devices, but also provides for industries a unique chance to develop standardized products that will work anywhere in the world. This reduces the risk that investors face and allows for a decent cost cut in their products compared with proprietary solutions, which more often than not result in fragmented market and high cost. While maintaining its simplicity, 802.15.4 also strives to provide flexibility in its design. An 802.15.4 network can choose to work in one of the 27 channels depending on the availability, congestion state as well as data rate of each channel. Different data rates offer better choices for different applications, in terms of energy and cost efficiency. For example, while 250 Kbps is required for some computer peripherals and interactive toys, a lower data rate like 20 Kbps can meet the

requirements of many other envisioned applications such as various sensors, smart tags, and consumer electronics. 802.15.4 even includes a guaranteed time slot (GTS) mechanism in the attempt to support time-constraint applications. Nevertheless, GTS is a costly approach for low data applications as shown by our experimental results and its feasibility yet remains to see.

3.3.2 Supporting Simple Devices

Featured low data rate, low power consumption, and short transmission range, 802.15.4 is in a good position for supporting simple devices. There are 14 PHY primitives and 35 MAC primitives defined in 802.15.4. This total number of primitives is only about one third the number of primitives defined in Bluetooth, which makes 802.15.4 very suitable for simple devices with limited memory and computational capacity. Two different types of devices are defined in an 802.15.4 network, a full function device (FFD) and a reduced function device (RFD). An FFD is required to support all the 49 primitives while an RFD is only required to support 38 out of the 49 primitives under its minimal configuration. An FFD can talk to RFDs and other FFDs, and operate in three modes serving either as a PAN coordinator, a coordinator or a device. An RFD can only talk to an FFD and is intended for extremely simple applications.

3.3.3 Different Data Transmission Methods and Low Power Consumption

In 802.15.4, data transfer can happen in three different ways: (1) from a device to a coordinator; (2) from a coordinator to a device, or (3) from one peer to another in a peer-to-peer multi-hop network. Nevertheless, for describing its low power consumption feature, we classify the data transfer into the following three types:

- *Direct data transmission:* This applies to all data transfers, either from a device to a coordinator, from a coordinator to a device, or between two peers. Unslotted carrier sense multiple access with collision avoidance (CSMA-CA) or slotted CSMA-CA is used for data transmission, depending whether non-beacon enabled mode or beacon enabled mode is used (see subsection 3.3.6).
- *Indirect data transmission:* This is only applicable to data transfer from a coordinator to its devices. In this mode, a data frame is kept in a transaction list by the coordinator, waiting for extraction by the corresponding device. A device can find out if it has a packet pending in the transaction list by checking the beacon frames received from its coordinator. Occasionally, indirect data transmission can also happen in non-beacon enabled mode such as during an association procedure. Unslotted CSMA-CA or slotted CSMA-CA is used in the data extrac-

tion procedure.

- *GTS data transmission:* This only applies to data transfer between a device and its coordinator, either from the device to the coordinator or from the coordinator to the device. No CSMA-CA is needed in GTS data transmission.

802.15.4 would not be expected to go too far without its most acclaimed feature, low power consumption. For those battery powered simple devices, the replacement of batteries could cost a user more than the devices themselves. It is not only very cumbersome but also practically impossible to replace the batteries in some applications such as in a pressure sensor embedded into the tire of an automobile or in a densely deployed large scale sensor network. Low power consumption is one of most important research topics in wireless networks and much research has been done in this area [23, 25, 26, 38, 43]. Several mechanisms are incorporated in 802.15.4 to prolong the device battery lifetime.

Most power-saving mechanisms in 802.15.4 are based on beacon enabled mode. In direct data transmission, if the *BatteryLifeExtension* option is set to TRUE, the receiver of the beaconing coordinator is disabled after *macBattLifeExtPeriods* (default value 6) backoff periods following the inter-frame space (IFS) period of the beacon frame. Using default configuration, this means that the transceiver of a coordinator or a device is required to be turned on for only about 1/64 of the duration of a superframe, if no data to

be exchanged. In indirect data transmission, a device can enter a low power state, like sleeping state, if it finds there are no pending packets by checking the beacon received from its coordinator. GTS data transmission also has a low duty cycle, nonetheless it is too expensive for low data rate applications.

Some other mechanisms that help further reduce the power consumption are small CSMA-CA backoff period and short transceiver warmup time. In low power applications, reception power is normally larger than transmission power, since more circuits are involved in the signal processing in receptions than in transmissions [23]. This suggests that a short reception time is preferred. The default backoff period used in 802.15.4 is 160 symbols (equivalent to the time needed to transmit 80 bytes in the 2.4 GHz band or 20 bytes in the 868/915 MHz bands) for the first backoff and is doubled each time a backoff is retried, up to 640 symbols. In consideration of low data rate and low power consumption, 802.15.4 also drops the request-to-send (RTS) and clear-to-send (CTS) mechanism in its CSMA-CA. Low duty cycle applications can suffer significant energy loss in their transceiver start-up procedures [38]. If a device needs to go to sleep frequently and the warmup time of its transceiver is not short enough, then the reduction of duty cycle can not guarantee overall low power consumption. Settling time in the channel filter is one of the dominant factors that affect the warmup time of a transceiver. Wideband techniques, such as 16-ary quasi-orthogonal modulation and direct sequence spread spectrum (DSSS) used in 802.15.4,

have short settling times in their wide channel filters.

3.3.4 Reliable and Unreliable Data Delivery

Various mechanisms are included in 802.15.4 to enhance the reliability of data delivery, such as CSMA-CA channel access, frame acknowledgment and retransmission, modulation and spreading techniques, as well as data verification by using a 16-bit cyclic redundancy check (CRC) code. In 802.15.4, both unslotted CSMA-CA and slotted CSMA-CA are used for channel access, with the former used in non-beacon enabled mode and the latter used in beacon enabled mode. When needed, a sender can choose to request the receiver to acknowledge its reception of a certain frame and will retransmit the frame if such an acknowledgment not received within a reasonable period. For better interference resistance and error correction, the 2.4 GHz PHY employs a 16-ary quasi-orthogonal modulation technique, in which each four information bits are mapped into a 32-chip pseudo-random noise (PN) sequence. The PN sequences for successive data symbols are then concatenated and modulated onto the carrier using offset quadrature phase shift keying (O-QPSK). The 868/915 MHz PHY, on the other hand, employs direct sequence spread spectrum (DSSS) with binary phase shift keying (BPSK) used for chip modulation and differential encoding used for data symbol encoding. Each data symbol is mapped into a 15-chip PN sequence and the concatenated PN sequences are then modulated onto the

carrier using BPSK with raised cosine pulse shaping. Moreover, a 16-bit ITU-T CRC code is used for bit error detection.

Unreliable data delivery is also supported in 802.15.4. Besides the broadcast frames that are traditionally not acknowledged, unicast frames can also be transmitted with their acknowledgment requirement flags turned off. Since it has the potential to cut down the traffic overhead, power consumption, and delivery latency, unreliable data delivery may be good enough in some applications where high reliability is not required. For example, missing one or two sampling data once in a while in a humidity sensor or thermometer is acceptable in most cases.

3.3.5 Secure Data Transfer

Secure data transfer is achieved by using proprietary approaches in some products. However, for a global standard like 802.15.4, security is a big concern. For security purpose, the MAC sublayer maintains an access control list (ACL) in its MAC PAN information base (MPIB). By specifying a security suite in the ACL for a communication peer, a device can indicate what level security should be used (e.g., none, access control, data encryption, frame integrity, etc.) for communications with that peer. The building block of 802.15.4 security is the Advanced Encryption Standard (AES) [17]. AES can be used to protect data payload and prevent attackers from impersonating legitimate devices by utilizing an "integrity code" (IC), but it can

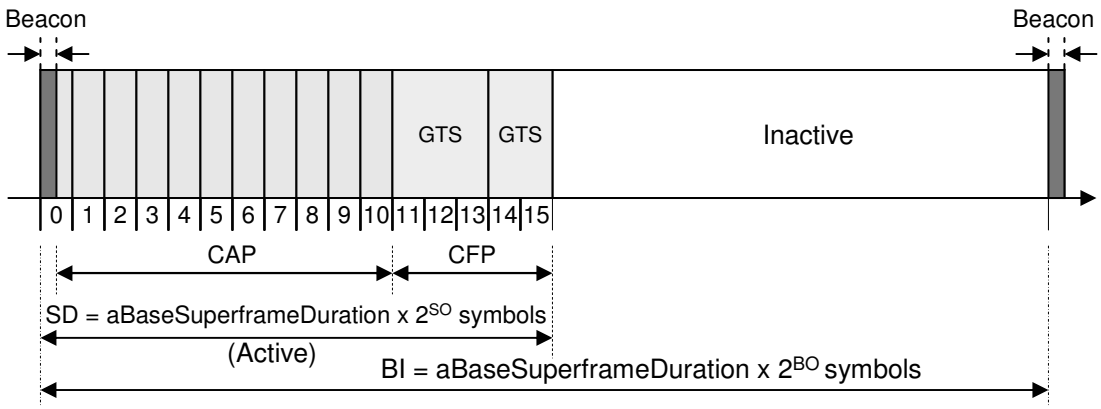


Figure 3.4: An Example of the Superframe Structure

not prevent an attacker from intercepting the symmetric key by eavesdropping at the time when the key is exchanged between communication peers. To prevent this type of attacks, public key cryptography can be used for distributing the AES symmetric key. To minimize the cost for devices, the key distribution method is not included in the 802.15.4 specifications, rather the upper layers are assumed to do the job when needed. We will describe 802.15.4 security services, together with ZigBee security services, in more detail in subsection 3.4.2.5.

3.3.6 Beacon Mode and Superframe Structure

An 802.15.4 network can work in either beacon enabled mode or non-beacon enabled mode. In beacon enabled mode, a coordinator broadcasts beacons periodically to synchronize the attached devices and for other purposes. In non-beacon enabled mode, a coordinator does not broadcast beacons periodically, but may unicast a beacon to a device that is soliciting

beacons.

A superframe structure is used in beacon enabled mode. The format of the superframe is defined by the coordinator. From Figure 3.4, we can see the superframe comprises an active part and an optional inactive part, and is bounded by network beacons. The length of the superframe (a.k.a. beacon interval, BI) and the length of its active part (a.k.a. superframe duration, SD) are determined by the beacon order (BO) and superframe order (SO) respectively. The active part of the superframe is divided into *aNumSuperframeSlots* (default value 16) equally sized slots and the beacon frame is transmitted in the first slot of each superframe. The active part can be further broken down into two periods, a contention access period (CAP) and an optional contention free period (CFP). The optional CFP may accommodate up to seven so-called guaranteed time slots (GTSs), and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP shall remain for contention based access of other networked devices or new devices wishing to join the network. A slotted CSMA-CA mechanism is used for channel access during the CAP. All contention based transactions shall be complete before the CFP begins. Also all transactions using GTSs shall be done before the time of the next GTS or the end of the CFP.

3.3.7 Self-Configuration and Orphaning

To support self-configuration, 802.15.4 embeds association and disassociation functions in its MAC sublayer. This not only enables a star to be setup automatically, but also allows for the creation of a self-configuring, peer-to-peer network. Various configurations are also done during the association procedure, such as selecting a channel and an identifier (ID) for the PAN, determining whether beacon enabled mode or non-beacon enabled mode to be used, choosing the beacon order and superframe order in beacon enabled mode, assigning a 16-bit short address for a device, setting the *BatteryLifeExtension* option and many other options in the MAC layer PAN information base (MPIB).

A device is considered orphaned if it misses *aMaxLostBeacons* (default value 4) beacons from its coordinator in a row. Orphaning mechanism is not used by devices that are in non-beacon enabled mode or devices that are in beacon enabled mode but not tracking beacons. Orphaning offers a way to detect link and/or node failures. When orphaning happens, the device will try to relocate its coordinator through a coordinator realignment procedure.

3.4 Ongoing Mesh Standards Activities of IEEE 802.15.5 and ZigBee

3.4.1 IEEE 802.15.5

According to the Project Authorization Request (PAR) of IEEE 802.15.5 [18], IEEE 802.15.5 tries to "provide a recommended practice to provide the architectural framework enabling WPAN devices to promote interoperable, stable, and scalable wireless mesh topologies and, if needed, to provide the amendment text to the current WPAN standards that is required to implement this recommended practice". The technical requirements of IEEE 802.15.5 [19] include:

- *Topology*: The IEEE 802.15.5 shall support all types of logical topologies, including star, tree, and mesh. The network configuration may be static or dynamic.
- *Power*: The device (complete communication system including alt-PHY and MAC) must operate while supporting a battery life of months or years without intervention. Therefore very efficient power saving modes are desirable, in particular for devices that transmit sporadically. In addition the coordination of nodes must not induce frequent wake up of nodes.
- *Delay*: Maximal end-to-end latency is 250 ms.

- *Message Flow*: Need to support unicast, multicast, and broadcast.
- *Mobility*: This is a mandatory feature related to Intra-PAN mobility, not to roaming or handover. Nodes shall be capable of reliable communication when in the move, at least for tracking. It is admitted that limited communication performance (e.g. data rate) can be tolerated in such cases. The considered applications may involve pedestrian, industrial vehicle and optionally higher speed vehicle mobility.
- *Network Size*: Maximal parameters:
  - Home applications: 500 nodes
  - Commercial applications: 10,000 nodes

  Need scalability support in network density, network size and hop number.
- *Complexity*: Complexity should be minimal to enable mass commercial adoption for a variety of cost sensitive products.
- *Network Management*: Self-organization, self-healing, allowing dynamic routing adaptive to environment; continuum of distributed to centralized management

The IEEE 802.15.5 task group is still in the initial state of call for proposals. In the following, we briefly describe the proposal submitted as a baseline document for wireless mesh personal area networks (WMPANs) by the Joint

Lab of Samsung Advanced Institute of Technology (SAIT) and The City University of New York (CUNY).

The proposal is based on meshed tree approach; it addresses meshed tree routing, multicasting, and key pre-distribution. The meshed tree approach is summarized in this subsection. The complete proposal can be found in [28].

3.4.1.1 Adaptive Robust Tree

The tree defined in the proposal is called adaptive robust tree (ART), for the fact that logical addresses are adaptively assigned during the tree formation procedure to reflect the actual network topology and that the tree is free of single point of failures (SPFs). In ART, each node keeps an ART table (ARTT) to track its branches. Each branch is assigned a block of consecutive addresses. But there is no need to assign consecutive blocks to those branches. This feature allows adding new branches later as well as repairing the tree efficiently. Different priorities are used to determine which branch should be selected for relaying a packet in case that the destination address (or source address) of a packet falls in (or out of) multiple address blocks. The parent of a node is treated as an implicit branch (i.e., not recorded in the ARTT of the node), which has the lowest priority.

Functionally, three phases are defined in ART: initialization (or configuration) phase, operation phase, and recovery phase.

- During initialization phase, nodes join the network and an ART tree is

formed.

- After initialization, the network enters operation phase, in which normal communications start. During operation phase, new nodes are still allowed to join the network, but the number of new nodes is expected to be relatively small compared with the number of nodes already in the network. If there is a substantial change of either the number of nodes or the network topology, the network may need to be re-configured.
- If the tree is broken, then the recovery phase is triggered. Notice that recovery phase is different from the other two phases in that only the affected part of the tree needs to enter the recovery phase (other unaffected parts are still in operation phase). And only failures (either node failures or link failures) happened after initialization will trigger recovery phase. Failures during initialization should be handled by initialization phase itself.

An ART tree is formed during initialization phase. The ART tree formation is functionally divided into two stages: association and address assigning. During association stage, beginning from the root (normally designated manually, for example, by pressing a button), nodes gradually join the network and a tree is formed. But this tree is not an ART tree yet, since no node has been assigned a logic address. There is no limitation on the number of children a node can accept. A node can determine by itself how many nodes (therefore, how many branches) it will accept according to its capa-

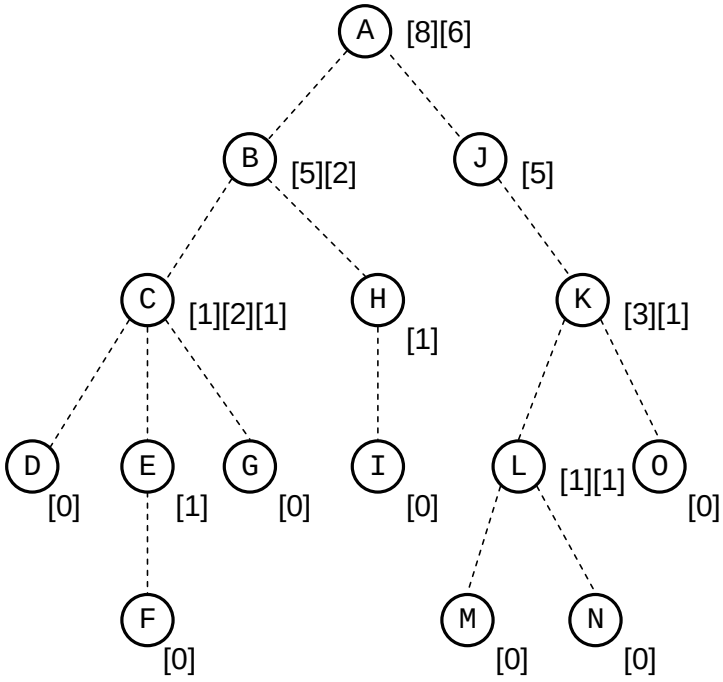


Figure 3.5: Calculation of Number of Nodes along Each Branch

bility and other factors. After the tree reaches its bottom, that is, no more nodes wait for joining the network (a proper timer can be used for this purpose), a down-top procedure is used to calculate the number of nodes along each branch, as shown in Figure 3.5. The numbers in brackets indicate the numbers of nodes within branches below a certain node.

When numbers of nodes are reported from bottom to top, each node can also indicate a desirable number of addresses. For example, node *F* can indicate it wishes to get 3 addresses (2 for possible future extension), though currently only one address is enough (for itself); node *E* can indicate it wishes to get 5 addresses (3 for node *F*, 1 for itself, and 1 for future extension); and so on. After the root, node *A*, receives the information from all

the branches, it can begin to assign addresses, taking into account the actual number of nodes and the wished number of addresses of each branch below it (weighing those numbers into the address assignment is out of the scope of the proposal). The ultimate result of address assignment is that each node has an ARTT built.

During operation phase, a packet can be easily routed using the ARTT. For example, when a packet is received or generated by a node, the node will check if the destination belongs to one of its branches. If not, it will further check if the source belongs to one of its branches. If such a branch is found, the packet will be relayed to that branch; otherwise the packet will be routed to the parent of the node.

During operation phase, link failures or routing node failures will trigger recovery phase. The ARTT is constructed in a way such that tree repair and recovery can be accomplished without changing any assigned address. First, there is no specific relationship between two different branches of a node. The address blocks of different branches can even be overlapped due to tree repair, where a priority value will be applied to determine which branch is actually used in such a case. This means one is free to move a branch from a place to another place without changing any address within the branch. Second, a packet is either routed through one of the branches or through the parent. Since a branch can be routed through a node that is not necessarily within the branch, it can be removed from its original attaching point to a

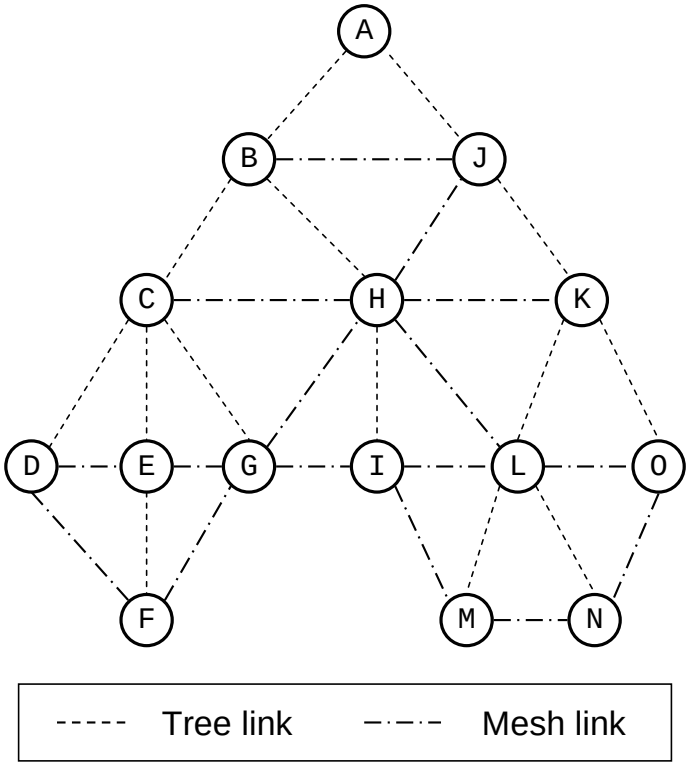


Figure 3.6: Meshed ART

new attaching point without changing any address within the branch. The details of tree repair and recovery can be found in [28].

3.4.1.2 Meshed Adaptive Robust Tree

A meshed ART (MART) can be formed on top of an ART. In Figure 3.6, besides the lines that represent tree links, additional lines (referred to as mesh lines here) are added so that the network now looks more like a mesh than a tree. But from each individual node's point of view, the network is still a tree. Any two nodes connected through a mesh line treats each other as a child and adds an ARTT entry for each other. For example, node *K* treats node *H* as a child, and vise versa. Note that ancestors and descendents,

no matter they are one level or multiple levels away from the node, are not meshed (i.e., not connected to the node through mesh lines).

By forming a MART, it is possible to route a packet through a shorter path. For example, a packet from node *M* to node *I* can be transmitted directly (Figure 3.6), since from node *M*'s point of view, node *I* is its child. On the other hand, the original path in the corresponding ART is *M*-*L*-*K*-*J*-*A*-*B*-*H*-*I*. While this is an extreme example, there is a high probability that a shorter path can be found in a MART than in the corresponding ART.

Another advantage of MART is that some SPFs are removed. For instance, if the link between nodes *J* and *K* is broken, packets from node *K* to node *H* or *I* can still be routed. MART also facilitates tree repair, since existing connections such as *K*–*H* can be immediately used for tree repair purpose.

3.4.2 ZigBee

In this subsection, we brief the standards activities of WMPANs in the ZigBee Alliance.

3.4.2.1 Stack Architecture

The ZigBee stack architecture is depicted in Figure 3.7. The IEEE 802.15.4 [5] standard defines the physical (PHY) layer and the medium access control (MAC) sublayer for LR-WPANs. And the ZigBee Alliance has

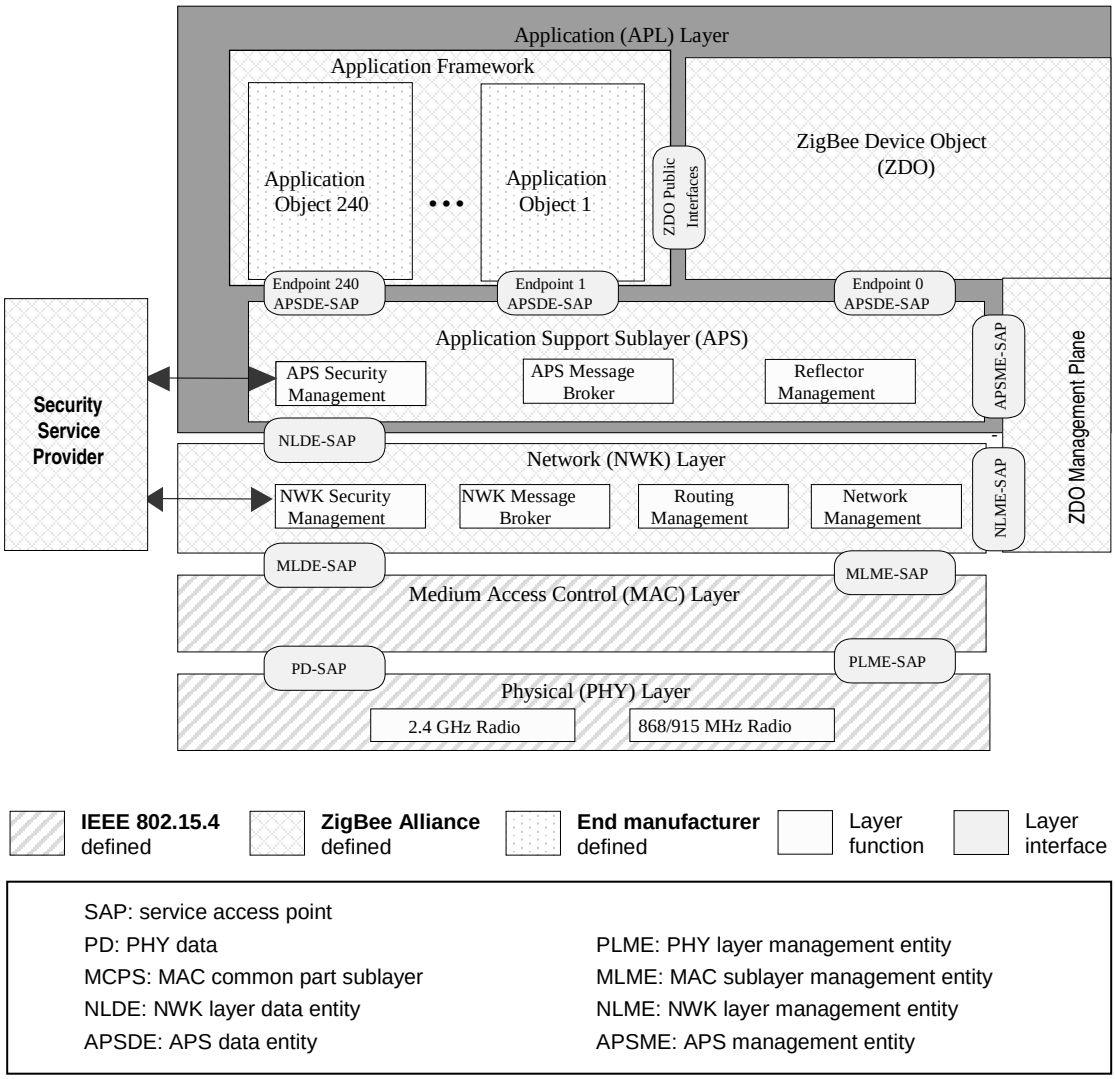


Figure 3.7: ZigBee Stack Architecture

been working on the application (APL) layer, the network (NWK) layer, and the security services; each of them will be discussed briefly in the following subsections.

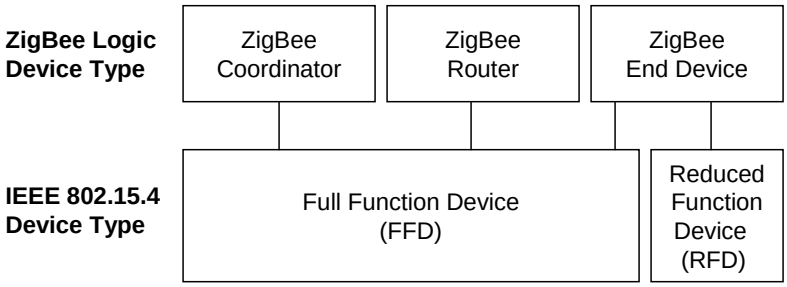


Figure 3.8: Device Type

3.4.2.2 Device Type

There are three logic device types defined in a ZigBee network, namely, ZigBee coordinator, ZigBee router, and ZigBee end device (Figure 3.8). The ZigBee coordinator is the IEEE 802.15.4 [5] PAN coordinator and must be a full function device (FFD). A ZigBee router is an FFD participating in a ZigBee network, which is not the ZigBee coordinator but may act as an IEEE 802.15.4 coordinator within its personal operating space. A ZigBee end device is either an FFD or a reduced function device (RFD) participating in a ZigBee network, which is neither the ZigBee coordinator nor a ZigBee router. The capabilities and functionalities of these three types of devices are different in the network.

3.4.2.3 Application Layer

The application (APL) layer includes the application support (APS) sublayer, the ZigBee device objects (ZDO), and the manufacturer-defined application objects. The APS provides an interface between the NWK layer

and the APL layer through two service access points: the APS data entity (APSDE) service access point (APSDE-SAP) and the APS management entity (APSME) service access point (APSME-SAP), and is responsible for maintaining tables for binding (i.e., matching two devices together based on their services and their needs) and forwarding messages between bound devices. Total 255 endpoints can be defined upon the APS: endpoints 1–240 are used for distinct application objects (e.g., sensors, switches, etc.), endpoint 0 is reserved for the data interface to the ZDO, endpoint 255 is reserved for the data interface function to broadcast data to all application objects, and endpoints 241–254 are reserved for future use. The endpoint number, together with the node (or radio) address, uniquely identifies each endpoint in the network, thus enabling binding two or more endpoints for certain applications (e.g., binding switches with lamps in a lighting application). Inside the application framework, the application objects send and receive data through the APSDE-SAP. And the ZDO is in charge of the control and management of the application objects, including defining the role of the device within the network (e.g., ZigBee coordinator or end device), discovering devices on the network and determining which application services they provide, initiating and/or responding to binding requests and establishing a secure relationship between network devices.

Profiles are developed by ZigBee vendors to address solutions to specific technology needs. A profile is a collection of device descriptions, and

contains one or more so called clusters. Each cluster is further a container for one or more attributes and is identified by a cluster identifier, which is unique within the scope of a particular profile. Profiles are used in service discovery and binding operations. The service discovery process is key to interfacing devices. Service discovery can be done through specific requests for descriptors on specified nodes or broadcast requests for service matching. Binding, the creation of logical links between complementary application devices and endpoints, is accomplished by matching an output cluster identifier to an input cluster identifier, assuming both are within the same profile. The APS of the ZigBee coordinator or a ZigBee router may maintain a binding table, which is needed in indirect addressing (discussed below). The binding table implements the following mapping:

$$(a_s,\, e_s,\, c_s) \;=\; \{(a_{d1},\, e_{d1}),\, (a_{d2},\, e_{d2}),\, \cdots,\, (a_{dn}, e_{dn})\}$$

where,

a_s = the address of the binding source device,

e_s = the endpoint identifier of the binding source device,

c_s = the cluster identifier used in the binding link,

a_{di} = the i^{th} address of the binding destination device,

e_{di} = the i^{th} endpoint identifier of the binding destination device.

There are three addressing modes defined in a ZigBee network: direct

addressing, indirect addressing, and broadcast addressing. Direct addressing defines a means of transmitting messages to a destination by including its full address and endpoint information. To support extremely simple devices (e.g., lamp switches), which may not have enough storage space for a binding table, indirect addressing is defined and employed to send messages to destinations for which they do not have an address. An indirect addressed message is relayed by the ZigBee coordinator or a ZigBee router who has a binding table. The included source address, source endpoint, and cluster identifier in the message are translated via the binding table to those of the destination device(s) and the message is relayed to each indicated destination. Broadcast addressing is also called application broadcast, which allows data to be broadcast by setting the destination address to the 16-bit network broadcast address and the delivery mode sub-field in the packet control field to *Broadcast*.

3.4.2.4 Network Layer

The NWK layer provides the data service and the management service to the APL layer. The data service includes:

- *Network protocol data unit (NPDU) generation*: The ability to generate an NPDU from an APS PDU through the addition of an appropriate protocol header.
- *Routing*: The ability to transmit an NPDU to an appropriate device

that is either the final destination of the communication or the next step towards the final destination in the communication chain.

The management service covers:

- *Configuring a new device*: The ability to configure the stack for operation as required. Configuration options include beginning operation as a ZigBee coordinator or joining an existing network.
- *Starting a network*: The ability to establish a new network.
- *Joining and leaving a network*: The ability to join or leave a network as well as the ability for a ZigBee coordinator or ZigBee router to request that a device leave the network.
- *Addressing*: The ability of ZigBee coordinators and routers to assign logic addresses to devices joining the network.
- *Neighbor discovery*: The ability to discover, record, and report information pertaining to the one-hop neighbors of a device.
- *Route discovery*: The ability to discover and record paths through the network whereby messages may be efficiently routed.
- *Reception control*: The ability for a device to control when the receiver is activated and for how long, enabling MAC sublayer synchronization or direct reception.

In what follows, we describe in a little more detail some of the NWK layer functions, including logic address assignment, routing, and data broadcast.

Through the association primitive supported by IEEE 802.15.4 [5], a logical tree, referred to as cluster-tree [29], can be formed along with the setup of an LR-WPAN. The first node in the network will designate itself as the ZigBee coordinator and begin to accept association requests from other nodes. Any node already in the network can determine whether to allow other nodes to join it, that is, whether to act as a ZigBee router, depending on the availability of its resources such as memory and energy. In the cluster-tree, a node can have a maximum number of C_m children and a node can be at most L_m levels (i.e., hops) away from the root of the tree (C_m and L_m are two network-wide constants determined by the ZigBee coordinator). In the original cluster-tree scheme, a node with a logic address s is in charge of assigning logic addresses to its children according to the following algorithm: assign logic address $s+1$ to the first child, $s+1+C_{skip}(L_s)$ to the second child, and $s+1+(n-1)\times C_{skip}(L_s)$ to the n^{th} child, up to the $(C_m)^{th}$ child. And $C_{skip}(L_s)$ is calculated as follows:

$$C_{skip}(L_s) = \left\lfloor \frac{B - \sum_{k=0}^{L_s}(C_m)^k}{(C_m)^{L_s+1}} \right\rfloor \tag{3.1}$$

where B is the address block size of the whole network and L_s is the level of the node. For a full block, B can be calculated using C_m and L_m as follows:

$$B = \sum_{k=0}^{L_m}(C_m)^k \tag{3.2}$$

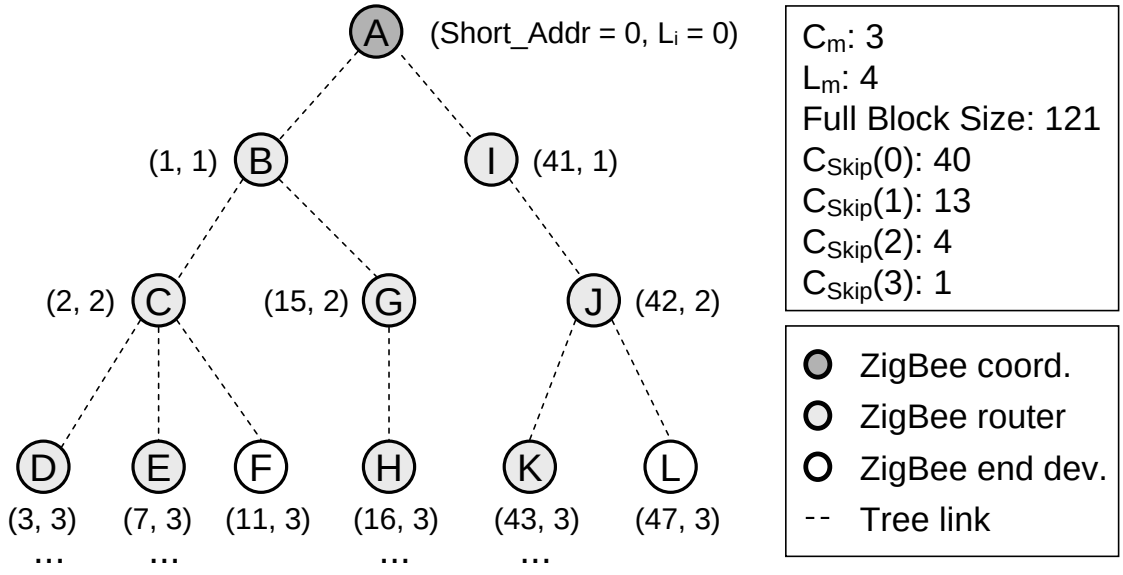


Figure 3.9: An Example of Cluster-tree

or otherwise designated for a non-full block.

Figure 3.9 is an example of cluster-tree with $C_m = 3$ and $L_m = 4$. Node *A* is the ZigBee coordinator with a logic address 0. Since $C_{skip}(0) = 40$, node *A* assigns the logic addresses 1 and 41 to its two children *B* and *I* respectively. Similarly, node *B* assigns the logic addresses 2 and 15 to its two children *C* and *G* respectively, using $C_{skip}(1) = 13$. This procedure continues until the network reaches the maximum L_m levels. Some branches may terminate at a level less than L_m if the nodes at the end of those branches (e.g., node *F* and *L* in Figure 3.9) are ZigBee end devices.

As of ZigBee Network Specification V 1.0 [30], a slightly different version of cluster-tree algorithm is used. Compared with the above cluster-tree routing, the new version distinguishes two types of devices when assigning logic addresses, i.e., routers and end devices. A router is still assigned an

address sub-block, which can be further assigned to its children, but an end device only gets a single address and thus can not have any children. In the new version, denote R_m the maximum number of routers a parent may have as children, Equation (3.1) becomes

$$C_{skip}(L_s) = \begin{cases} 1 + c_m \times (L_m - L_s - 1) & \text{if } R_m = 1 \\ \frac{1 - C_m - R_m - C_m \times R_m^{L_m - L_s - 1}}{1 - R_m} & \text{otherwise} \end{cases} \tag{3.3}$$

for router-capable children. For end devices, logic addresses are assigned in a sequential manner with the n^{th} address, A_n, given by the following equation:

$$A_n = A_{parent} + C_{skip}(L_s) \times R_m + n \tag{3.4}$$

where $1 \leq n \leq (C_m - R_m)$ and A_{parent} represents the address of the parent.

Based on the above logic address assignment, a node can easily determine how to forward a data packet by looking at the destination address, that is, whether forward the data packet to one of its end device children or to one of its router-capable children or to its parent. This routing scheme is referred to as cluster-tree routing.

With the cluster-tree, a device can immediately begin to transmit data packets to other devices once it joins the network, without going through the route discovery procedure. However, as we can see from Figure 3.9, most cluster-tree routes are not optimal in terms of hop count. Cluster-tree routing also results in uneven traffic distribution. That is, a node at a smaller tree level normally needs to handle more traffic than a node at a larger tree level. As such, a node at a smaller tree level dies more quickly than other nodes due to its quick battery depletion. Without other mechanisms, single point of failure (SPF) and network partition could easily happen. Therefore, in ZigBee networks, the cluster-tree routing is combined with another on-demand table-driven routing, which is currently based on the ad hoc on-demand distance vector junior (AODVjr) [31]. AODVjr is a simplified version of AODV [11]. It is capable of finding optimal or near-optimal routes, and thus helps reduce the message delivery latency. Nevertheless, compared with cluster-tree routing, it requires more memory to store routing entries and also incurs much control overhead. As most routes are formed on demand, the initial latency caused by route discovery is high. In general, AODVjr is suitable for devices with sufficient memories, and favors long communication sessions. The ZigBee routing combines the cluster-tree routing and the AODVjr routing and makes tradeoff between them according to the network conditions and application requirements.

In addition to unicast, broadcast can be used to transmit any network layer data packets from one device to other devices in a ZigBee network. In a non-beacon enabled ZigBee network, a passive acknowledgment mechanism is employed in data broadcast to achieve reliability. For passive acknowledgment mechanism to work, each device keeps a broadcast transaction table (BTT), each entry of which is a distinct broadcast transaction record (BTR), to track if its neighbors have relayed the broadcast packet. It will retransmit a previous broadcast packet if any of its neighbors has not relayed the broadcast packet within a certain time-out period, to a predetermined maximum times. A radius counter (RC) is included in the broadcast packet by the source node, which is decreased by one by each intermediate node relaying the broadcast packet. When the RC reaches zero, the broadcast packet will not be relayed anymore.

In a beacon enabled ZigBee network, devices may go to sleep. There is an attribute in the MAC PAN information base (MPIB), *macRxOnWhenIdle*, which determines whether a device should turn on (if *macRxOnWhenIdle* is TRUE) or off (if *macRxOnWhenIdle* is FALSE) its transceiver when it is idle, i.e., neither in transmission nor in reception. Different from in a non-beacon enabled ZigBee network, a ZigBee router with the *macRxOnWhenIdle* set to FALSE will retransmit a received broadcast data packet to each of its neighbors individually, using a MAC layer unicast. Similarly, a router with the *macRxOnWhenIdle* set to TRUE, which has one or more

Table 3.1: IEEE 802.15.4 Security Suites

Security suite name	Access control	Data encryption	Frame integrity	Sequential freshness (optional)
None				
AES-CTR	x	x		x
AES-CCM-128	x	x	x	x
AES-CCM-64	x	x	x	x
AES-CCM-32	x	x	x	x
AES-CBC-MAC-128	x		x	
AES-CBC-MAC-64	x		x	
AES-CBC-MAC-32	x		x	

neighbors with the *macRxOnWhenIdle* set to FALSE, will retransmit the broadcast data packet to each of these neighbors in turn as a MAC layer unicast in addition to performing the more general broadcast procedure outlined above.

3.4.2.5 Security Services

IEEE 802.15.4 [5] provides link layer security for LR-WPANs, including access control, confidentiality, message integrity, and optional message freshness, as outlined in Table 3.1. Access control is supported by all security suites except *None* and it provides the ability for a device to select the other devices with which it is willing to communicate. For security purposes, each device keeps an ACL in its MAC sublayer PAN Information Base (MPIB). The ACL contains up to 255 entries, one for each destination device. Each ACL entry consists of the destination address (IEEE address

and optional logical address), security suite identifier, and other security materials. By default, security is not enabled in 802.15.4. To enable security, upper layers should specify a security suite other than *None* in the ACL entry corresponding to the destination. However, acknowledgment frame is required to always use security suite *None*, and thus not protected.

The AES-CTR security suite provides confidentiality protection by encrypting the payload of a frame, using the Advanced Encryption Standard (AES) [17] block cipher [32] with counter mode. The AES-CBC-MAC security suite, on the other hand, provides integrity protection, using the Cipher Block Chaining Message Authentication Code (CBC-MAC) [33]. And the AES-CCM security suite provides both confidentiality and integrity protection, using the Counter with CBC-MAC (CCM) [34]. Both AES-CBC-MAC and AES-CCM have three variants (see Table 3.1), depending on the size of the MAC[3] used.

Upon IEEE 802.15.4, the ZigBee Alliance defines the NWK and application layer security services [30], based on CCM*, a minor modification of CCM [34]. Besides all the features of CCM, CCM* additionally offers encryption-only and integrity-only capabilities, thus eliminates the need for CTR and CBC-MAC modes. Also, CCM* allows using a single key for all CCM* security levels (see Table 3.2). This is different from the MAC sublayer security modes, which require different keys for different security

[3]Here MAC is short for Message Authentication Code, not for Medium Access Control.

Table 3.2: Security Levels Available in ZigBee

Secu-level	Secu-attribute	Data encryption	Frame integrity
'000'	None		
'001'	MIC-32		x
'010'	MIC-64		x
'011'	MIC-128		x
'100'	ENC	x	
'101'	ENC-MIC-32	x	x
'110'	ENC-MIC-64	x	x
'111'	ENC-MIC-128	x	x

levels. As a result, different layers in a ZigBee network can reuse some keys. Another design feature of ZigBee is to use the so-called open trust mode, in which different layers of the communication stack and all applications running on a single device trust each other.

ZigBee uses a 128-bit link key (more actually its derivatives, see details below) to secure pairwise communications, probably multiple hops away, and a 128-bit Network key to secure broadcast communications. A device can acquire link keys and a Network key via key-transport or pre-installation. Link keys can also be obtained through key-establishment technique, based on a 'master' key, which itself can be obtained via key-transport or pre-installation. The ultimate security between devices depends on the secure initialization and installation of these keys. The Network key may be shared by the MAC, NWK, and APS, but the link and master keys may be used only by the APS. To avoid security leaks due to unwanted in-

teractions between different security services, ZigBee also uses a one-way function to derive various service-specific keys from the link key, including the key-load key, key-transport key, and data key. The key-load key, key-transport key, and data key are used to protect frames containing transported master keys, frames containing other transported keys, and all other frames that need to be secured respectively.

ZigBee performs centralized security control via a trust center. There is exactly one trust center in each secure network. The trust center is responsible for distributing and maintaining the Network key to devices as well as binding two applications and enabling end-to-end security between devices (e.g., by distributing master keys or link keys). To meet different security needs, the trust center can be configured to work in either commercial mode or residential mode. In commercial mode, the trust center *shall* maintain a list of devices, master keys, link keys, and Network keys that it needs to control and enforce the policies of Network key updates and network admittance. In this mode, the memory required for the trust center grows with the number of devices in the network. In residential mode, the trust center *may* maintain a list of devices, master keys, or link keys with all the devices in the network; however, it *shall* maintain the Network key and controls policies of network admittance.

In both IEEE 802.15.4 and ZigBee, a frame counter, which is a monotonically increasing 4-octet sequence number bound to an encryption key, is

used to prevent replay attacks.

3.5 Conclusions

This chapter describes the features, functions, and feasibility of IEEE 802.15.4 standard with respect to ubiquitous networking. The related ongoing mesh standards activities of IEEE 802.15.5 and ZigBee are also presented.

Many features of WMPANs, such as using globally available ISM frequency bands, power conservation, self-configuration, secure data transfer and so on, make them a technology having the promise to unify all those simple devices from different manufacturers and bring networks to the level of each person.

The number of applications that can benefit from WMPANs is enormous. From home to office, industry to agriculture, civilian activities to military operations, and indoors to outdoors, WMPANs will find their way there. By extending networks to cover all the simple devices and with the emerging of many interesting and wonderful applications, we are stepping closer to a ubiquitous networking epoch.

Part II

Lower Layer Issues – MAC, Scheduling, and Midware

Chapter 4

A Comprehensive Performance Study of IEEE 802.15.4

Chapter 4

A Comprehensive Performance Study of IEEE 802.15.4

IEEE 802.15.4 is a new standard uniquely designed for low rate wireless personal area networks (LR-WPANs). It targets low data rate, low power consumption, and low cost wireless networking, and offers device level wireless connectivity. We develop an NS2 simulator for IEEE 802.15.4 and conduct several sets of experiments to study its various features, including: (1) star topology and peer-to-peer topology; (2) beacon enabled mode and non-beacon enabled mode; (3) association, tree formation and network autoconfiguration; (4) orphaning and coordinator relocation; (5) carrier sense multiple access with collision avoidance (CSMA-CA), both unslotted and slotted; and (6) direct, indirect and guaranteed time slot (GTS) data transmissions. In non-beacon enabled mode and under moderate data rate, the new IEEE 802.15.4 standard, compared with IEEE 802.11, is more efficient in terms of overhead and resource consumption. It also enjoys a low hop delay (normalized by channel capacity) on average. However, its packet delivery ratio suffers as data rate increases. In beacon enabled mode, an LR-WPAN can be flexibly configured to meet different needs, such as link failure detection and self-recovery from disruption by using orphaning and coordinator relocation mechanism, and low duty cycle and energy efficiency by adopting indirect data transmission or turning on the *BatteryLifeExtension* feature in direct data transmission. GTS also has a low duty cycle, nonetheless it is an expensive approach for low data rate applications. In both beacon enabled mode and non-beacon enabled mode, association and tree formation proceed smoothly and the network can shape up efficiently by itself. We also identify some issues that could degrade the network performance if not handled properly.

4.1 Background and Motivation

Compared with wired networks, wireless networks provide advantages in deployment, cost, size, and distributed intelligence. Wireless technology not only enables users to set up a network quickly, but also enables them to set up a network where it is inconvenient or impossible to wire cables. The "care free" feature and convenience of deployment make a wireless network more cost-efficient than a wired network in general.

The release of IEEE 802.15.4 (referred to as 802.15.4 hereinafter), "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs)" [5][1], represents a milestone in wireless personal area networks and wireless sensor networks. 802.15.4 is a new standard uniquely designed for low rate wireless personal area networks. It targets low data rate, low power consumption and low cost wireless networking and offers device level wireless connectivity. A host of new applications can benefit from the new standard, such as those using sensors that control lights or alarms, wall switches that can be moved at will, wireless computer peripherals, controllers for interactive toys, smart tags and badges, tire pressure monitors in cars, inventory tracking devices.

802.15.4 distinguishes itself from other wireless standards such as IEEE 802.11 (referred to as 802.11 hereinafter) [20] and Bluetooth [21] by some

[1] All results in this chapter apply to the IEEE 802.15.4 draft D18 [5]

unique features (see section 4.2). However, there are no simulations or implementations available so far to test these new features. We develop an NS2 simulator for 802.15.4 and carry out several sets of experiments to evaluate its performances, in hopes of helping IEEE to verify and/or improve the design, and facilitating researchers and manufacturers to develop products based upon this new standard. 802.15.4 has been designed as a flexible protocol in which a set of parameters can be configured to meet different requirements. As such, we also try to find out how users can tailor the protocol to their needs and where the trade-off is for some applications.

The rest of the chapter is structured as follows. In section 4.2, we give a brief description of 802.15.4. Next, in section 4.3, we outline the NS2 simulator for 802.15.4. In section 4.4, we define a set of performance metrics and present the experimental setup. Then, in section 4.5, we give out the experimental results with discussions. In section 4.6, we propose some enhancements for 802.15.4. Finally, in section 4.7, we conclude.

4.2 A Brief Description of IEEE 802.15.4

The new IEEE standard, 802.15.4, defines the physical (PHY) layer and medium access control (MAC) sublayer specifications for low data rate wireless connectivity among relatively simple devices that consume minimal power and typically operate in the Personal Operating Space (POS) of

10 meters or less. An 802.15.4 network can simply be a one-hop star, or, when lines of communication exceed 10 meters, a self-configuring, multi-hop network. A device in an 802.15.4 network can use either a 64-bit IEEE address or a 16-bit short address assigned during the association procedure, and a single 802.15.4 network can accommodate up to 64k (2^{16}) devices. Wireless links under 802.15.4 can operate in three license free industrial scientific medical (ISM) frequency bands. These accommodate over air data rates of 250 Kbps (or expressed in symbols, 62.5 Ksym/sec) in the 2.4 GHz band, 40 Kbps (40 Ksym/sec) in the 915 MHz band, and 20 Kbps (20 Ksym/sec) in the 868 MHz. Total 27 channels are allocated in 802.15.4, with 16 channels in the 2.4 GHz band, 10 channels in the 915 MHz band, and 1 channel in the 868 MHz band.

Wireless communications are inherently susceptible to interception and interference. Some security research has been done for WLANs and wireless sensor networks [40–42, 47, 49, 120], but pursuing security in wireless networks remains a challenging task. 802.15.4 employs a fully handshaked protocol for data transfer reliability and embeds the Advanced Encryption Standard (AES) [17] for secure data transfer.

In the following subsections, we give a brief overview of the PHY layer, MAC sublayer and some general functions of 802.15.4. Detailed information can be found in [5].

4.2.1 The PHY Layer

The PHY layer provides an interface between the MAC sublayer and the physical radio channel. It provides two services, accessed through two service access points (SAPs). These are the PHY data service and the PHY management service. The PHY layer is responsible for the following tasks:

- *Activation and deactivation of the radio transceiver:* Turn the radio transceiver into one of the three states, that is, transmitting, receiving, or off (sleeping) according to the request from MAC sublayer. The turnaround time from transmitting to receiving, or vice versa, should be no more than 12 symbol periods.
- *Energy detection (ED) within the current channel:* It is an estimate of the received signal power within the bandwidth of an IEEE 802.15.4 channel. No attempt is made to identify or decode signals on the channel in this procedure. The energy detection time shall be equal to 8 symbol periods. The result from energy detection can be used by a network layer as part of a channel selection algorithm, or for the purpose of clear channel assessment (CCA) (alone or combined with carrier sense).
- *Link quality indication (LQI) for received packets:* Link quality indication measurement is performed for each received packet. The PHY layer uses receiver energy detection (ED), a signal-to-noise ratio, or a

combination of these to measure the strength and/or quality of a link from which a packet is received. However, the use of LQI result by the network or application layers is not specified in the standard.

- *Clear channel assessment (CCA) for carrier sense multiple access with collision avoidance (CSMA-CA):* The PHY layer is required to perform CCA using energy detection, carrier sense, or a combination of these two. In energy detection mode, the medium is considered busy if any energy above a predefined energy threshold is detected. In carrier sense mode, the medium is considered busy if a signal with the modulation and spreading characteristics of IEEE 802.15.4 is detected. And in the combined mode, both conditions aforementioned need to be met in order to conclude that the medium is busy.
- *Channel frequency selection:* Wireless links under 802.15.4 can operate in 27 different channels (but a specific network can choose to support part of the channels). Hence the PHY layer should be able to tune its transceiver into a certain channel upon receiving the request from MAC sublayer.
- *Data transmission and reception:* This is the essential task of the PHY layer. Modulation and spreading techniques are used in this part. The 2.4 GHz PHY employs a 16-ary quasi-orthogonal modulation technique, in which each four information bits are mapped into a 32-chip pseudo-random noise (PN) sequence. The PN sequences for successive

data symbols are then concatenated and modulated onto the carrier using offset quadrature phase shift keying (O-QPSK). The 868/915 MHz PHY employs direct sequence spread spectrum (DSSS) with binary phase shift keying (BPSK) used for chip modulation and differential encoding used for data symbol encoding. Each data symbol is mapped into a 15-chip PN sequence and the concatenated PN sequences are then modulated onto the carrier using BPSK with raised cosine pulse shaping.

4.2.2 The MAC Sublayer

The MAC sublayer provides an interface between the service specific convergence sublayer (SSCS) and the PHY layer. Like the PHY layer, the MAC sublayer also provides two services, namely, the MAC data service and the MAC management service. The MAC sublayer is responsible for the following tasks:

- *Generating network beacons if the device is a coordinator:* A coordinator can determine whether to work in a beacon enabled mode, in which a superframe structure is used. The superframe is bounded by network beacons and divided into *aNumSuperframeSlots* (default value 16) equally sized slots. A coordinator sends out beacons periodically to synchronize the attached devices and for other purposes (see subsection 4.2.3).

- *Synchronizing to the beacons:* A device attached to a coordinator operating in a beacon enabled mode can track the beacons to synchronize with the coordinator. This synchronization is important for data polling, energy saving, and detection of orphanings.
- *Supporting personal area network (PAN) association and disassociation:* To support self-configuration, 802.15.4 embeds association and disassociation functions in its MAC sublayer. This not only enables a star to be setup automatically, but also allows for the creation of a self-configuring, peer-to-peer network.
- *Employing the carrier sense multiple access with collision avoidance (CSMA-CA) mechanism for channel access:* Like most other protocols designed for wireless networks, 802.15.4 uses CSMA-CA mechanism for channel access. However, the new standard does not include the request-to-send (RTS) and clear-to-send (CTS) mechanism, in consideration of the low data rate used in LR-WPANs.
- *Handling and maintaining the guaranteed time slot (GTS) mechanism:* When working in a beacon enabled mode, a coordinator can allocate portions of the active superframe to a device. These portions are called GTSs, and comprise the contention free period (CFP) of the superframe.
- *Providing a reliable link between two peer MAC entities:* The MAC sublayer employs various mechanisms to enhance the reliability of the

link between two peers, among them are the frame acknowledgment and retransmission, data verification by using a 16-bit CRC, as well as CSMA-CA.

4.2.3 General Functions

The standard gives detailed specifications of the following items: type of device, frame structure, superframe structure, data transfer model, robustness, power consumption considerations, and security. In this subsection, we give a short description of those items closely related to our performance study, including type of device, superframe structure, data transfer model, and power consumption considerations.

Two different types of devices are defined in an 802.15.4 network, a full function device (FFD) and a reduced function device (RFD). An FFD can talk to RFDs and other FFDs, and operate in three modes serving either as a PAN coordinator, a coordinator or a device. An RFD can only talk to an FFD and is intended for extremely simple applications.

The standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. From Figure 4.1, we can see the superframe comprises an active part and an optional inactive part, and is bounded by network beacons. The length of the superframe (a.k.a. beacon interval, BI) and the length of its active part (a.k.a. superframe duration, SD) are defined as follows:

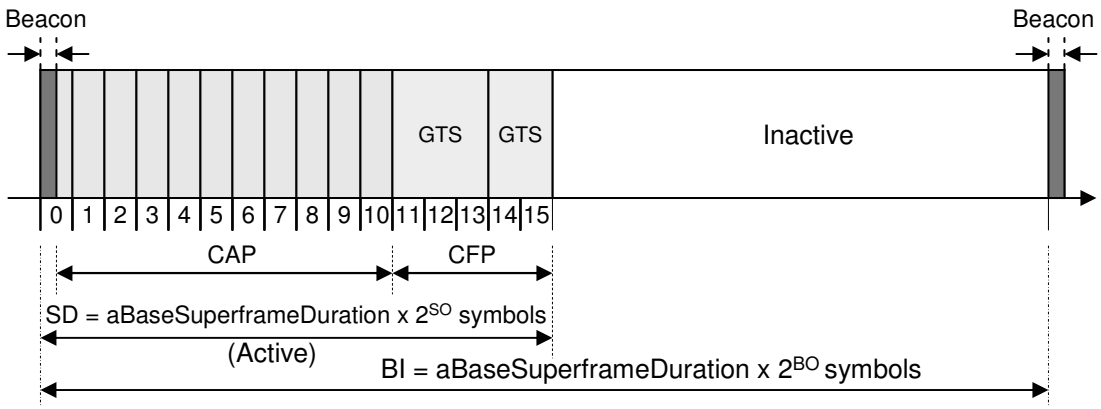


Figure 4.1: An Example of the Superframe Structure

$$BI = aBaseSuperframeDuration * 2^{BO}$$

$$SD = aBaseSuperframeDuration * 2^{SO}$$

Where,

aBaseSuperframeDuration = 960 symbols

BO = beacon order

SO = superframe order

The values of BO and SO are determined by the coordinator. The active part of the superframe is divided into *aNumSuperframeSlots* (default value 16) equally sized slots and the beacon frame is transmitted in the first slot of each superframe. The active part can be further broken down into two periods, a contention access period (CAP) and an optional contention free period (CFP). The optional CFP may accommodate up to seven so-called guaranteed time slots (GTSs), and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP shall remain for contention based access of other networked devices or new devices wishing to join the

network. A slotted CSMA-CA mechanism is used for channel access during the CAP. All contention based transactions shall be complete before the CFP begins. Also all transactions using GTSs shall be done before the time of the next GTS or the end of the CFP.

Data transfer can happen in three different ways: (1) from a device to a coordinator; (2) from a coordinator to a device; and (3) from one peer to another in a peer-to-peer multi-hop network. Nevertheless, for our performance study, we classify the data transfer into the following three types:

- *Direct data transmission:* This applies to all data transfers, either from a device to a coordinator, from a coordinator to a device, or between two peers. unslotted CSMA-CA or slotted CSMA-CA is used for data transmission, depending whether non-beacon enabled mode or beacon enabled mode is used.
- *Indirect data transmission:* This only applies to data transfer from a coordinator to its devices. In this mode, a data frame is kept in a transaction list by the coordinator, waiting for extraction by the corresponding device. A device can find out if it has a packet pending in the transaction list by checking the beacon frames received from its coordinator. Occasionally, indirect data transmission can also happen in non-beacon enabled mode. For example, during an association procedure, the coordinator keeps the association response frame in its transaction list and the device polls and extracts the association response

frame. Unslotted CSMA-CA or slotted CSMA-CA is used in the data extraction procedure.

- *GTS data transmission:* This only applies to data transfer between a device and its coordinator, either from the device to the coordinator or from the coordinator to the device. No CSMA-CA is needed in GTS data transmission.

Power conservation has been one of research focuses for wireless networks [23,25,26,38,43,46,48], since most devices in wireless networks are battery powered. The standard was developed with the limited power supply availability in mind and favors battery powered devices. The superframe structure, the indirect data transmission and the *BatteryLifeExtension* option are all examples. If the *BatteryLifeExtension* is set to TRUE, all contention based transactions are required to begin within *macBattLifeExtPeriods* (default value 6) full backoff periods after the inter-frame space (IFS) period of the beacon frame.

4.3 NS2 Simulator

The 802.15.4 NS2 [22] simulator developed at the Joint Lab of Samsung and the City University of New York confirms to IEEE P802.15.4/D18 Draft[2]. Figure 4.2 outlines the function modules in the simulator, and a brief

[2]The source code of the IEEE 802.15.4 NS2 simulator has been included in the release of NS 2.28 and is also available at http://www-ee.ccny.cuny.edu/zheng/pub.

description is given below for each of the modules.

- *Wireless Scenario Definition:* It selects the routing protocol; defines the network topology; and schedules events such as initializations of PAN coordinator, coordinators and devices, and starting (stopping) applications. It defines radio-propagation model, antenna model, interface queue, traffic pattern, link error model, link and node failures, superframe structure in beacon enabled mode, radio transmission range, and animation configuration.
- *Service Specific Convergence Sublayer (SSCS):* This is the interface between 802.15.4 MAC and upper layers. It provides a way to access all the MAC primitives, but it can also serve as a wrapper of those primitives for convenient operations. It is an implementation specific module and its function should be tailored to the requirements of specific applications.
- *802.15.4 PHY:* It implements all 14 PHY primitives.
- *802.15.4 MAC:* This is the main module. It implements all the 35 MAC sublayer primitives.

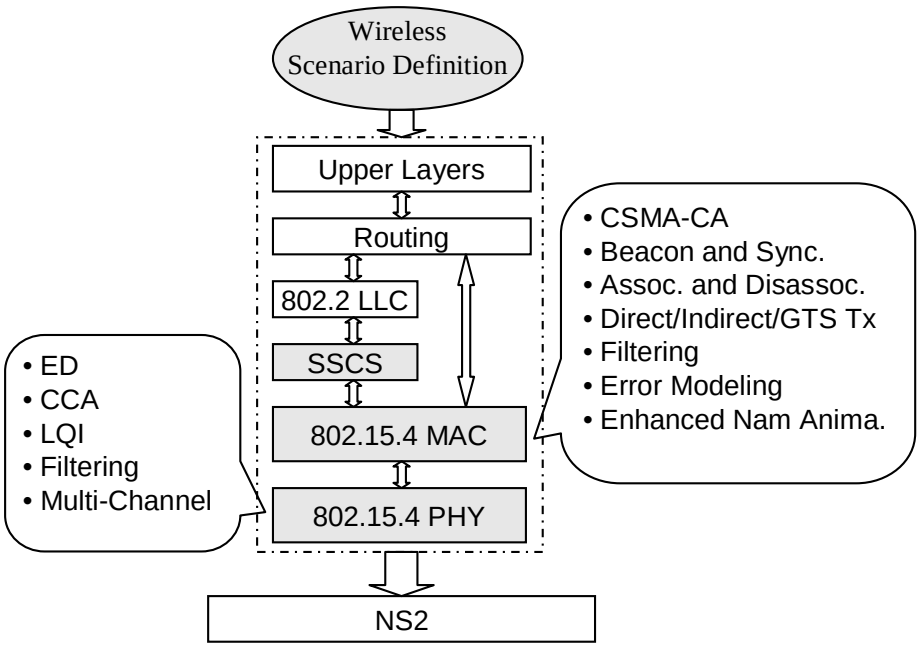


Figure 4.2: NS2 Simulator for IEEE 802.15.4

4.4 Performance Metrics and Experimental Setup

4.4.1 Performance Metrics

We define the following metrics for studying the performance of 802.15.4. All metrics are defined with respect to MAC sublayer and PHY layer in order to isolate the effects of MAC and PHY from those of upper layers.

- *Packet delivery ratio:* The ratio of packets successfully received to packets sent in MAC sublayer. This metric does not differentiate transmissions and retransmissions, and therefore does not reflect what percentage of upper layer payload is successfully delivered, although they

are related.

- *Hop delay:* The transaction time of passing a packet to a one-hop neighbor, including time of all necessary processing, backoff as well as transmission, and averaged over all successful end-to-end transmissions within a simulation run. It is not only used for measuring packet delivery latency, but also used as a negative indicator of the MAC sublayer capacity. The MAC sublayer has to handle the packets one by one and therefore a long delay means a small capacity.
- *RTS/CTS overhead:* The ratio of request-to-send (RTS) packets plus clear-to-send (CTS) packets sent to all the other packets sent in 802.11. This metric is not applicable to 802.15.4, in which RTS/CTS mechanism is not used. We compare the performances of 802.11 and 802.15.4 to justify the dropping of RTS/CTS mechanism in 802.15.4.
- *Successful association rate:* The ratio of devices successfully associated with a coordinator to the total devices trying to associate with a coordinator. In our experiments, a device will retry in one second if it fails to associate with a coordinator in the previous attempt. The association is considered successful if a device is able to associate with a coordinator during a simulation run, even if multiple association attempts have been made.
- *Association efficiency:* The average number of attempts per successful association.

- *Orphaning rate:* A device is considered orphaned if it misses *aMaxLostBeacons* (default value 4) beacons from its coordinator in a row. The orphaning rate is defined as the ratio of devices orphaned at least once to the total devices that are in beacon enabled mode and keep tracking beacons. This metric is not applicable to devices in non-beacon enabled mode or devices in beacon enabled mode but not tracking beacons. In our experiments, all devices in beacon enabled mode track beacons.
- *Orphaning recovery rate:* Two different versions are defined for this metric. One is the ratio of orphaned devices that have successfully relocated their coordinators, i.e., have recovered from orphaning, to the total orphaned devices. The other is the ratio of recovered orphanings to the total orphanings, in which multiple orphanings of a device are counted. No further attempt is made if the orphaning recovery procedure fails.
- *Collision rate:* The total collisions during a simulation run.
- *Collision rate between hidden terminals:* The total collisions that occur between hidden terminals during a simulation run. Hidden terminals prevent carrier sense from working effectively, and therefore transmissions from them are likely to collide at a third node [50]. In 802.11, the request-to-send (RTS) and clear-to-send (CTS) mechanism is used to tackle this problem [20].

- *Repeated collision rate:* The total collisions that happen more than once between the same pair of packets during a simulation run.
- *Collision distribution:* The time distribution, within a superframe, of collisions. This metric is only used in beacon enabled mode.
- *Duty cycle:* The ratio of the active duration, including transmission, reception and carrier sense time, of a transceiver to the whole session duration.

4.4.2 Experimental Setup

Five sets of experiments are designed to evaluate the various performance behaviors of 802.15.4, including those applicable to all wireless networks (such as packet delivery ratio, packet delivery latency, control overhead, and transmission collision) as well as other behaviors specific to LR-WPANs (such as association, orphaning, and different transmission methods). The first set is for non-beacon enabled mode, the second and third sets are for mixed mode, that is, a combination of beacon enabled mode and non-beacon enabled mode, and the fourth and fifth sets are for beacon enabled mode. The first three sets run in a multi-hop environment (Figure 4.3 (a)), and the other two sets run in a one-hop star environment (Figure 4.3 (b)). Although a specific network can take a quite different topology, the two topologies used in our experiments represent the topologies currently supported by 802.15.4 and are enough for performance study purpose.

General parameters: Assuming a 10^{-6} to 10^{-5} link bit error rate (BER), we apply a 0.2% statistical packet error rate (PER) to all our experiments. The simulation duration is 1000 seconds, and the application traffic runs from 20 to 900 second, leaving enough time for the experiment to shut down gracefully. Since the popular constant bit rate (CBR) traffic used in most simulations is too deterministic for non-mobile wireless networks, Poisson traffic is used for all application sessions in our experiments. The application packet size is 90 bytes. Except the fifth set of experiments, all the other experiments use direct data transmission. The radio propagation model adopted in all our experiments is two-ray ground reflection. Beacon order (BO) and superframe order (SO) take the same value in all beacon enabled modes, that is, the optional inactive part is not included in superframes. Most experiments run 10 times with random seeds, but those with a traffic load of 0.2 packet per second (pps) and those with a traffic load of 0.1 pps run 20 times and 40 times respectively. Other experiment specific configuration information is given in the following paragraphs corresponding to each set of experiments.

Experiment set 1 – Comparing 802.15.4 with 802.11: The first set of experiments are used to compare the performances of 802.15.4 and 802.11. Although 802.15.4 and Bluetooth bear more similarities from the application point of view, 802.15.4 and 802.11 are more comparable as far as our performance study is concerned. Both 802.15.4 and 802.11 support multi-

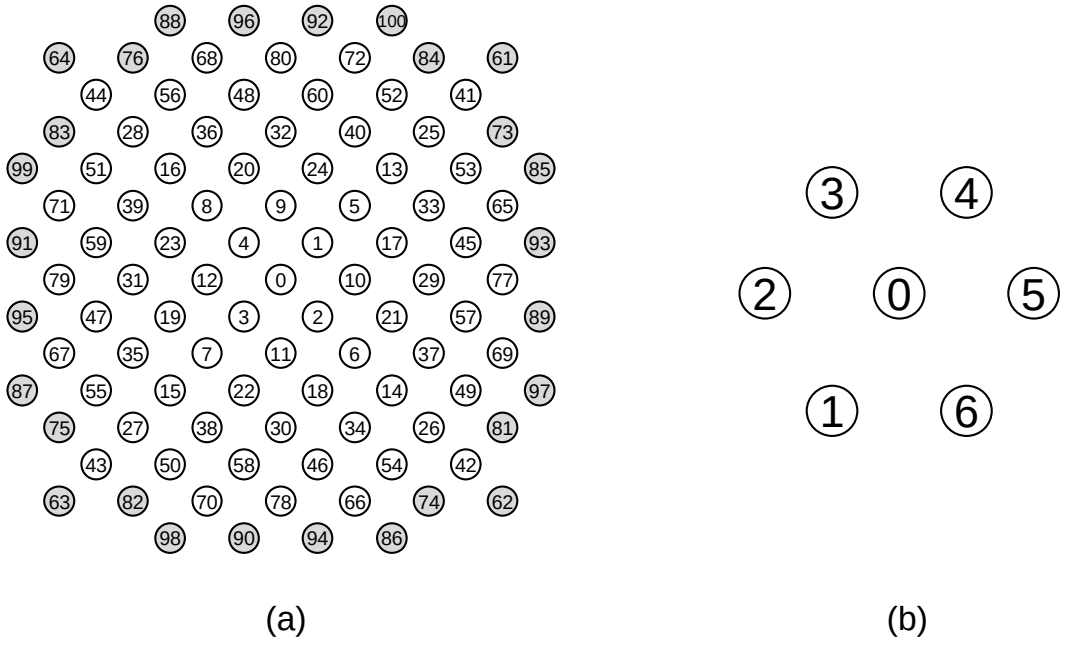


Figure 4.3: Experiment Scenarios

hop network topology and peer-to-peer communications, which are used in our first set of experiments. The dominant topology in Bluetooth, on the other hand, is one-hop star or so-called piconet, which consists of one coordinator and up to seven devices. In a piconet, a device only communicates with its coordinator. Although scatternets can be used to extend the coverage and the number of devices of a Bluetooth network, our research work showed that there are scalability problems in scatternets [36]. Furthermore, all the devices in either 802.15.4 or 802.11 share a single chip code for spread spectrum, while different devices in Bluetooth are assigned different chip codes. Based on the above facts, we select 802.11 instead of Bluetooth for comparison. The performance is evaluated with respect to the following

parameters as well as those listed in the previous paragraph:

- 101 nodes evenly distributed in an 80 x 80 m^2 area (Figure 4.3 (a)).
- 9 meter transmission range, which only covers the neighbors along diagonal direction.
- 802.15.4 operates at an over air data rate of 250 Kbps (in the 2.4 GHz ISM band) and in non-beacon enabled mode, and 802.11 operates at a data rate of 2 Mbps.
- Poisson traffic with the following average packet rates: 0.1 packet per second (pps), 0.2 pps, 1 pps, 5 pps and 10 pps.
- We apply two types of application traffic: (1) peer-to-peer application traffic, which consists of six application sessions between the following nodes: 64 $\rightarrow$ 62, 63 $\rightarrow$ 61, 99 $\rightarrow$ 85, 87 $\rightarrow$ 97, 88 $\rightarrow$ 98, and 100 $\rightarrow$ 86, and (2) multiple-to-one application traffic, which consists of twelve application sessions from nodes 64, 62, 63, 61, 99, 85, 87, 97, 88, 98, 100 and 86 to node 0. The first type of application traffic is used to study the general peer-to-peer behavior of 802.15.4 and, for comparison, it is applied to both 802.15.4 and 802.11. The second type of application traffic targets the important application of 802.15.4, wireless sensor networks, where traffic is typically between multiple source nodes and a sink. It is only applied to 802.15.4. Although the second type of application traffic is not used for comparing 802.15.4

with 802.11, we include it here to facilitate the comparison of 802.15.4 behaviors under different application traffic. We refer to the second type of application traffic as sink-type application traffic hereinafter.

Experiment set 2 – Association efficiency: The second set of experiments are designed to evaluate the association efficiency under different number of beaconing coordinators and different beacon orders. The same network topology, transmission range, frequency band, data rate, and peer-to-peer application sessions are used as in the first set of experiments. Except node 0, which is the PAN coordinator, and the leaf nodes depicted in grey, which are pure devices, all the other nodes serve as both a coordinator (to its children) and a device (to its parent). So we have 73 coordinators and 100 devices. This set of experiments run in a mixed mode, with different percentage of coordinators beaconing (0%, 25%, 50%, 75% and 100%). The beacon order varies and takes the values of 0, 1, 2, 3, 4, 5, 6 and 10. The application traffic is fixed at 1 pps.

Experiment set 3 – Orphaning: The third set of experiments are used to study the device orphaning behavior, namely, how often orphanings happen and what percentage of orphanings, in terms of number of orphaned devices or number of orphanings, can be recovered. The experimental setup is the same as that of the second set of experiments.

Experiment set 4 – Collision: The fourth set of experiments target the collision behavior of 802.15.4. The experiments run in a beacon enabled

star environment. Nevertheless, except some beacon specific metrics, most of the metrics extracted from this set of experiments are general and can serve for both beacon and non-beacon enabled modes. Besides the general parameters given above, the following parameters are used in the experiments:

- 7 nodes form a star with a radius of 10 meters, with one coordinator at the center and six devices evenly distributed around it (Figure 4.3 (b)).
- 15 meter transmission range, which enables the coordinator to reach all the devices. However, a device can only reach the coordinator and two devices adjacent to it. In other words, devices are hidden from each other unless they are adjacent to each other.
- Operates at an over air data rate of 250 Kbps (in the 2.4 GHz ISM band).
- Poisson traffic with the average packet rate of 1 pps.
- Six application sessions, one for each device, are setup from the devices to the coordinator.
- The beacon order changes from 0 to 8.

Experiment set 5 – Direct, indirect and GTS data transmissions: The last set of experiments are used to investigate the different features of the three data transmission methods in 802.15.4. We compare the packet delivery ratio, hop delay and duty cycle of the three different methods. All the pa-

rameters are the same as those in the fourth set of experiments, except that only two application sessions originating from adjacent devices are used, and that three different data transmission methods are used.

4.5 Experimental Results

4.5.1 Comparing IEEE 802.15.4 with IEEE 802.11

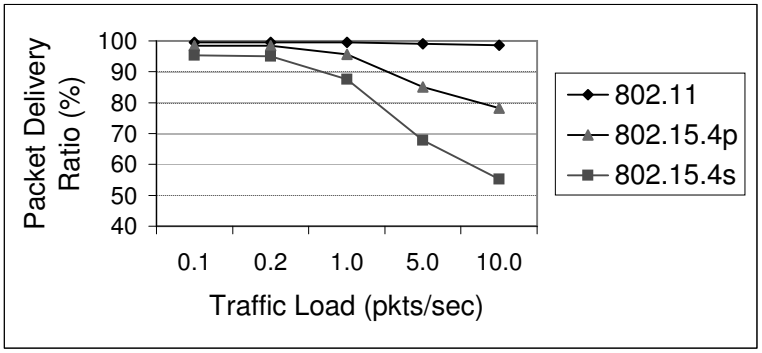


Figure 4.4: Comparing 802.15.4 with 802.11: Packet Delivery Ratio

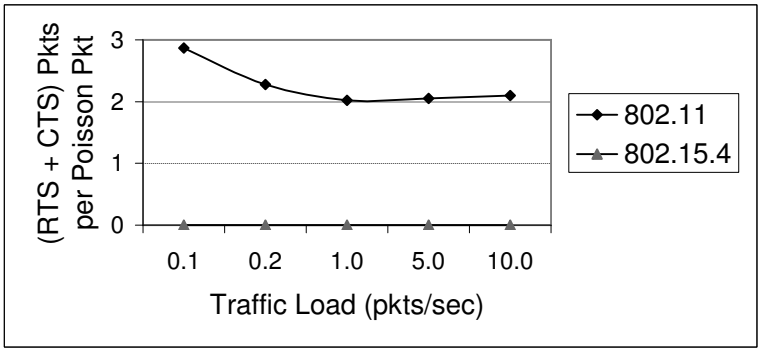


Figure 4.5: Comparing 802.15.4 with 802.11: RTS /CTS Overhead

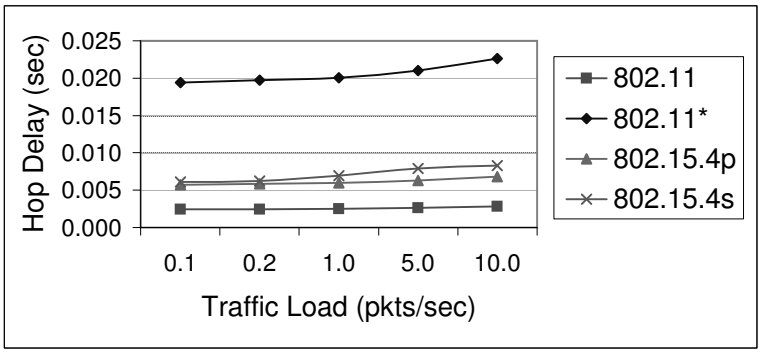


Figure 4.6: Comparing 802.15.4 and 802.11: Hop Delay

To distinguish experiment results for 802.15.4 with different application traffic, we use 802.15.4p and 802.15.4s to denote the data series corresponding to peer-to-peer application traffic and sink-type application traffic respectively (see Figure 4.4 and Figure 4.6). However, when experiment results are not specific to a certain application traffic (e.g., the data series 802.15.4 in Figure 4.5) or only one application traffic is applied (e.g. for 802.11), the protocol name is used only to denote the corresponding data series.

For peer-to-peer application traffic, as shown in Figure 4.4, the packet delivery ratio of 802.11 decreases slowly from 99.53% to 98.65% when the traffic load changes from 0.1 packet per second (pps) to 10 pps. On the other hand, the packet delivery ratio of 802.15.4 drops from 98.51% to 78.26% for the same traffic load change (data series 802.15.4p in Figure 4.4). For sink-type application traffic, the packet delivery ratio of 802.15.4 drops more sharply from 95.40% to 55.26% when the traffic load changes from 0.1

pps to 10 pps (data series 802.15.4s in Figure 4.4). In general, 802.15.4 maintains a high packet delivery ratio for application traffic up to 1 pps (95.70% for 802.15.4p and 87.58% for 802.15.4s), but the value decreases quickly as traffic load increases.

The difference of packet delivery ratio between 802.15.4 and 802.11 comes from the fact that the former does not use RTS/CTS mechanism while the latter does. This RTS/CTS overhead proves to be useful when traffic load is high, but obviously too expensive for low data rate applications as of the case of LR-WPANs for which 802.15.4 is designed. From Figure 4.5, we can see the ratio of (RTS+CTS) packets to Poisson data packets is within the scope [2.02, 2.78], which can not be justified in 802.15.4, considering the less than 4% increase of packet delivery ratio for application traffic up to 1 pps. Note that, even under collision-free condition, the ratio of (RTS+CTS) packets to Poisson data packets is larger than 2.0, because RTS/CTS packets are also used for transmissions of other control packets such AODV packets. It is clear that the high ratio of (RTS+CTS) packets to Poisson data packets for 0.1 pps must come from the high ratio of other control packets to Poisson data packets, since collisions are ignorable under such low traffic load.

The RTS/CTS mechanism also affects the network latency. We measure the average hop delay for both protocols in comparison, and the results are depicted in Figure 4.6. The initial results show that 802.11 enjoys a lower

delay than 802.15.4 (data series 802.11 and 802.15.4p in Figure 4.6). Nevertheless, this comparison is unfair to 802.15.4, since it operates at a data rate of 250 Kbps while 802.11 operates at 2 Mbps in our experiments. Taking this into account, we normalize the hop delay according to the media data rate, which gives us a different view that the hop delay of 802.11 is around 3.3 times of that of 802.15.4 (data series 802.11* and 802.15.4p in Figure 4.6). The hop delay for sink-type application traffic is 6.3% (for 0.1 pps) to 20.9% (for 10 pps) higher than that for peer-to-peer application traffic (data series 802.15.4s and 802.15.4p in Figure 4.6). The increment of delay is expected, since all the traffic flows now need to converge on the sink node.

4.5.2 Association Efficiency

Table 4.1: Successful Association Rate vs. Beaconing Coordinator Ratio

Beaconing coordinator ratio (%)	0	25	50	75	100
Successful association rate (%)	100	100	100	99	100

The typical scenario of an LR-WPAN is a densely distributed unattended wireless sensor network. Self-configuration in deployment and auto-recovery from failures is a highly desirable feature in such a network [37]. For this purpose, 802.15.4 includes an association and disassociation mech-

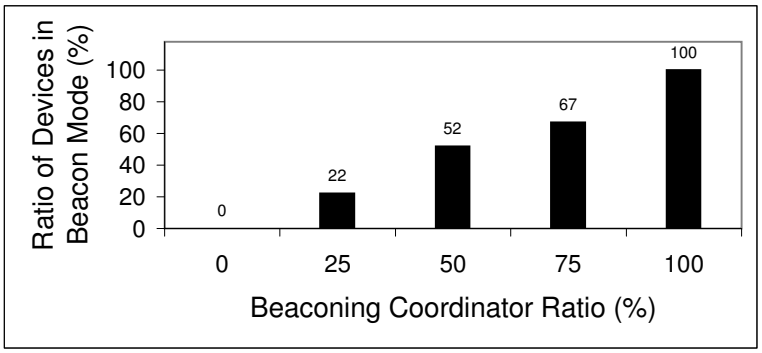


Figure 4.7: Devices Associated with Beaconing Coordinators

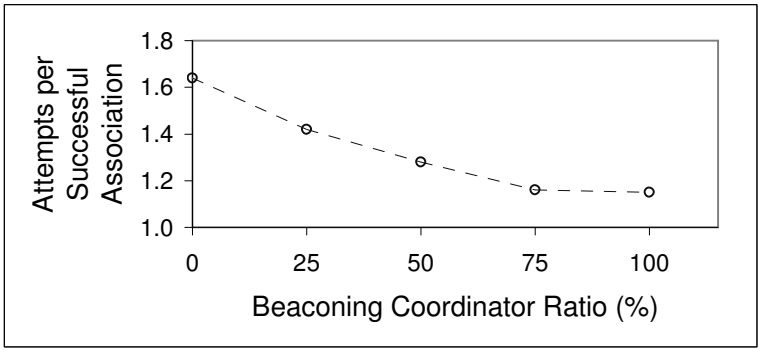


Figure 4.8: Association Efficiency vs. Beaconing Coordinator Ratio

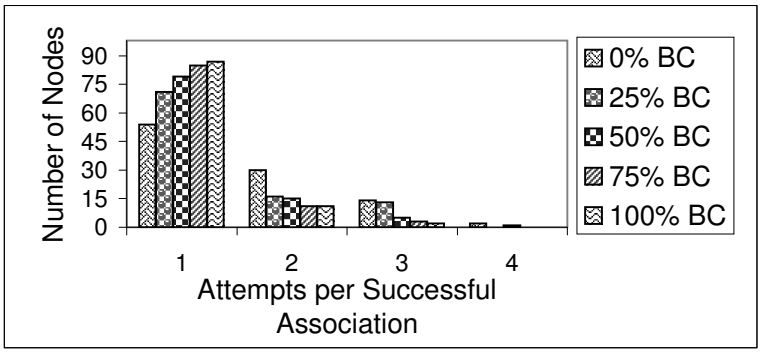


Figure 4.9: Attempts per Successful Association vs. Beaconing Coordinator (BC) Ratio

Table 4.2: Distribution of Association Attempts (expressed in number of devices)

	1 attempt	2 attempts	3 attempts	4 attempts
0% beaconing coordinators	54	30	14	2
25% beaconing coordinators	71	16	13	–
50% beaconing coordinators	79	15	5	1
75% beaconing coordinators	85	11	3	–
100% beaconing coordinators	87	11	2	–

anism together with an orphaning and coordinator relocation mechanism in its design. We give out the experimental results of association in this subsection, while the experimental results of orphaning will be given in next subsection.

To associate with a coordinator, a device will perform an active channel scan, in which a beacon request frame is sent, or a passive channel scan, in which no beacon request frame is sent, to locate a suitable coordinator. Active channel scan is used in our experiments, since a device needs to explicitly request for beacons in non-beacon enabled environment. When a coordinator receives the beacon request frame, it handles it differently depending on whether itself is in beacon enabled mode or non-beacon enabled mode. If the coordinator is in beacon enabled mode, it discards the frame silently, since beacons will be bent periodically anyway. Otherwise, the coordinator needs to unicast a beacon to the device soliciting beacons. In our experiments, we vary the percentage of beaconing coordinators to see the different effects of beaconing coordinators and non-beaconing coordinators.

In general, the successful association rate is very high (more than 99%) for different combinations of beaconing coordinators and non-beaconing coordinators, as illustrated in Table 4.1. From Figure 4.7, we can see that a device gets an almost equal chance to associate with a beaconing coordinator or a non-beaconing coordinator. However, this result is obtained for beacon order 3 and it may be different for other beacon orders. Normally, a beaconing coordinator with a larger beacon order (i.e., longer superframe) reacts slowly to a beacon request, which means it will not get the same chance to serve as a coordinator for a certain device, when competing with other non-beaconing coordinators or beaconing coordinators with smaller beacon orders.

The association efficiency shown in Figure 4.8, in terms of attempts per successful association, is high. The association procedure is a multi-step procedure as briefly described by the following pseudo code (for device part only):

```
1:  channel scan
2:  if coordinators not found
3:          association fail
4:  elseif no coordinators permit association
5:          association fail
6:  else
7:          select a proper coordinator
8:          send association request to the coord.
9:          wait for ACK
10:         if ACK not received
11:                 association fail
```

```
12:     else
13:         send data request to the coord.
14:         wait for ACK
15:         if ACK not received
16:             association fail
17:         else
18:             wait for association response
19:             if asso. response not received
20:                 association fail
21:             elseif association not granted
22:                 association fail
23:             else
24:                 association succeed
```

If there are multiple non-beaconing coordinators around, they all will try to unicast a beacon, using unslotted CSMA-CA, to the device asking for beacons. These beacons are likely to collide at the device due to the hidden terminal problems as a fact of lacking RTS/CTS, that is, even the first step of the association may fail. The situation is better if there are multiple beaconing coordinators around, since they will continue beaconing as usual even if a beacon request is received. Of course, if beacons are sent with high frequency (low beacon order), then the collisions will increase, which will bring down the association efficiency. In summary, non-beaconing coordinators are likely to affect the first step of the association procedure, while the beaconing coordinators can affect all the steps. As revealed by our experimental results, beaconing coordinator as a whole is a better choice regarding association efficiency, provided the beacon order is not too small.

Table 4.3: Successful Association Rate vs. Beacon order

Beacon order	0	1	2	3	4	5	6	10
Successful association rate (%)	99	96	95	100	99	100	100	99

Table 4.2 gives out the distribution of association attempts, which shows that most of the devices succeed in their first association attempt, a small part of the devices try twice or three times, and three devices try four times.

Association is the basis of tree formation in a peer-to-peer multi-hop network. The efficiency of tree formation is directly related to association efficiency. Tree is a useful structure and can be used by network layer, especially for routing purpose. In this set of experiments, a tree is quickly formed thanks to the high association efficiency. Various configurations are also done during this procedure, such as select a channel and an identifier (ID) for the PAN, determine whether beacon enabled mode or non-beacon enabled mode to be used, choose the beacon order and superframe order in beacon enabled mode, assign a 16-bit short address for a device, set the *BatteryLifeExtension* option and many other options in the MAC layer PAN information base (MPIB). The smooth procedure of association and tree formation indicates that an 802.15.4 network has a feature of self-configuration and can shape up efficiently.

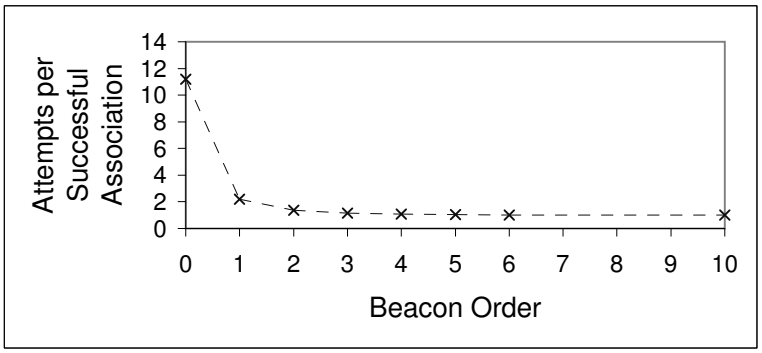


Figure 4.10: Association Attempts vs. Beacon order

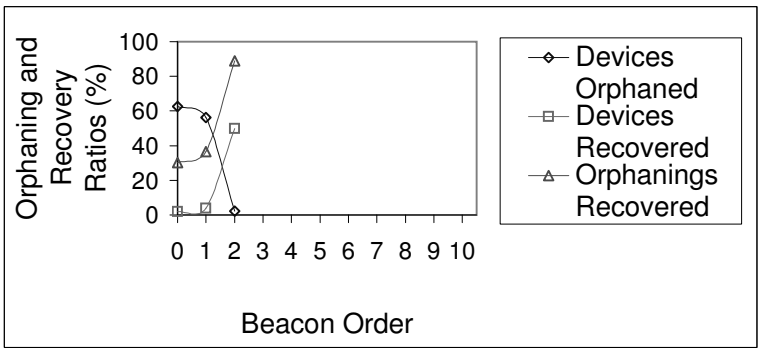


Figure 4.11: Orphaning and Recovery

4.5.3 Orphaning

The orphaning study is conducted in an environment with all coordinators beaconing. Specifically we examine the orphaning behavior for different beacon orders. Orphaning mechanism works only if a device is successfully associated with a beaconing coordinator, and the device keeps tracking the beacons from the coordinator. Since orphaning is related to association, here we also give out the association results. Table 4.3 and Figure 4.10

suggest that the performance of beacon enabled modes with small beacon orders is not so good as that with large beacon orders. For example, the attempts per successful association for beacon order 0 is "outstanding" among its peers. And the successful association rate for beacon order 1 and beacon order 2 is also slightly lower than others.

Unsurprisingly, orphaning is also more serious in those beacon enabled modes with smaller beacon orders (Figure 4.11). The percentage of devices orphaned in beacon order 0 or beacon order 1 is about the same (around 58%), and is 29 times of that in beacon order 2. There is no orphaning in beacon order 3 or up. In an environment with high rate of orphaning, the chance an orphaned device successfully recovers from all orphanings is very low (2% for beacon order 0 and 4% for beacon order 1 as shown by data series "Devices Recovered"), but the recovery rate of orphaning itself is not as bad (from 30% to 89% as shown by data series "Orphanings Recovered"). One point worth mentioning is that, a device failed to recover from all orphanings still benefits from the recovery mechanism, since its association with the coordinator is prolonged, though not to the end of the session.

4.5.4 Collision

It is clearly shown in Figure 4.12 that more collisions happen in low beacon orders than in high beacon orders. And the network virtually loses

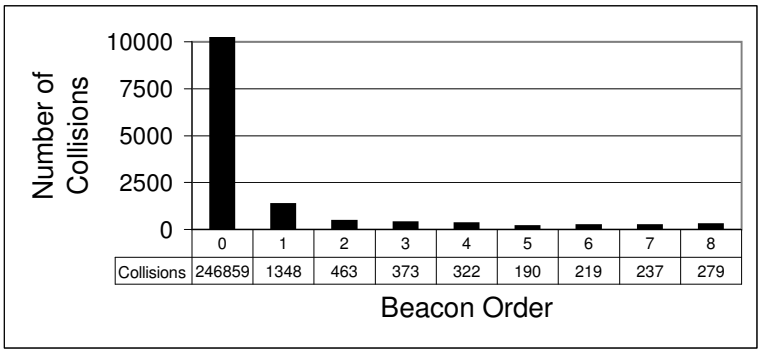


Figure 4.12: Collisions vs. Beacon Order

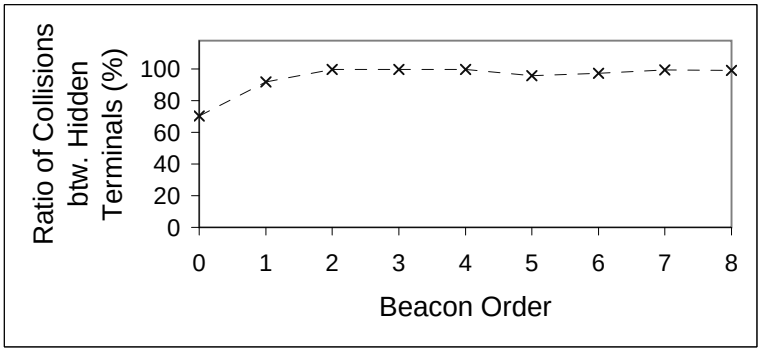
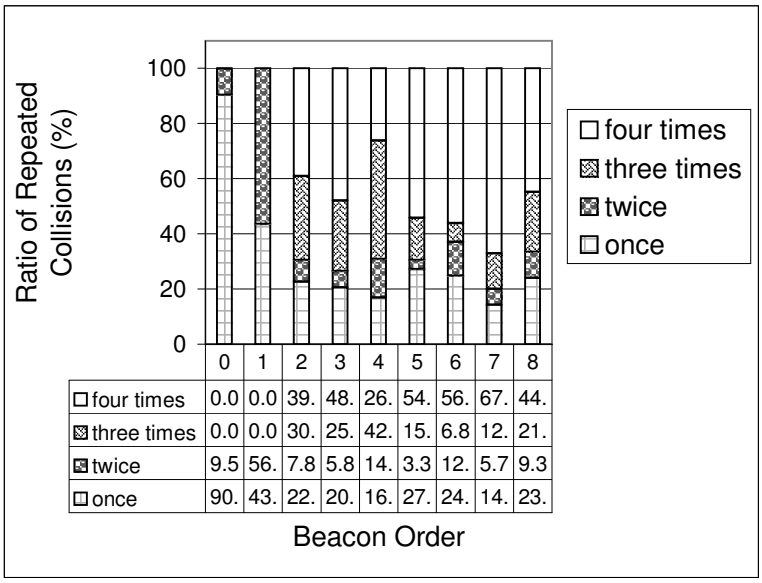


Figure 4.13: Ratio of Collisions between Hidden Terminals

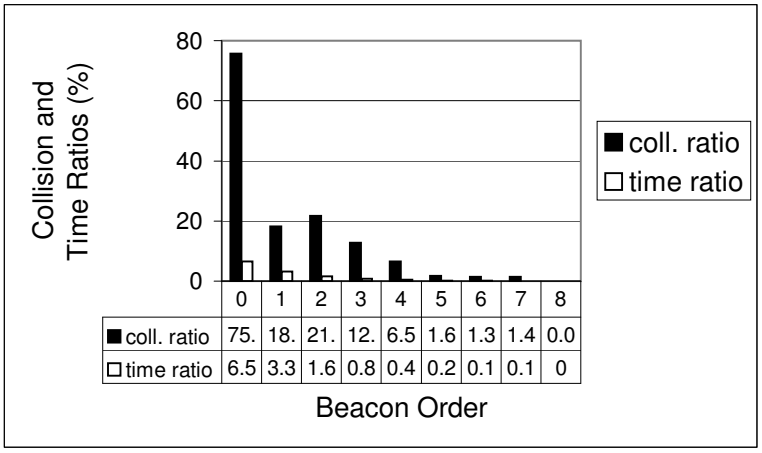
its control in beacon order 0, due to large number of collisions. This type of "Beacon Storm" problem is alleviated in high order beacons. Due to the broadcast nature of wireless networks, broadcast-based storm is not a rare phenomenon [44]. It necessitates careful handling.

As expected, the majority of collisions happen between hidden terminals (Figure 4.13), that is, between any two devices not adjacent to each other in our experiments (see subsection 4.4.2). However, probability of collisions between non-hidden terminals in low beacon orders is not trivial either. This



	0	1	2	3	4	5	6	7	8
four times	0.0	0.0	39.	48.	26.	54.	56.	67.	44.
three times	0.0	0.0	30.	25.	42.	15.	6.8	12.	21.
twice	9.5	56.	7.8	5.8	14.	3.3	12.	5.7	9.3
once	90.	43.	22.	20.	16.	27.	24.	14.	23.

Figure 4.14: Ratio of Repeated Collisions



	0	1	2	3	4	5	6	7	8
coll. ratio	75.	18.	21.	12.	6.5	1.6	1.3	1.4	0.0
time ratio	6.5	3.3	1.6	0.8	0.4	0.2	0.1	0.1	0

Figure 4.15: Ratio of Collisions within the First Millisecond of a Superframe

means the slotted CSMA-CA can no longer work effectively if the beacon order is very small, and the chance that two non-hidden terminals jump to the channel simultaneously is significantly increased.

Unexpectedly, the ratio of repeated collisions is very high, as manifested in Figure 4.14. By tracking these collisions, we find the reason is that the suggested backoff length in 802.15.4 is too short, especially for long frames (Physical Protocol Data Unit larger than 100 bytes). This short backoff length results from the consideration of energy conservation, but a too short backoff length will cause repeated collisions and defeat the initial design goal. The fact that no collisions repeated more than twice in beacon order 0 and beacon order 1 is somewhat misleading. It is not because that the collisions can be resolved within the first two backoffs, but that the enormous number of collisions make it impossible in effect for a packet to collide with another packet more than twice before it reaches its retransmission threshold.

The last metric we extract from this set of experiments is the time distribution of collisions within a superframe. In beacon enabled mode, a transaction (transmission of a frame as well as reception of an acknowledgment frame if required) using slotted CSMA-CA is required to be completed before the end of the contention access period (CAP). Otherwise, the transaction should be delayed until the beginning of next superframe. In such a design, more collisions are expected at the beginning of a superframe, especially a short superframe (low beacon order) in which more transactions are likely to be delayed until the beginning of next frame. This is confirmed by our experimental results shown in Figure 4.15. For beacon order 0, for

example, about 75% of collisions happen within the first millisecond of a superframe (but one millisecond is only about 6.5% of a superframe of beacon order 0).

4.5.5 Direct, Indirect, and GTS Data Transmissions

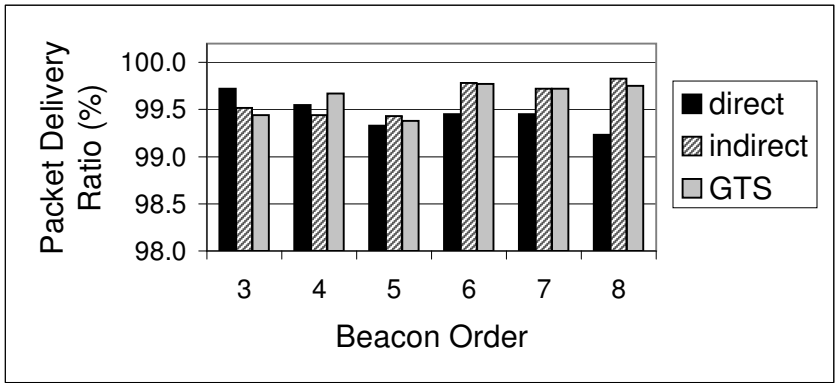
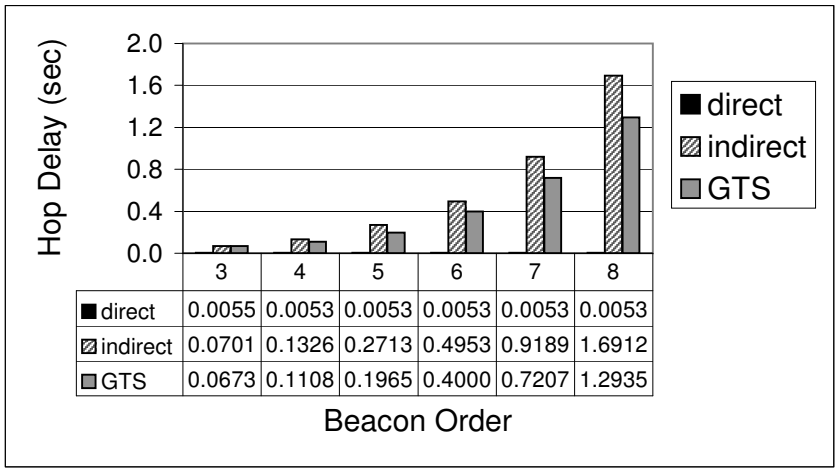


Figure 4.16: Different Data Transmission Methods: Packet Delivery Ratio



	3	4	5	6	7	8
direct	0.0055	0.0053	0.0053	0.0053	0.0053	0.0053
indirect	0.0701	0.1326	0.2713	0.4953	0.9189	1.6912
GTS	0.0673	0.1108	0.1965	0.4000	0.7207	1.2935

Figure 4.17: Different Data Transmission Methods: Hop Delay

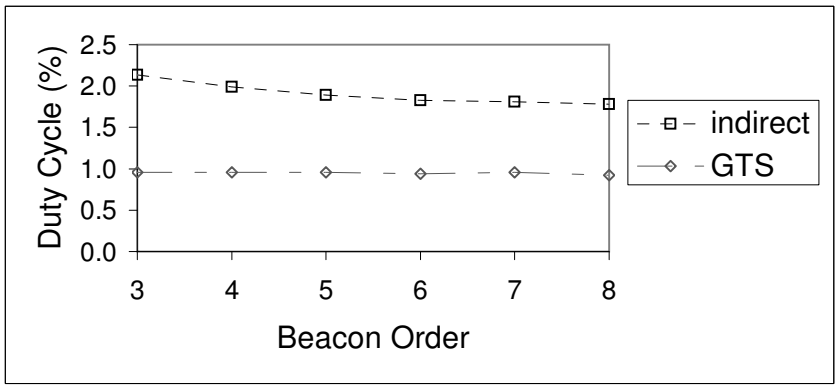


Figure 4.18: Different Data Transmission Methods: Duty Cycle

In this set of experiments, we compare three different data transmission methods, i.e., direct, indirect and guaranteed time slot (GTS) data transmissions (DIG). The focus is latency (Figure 4.17) and duty cycle (Figure 4.18), but packet delivery ratio is also given (Figure 4.16), for the sake of completion. Small beacon orders 0, 1 and 2 are not shown in the above figures, since, in GTS data transmission, we only allocate one slot for each device and the slot is too short for holding a data frame.

No significant difference has been observed in the packet delivery ratio among the three data transmission methods. Nevertheless, the hop delay varies, which will definitely affect the packet delivery ratio in upper layers. The hop delay in direct data transmission is much shorter than those in indirect and GTS data transmissions.

One fundamental aspect of 802.15.4 is low power consumption, which is very desirable in a wireless sensor network, as the replacement of batter-

ies is very cumbersome due to the large number of sensors. Most power-saving mechanisms in 802.15.4 are based on beacon enabled mode. In direct data transmission, if the *BatteryLifeExtension* option is set to TRUE, the receiver of the beaconing coordinator is disabled after *macBattLifeExtPeriods* (default value 6) backoff periods following the inter-frame space (IFS) period of the beacon frame. Using default configuration, this means that the transceiver of a coordinator or a device is required to be turned on for only about 1/64 of the duration of a superframe, if no data to be exchanged. If the value of *BatteryLifeExtension* is FALSE, the receiver of the beaconing coordinator remains enabled for the entire CAP. In indirect data transmission, a device can enter a low power state, like sleeping state, if it finds there are no pending packets by checking the beacon received from its coordinator.

As shown in Figure 4.18, the duty cycle is around 2% in indirect data transmission, and about 1% in GTS data transmission. However, there are two slots or 12.5% of a superframe allocated for GTS data transmission in our experiments, which means that $(12.5 - 1)/12.5 = 92\%$ of the allocated GTS slots are wasted. This result shows that GTS is too expensive for low data rate applications.

The above duty cycle measurement is based on the traffic load of one packet per second, and it shall vary when traffic load changes. Perfect synchronization among devices is also assumed in the measurement, which is generally not true in practice. Some margin should be provided for the non-

perfect synchronization, which means an increment in duty cycle. One more point about power conservation is that, it is acquired at the cost of delay, as clearly shown in Figure 4.17. The power consumption mechanisms employed in 802.15.4 are based on the assumption of low data rate and should be used properly.

4.6 Possible Enhancements of IEEE 802.15.4

In section 4.5, we identified some issues that could degrade the network performance if not handled properly. The performance will deteriorate if the traffic load is high, due to the hidden terminal problems. The backoff period should be adjusted to avoid repeated collisions. Superframes with low beacon orders can lower the backoff efficiency of slotted CSMA-CA and lead to high collision probability at the beginnings of superframes. Besides those given in section 4.5, there are some other issues [45], including non-atomic transactions, insufficient support of multi-hop beacon enabled networks, and improper default status of transceiver. In this section, we propose some possible enhancements with regard to those identified issues.

Repeated collisions: CSMA-CA does not work in the case of hidden terminals. A large backoff period can help alleviate the problem. The backoff period in IEEE 802.15.4, however, is too small. In current 802.15.4, the

backoff period is:

$$aUnitBackoffPeriod \times (2^{BE} - 1)$$

where

$aUnitBackoffPeriod = 20\, symbols$

BE (backoff exponent) $= 2$ to 5 for beacon enabled mode, or 3 to 5 for non-beacon enabled mode.

So the maximum backoff period is 620 symbols = 310 bytes for 2.4 GHz band, or 77.5 bytes for 868/915 MHz bands. In case of hidden terminal problems, CSMA-CA will sense the channel as being idle. So only BE =2 (in beacon enabled mode) or 3 (in non-beacon enabled mode) is used, which is much smaller than the above maximum backoff period. Since the PHY protocol data unit (PPDU) can be as large as 127 bytes, two collided frames have a very high chance to collide again during retransmissions. One possible solution is to relate BE to the retransmission status. Let $txRetry$ denote the number of retransmissions for a certain frame, then set BE according to the following:

$$BE = \begin{cases} (2 + txRetry) \text{ to } (5 + txRetry) & \text{(beacon enabled)} \\ (3 + txRetry) \text{ to } (5 + txRetry) & \text{(non-beacon enabled)} \end{cases}$$

Non-atomic transactions: Here we focus on non-atomic problems caused by CCA and interframe space (IFS), though hidden terminal problems can

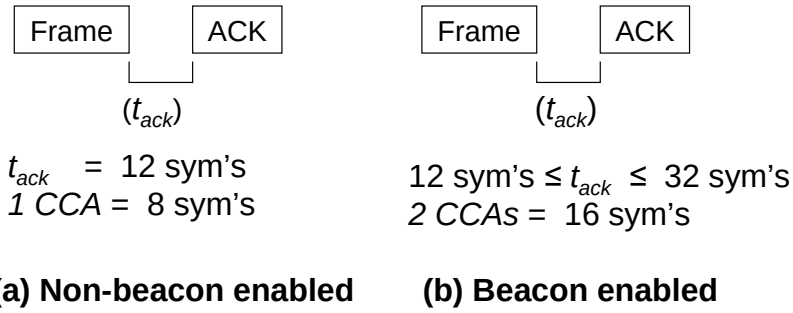


Figure 4.19: Acknowledgement

also lead to non-atomic problems. Denote t_ack the time interval between the reception of a frame and the transmission of the corresponding acknowledgement (ACK) (when ACK required) (Figure 4.19). Then for non-beacon enabled mode, $t\_ack = 12$ symbols, while CSMA-CA requires 1 CCA $= 8$ symbols. For beacon enabled mode, 12 symbols $\leq t\_ack \leq 32$ symbols, and CSMA-CA requires 2 CCAs $= 16$ symbols. Therefore another transmission can happen between the transmissions of a frame and its ACK in both non-beacon enabled and beacon enabled modes. To solve the problem, we need to increase CCA duration so that it is larger than t_ack. For instance, letting CCA $= 13$ symbols will solve the problem in non-beacon enabled mode. For beacon enabled mode, we need to perform two CCAs. One possible solution is: CCA (13 symbols) + delay (7 symbols) + CCA (13 symbols) $> t\_ack$.

Insufficient support of multi-hop beacon enabled networks: Beacons are important to network operations and their transmissions do not follow CSMA-CA. Current 802.15.4 does not provide any mechanism at the MAC

sublayer to avoid beacon collisions or collisions between beacons and data packets. In a multi-hop beacon enabled network, each node has to prevent its beacons and data packets from destroying the beacons from all its neighbors. To support multi-hop beacon enabled networks, we may modify the CSMA-CA as follows:A node shall begin to transmit a frame only if all the following conditions are satisfied:

- The channel is sensed as being idle;
- The transaction can be finished before the end of the current CAP corresponding to its beaconing parent or the end of the superframe corresponding to any of its beaconing children, whichever arrives first;
- If required, beaconing sibling nodes should also be considered.

Otherwise, the node should wait for next superframe and restart CSMA-CA procedure.

Improper default status of transceiver: The attribute $macRxOnWhenIdle$ in MPIB determines whether the receiver should be turned on when the device is idle. In non-beacon enabled mode, devices can not go to sleep. Thus, the default value of $macRxOnWhenIdle$, *false*, does not make sense in non-beacon enabled mode; it should be changed to *true*.

4.7 Conclusions

At its heart, the new IEEE 802.15.4 standard, which is designed for low rate wireless personal area networks (LR-WPANs), is an enabling standard. It brings to light a host of new applications as well as changes many other existing applications. It is the first standard to allow simple sensors and actuators to share a single standardized wireless platform.

To evaluate the general performance of this new standard, we develop an NS2 simulator, which covers all the 802.15.4 PHY and MAC primitives, and carry out five sets of experiments, that is, experiments of: (1) comparing the performance between 802.15.4 and 802.11; (2) association and tree formation study; (3) orphaning and coordinator relocation investigation; (4) examination of unslotted CSMA-CA and slotted CSMA-CA behaviors; and (5) comparing three different data transmissions, namely, direct, indirect and guaranteed time slot (GTS) data transmissions. Detailed experimental results are presented, and analyses and discussions are given.

In non-beacon enabled mode and for low rate applications (traffic load $\leq$ one packet per second), the packet delivery ratio of 802.15.4 is similar to that of 802.11. However, 802.15.4 shows clear advantage over 802.11 regarding control overhead and transaction latency. The experimental results endorse the non RTS/CTS CSMA-CA approach.

Association and tree formation in 802.15.4 proceed smoothly in both

beacon enabled mode and non beacon enabled mode, which implies 802.15.4 possesses a good self-configuration feature and is able to shape up efficiently without human intervention. The orphaning and coordinator relocation (recovery from orphaning) mechanism provides for a device a chance of self-healing from disruptions. The orphaning recovery probability is about 30% for the worst case and about 89% for the best case in our experiments. Notwithstanding, the chance that an orphaned device is completely recovered, that is, it recovers each time it is orphaned, is very low.

For the lack of RTS/CTS, 802.15.4 is expected to suffer from hidden terminal problems. Our experiment results match this expectation. But for low data rates up to one packet per second, the performance degradation is minor. The default CSMA-CA backoff period in 802.15.4 is too short, which leads to frequent repeated collisions. Superframes with low beacon orders can also lower the slotted CSMA-CA backoff efficiency and lead to high collision probability at the beginnings of superframes.

Our study shows that 802.15.4 is an energy-efficient standard favoring low data rate and low power consumption applications. GTS data transmission is an expensive approach for low data rate applications, as can be seen from our experimental results.

Chapter 5

Scheduling and Midware

Chapter 5

Scheduling and Midware

Two major problems faced by wireless medium access are the hidden terminal (HT) and exposed terminal (ET) problems. HT problem results in collisions and ET causes unnecessary delay. HT problem is more serious to most applications, as packets may be dropped due to collisions, which reduces the network throughput. Wi-Fi uses request-to-send (RTS) and clear-to-send (CTS) scheme to cope with HT/ET problems. Since RTS/CTS control messages themselves still suffer from ET/HT problems, the possible failure of RTS/CTS transmissions means ET/HT problems can not be completely eliminated by the RTS/CTS scheme. And mobility can substantially reduce the reliability of the RTS/CTS scheme. Another shortcoming of the RTS/CTS scheme is that it can only handle unicast communications. The RTS/CTS scheme is not adopted in IEEE 802.15.4 for the consideration of low data rate, low cost, and low power consumption. This, however, limits the network throughput. Two schemes are proposed in this chapter to handle the HT problem in wireless mesh PANs (WMPANs): (1) receiver oriented time division multiple access (TDMA), which can be used to remove most collisions; (2) medium access scheduling midware, which can be employed to eliminate all collisions.

5.1 Introduction

Two major problems faced by wireless medium access are the hidden terminal (HT) and exposed terminal (ET) problems. HT problem results in collisions and ET problem causes unnecessary delay. HT problem is more

serious to most applications, as packets may be dropped due to collisions, which reduces the network throughput. Collisions caused by HT problem can be classified into two types. One is that a collision happens at the common destination of two or more packets. The other is that a collision happens at a node that is the destination of one of the packets involved in the collision. Here we call the first type of collision *primary collision* and the second type of collision *secondary collision*. All packets are destroyed in a primary collision, while only packet(s) destined for the node where collision happens is destroyed in a secondary collision. Since most collisions happen between two packets, a secondary collision has a good chance to be resolved by one retransmission (if no other transmission happens during the resolution period, the collision will certainly be resolved by one retransmission). In general, a primary collision needs more retransmissions to resolve. In some networks using short backoff (such as WMPANs), repeated primary collisions are likely to happen and packets has a high probability to be dropped.

Wi-Fi uses request-to-send (RTS) and clear-to-send (CTS) scheme to cope with HT/ET problems. Since RTS/CTS control messages themselves still suffer from ET/HT problems, the possible failure of RTS/CTS transmissions means ET/HT problems can not be completely eliminated by the RTS/CTS scheme. And mobility can substantially reduce the reliability of the RTS/CTS scheme. Another shortcoming of the RTS/CTS scheme is that

it can only handle unicast communications. The RTS/CTS scheme is not adopted in IEEE 802.15.4 for the consideration of low data rate, low cost, and low power consumption. This, however, limits the network throughput. Two schemes are proposed in this chapter to handle the HT problem in WMPANs: (1) receiver oriented time division multiple access (TDMA), which can be used to remove all primary collisions; (2) medium access scheduling midware, which can be employed to eliminate both primary and secondary collisions.

5.2 Receiver Oriented TDMA

The simple receiver oriented time division multiple access (TDMA) scheme presented here can be used to eliminate all primary collisions. While it is possible to eliminate both primary and secondary collisions using more sophisticated TDMA schemes, nodes will suffer more serious ET problem as more time slots are generally needed. As noted above, secondary collisions can be resolved using one retransmission in most cases; the network throughput by using the simple receiver oriented TDMA (ROT) scheme is expected to be close to that by using sophisticated TDMA schemes. Transmission latency caused by retransmissions in ROT is (at least partially) compensated by that less time slots are needed compared with other sophisticate TDMA schemes. Compared with a collision-free TDMA scheme, retrans-

missions in ROT will consume more network resource such as frequency bandwidth and energy. But fortunately, a node does not need to retransmit twice for most collisions, which may prove to be a reasonable cost paid for the simplicity of ROT. Our simulation results show that this simple scheme is very efficient for those networks such as WMPANs where the primary collision problem is serious.

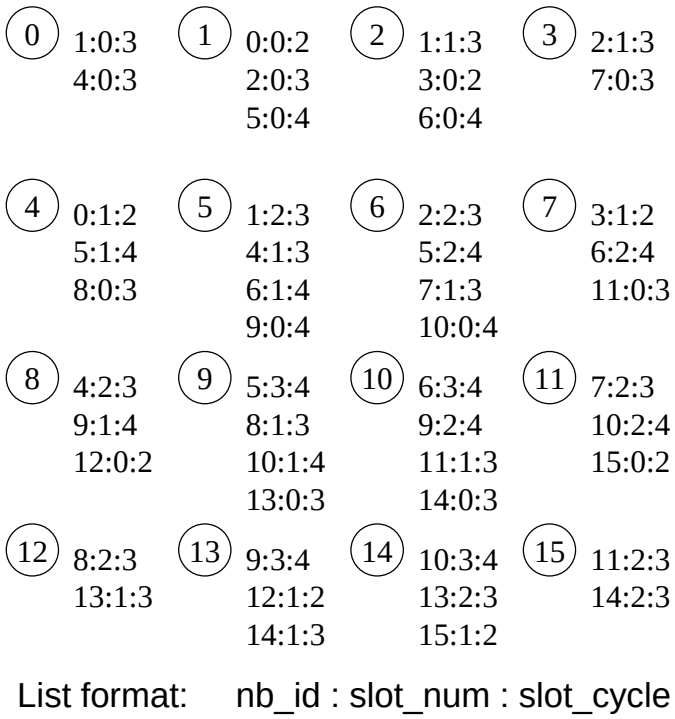


Figure 5.1: An Example of Receiver Oriented TDMA

In ROT, two-hop neighbor information is exchanged among nodes. To do this, each node transmits one-hop *Hello* messages, in which information of all one-hop neighbors is included. When a node receives a Hello message from a one-hop neighbor, it adds the neighbor into its neighbor list and also calculates the time slot and slot cycle it should use to transmit packets to this

neighbor. A simple way to calculate the time slot is to sort identifiers (IDs) (e.g., addresses) of all the neighbors of the neighbor where the Hello message comes in. Based on the order its ID appears in the sorted ID list, the node knows its time slot. For example, in Figure 5.1, node 5 has four one-hop neighbors (1, 4, 6, 9). If include each neighbor's neighbors, the node has a view of two-hop neighbors (1 [0, 2, **5**], 4 [0, **5**, 8], 6 [2, **5**, 7, 10], 9 [**5**, 8, 10, 13]). Thus, node 5 has a time slot table (TST), (1:2:3/4:1:3/6:1:4/9:0:4), each entry of which is in the format of *neighbor_id*:*slot_number*:*slot_cycle*. The TST of node 5 tells that node 5 should use slot 2 (the third slot – slots are numbered from 0) (modulo slot cycle 3), 1 (modulo slot cycle 3), 1 (modulo slot cycle 4), and 0 (modulo slot cycle 4) to transmit packets to neighbors 1, 4, 6, and 9 respectively. Each time a node receives a Hello message, it will check if the TST needs to be updated. Synchronization is needed in ROT, as in any other TDMA scheme. But as we have noticed, ROT is a fully distributed receiver oriented scheme, which means no network-wide synchronization is needed. So the synchronization problem is simply reduced to that a node should know the clock of each its neighbor. Therefore, the synchronization problem can be easily solved by including the clock information in the Hello message. To compensate for clock inaccuracy and clock drift, some small guard time duration (GTD) can be added into each time slot and re-synchronization should be performed before the clock drift exceeds the GTD. Note that the synchronization clock, slot number, and slot

cycle used for one neighbor is independent to those for other neighbors.

While ROT eliminates all primary collisions, it suffers from secondary collisions. This is partially compensated by that ROT has a smaller time slot cycle (TSC) (equivalently, less serious ET problem and less delay) compared with a more sophisticated TDMA scheme. To see this, let us compare ROT with any TDMA scheme that is able to eliminate both primary and secondary collisions. To eliminate both primary and secondary collisions, a node can not share a time slot with any neighbor that is within two hops away. Based on this, we can calculate the time slot reuse distance (RD) (in terms of number of nodes, or equivalently in terms of area) for the network topology given in Figure 5.1. To reuse a time slot, two nodes must be three hops away. There are two types of neighbors that are three hops away. One is that the two nodes locate three hops away horizontally or vertically; the other is that the two nodes locate at the two opposite corners of a 1×2 rectangular area. The first case has a time slot RD 9 and the second case has a time slot RD $9/2 = 4.5$. Since each node can have the same number of three-hop neighbors for each case, the average RD is $(9 + 4.5)/2 = 6.75$. This means that at least 7 time slots are needed. This value is larger than the total number of time slots needed in ROT, which is 4 (see Figure 5.1).

When traffic is not heavy, ROT can be combined with some existing medium access schemes such as carrier sense multiple access with collision avoidance (CSMA-CA). For example, CSMA-CA is used for the first trans-

mission of a packet and ROT is used for retransmissions only. In this way, CSMA-CA helps to reduce the first transmission delay that is otherwise incurred by ROT. And ROT helps to efficiently resolve collisions.

5.3 Medium Access Scheduling Midware

5.3.1 Overview

TDMA schemes are in general free of HT problem. Nevertheless, when traffic load is light, TDMA results in unnecessary delay, which can be viewed as a special ET problem. The medium access scheduling midware (MASM) approach proposed here tries to eliminate HT problem and, at the same time, minimize the effect of ET problem. MASM sits on top of the MAC sublayer and works as a Plug and Play (PnP) midware (i.e., a shim sublayer). As a midware, it goes with different wireless networks and requires no or minimal modifications to existing network protocols such as MAC and routing protocols. It can be used for various purposes (for example, data transmission, beacon scheduling, sleeping scheduling, etc.). Following summarizes the main features of MASM:

1. Each node is assigned one or more time slots that are distinct from those of its neighbors within two hops, thus eliminating HT problems for both unicast and broadcast communications.
2. Time slot assignment is done in distributed fashion, and priority can

be applied during this procedure (a node with a high priority can select slot(s) first and can require more time slots than other nodes).

3. Use multi-level scheduling (with or without using multiple channels) to minimize the effect of ET problem. Different levels can use different time slot durations. For example, multiple mini slots can be scheduled within one reserved common time slot of another level. Also different levels can use different frequency channels, if available. Multi-level approach also favors applications that need to support sleeping mode.
4. Without incurring additional delay, large TSC is used to
   (a) simplify time slot assignment;
   (b) handle unevenly distributed network topologies;
   (c) maintain fairness among nodes with the same priority;
   (d) facilitate the optimization of time slot assignment;
   (e) cope with dynamic activities (node joining, node leaving/failure, mobility, and sleeping mode).
5. The self-correcting ability enables a node to recover from the loss of synchronization due to clock drift or other problems. This feature also expedites link/node failure detection and time slot recycling as well as the handling of various dynamic activities mentioned above.

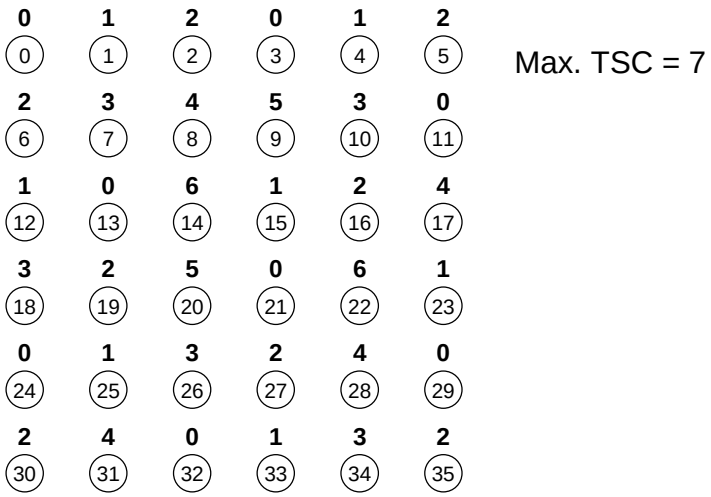
5.3.2 The Basic Scheme

In the basic scheme, two-hop neighbor information, which includes time slot assignment data, is first exchanged among nodes. Then each node determines if it has the highest priority among all its neighbors that are within two hops (referred to as two-hop neighbors here) and that have not chosen a time slot. To do this, a node starts a timer if it finds it has the highest priority. If the node receives a message from any two-hop neighbor with a higher priority than itself before the timer expires, it stops the timer. When the timer expires, it selects the smallest time slot that has not been used by any its two-hop neighbor[1]

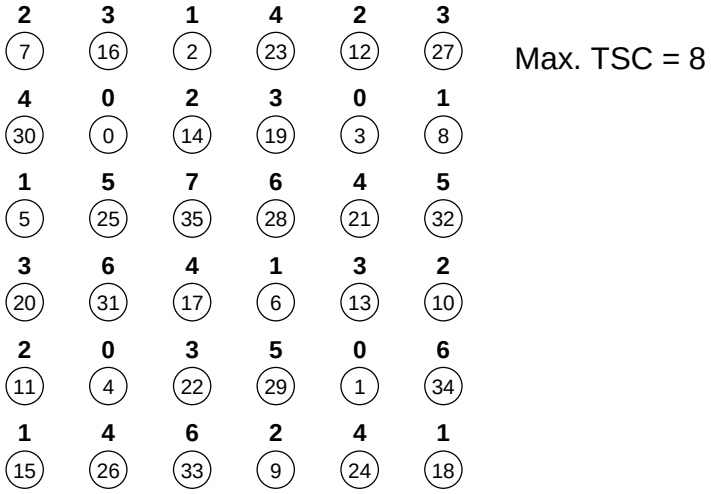
Figure 5.2 shows two examples of time slot assignment using the basic scheme. In both examples, priority is simply determined according to node ID, and a smaller ID enjoys a higher priority. In Figure 5.2 (a), node IDs are orderly distributed; and in Figure 5.2 (b), node IDs are randomly distributed. The time slot assignment results, in terms of total number of time slots used (i.e., time slot cycle), are roughly the same for the two examples. Except node 35 in Figure 5.2 (b) who has a time slot 7, all other nodes in both Figure 5.2 (a) and (b) have a time slot between 0 and 6.

We know, from section 5.2, the time slot reuse distance (RD) for the network topology given in Figure 5.2 is 6.75. This means that at least 7 time slots are needed. In general, RD can be calculated as follows:

[1] An alternative way is that a node never stops a running timer, instead it checks if it still has the highest priority when the timer expires.



(a) Node IDs orderly distributed



(b) Node IDs randomly distributed

Figure 5.2: Examples of Time Slot Assignment Using the Basic Scheme

$$RD = \sum_{i}^{k} p_i d_i \tag{5.1}$$

where

k is the total number of types of three-hop neighbors a node can have;

d_i is the time slot RD of type i;

p_i is the probability that a three-hop neighbor belongs to type i and it can be calculated using degree of connectivity information as follows:

$$p_i = \frac{c_i}{\sum_j^k c_j} \tag{5.2}$$

where

c_i is the degree of connectivity of type i three-hop neighbors;

C is the total degree of connectivity of all three-hop neighbors.

Note that, when nodes are unevenly distributed, RD varies with location. Here "unevenly distributed" means the degree of connectivity is not a constant through the whole network. A physically evenly distributed network is not a strict evenly distributed network regarding RD, because the degree of connectivity around the network boundary is smaller than that in the middle of the network.

The time slot cycle (TSC) in Figure 5.2 (a) is 7, which is the number given by equation (5.1). However, 8 time slots are needed in Figure 5.2 (b). We will show in subsection 5.3.3 that the difference in TSC does not affect the performance of the network. Actually for various reasons, we will intentionally use a TSC which is much larger than the value given by equation (5.1).

5.3.3 The Advanced Scheme

There are some practical problems that are not addressed by the basic scheme presented above. In this subsection we study those problems and their solutions in detail. As a result, a more advanced scheme is proposed.

5.3.3.1 Time slot assignment and time slot cycle

To determine when to transmit a packet, a node needs to know its time slot as well as the TSC. Obviously, all nodes should use the same TSC, even if they have different TSCs within the scope of two-hop neighbors. This is simply because that a node normally is the two-hop neighbor of multiple nodes, and therefore a time slot (e.g., slot 3) will be calculated differently by different nodes if they use different TSCs (e.g., $n \times TSC + 3$ gives different values if different TSCs are used). From above discussion, it is clear that each node needs to know the maximum TSC (*maxTSC*) that has been used in the network. However, the *maxTSC* depends on the node distribution and normally it is difficult to know before the time slot assignment is done. For example, although both have the same physical topology, the *maxTSC* is different in Figure 5.2 (a) and (b). One intuitive solution to let each node know the *maxTSC* is to exchange local TSC information after the time slot assignment is done. But this incurs additional overhead and latency. We will discuss this later and show that a better solution is available.

Another problem revealed by the fact that the *maxTSC* is different in

Figure 5.2 (a) and (b) is that the *maxTSC* is not uniquely determined by the physical topology. The order by which nodes choose time slots makes a difference to the time slot assignment, including the *maxTSC*. From equation (5.1), we know the minimum slots needed for a grid topology is 6.75. Since a node has 12 two-hop neighbors for a grid topology, theoretically it may have a local TSC as large as 13 in the worst case. This means all nodes may need to use a TSC of 13 even if most of them have a local TSC of 7. One may argue that the worst case is unlikely to happen. Even so, the problem persists as it is likely that node density of the network is not uniform. If a small part of the network has a high node density, then the whole network has to use a large TSC. If the *maxTSC* used in the network is higher than a node's local TSC, then some slots are wasted in the basic scheme. For example, the *maxTSC* in Figure 5.2 (a) is 7, but node 0 has a local TSC of 4. So if only one slot is assigned to each node, some slots are wasted.

To fully utilize all available time slots, some nodes will be assigned multiple time slots. Figure 5.3 shows the final time slot assignment result for the topology given in Figure 5.2 (b). After the first round of time slot assignment, the *maxTSC* is determined and announced to all nodes. Then the second round of time slot assignment begins. Different from the first round of time slot assignment in which each node needs to select the smallest possible slot, from second round on, a node may face three situations: it can not find any unused slot; it finds some unused slot(s), but decides not to select

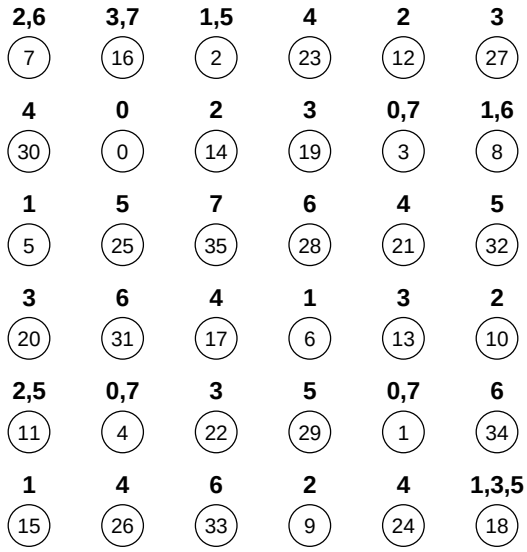


Figure 5.3: Full Utilization of Time Slots

any slot; it finds some unused slot(s) and selects one slot. In any case, the node needs to notify other nodes so that those nodes with lower priority can continue their time slot selection. Note that a node does not need to select the smallest available slot from second round on. And, for fairness reason, a node should not try to occupy all the available slots (in above example, a node is not allowed to select more than one slot during each round). Multiple rounds may be needed before all slots are assigned.

In general a node in a sparse area will be assigned more time slots than a node in a dense area. This will result in a smaller effective TSC for those nodes located in a sparse area. Apparently, a node's effective TSC closely reflects its local TSC. This is natural, as a node located in a sparse area should not be affected too much by another node located in a remote dense area. This result also means what *maxTSC* is used is not critical

in terms of time slot utilization. For example, if we triple the *maxTSC* from 3 to 9, then the original time slot assignment, for example (node 0:slot 0/node 1:slot 1/node 2:slot 2) or in short (0:0/1:1/2:2), will become (0:0/1:1/2:2/0:3/1:4/2:5/0:6/1:7/2:8), which is virtually the same as before. Based on this fact, we propose to use a *maxTSC* that is much larger than otherwise needed. The motivation is that:

1. A predetermined *maxTSC* precludes the need for exchanging local TSC information to get the *maxTSC* after the first round of time slot assignment. As such, both overhead and delay can be reduced. When *maxTSC* is predetermined, a node does not need to select the smallest possible time slot in the first round of time slot assignment. It can select any available slot. This facilitates the optimization of time slot assignment (discussed later).
2. A large enough *maxTSC* guarantees there are enough time slots to be assigned even for a very irregular network topology (e.g., the node density is very high in some area). Yet this large *maxTSC* will not introduce additional delay as can be seen from the example of tripling the *maxTSC*.
3. A small *maxTSC* leads to unfairness in time slot assignment. This can be seen from Figure 5.3. To understand this, let us compare the different results of 7 slots allocated to 6 nodes and 70 slots allocated to 6 nodes. Apparently, we can do much better in the second case in

terms of fairness.

4. With a large *maxTSC*, dynamic activities such as node joining, node leaving/ failure, mobility, and sleeping mode can be better handled. All those activities require the adjustment of time slot assignment. A large *maxTSC* not only makes such adjustment possible, but also easier and smoother.

5.3.3.2 Acknowledgment issue

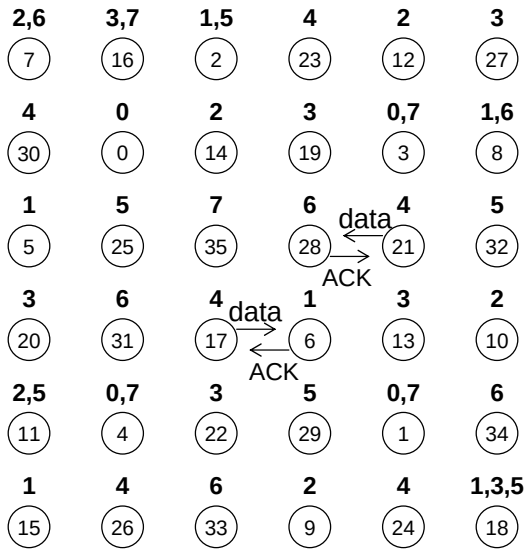


Figure 5.4: Acknowledgment Issue

When an immediate acknowledgment (ACK) is required for a successful data transmission, the scheme presented in subsection 5.3.3.1 may fail. For example, in Figure 5.4 node 17 and node 21 both own time slot 4 and they are allowed to transmit data simultaneously. Since they are three hops away, data packets from these two nodes will not collide with each other. The

corresponding ACK packets, although transmitted from two neighboring nodes, will not collide with each other either, as an ACK packet will not be able to reach the destination of another ACK packet. The only problem is the collision between a data packet and an ACK packet. In principle, the problem can be solved by making data transmission and ACK transmission non-overlapping (not necessary in time). This is not a problem for a new MAC protocol. However, for an existing MAC protocol, it could be difficult to find a satisfactory solution for the problem without modifying the MAC protocol.

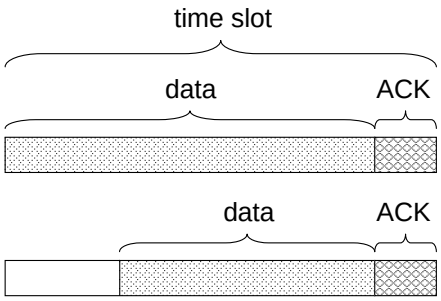


Figure 5.5: Non-overlapping Data Transmission and ACK Transmission within a Time Slot

Most MAC schemes do not allow the ACK to be delayed, that is, the ACK should be sent within a certain short period after a data packet is received. This makes it impossible to transmit the data and its corresponding ACK in separate time slots. If not using different frequency channels or different spreading codes (as in code division multiple access) for the data transmission and the corresponding ACK transmission, then we may have to choose one of the following two solutions. One is to assign distinct time

slots within three-hop neighbors instead of two-hop neighbors. The other is to separate a time slot into two non-overlapping parts, one for data transmission and the other for ACK transmission (Figure 5.5). The first solution always works, but may significantly reduce the bandwidth efficiency. The second solution is much better, but may not work with all existing MAC protocols.

As shown in Figure 5.5, the whole time slot is divided into two parts: one for the transmission of data, the other for the transmission of ACK. Since an ACK needs to be transmitted within a short period after a data packet is received, the transmission of the data should be scheduled in such a way that it ends at the boundary of the two parts used for transmitting data and ACK. One difficulty is that most MAC protocols employ some random backoff scheme for the purpose of medium access, which means an upper layer will not be able to accurately schedule the data transmission. Fortunately some MAC protocols allow configuring the backoff parameters. For example, in IEEE 802.15.4 [5], an upper layer can set the minimum backoff exponent, *macMinBE*, in the MAC PAN information base (MPIB) to 0, thus leading to the first backoff equal to $(2^{macMinBE} - 1) \times aUnitBackoffPeriod = 0$.

5.3.3.3 Multi-level time slot assignment

One drawback of any TDMA-based scheme is that time slots are assigned regardless of the actual traffic need. As a consequence, some nodes may

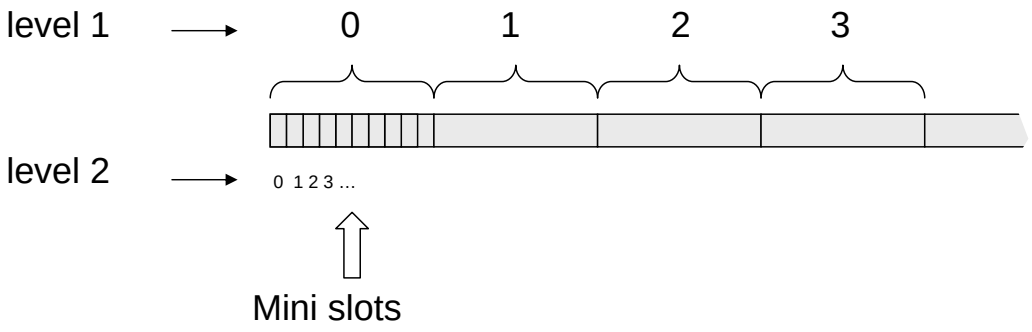


Figure 5.6: Multi-level Time Slot Assignment

have more time slots than they need while others are running out of time slots. One way to mitigate this is to use multi-level time slot assignment. Figure 5.6 illustrates a two-level time slot assignment example. Slot 0 of level 1 is reserved for all nodes. This reserved slot is further divided into several level 2 mini slots, which are assigned as usual. These mini slots can be used by nodes to borrow slots from one another. For example, a node can return its slot(s) to a slot pool (SP) if it does not have data to transmit during a certain period. The SP lists all the available time slots together with their original owners and cycles they will be available. A node can borrow slots from the SP. The rule of borrowing slots is like that of slot assignment. That is, a node can borrow a slot listed in the SP if no two-hop neighbor is using a slot with the same number. How many level 1 slots should be reserved during each TSC depends on the specific application. This approach also favors applications that need to support sleeping mode. If a node finds out (e.g., using level 2 communications) that it has no data to transmit and there

is no data to be transmitted from any neighbor to it for some duration in the future, it can go to sleep.

Instead of reserving slot(s) of one level for the use of another level, another approach is to use multiple channels, one for each level. In above example, level 2 can use a channel that is different from that of level 1; and the bandwidth of the channel used by level 2 can be much smaller than that used by level 1.

5.3.3.4 Optimization of time slot assignment

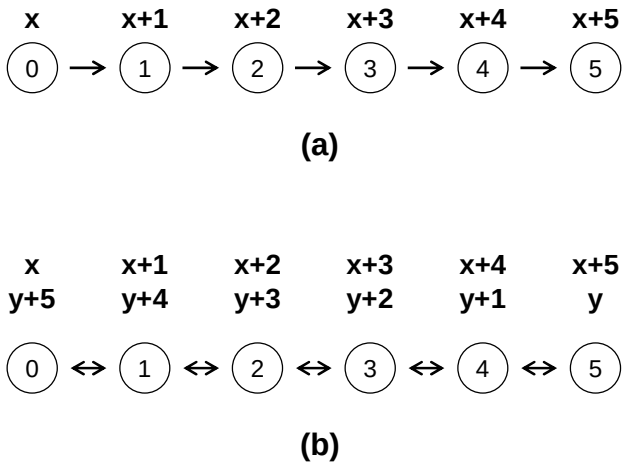


Figure 5.7: Optimization of Time Slot Assignment

If time slots are allocated in such a way that each node along a traffic flow from the source to the destination owns a slot that is slightly larger than that owned by the previous hop (an example shown in Figure 5.7 (a)), then the end-to-end packet delivery latency can be minimized. For bi-directional traffic, a node can choose some optimal slots for one direction and some

others for the other direction if multiple slots can be chosen (an example shown in Figure 5.7 (b)). If only one slot can be selected and the traffic rates of two-directions are similar, then a distance of half TSC between a node's slot and the slot of its neighbor is reasonable. For instance, if the TSC is 8, then the slots of two neighboring nodes is better to have a distance of 4 slots.

In a mesh network where bi-directional traffic may flow between one node and any of its neighbors, it is difficult to find a slot that is optimal for all traffic flows. In this case, average result should be used and tradeoff is often needed. The situation of pure tree routing (e.g., cluster tree routing [29]) is a little better, as traffic can only go up or down along the tree. In this case, each node can select two time slots, one optimal for upward traffic and the other optimal for downward traffic. If each node can only select one slot, then the above half TSC distance rule can be applied. An alternative way is to use two channels, one for upward traffic and the other for downward traffic.

5.3.3.5 Synchronization and self-correcting/adaptation

For synchronization purpose, one node's clock should be used as the "standard" clock. Practically a powerful node such as the base station in a Wi-Fi or the PAN coordinator in a wireless personal area network (WPAN) is chosen for this purpose, but it is free to select any other node. The node

with the "standard" clock first broadcasts a synchronization message, in which its clock information is included. All neighbors are then synchronized with this node. Next, each neighbor tries to synchronize its neighbors by broadcasting another synchronization message, in which its synchronized clock information is included. This procedure continues until all nodes are synchronized. Due to the propagation delay and clock drift, some small GTD may be added to each time slot.

To handle clock drift problem, a re-synchronization message is periodically broadcast from the node with the "standard" clock. This message spreads through the network and is rebroadcast during each relaying node's time slot. A node may not be able to receive and relay this broadcast message due to the loss of synchronization or other problems. To recover from loss of synchronization, each node keeps a timer, which expires after a certain number of re-synchronization broadcast periods. Each time a re-synchronization message is received, the timer is reset. When the timer expires, the node in problem should stop any transmission until it receives a re-synchronization message. To recover from loss of synchronization quickly, a node can also try to overhear transmissions of its neighbors and then synchronize with them. Normally it takes much less time to overhear transmissions than wait for a re-synchronization message.

The self-correcting scheme used for synchronization recovery can also be used to handle link/node failures. If a node finds that one of its neighbors

has not been relaying re-synchronization messages for certain duration, it concludes the link to the neighbor or the neighbor itself has failed and it should update all its neighbors of the failure. If a failure is reported by more than one neighbor, the failure is likely to be node failure. Any node detects a failure and/or receives a failure report will update its two-hop view and check if any time slot can be recycled.

The proposed scheme can also adapt to other dynamic activities such as node leaving, node joining, mobility, and sleeping mode. Node leaving can be handled similarly as node failure. For node joining, a joining node first asks its two-hop neighbors to send their slot information to it. Based on this, it can find out if any unused time slot is available (often true if the joining node is at the boundary of the network). If there is no unused time slot or the number of unused time slots is not enough, the joining node may request an adjustment or re-assignment of time slots among its two-hop neighbors. If any node's slot assignment is changed, it should update all its two-hop neighbors so that any reclaimed time slot by the change can be reused. Note that each node has a different two-hop view. So a reclaimed time slot may be reassigned to one neighbor, but not to another. If two-hop neighbor information is updated promptly, then time slot assignment can be dynamically adjusted to reflect the effect of mobility. Another related application scenario that requires dynamic adjustment of time slots is the dense wireless sensor network where sensors need to go to sleep or may

die due to energy depletion. For example, 5 times of sensors than needed at any time may be deployed in an area. Then only 1/5 of the sensors are turned into normal state and all other sensors go to sleep. At the time the energy of the sensors in normal state is about to be depleted, another 1/5 of the sensors are turned into normal state. This is a way to prolong the life of wireless sensor networks.

Note that multi-level time slot assignment (with or without using multiple channels) discussed above can be used to further facilitate the handling of all the dynamic activities addressed in this subsection.

Chapter 6

Parallel and Accumulative Reception using CDMA

Chapter 6

Parallel and Accumulative Reception using CDMA

This chapter presents a new medium access control (MAC) scheme, which targets high rate wireless personal area networks (HR-WPANs) based on IEEE 802.15.3a (a.k.a. ultra wideband (UWB)). The new scheme exploits the code division multiple access (CDMA) technology to provision many-to-one simultaneous wireless communication service without utilizing any time division multiple access (TDMA) scheme. This, combined with existing one-to-one (i.e., unicast) and one-to-many (i.e., multicast or broadcast) wireless communication services, brings into life a new communication paradigm, that is, mesh communications. The new scheme also supports accumulative receptions, namely, a receiver can buffer partially damaged (re-)transmitted frames and add those frames efficiently to form an error-free frame. What sets the new scheme apart from other CDMA-based medium access schemes is that neither spreading code assignment nor power control, both of which are the essential research topics of other CDMA-based schemes, are needed. Our simulation results show that the new scheme outperforms IEEE 802.11 in both non-mobile and mobile environments.

6.1 Introduction

6.1.1 Background

Code division multiple access (CDMA) allows multiple users to simultaneously access the same bandwidth without significantly interfering with one another. This is done by assigning different spreading codes (a.k.a. chip codes) to different transmitter-receiver pairs that are not spatial separated. As a result, CDMA achieves a much higher throughput. For example, Gilhousen *et al.* showed that CDMA can provide up to six times the capacity of time division multiple access (TDMA) or frequency division multiple access (FDMA) in cellular systems [53]. Some of the important features of CDMA are inherent interference rejection, resistance to multipath fading, and graceful signal degradation.

In CDMA, a digital signal is first spread at the transmitter side, that is, multiplied by a chip code. The spread signal can then be despread through cross-correlation with the same chip code at the receiver side. When a spread signal is cross-correlated with a chip code other than the one used for spreading, the result is either a zero or a very small amount of wideband noise, depending on whether these two chip codes are orthogonal or quasi-orthogonal. While at first sight orthogonal chip codes appear very attractive due to their immune to multiple access interference (MAI) [54, 55], they are seldom used in practice. The main problem with orthogonal chip

codes is that strict synchronization is required between nodes, which is either impossible or too expensive in most wireless environments. For most existing orthogonal chip codes, the cross-correlation is non-trivial when they are shifted (i.e., not well synchronized). In light of this, we will only consider quasi-orthogonal chip codes here, specifically, the pseudorandom noise (PN) codes.

CDMA has been successfully implemented in many cellular systems, including third-generation (3G) systems [56], whereas its applications to packet radio networks (PRNs) largely remain in research stage. To date the difficulties of applying CDMA to those networks have come from two main factors. First, an efficient PN code assignment scheme is needed for supporting concurrent transmissions. And second, the notorious near-far problem caused by non-zero cross-correlation between different PN codes necessitates the use of power control, interference cancellation, and/or access control. The goal of code assignment is to eliminate or reduce *primary collisions*. A collision is said to be *primary* if it involves two or more transmissions that are spread using the same code. Besides primary collisions, another type of collisions, called *secondary collisions,* can happen even if distinct codes are assigned to different transmitter-receiver pairs. Secondary collisions are caused by MAI, which results from the non-zero cross-correlation between different PN codes. They can be caused by too many undesired transmissions in a receiver's vicinity, or even by a single

undesired transmission that is much closer to the receiver than that from the transmitter (i.e., a near-far problem). Power control can mitigate the near-far problem, but can not get rid of it altogether in PRNs, especially in mobile ones. Communications in a PRN generally follow a peer-to-peer pattern and, as a consequence, one power control requirement may conflict with another. Thereby, access control is generally still needed in those networks. Some research work has been done in an effort to solve the primary collision and secondary collision problems in PRNs (see section 6.2).

6.1.2 Motivation

Communications in a PRN are by nature many-to-one and one-to-many. This feature, however, is neither reflected in non-concurrent transmission schemes nor in concurrent transmission schemes. Although both IEEE 802.11 [20] and 802.15.4 [5] standards use spread spectrum (SS) techniques, only a single chip code is used for suppressing interferences in a heterogeneous environment rather than supporting concurrent transmissions in the vicinity of a receiver. Other CDMA-based MAC protocols (see subsection 6.2) support concurrent transmissions, but all follow a one-to-one transmission pattern. To improve the performance of PRNs, we propose a new scheme in this chapter. Besides normal concurrent transmissions, the new scheme also allows a node to receive data from multiple transmitters simultaneously.

For most MAC protocols, partially damaged frames are simply dropped. This implies that a completely error-free (re-)transmission is required to transmit a frame. Nevertheless, due to the variety of retransmissions, it is possible to construct a completely error-free frame from several partial damaged frames. Our new scheme supports accumulative receptions, namely, a receiver can buffer partially damaged frames and add them efficiently to construct a completely error-free frame.

Another common practice in most MAC protocols is that no feedback is sent from the receiver to the transmitter if a frame is damaged. It's the transmitter's responsibility to retransmit the frame. This practice can not be justified, considering the fact that collisions account for most transmission failures and a receiver is usually in a better position to resolve a collision than transmitters. On one hand, the receiver can access all the transmitters, but transmitters may hide from one another. On the other hand, the receiver may still be able to extract some information from partially damaged frames, which can be used not only to support accumulative receptions, but also to help prevent collisions in ensuing retransmissions. In our scheme, a receiver actively helps transmitters to recover damaged frames, by means of accumulative receptions. Accumulative receptions can be either active or passive. In active accumulative receptions, a receiver tries to send back a negative acknowledgment (NACK) to the transmitter in case of a transmission failure, indicating what part of the frame is damaged so that the

transmitter only needs to retransmit that part. In passive accumulative receptions, no information about partial damage is sent back to the transmitter and a damaged frame is always retransmitted in full. In general, active accumulative receptions favor transmissions using large physical protocol data unit (PPDU), whereas passive accumulative receptions are more suitable for transmissions using small PPDU.

For its support of parallel and accumulative receptions, our new scheme is referred to as PAR-CDMA in what follows.

6.1.3 Features of the Proposed Scheme

This subsection summarizes some features of PAR-CDMA. Detailed discussions are left to section 6.3.

- Parallel receptions and accumulative receptions in PRNs are supported.
- For each node, multiple sets of shift registers are used for the purpose of parallel receptions. And each set of shift registers are in one of the three states at any time: stand-by, reception, and sleeping. For description convenience, we call each set of shift registers a soft channel.
- Based on the concept of code-time space, no code assignment is needed. Although only one universal code is used throughout the network, the probability of primary collisions is negligible.
- For simplicity, no request-to-send (RTS) and clear-to-send (CTS) control frames are used. And no power control is performed. The sec-

ondary collisions are prevented or resolved via efficient overhearing and a transmission jitter.

- Train of synchronization sequences are used for reserving soft channels. This allows for differential treatment of data frames and control frames such as acknowledgments (ACKs), and the support of quality of service (QoS).
- Besides ACKs, NACKs can optionally be sent from receivers to transmitters. Both ACKs and NACKs can be accumulated and transmitted in a batch, via either unicast or broadcast.
- An m-value ($m > 2$) decision circuit is used in place of a common 2-value decision circuit for despreading, which makes it possible to quantify the reliability of the reception of a certain data bit rather than merely give out the binary result "1" or "0". As a result, a receiver will be able to estimate what part of a damaged frame is corrupted. This is the basis of accumulative receptions.

The rest of the chapter is organized as follows. Some related work is given in section 6.2. The PAR-CDMA scheme is described in section 6.3. Performance of PAR-CDMA is evaluated and compared with that of IEEE 802.11 in section 6.4. Finally, conclusion and future work are given in section 6.5.

6.2 Related Work

To apply CDMA technologies to PRNs, two problems have to be addressed: primary collisions and secondary collisions. The first problem is basically a chip code assignment problem; the second problem is an MAI problem. The near-far problem is in nature an MAI problem, but the MAI effect is exacerbated by the fact that the undesired source is much closer to the receiver than the desired source.

L. Hu [58] proposed several two-phase distributed code assignment algorithms for CDMA PRNs. In the first phase, a random code is assigned to each node or to each link to make the network operational immediately. Then in the second phase, the algorithms gradually correct the codes to improve the throughput performance. These two-phase algorithms minimize the time complexity in the first phase and minimize the number of control packets needed to be exchanged in the second phase.

A. Bertossi *et al.* [59] studied the problem of minimizing the codes needed to eliminate hidden terminal interferences in a PRN with transmitter-oriented code assignment (TOCA) MAC protocols [60]. Through a graph-theoretic investigation, they showed that this problem is NP-complete. Optimal algorithms for restricted topologies, as well as fast suboptimal centralized and distributed heuristic algorithms were proposed. R. Battiti *et al.* [61] investigated the upper and lower bounds of code assignments for hidden and

primary collision avoidance (HP-CA) and hidden collision avoidance only (H-CA). Optimal assignments for special topologies and heuristics for general topologies were proposed.

A distributed code assignment algorithm for a dynamic, multi-hop wireless radio network was given in [62]. The implementation of the code assignment algorithm as part of the MAC and routing protocols of a multi-hop PRN was discussed. Wieselthier *et al.* [63] proposed the use of small control packets to allocate frequency slots for data packets. It is assumed no feedback is sent back from the receiver to the transmitter. So in case two or more senders try to reserve the same frequency slot, it is solely up to the receiver to decide which of these signals it will monitor. In summary, a receiver will selectively choose one signal (that is, a frequency-hopping pattern or PN-sequence) to monitor. A medium-access and connection-establishment protocol for an ad-hoc quasi-synchronous PRN (QSPNET) was propose in [64]. Transmitted waveforms in the QSPNET are made quasi-synchronous by using a local Global Positioning System (GPS) clock. Channel in QSPNET is divided into data channel and control channel, and both channels are further divided into time slots. In the same slot, up to a certain number of codes are available for use, thus allowing multiple users to share the same slot.

Power control is usually used to solve the MAI problem in PRNs. The power control problem in the context of strong connectivity in static multi-hop broadcast PRNs was studied by Chen *et al.* [65]. The authors addressed

how to adjust the power level used by each radio terminal so that the total amount of power used is minimized and, at the same time, the network is still strongly connected.

Herdtner *et al*. [66] studied a class of distributed asynchronous power control algorithms based on the schemes used in IS-95 inner loop power control. They showed that under certain conditions, this class of algorithms is stable and converges to the optimal power assignment to within controllable errors. Convergence property of the binary feedback power control algorithm is investigated in [67]. The author proved that, in a network adopting this simple power control algorithm, the received signal-to-interference ratio (SIR) of each user falls within a certain specific range that is determined by an SIR target and a power control step-size.

A source coding and modulation technique for reducing MAI as well as for reducing power consumption in Direct Sequence CDMA (DS-CDMA) wireless sensor network systems was proposed in [68]. In this approach, MAI is controlled by changing the number of redundant bits. For example, when number of users increases, the redundant bits are increased, which means the channel coding rate is reduced.

By analogy with CDMA networks, O. Dousse *et al*. [69] introduced some orthogonality factor γ, which can vary from 0 to 1, and investigated the impact of interferences on the connectivity of large-scale ad hoc networks based on TDMA scheme. The value γ quantifies the orthogonality of the

codes used in CDMA, and a value $\gamma = 0$ corresponds to perfect orthogonality. The main conclusion is that long range multi-hop communications are still possible in large scale ad hoc networks if γ is small enough (i.e., highly orthogonal), though not zero.

De *et al.* [70] studied the MAI in wireless CDMA sensor networks with uniformly random distributed nodes. The tradeoff between interference and connectivity was investigated for three competitive deterministic topologies, in terms of link-level and network-level (routing) performance.

A. Muqattash *et al*. [71] presented a CDMA-based power controlled medium access protocol for mobile ad hoc networks (MANETs). The required transmission power for data packets is estimated by overhearing RTS and CTS packets, which are transmitted in a separate control channel. By adjusting the transmission power accordingly, interference-limited simultaneous transmissions are made possible in the neighborhood of a receiving terminal.

Besides power control, topology control is also used to handle the MAI problem. Since transmission power decreases as the signal propagates away from the source, topology control effectively is another way of power control. Some topology control schemes were proposed in [72–74]. Another technology that holds great promise in reducing MAI is interference cancellation. Large amount of research work has been conducted in this area, including successive interference cancellation [75–79], parallel interference

cancellation [80–85], and iterative interference cancellation [86–90]. Approaches that combine power control and interference cancellation have also been proposed [91–94].

6.3 The Proposed Scheme

In this section, we present the PAR-CDMA scheme in detail.

6.3.1 Parallel Receptions

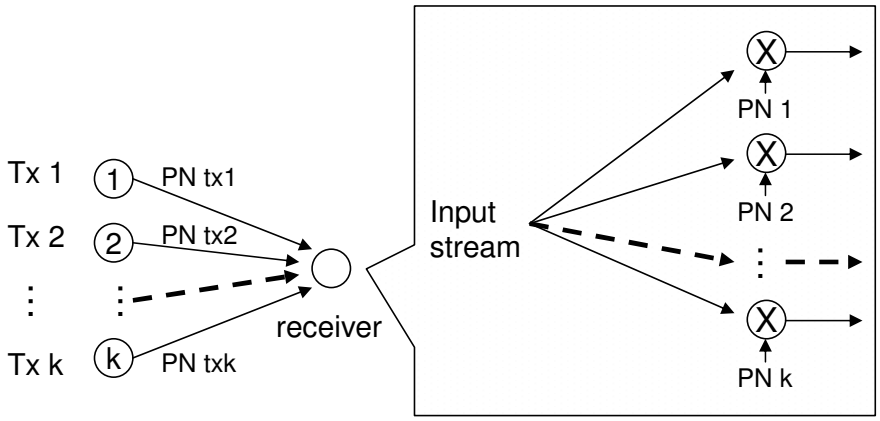


Figure 6.1: Parallel Receptions using Multiple Correlators

As shown in Figure 6.1, by using multiple correlators and distinct PN codes, it is possible for a node to receive data transmitted in a common frequency band from several other nodes simultaneously. We envision that there are multiple soft channels, each corresponding to a correlator using a distinct PN code. By soft, we mean those channels, unlike in FDMA, are not completely orthogonal.

Some issues to be addressed for such a parallel reception scheme are chip code generation and assignment, medium access, channel coding rate, and handling of acknowledgments.

6.3.1.1 Chip code

In CDMA, chip codes are used for spreading and despreading signals, and hence are of great importance. Roughly, there are two types of codes: orthogonal codes and non-orthogonal codes. A typical example of orthogonal codes is the Walsh codes generated by using Hadamard matrix. Orthogonal codes, as we pointed out in subsection 6.1.1, are seldom used in CDMA systems, with the exception of IS-95 standard for cellular CDMA networks [57], in which Walsh codes are used. Of our interest are the so-called pseudorandom noise (PN) codes generated using maximal-length linear shift register sequence. PN codes fall into the type of non-orthogonal codes, but with the following features [52]:

1. The number of +1's differs from the number of -1's by exactly one.
2. Half of the runs of the same sign have length 1, one fourth have length 2, one eighth have length 3, and so forth. Also the number of positive runs equals the number of negative runs.
3. The auto-correlation is two-valued. That is, for an N-bit PN code,

$$C(k) = \sum_{n=1}^{N} a_n a_{n+k} \tag{6.1}$$
$$= \begin{cases} N & (\text{if } k = N,\ 2N,\ \cdots) \\ -1 & (\text{otherwise}) \end{cases}$$

From equation (6.1), we can effectively view a shifted PN code as another code, which is quasi-orthogonal to the original one. For distinction, we use cross-correlation instead of auto-correlation to describe the correlation between a PN code and its shifted version. In practice, a PN code is very long and only partial auto-correlation or cross-correlation is used. For partial cross-correlation, the result is no longer exactly -1, but remains trivial compared with the corresponding partial auto-correlation.

6.3.1.2 PN code assignment and synchronization

To communicate using CDMA, two nodes need to agree on what PN code to use and how to synchronize with each other. For what PN code to use, a node can either randomly select one from a set of predetermined well-known PN codes or select a PN code (not necessary to be well-known) through negotiation with its communication peer. Synchronization can be achieved either via some common clock or, if such common clock is not available, using some synchronization sequence included in each frame header. In the following, we present two different approaches.

A. Approach based on multiple distinct PN codes

Assume a node has k sets of correlators, each controlled by a set of shift registers. Furthermore, all the correlators are physically the same, that is, all sets of shift registers are described by the same irreducible polynomial. If all those sets of shift registers begin with the same initial value, then they will produce the same chip sequence. Effectively we say that all of them use the same PN code. However, if those sets of shift registers begin with different initial values, they will give out different chip sequences, i.e., different PN codes, and one is the shift version of another.

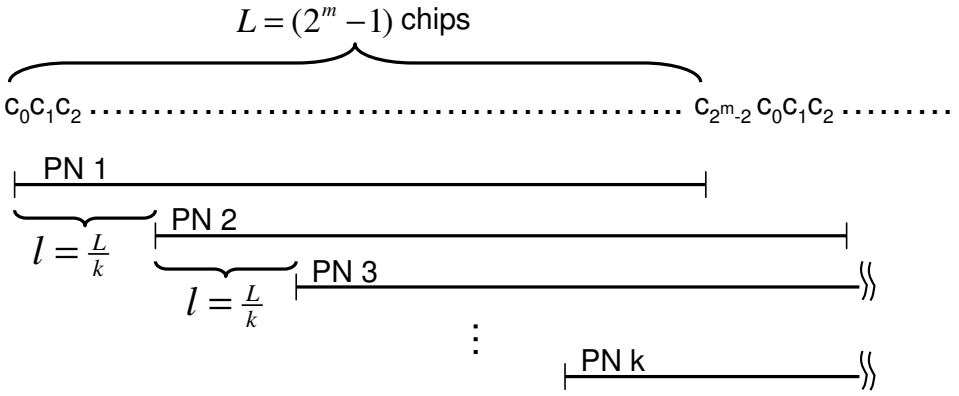


Figure 6.2: Multiple Distinct PN codes

For a set of m-stage shift registers, we can select k PN codes from all the $2^m - 1$ PN codes, for example, with an equal distance $l = (2^m - 1)/k$ between two adjacent PN codes (Figure 6.2). Those fixed PN codes are made well-known to all the nodes. Note that l chips are generally more than enough for spreading a frame, and that the PN sequence is reset each time a frame is transmitted. So different PN codes guarantee the corresponding

frames are spread using non-overlapping chip sequences. In non-mobile wireless environments, a node can assign PN codes to its neighbors through negotiation (e.g., by exchanging Hello messages). This scheme is highly reliable and PN code duplication can be minimized. The drawback is the control overhead resulted from negotiation. In this scheme, it is not necessary to limit the PN codes to those well-known ones or limit the number of PN codes to k. In mobile environments, PN code assignment is no longer efficient due to mobility. So a node may just randomly select a well-known PN code. This scheme does not incur additional control overhead, but PN code duplication is likely to happen. Following is the algorithm we used for selecting a PN code in mobile environments:

```
1:  if (no previous PN code)
2:      randomly select a PN code
3:  elseif (last tx successful)
4:      if (previous PN code is fresh)
5:          continue using previous PN code
6:          previous PN code becomes good
            (i.e., at least succeeded once)
7:      else
8:          r = rand(2)
9:          if r equals 0
10:             continue using previous PN code
                (prev. PN code doesn't become good)
11:         else
12:             randomly select a PN code
13: else //last tx failed
14:     if (previous PN code is not good or not fresh)
15:         randomly select another PN code
```

```
16:     else if (first failure of previous PN code)
17:         continue using previous PN code
18:     else
19:         randomly select another PN code
```

The above algorithm applies to transmissions from one node to all its neighbors, irrespective of the destination. The rationale is that, no matter the transmission is destined for which neighbor, it will affect all the neighbors if omni-antenna is used.

To support broadcast, at least one PN code should be made well-known and be supported by all nodes. If multiple PN codes are used for broadcast, then parallel reception of broadcast frames is also possible.

To increase the reliability of transmissions of ACKs, a soft channel is always reserved if a node is expecting an incoming ACK. The reservation works as follows: When a node first transmits a frame to another node, it will randomly select a PN code, which should be different from all well-known PN codes, and include the code in the outgoing frame. Then the node will tune one of its soft channels to this PN code, waiting for the ACK. The destination will send back the ACK using this PN code. Due to the large number of PN codes ($2^m - 1$), it is extremely unlike that this PN code will collide with another. So ACKs are better protected than data. When sending back the first ACK, the destination can also suggest the source to use another PN code that has been randomly selected by it for later recep-

tions of ACKs. By this way, a destination can notify all the sources to use a single PN code, which will enable the destination to broadcast/multicast ACKs. A broadcast/multicast ACK can include various information, e.g., list of sources and their transmission statuses, partial retransmission scope, current used or available PN codes, etc. Both the unicast PN code (selected by a source) or the broadcast/multicast PN code (selected by a destination) used for transmissions of ACKs can be changed at any time.

B. Approach based on a single PN code

By different PN codes, we mean that different sets of shift registers of a receiver are at different offsets during receptions. Simply assigning different PN codes to different soft channels is insufficient to ensure that they are using different PN codes; those soft channels in general begin their receptions at different points in time and there is a chance that two or more sets of shift registers are at the same offset during receptions. Likewise, assigning a single PN code to all the soft channels does not mean that all sets of shift registers will have the same offset during receptions. The above issue relates to both code and time, and therefore should be addressed in the code-time space. To avoid primary collisions, either multiple non-overlapping PN codes should be used (i.e., code space separation), or multiple receptions with a single PN code should begin at different points in time (i.e., time space separation).

Based on the above, we can use a single PN code for all soft channels. That is, each node uses the same PN code for the transmission of each frame. In this approach, primary collisions due to PN code duplication only happen when two or more soft channels begin their receptions at the same point in time or too close, which is rare in practice. The efficient overhearing scheme described in subsection 6.3.1.3 can further help prevent two or more receptions from happening at the same point in time or too close.

To demonstrate the feasibility of achieving parallel receptions by using a single PN code, we give an implementation example here. Assume a node has 4 soft channels. At any moment, one soft channel, if available, is in the state of *stand-by*, waiting for a new incoming frame. Once an incoming frame is detected, the channel will switch its state from *stand-by* to *reception* and, at the same time, awake another soft channel that is in the state of *sleeping*, if available, to enter *stand-by* state. The period from the moment the first bit of the incoming frame arrives to the moment another soft channel is turned into *stand-by* state is a *blind* period. A second frame arriving within this period will not be detected. This blind period tells what *two or more receptions are too close* means. To reliably determine whether there is an incoming frame, it normally requires the receiver to match the whole synchronization sequence, which could be several to tens of bytes in size. But in order to reduce the blind period, the receiver can make a decision much earlier than that, for example, after the reception of the first byte.

Such a decision is not reliable. However, a wrong decision merely means that the receiver should terminate the reception and make the corresponding soft channel available for later receptions. Note that, if incoming signals are allowed to be buffered, the blind period can be reduced or even eliminated.

The single PN code approach not only precludes the need for code assignment, thus significantly simplifies the protocol, but also results in a better performance, since the primary collisions due to PN code collisions are reduced to a trivial level. Unless otherwise stated, all the discussions are with respect to the single PN code approach from now on.

6.3.1.3 Medium access

Besides PN code assignment, another problem in CDMA-based MAC is the MAI problem, the worst case of which is the near-far problem. One approach to handle MAI is to employ the carrier sense multiple access with collision avoidance (CSMA-CA) scheme, which has been widely adopted in other MAC schemes. But a strict CSMA-CA scheme would mean that no concurrent transmissions will be allowed, which defeats the main objective of the design of CDMA-based PRNs. As a result, most CDMA-based MAC protocols that perform medium access control use a simpler version of CSMA-CA on top of power control. Those schemes first try to support concurrent transmissions through power control. But as we have pointed out, power-control can not always be done in PRNs. In that case, access

control will be necessary.

To simplify the protocol design, the normal CSMA-CA is not used in PAR-CDMA. Instead, an efficient overhearing scheme together with a transmission jitter is used to handle the MAI problem. Among all the soft channels, one channel is reserved[1]. There are two functions of this reserved soft channel. If a node is expecting an incoming ACK, this soft channel will be tuned to receive the ACK (see details below); otherwise this soft channel will be used to overhear nearby ongoing transmissions. For overhearing purpose, it suffices to overhear the frame header only. So, if transmissions of frame headers are not overlapping in time, the corresponding frames have a good chance to be overheard. The overhearing scheme is unique in the sense that it can be done even during normal data receptions, thanks to the use of multiple soft channels.

Reception failures can occur under several situations: Soft channels are used up; the receiver itself is in transmission and; two frames arrive at the receiver too close in time. For the first case, the number of soft channels (a physical parameter) determines the maximum possible number of parallel receptions allowed[2]. However, the maximum allowed number of parallel receptions is also affected by the channel coding rate (a.k.a. spread rate, i.e., the ratio of chip rate to data bit rate) and the near-far effect (see subsec-

[1]This channel does not need to be fixed. A reserved channel can be changed to a normal channel and begin to receive data, provided another channel is available to replace the role of the previous reserved channel.

[2]In our case, one soft channel is reserved and not used for data transmissions. So the number of soft channels is not equal to the number of parallel receptions allowed.

tion 6.3.1.4).

Too many simultaneous transmissions or severe near-far effects can result in a large bit error rate (BER).

Problems caused by transmitting a frame to a node who itself is in transmission or by transmitting multiple frames too close in time are either prevented through overhearing or, if overhearing fails, solved by retransmissions using a jitter. The jitter uses the same backoff algorithm used in IEEE 802.11 [20].

6.3.1.4 Channel coding rate

The number of simultaneous transmissions is relatively small in a PRN compared with that in the uplink of a wireless cellular network. Therefore, a relatively small channel coding rate (i.e., spread ratio) can be used in a PRN and interference cancellation (IC) is not a must-have function. In this subsection, we investigate the impact of the channel coding rate on the maximum allowed number of parallel receptions.

We consider the scenario that K nodes are transmitting simultaneously within the vicinity of another node d over a common additive white Gaussian noise (AWGN) channel, each with a binary phase-shift keying (BPSK) data modulation. Without loss of generality, we consider the decision statistic of node d for the first bit from the first node among the K nodes, which

is

$$\tilde{I}_1 = \int_0^{T_b} \mathcal{A}(t) a_1(t) \cos(\omega_c t) dt \tag{6.2}$$

where $\mathcal{A}(t)$ is the received amplitude profile, i.e.,

$$\mathcal{A}(t) = \sum_{i=1}^{K} A_i a_i(t - \tau_i) b_i(t - \tau_i) \cos(\omega_c t + \varphi_i) + n(t) \tag{6.3}$$

and

$$\begin{aligned}
T_b &= \text{bit period} \\
A_i &= \text{amplitude of } i^{th} \text{ node} \\
a_i(t) &= \text{chip sequence of } i^{th} \text{ node} \\
b_i(t) &= \text{bit sequence of } i^{th} \text{ node} \\
\omega_c &= \text{carrier angular frequency} \\
\tau_i &= \text{time delay of } i^{th} \text{ node relative to the first node} \\
\varphi_i &= \text{phase offset of } i^{th} \text{ node relative to the first node} \\
n(t) &= \text{additive white Guassian noise}
\end{aligned}$$

The PN chip sequence $a_i(t)$ is of the form

$$a_i(t) \;=\; \sum_{j=-\infty}^{\infty} \sum_{k=0}^{M-1} a_{i,\,k}\mathrm{u}\left(\frac{t-(k-jM)T_c}{T_c}\right)$$

where $a_{i,\,k} \in \{-1,\,1\}$, M is the cycle of the PN chip sequence, and $\mathrm{u}(t)$ represents the unit pulse function

$$\mathrm{u}(t) \;=\; \begin{cases} 1 & 0 \le t < 1 \\ 0 & \text{otherwise} \end{cases}$$

Equation (6.2) may be expressed as

$$\tilde{I}_1 \;=\; I_1 + \xi + \eta \tag{6.4}$$

where

$$\begin{aligned} I_1 &= \int_0^{T_b} A_1 a_1^2(t) b_1(t) \cos^2(\omega_c t)dt \\ &= \frac{1}{2}A_1 b_1 \int_0^{T_b} (1+\cos(2\omega_c t))dt \\ &= \frac{1}{2}A_1 b_1 T_b \qquad b_1 \in \{-1,\,1\} \\ \xi &= \int_0^{T_b} \left(\sum_{i=2}^{K} A_i a_i(t-\tau_i) b_i(t-\tau_i)\cos(\omega_c t+\varphi_i)\right) \\ &\quad a_1(t)\cos(\omega_c t)dt \\ \eta &= \int_0^{T_b} n(t) a_1(t)\cos(\omega_c t)dt \end{aligned}$$

are the desired contribution, the MAI, and the thermal noise contribution respectively. For an AWGN η with two-sided power spectral density $N_0/2$, the mean is 0 and the variance is approximately $N_0 T_b/4$ when $\omega_c \gg 2/T_b$.

Denote

$$
\begin{aligned}
S_i &\triangleq a_i(t-\tau_i)\cos(\omega_c t+\varphi_i) && (6.5)\\
R_{ij} &\triangleq \cos(\omega_c t+\varphi_i)\cos(\omega_c t+\varphi_j) && (6.6)\\
C_{ij} &\triangleq \int_0^{T_b} b_i(t-\tau_i)S_iS_j dt \\
&= \sum_{k=0}^{N-1} \\
&\quad \left[a_{i,\,k}\int_{\tau_i+kT_c}^{\tau_i+(k+1)T_c} b_i(t-\tau_i)\cos(\omega_c t+\varphi_i)S_j dt\right] \\
&= \sum_{k=0}^{N-1}\left[a_{i,\,k}\sum_{l=0}^{N-1}\left(a_{j,\,l}\int_{t_1}^{t_2} b_i(t-\tau_i)R_{ij}dt\right)\right] && (6.7)
\end{aligned}
$$

where

$$
\begin{aligned}
i &\neq j \\
N &= \frac{T_b}{T_c} \quad (T_c \text{ is the chip period}) \\
[t_1,\, t_2] &= [\tau_i + kT_c,\, \tau_i + (k+1)T_c] \\
&\quad \cap[\tau_j + lT_c,\, \tau_j + (l+1)T_c]
\end{aligned}
$$

Notice that each $a_{j,\,l}$ in $\sum_{l=0}^{N-1} a_{j,\,l}$ may at most overlap in time with

two adjacent $a_{i,k}$'s in $\sum_{k=0}^{N-1} a_{i,k}$. Assume $a_{j,l}$ overlaps with $a_{i,l+s}$ in $[t_1 + lT_c,\, t_m + lT_c]$ and with $a_{i,l+s+1}$ in $[t_m + lT_c,\, t_1 + (l+1)T_c]$, then, if denote

$$\gamma_{ijls} \;=\; a_{i,l+s}a_{j,l}$$

we have

$$\begin{aligned} C_{ij} \;=\; & \sum_{l=0}^{N-1}\left(\gamma_{ijls}\int_{t_1+lT_c}^{t_m+lT_c} b_i(t-\tau_i)R_{ij}dt\right) + \\ & \sum_{l=0}^{N-1} \\ & \left(\gamma_{ijl(s+1)}\int_{t_m+lT_c}^{t_1+(l+1)T_c} b_i(t-\tau_i)R_{ij}dt\right) \end{aligned} \tag{6.8}$$

Since $b_i(t-\tau_i)$ is of the form

$$b_i(t-\tau_i) \;=\; \begin{cases} b_{i1} \in \{-1,\, 1\} & (0 \le k \le Q) \\ b_{i2} \in \{-1,\, 1\} & (Q < k \le N-1) \end{cases}$$

Equation (6.8) becomes

$$C_{ij} \;=\; \left(b_{i1}\sum_{l=0}^{Q}\gamma_{ijls} + b_{i2}\sum_{l=Q+1}^{N-1}\gamma_{ijls}\right)$$

$$
\begin{aligned}
& \int_{t_1+lT_c}^{t_m+lT_c} R_{ij}dt + \\
& \left(b_{i1} \sum_{l=0}^{Q} \gamma_{ijl(s+1)} + b_{i2} \sum_{l=Q+1}^{N-1} \gamma_{ijl(s+1)} \right) \\
& \int_{t_m+lT_c}^{t_1+(l+1)T_c} R_{ij}dt \\
= & \left(b_{i1} \sum_{l=0}^{Q} \gamma_{ijls} + b_{i2} \sum_{l=Q+1}^{N-1} \gamma_{ijls} \right) \\
& \int_{t_1}^{t_m} R_{ij}dt + \\
& \left(b_{i1} \sum_{l=0}^{Q} \gamma_{ijl(s+1)} + b_{i2} \sum_{l=Q+1}^{N-1} \gamma_{ijl(s+1)} \right) \\
& \int_{t_m}^{t_1+T_c} R_{ij}dt
\end{aligned} \tag{6.9}
$$

For $i = j$, Equation (6.7) is simply

$$
\begin{aligned}
C_{ii} &= \left(\sum_{k=0}^{N-1} a_{i,k} a_{i,k} \right) b_i \int_0^{T_c} R_{ii}dt \\
&= \frac{1}{2} N b_i T_c && (6.10) \\
&= \frac{1}{2} b_i T_b && (6.11)
\end{aligned}
$$

Following the notation of C_{ij} and C_{ii}, we have

$$
\begin{aligned}
I_1 &= A_1 C_{11} && (6.12) \\
\xi &= \sum_{i=2}^{K} A_i C_{i1} && (6.13)
\end{aligned}
$$

The desired contribution I_1 is proportional to the channel coding rate N. The MAI ξ, on the other hand, is not so sensitive to N, though statistically it will slowly increase as N increases. Denote by $P(e)$ the probability of event e, then the probability of correctly determining the value of a received bit is

$$\begin{aligned} P_r &= P(I_1 > 0)P(I_1 > -(\xi+\eta) \,|\, I_1 > 0) + \\ & \quad P(I_1 < 0)P(I_1 < -(\xi+\eta) \,|\, I_1 < 0) \end{aligned} \tag{6.14}$$

For symmetrically distributed I_1, ξ, and η, Equation (6.14) reduces to

$$\begin{aligned} P_r &= \frac{1}{2}P(I_1 > -(\xi+\eta) \,|\, I_1 > 0) \\ & \quad + \frac{1}{2}P(I_1 < -(\xi+\eta) \,|\, I_1 < 0) \\ &= \frac{1}{2}P(|I_1| > -(\xi+\eta)) \\ & \quad + \frac{1}{2}P(|I_1| > (\xi+\eta)) \\ &= P(|I_1| > (\xi+\eta)) \end{aligned} \tag{6.15}$$

Equation (6.15) gives the guidance for choosing a suitable channel coding rate N for supporting K simultaneous transmissions.

6.3.1.5 Handling of acknowledgments

A reception can not always be acknowledged immediately in PAR-CDMA. An ACK will be sent back to the source immediately upon the

reception of a frame requiring acknowledgment if there is no other ongoing receptions; otherwise the acknowledgment may need to be delayed. There is a maximum allowed delay time, *MaxACKDelay*, for acknowledgment. That is, upon the completion of a reception, any other ongoing receptions will be terminated if waiting for the completion of another reception will make any pending ACK to be delayed more than *MaxACKDelay*. Multiple pending ACKs can be broadcast and NACKs can be piggybacked in ACKs. How to select a proper value for *MaxACKDelay* is an implementation specific issue. A too small value will prevent parallel receptions from happening, while a too large value can result in unnecessary delay. Our simulation results show that a value equal to the transmission time of a maximum PPDU is a reasonable choice.

Since no full access control is performed and soft channels can be used up, transmissions of ACKs can fail. In PAR-CDMA, a soft channel is reserved for receptions of ACKs. One way to reserve a soft channel is to use a different PN code (randomly selected) and thus ignore incoming frames spread using the common PN code. However, in the single PN code approach, all the nodes can only overhear transmissions of frames spread using the common PN code. A different code will make it impossible to overhear the corresponding ACK frame. Therefore ACKs are still spread using the common PN code, but an additional randomly selected short synchronization sequence is added before the normal synchronization sequence for each

ACK. After the additional synchronization sequence has been transmitted, the PN sequence is reset so that the common synchronization sequence is transmitted as usual. The combined synchronization sequences are called a train of synchronization sequences. By matching the train of synchronization sequences, only the ACK will be detected by the reserved soft channel. But other nodes are still able to overhear the ACK frame. This technique can also be used to reserve soft channels for other purposes, for example, for support of QoS.

6.3.2 Accumulative Receptions

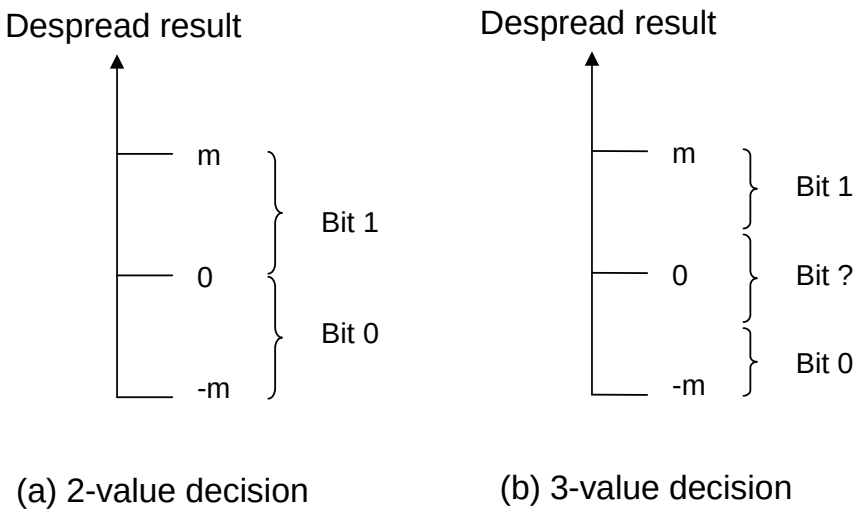


Figure 6.3: Reception Decision

If any transmitter that involves in a collision is identifiable, the receiver can buffer the collided result. A frame may be only partially damaged when colliding with another and it is possible to determine the damaged part(s)

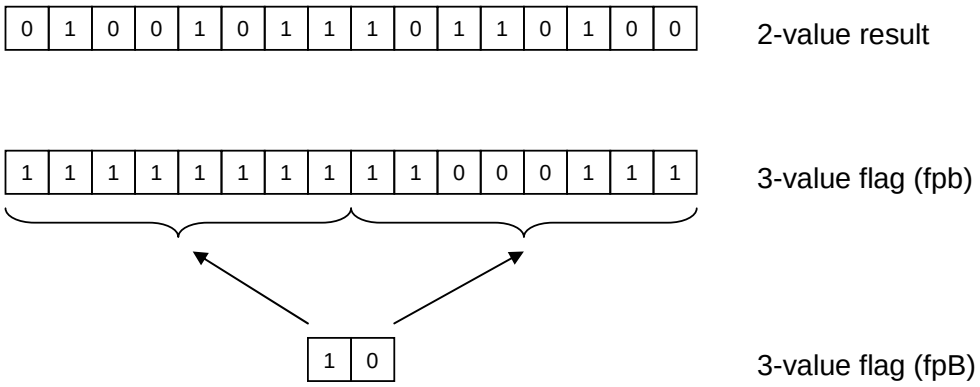


Figure 6.4: Locating Damages

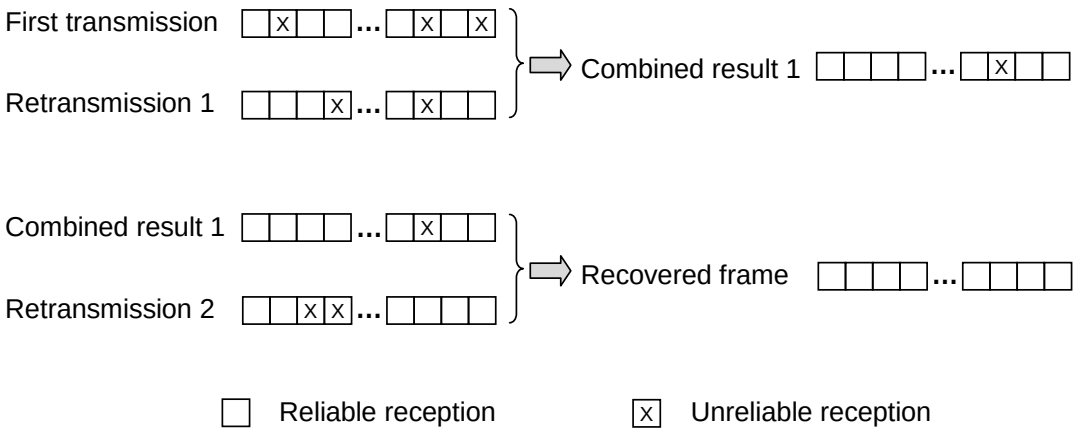


Figure 6.5: An Example of Frame Recovery from Partially Damaged (Re-)Transmitted Frames

using spread spectrum technologies.

A data bit is either 1 or 0, and therefore it is enough to make a 2-value decision at the receiver side (Figure 6.3 (a)). But the reliability of such a decision decreases as the despread result approaches 0. For reliability, we can make an m-value ($m > 2$) decision and use the decision result as a flag. Figure 6.3 (b) depicts a 3-value decision. The final decision results using 2-value decision and 3-value decision are compared in Figure .6.4.

The first row is the 2-value decision result. The second row is the 3-value decision flag. If the 3-value decision result is "?" (Figure 6.3 (b)), then the corresponding flag bit is marked as "0"; otherwise, it is marked as "1". It is not necessary to assign a flag bit for each bit of the decision result (flag per bit, or fpb) as in the second row; we can also assign a flag bit for each byte (fpB) as in the third row.

When a frame is received, the Cyclic Redundancy Check (CRC) is first examined using the 2-value result. If the CRC check fails, then the frame will be buffered, provided the source is identifiable. When a retransmitted frame is received, the receiver will perform a CRC check on the retransmitted frame if the 3-value flag indicates the reception is reliable; otherwise, the receiver will add the retransmitted frame to the buffered result in such a way that reliable parts (i.e., with flag 1's) replace unreliable parts (i.e., with flag 0's). Then the receiver will perform a CRC check on the sum to see if the complete frame has been recovered. This procedure continues until a frame is completely recovered or the buffered result is cleared when expires. In such a way, no information is wasted and it is not necessary to require a single transmission to be completely error-free. For some large frames, if the damaged part is very small, the receiver can even indicate (e.g., through NACKs) the source to only retransmit that small part. Note that at any moment only the combined result is buffered, and thus the requirement for memory is not affected by the number of (re-)transmissions.

Figure 6.5 illustrates one example.

This approach can be also used to increase the reliability of broadcast. In many cases, a node will receive duplicated broadcast frames. If a broadcast frame is damaged, it can be buffered and later added to other duplicated frames.

6.4 Performance Evaluations

6.4.1 Performance Metrics

Following metrics are defined for performance evaluation purpose.

- *Packet delivery ratio*: The ratio of packets successfully received to packets sent, averaged over all end-to-end transmissions within a simulation run.
- *Throughput*: The maximum packet delivery rate measured under sufficient heavy traffic load and expressed in packets per second per source node.
- *Hop delay*: The transaction time of passing a packet to a one-hop neighbor, including time of all necessary processing, backoff as well as transmission, and averaged over all successful end-to-end transmissions within a simulation run.
- *Transmission efficiency*: The ratio of the transmission time of all successfully delivered data packets plus their acknowledgments to the to-

tal transmission time within a simulation run.

6.4.2 Simulation Model

Performance of PAR-CDMA is evaluated and contrasted with that of IEEE 802.11 in both non-mobile and mobile environments via simulations using NS2 [22]. Simulations are based on Equation (6.15), but the noise term η in the equation is dropped by assuming that the communication channel is interference-limited. The key part of simulations is to compute the C_{ij} given in Equation (6.9). We can accurately compute C_{ij} using Equation (6.9) by knowing the details of the PN code and the distribution of data sequences. However, to avoid handling those details, we instead proceed as follows.

1. In Equation (6.9), assume terms with subscript j are those corresponding to the desired signal, then $\tau_j = 0$, $\varphi_j = 0$, $t_1 = 0$, $R_{ij} = \cos(\omega_c t + \varphi_i)\cos(\omega_c t)$, and

$$
\begin{aligned}
C_{ij} \;=\; & \left(b_{i1} \sum_{l=0}^{Q} \gamma_{ijls} + b_{i2} \sum_{l=Q+1}^{N-1} \gamma_{ijls} \right) \\
& \int_0^{t_m} \cos(\omega_c t + \varphi_i)\cos(\omega_c t) dt + \\
& \left(b_{i1} \sum_{l=0}^{Q} \gamma_{ijl(s+1)} + b_{i2} \sum_{l=Q+1}^{N-1} \gamma_{ijl(s+1)} \right)
\end{aligned}
$$

$$\int_{t_m}^{T_c} \cos(\omega_c t + \varphi_i) \cos(\omega_c t) dt \tag{6.16}$$

2. Assume each bit of a bit sequence is independent of other bits and has an equal probability to be 0 or 1. This means that, in Equation (6.16), b_{i2} has a probability of $\frac{1}{2}$ to be equal to b_{i1}. Thus, if denote $b_i = b_{i1} = b_{i2}$ (when $b_{i1} = b_{i2}$) and $\tilde{b_i} = b_{i1} = -b_{i2}$ (when $b_{i1} \neq b_{i2}$) and notice that $\varphi_i = -\frac{2\pi t_m}{T_c}$, Equation (6.16) becomes

$$\begin{aligned} C_{ij} \;=\; & \frac{1}{2}\left[\left(b_i \sum_{l=0}^{N-1} \gamma_{ijls}\right) + \right. \\ & \left. \tilde{b_i}\left(\sum_{l=0}^{Q} \gamma_{ijls} - \sum_{l=Q+1}^{N-1} \gamma_{ijls}\right)\right] \\ & \int_0^{t_m} \cos\left(\omega_c t - \frac{2\pi t_m}{T_c}\right) \cos(\omega_c t) dt + \\ & \frac{1}{2}\left[\left(b_i \sum_{l=0}^{N-1} \gamma_{ijl(s+1)}\right) + \right. \\ & \left. \tilde{b_i}\left(\sum_{l=0}^{Q} \gamma_{ijl(s+1)} - \sum_{l=Q+1}^{N-1} \gamma_{ijl(s+1)}\right)\right] \\ & \int_{t_m}^{T_c} \cos\left(\omega_c t - \frac{2\pi t_m}{T_c}\right) \cos(\omega_c t) dt \end{aligned} \tag{6.17}$$

3. To apply Equation (6.17), we need to determine the values of b_i, $\tilde{b_i}$, r_{ijls}, $r_{ijl(s+1)}$, Q, and t_m. The values of Q and t_m are respectively determined by the bit offset and chip offset between two arrived

spread sequences. And each of b_i, $\tilde{b}_i$, r_{ijls}, $r_{ijl(s+1)}$ can be treated as a random variable uniformly distributed in $\{-1, 1\}$.

6.4.3 Experimental Setup

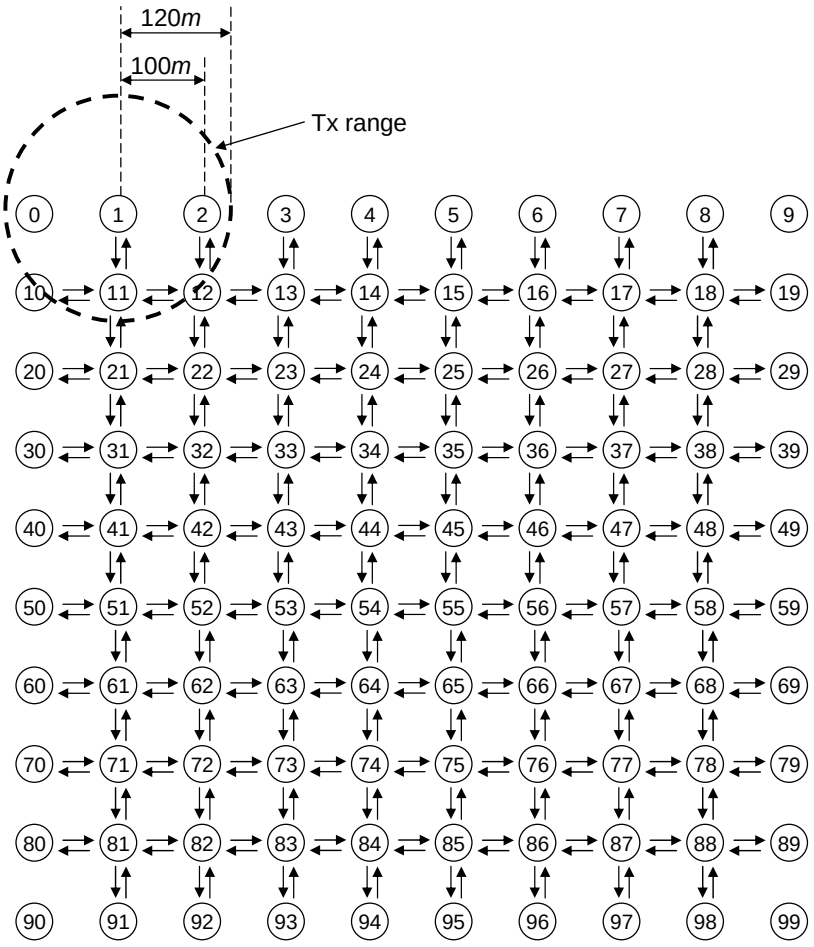


Figure 6.6: Simulation Scenario

All results for PAR-CDMA are based on the single PN code approach (see subsection 6.3.1.2). Simulations for non-mobile networks run in a multi-hop environment (Figure 6.6), where 100 nodes are evenly distributed within an area of $1000 \times 1000\, m^2$. And 32 traffic flows are set up between

boundary nodes as shown in Figure 6.6. The distance between two neighbors located horizontally or vertically is $100\,m$, and the transmission range is $120\,m$. Simulations for mobile networks run in a similar environment, but all the nodes except the 36 boundary nodes are moving. The reason to make all the boundary nodes fixed is that we want to have a better control over traffic flows. The traffic flows are set up as in non-mobile networks. If all nodes move, then the number of hops of a traffic flow is uncontrollable.

The simulation duration is 900 seconds for non-mobile networks and 2700 seconds for mobile networks. And the application traffic ends at 800 second and 2600 second respectively, leaving enough time for the experiment to shut down gracefully. Constant bit rate (CBR) traffic is used in both non-mobile and mobile networks. The CBR rate is 13 packets per second (pps) for non-mobile networks and 6 pps for mobile networks. The way-point mobility pattern [51] is used in all mobile simulations, with different maximum speeds but a fixed pause time 0 (i.e., no pause). It has been reported in [51] that, for some cases, the first 15 minutes of NS2 simulations may be unstable for the way-point mobility model. Therefore, the simulation results for mobile networks are recorded after 15 minutes. The log-normal shadowing radio propagation model is adopted in all simulations. Each non-mobile simulation runs 10 times and each mobile simulation runs 20 times, each run with a different random seed. Each node has 5 soft channels, 4 for data receptions and 1 for ACK receptions and overhearing. The

spread rate used in PAR-CDMA is two times of that used in 802.11. So the data bit rate of PAR-CDMA is half of that of 802.11 for the same available frequency bandwidth, which is 2 MHz for all simulations. Some other simulation parameters take the following values (unless otherwise stated): data packet size = 512 bytes; maximum mobility speed = 10 meters per second; shadowing deviation = 4; maximum allowed ACK delay = 512 bytes; active accumulative receptions are used.

6.4.4 Numerical Results

6.4.4.1 Effect of traffic load

To study the behavior of PAR-CDMA under different traffic loads, we vary the traffic load from 1 pps to 20 pps in non-mobile environments and from 1 pps to 10 pps in mobile environments. In non-mobile environments, PAR-CDMA has a throughput of about 13 pps, while 802.11 only reaches about 3 pps (Figure 6.7). When the traffic load is light (< 3 pps), the throughput is roughly the same for both PAR-CDMA and 802.11, which has a linear relation to the traffic load. However, when the traffic load is larger than 3 pps, PAR-CDMA performs much better than 802.11. The hop delay of PAR-CDMA roughly maintains at the same level when traffic load is within 8 pps and increases slowly after that (Figure 6.8); the hop delay of 802.11, on the other hand, shots up as the traffic load increases. In most cases, PAR-CDMA has a hop delay less than half of that of 802.11. The

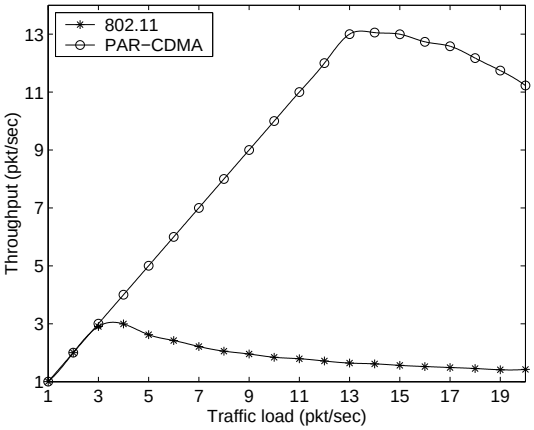


Figure 6.7: Throughput (non-mobile)

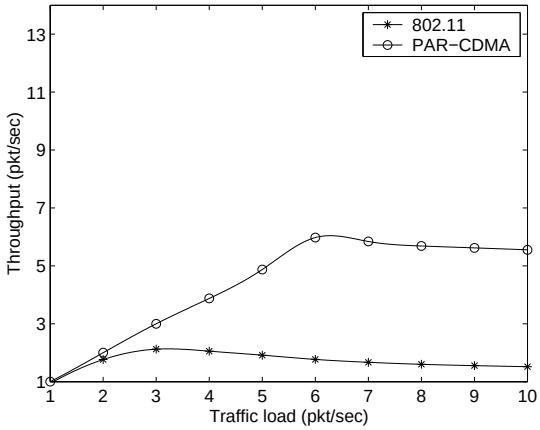


Figure 6.10: Throughput (mobile)

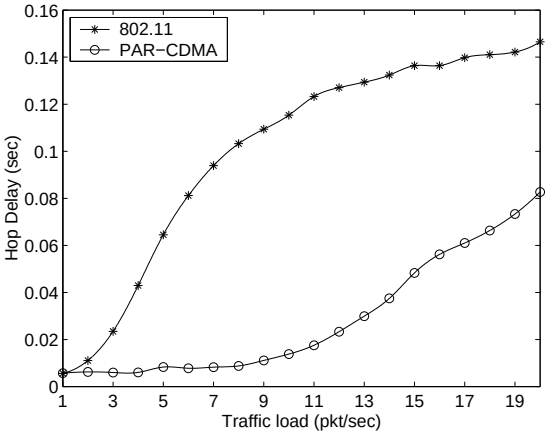


Figure 6.8: Hop Delay (non-mobile)

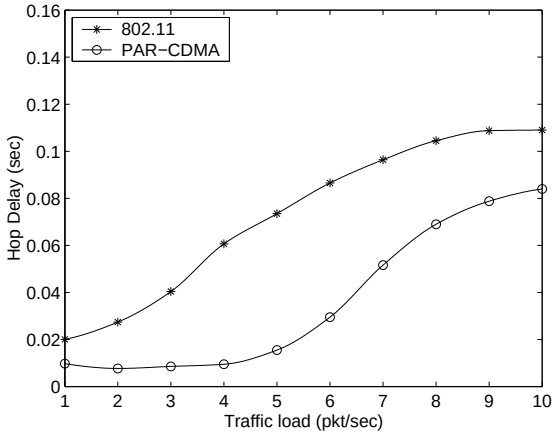


Figure 6.11: Hop Delay (mobile)

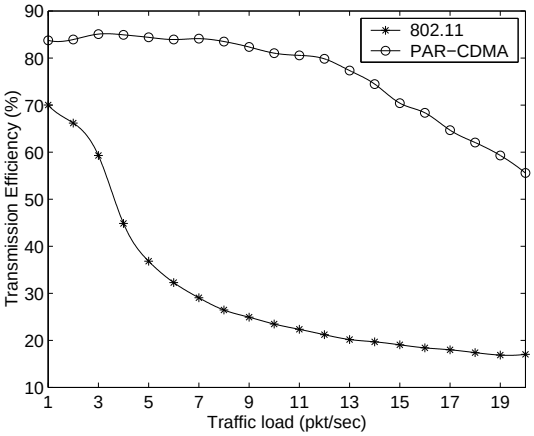


Figure 6.9: Transmission Efficiency (non-mobile)

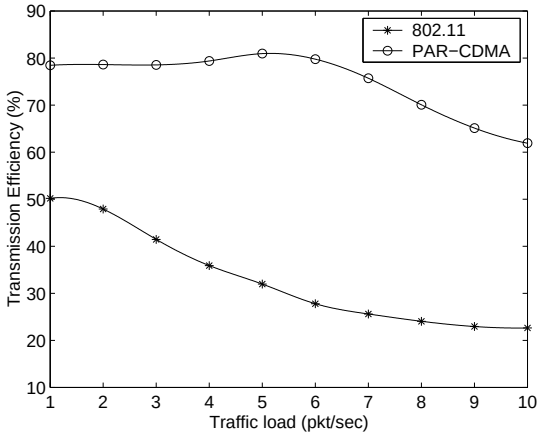


Figure 6.12: Transmission Efficiency (mobile)

transmission efficiency of PAR-CDMA ranges from 55.6% to 85.1% (Figure 6.9), but that of 802.11 drops quickly from 70.0% to 44.8% when traffic load increases from 1 pps to 4 pps and the lowest transmission efficiency is only 16.9%.

In mobile environments, PAR-CDMA has a throughput of about 6 pps and 802.11 has a throughput of about 2 pps (Figure 6.10). PAR-CDMA has a relatively stable hop delay when traffic load is less than 5 pps (Figure 6.11), which is only $21.1\% \sim 48.5\%$ of that of 802.11 for the same traffic load range. After 5 pps, the hop delay of PAR-CDMA increases slightly faster than that of 802.11, and it rises to 77.0% of that of 802.11 at 10 pps. Like in non-mobile environments, PAR-CDMA enjoys a much higher transmission efficiency than 802.11 in mobile environments (Figure 6.12). But the transmission efficiency of 802.11, when compared with that in non-mobile environments, decreases more smoothly as traffic load increases.

6.4.4.2 Passive and active accumulative receptions

Accumulative receptions in PAR-CDMA are studied by changing the shadowing deviation value from 4 to 12. As accumulative receptions only happen when transmissions fail, they play a more important role in an environment where transmissions are not reliable (here corresponding to a larger shadowing deviation). Four cases are compared: 802.11, PAR-CDMA without accumulative receptions (NoAccRx), PAR-CDMA with passive accu-

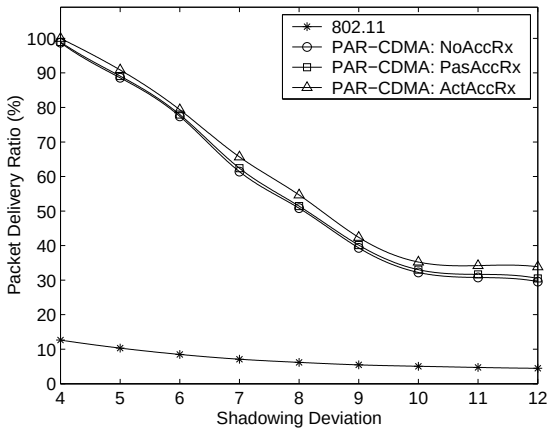


Figure 6.13: Packet Delivery Ratio (non-mobile)

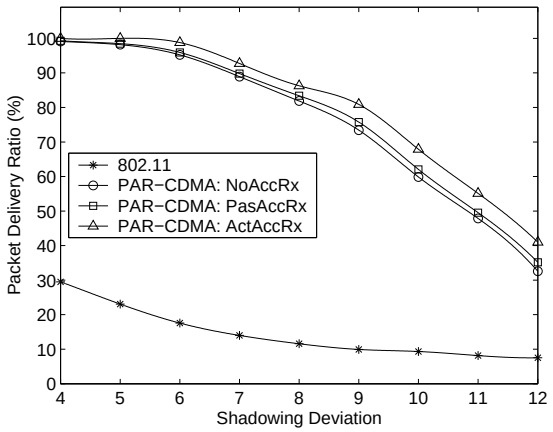


Figure 6.16: Package Delivery Ratio (mobile)

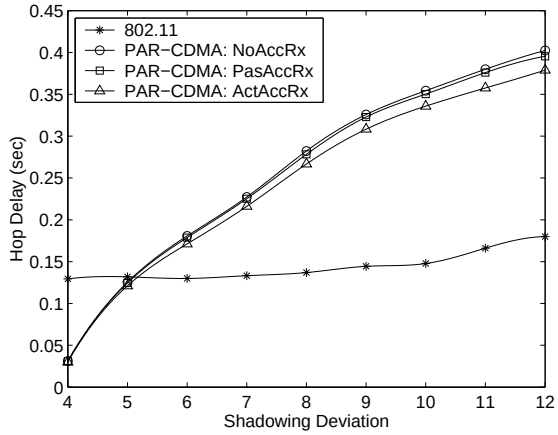


Figure 6.14: Hop Delay (non-mobile)

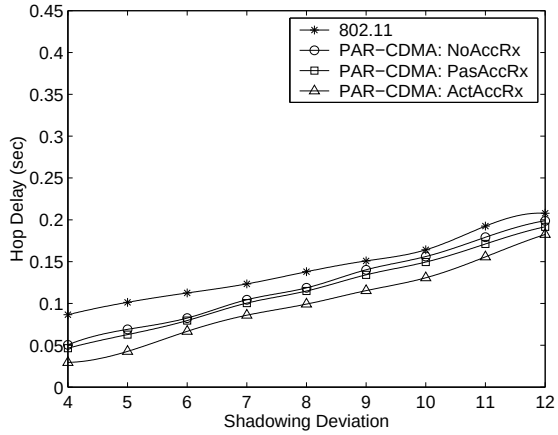


Figure 6.17: Hop Delay (mobile)

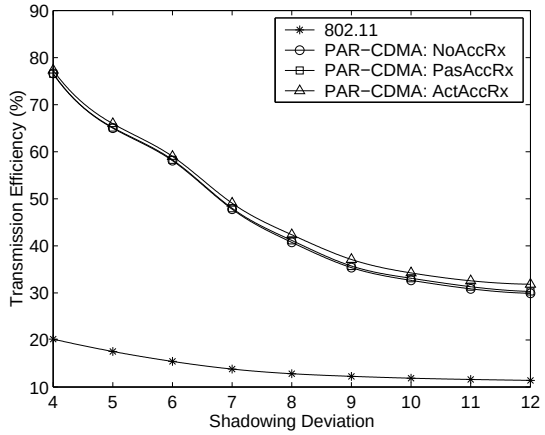


Figure 6.15: Transmission Efficiency (non-mobile)

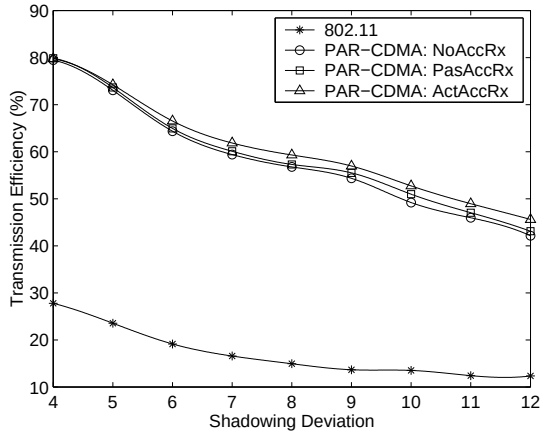


Figure 6.18: Transmission Efficiency (mobile)

mulative receptions (PasAccRx), and PAR-CDMA with active accumulative receptions (ActAccRx). In non-mobile environments, all three PAR-CDMA cases have a better performance than 802.11 in terms of packet delivery ratio and transmission efficiency (Figures 6.13, 6.15). For hop delay, it seems that 802.11 beats all three PAR-CDMA cases when the traffic load exceeds 5 pps (Figure 6.14). However, it is noteworthy that the packet delivery ratio of 802.11 is much smaller than those of the three PAR-CDMA cases and the hop delay is an average value calculated over all successful end-to-end transmissions. In mobile environments, all three PAR-CDMA cases outperform 802.11 in every respect (Figures 6.16, 6.17, 6.18). By using PAR-CDMA, there is a $25.1\% \sim 77.7\%$ absolute gain in packet delivery ratio and a $29.8\% \sim 51.7\%$ absolute gain in transmission efficiency. For hop delay, all three PAR-CDMA cases perform much better than 802.11 when the shadowing deviation is small, but the relative difference, in percentage, among four cases gets smaller as the shadowing deviation increases.

When comparing the three PAR-CDMA cases, one can see that in general ActAccRx has a better performance than PasAccRx, which in turn performs slightly better than NoAccRx. The performance improvement by using PasAccRx is not that significant in our simulations. For PasAccRx to work, several conditions must be met: the transmission of a certain packet must fail several times; no retransmission should be completely successful; and the damaged parts of several (re-)transmissions should be different.

The chance that all those conditions are met is very small when transmissions are relatively reliable. Nonetheless, the only condition for ActAccRx to work is that the header of a corrupted packet is readable at the receiver side. So ActAccRx is much more effective than PasAccRx. In general, the worse the communication environment is, the larger the performance gain by using either PasAccRx or ActAccRx is. For example, PasAccRx or ActAccRx is more effective for large shadowing deviation values or in mobile environments.

6.4.4.3 Maximum acknowledgment delay

The maximum allowed ACK delay, *MaxACKDelay*, is an important factor in PAR-CDMA. To see its impact on the network performance, we compare the performances for *MaxACKDelay* = 0.5 data packet (dp), 1 dp, and 2 dp's. In both non-mobile and mobile environments, we observe that 1 dp is a turning point and has the best performance (Tables 6.1, 6.2). This result shows that a too small *MaxACKDelay* will reduce the chance of parallel receptions and a too large *MaxACKDelay*, on the other hand, will incur more delay.

6.4.4.4 Mobility

Some simulations are run for evaluating the performance of PAR-CDMA under various mobility speeds. Figures 6.19, 6.20, and 6.21 show that PAR-

Table 6.1: Effect of Maximum ACK Delay (non-mobile)

Max ACK delay (data pkt)	0.5	1.0	2.0
Packet delivery ratio (%)	90.54	100.00	95.31
Hop delay (sec)	0.0667	0.0539	0.0701
Transmission Efficiency (%)	68.99	78.22	75.63

Table 6.2: Effect of Maximum ACK Delay (mobile)

Max ACK delay (data pkt)	0.5	1.0	2.0
Packet delivery ratio (%)	94.78	100.00	98.36
Hop delay (sec)	0.0528	0.0295	0.0424
Transmission Efficiency (%)	73.25	79.76	77.24

CDMA beats 802.11 in all cases. Recall that the data bit rate of PAR-CDMA is only half of that of 802.11, the same mobility speed will appear different to them. For example, more time is needed to transmit a frame in PAR-CDMA than in 802.11. Therefore, a node can move more distance in PAR-CDMA than in 802.11 during this period. As a result, PAR-CDMA is more susceptible to mobility than 802.11, as can be seen from the slopes of curves in all the three figures.

6.5 Conclusion and Future Work

A new MAC scheme, called parallel and accumulative reception using code division multiple access (PAR-CDMA), is proposed in this chapter. The scheme supports both parallel receptions and accumulative receptions without the need of code assignment. Performance evaluation results show

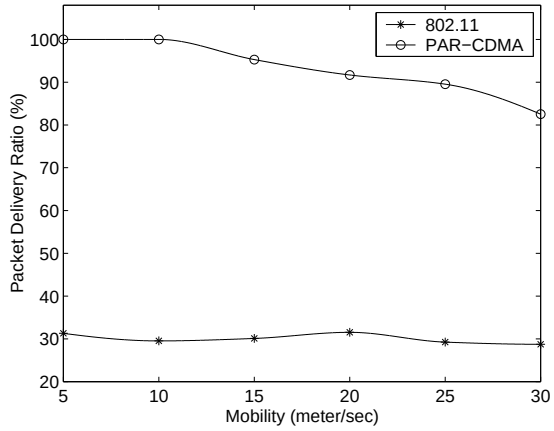


Figure 6.19: Packet Delivery Ratio

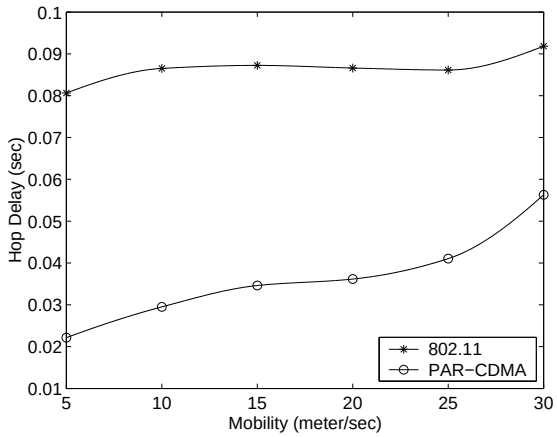


Figure 6.20: Hop Delay

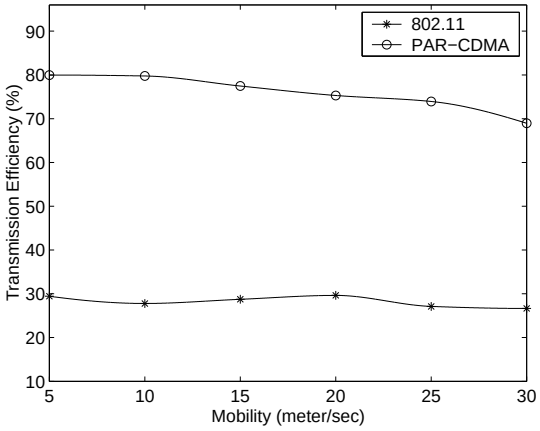


Figure 6.21: Transmission Efficiency

that, in terms of throughput, packet delivery ratio, hop delay, and transmission efficiency, PAR-CDMA towers above 802.11 in all scenarios.

To support parallel receptions, PAR-CDMA needs to use a smaller data bit rate than 802.11 for the same available frequency bandwidth. Therefore, the performance of PAR-CDMA is expected to be worse than that of 802.11 if parallel receptions do not happen that often. One way to solve this problem is to allow a node to dynamically change the channel coding rate via negotiation with its neighbors. That is, if no parallel receptions are needed or the number of parallel receptions is small, a node can try to increase its data bit rate by reducing the channel coding rate. Another way is to only apply parallel receptions to those short control packets such as request-to-send (RTS), clear-to-send (CTS), and acknowledgement (ACK) packets, while data packets are still transmitted as in 802.11. We intend to study those approaches in our future work.

Part III

Network Layer Issues – Addressing and Mesh Routing

Chapter 7

Meshed Adaptive Robust Tree Routing

Chapter 7

Meshed Adaptive Robust Tree Routing

By binding logic addresses to the network topology, routing can be done without going through route discovery. This eliminates the initial route discovery latency, saves storage space otherwise needed for routing table, and reduces the communication overhead and energy consumption. One example is the cluster tree routing. However cluster tree faces the following problems: (1) inflexible address assignment, which often results in huge waste of addresses; (2) single point of failure (SPF), and no efficient repair and recovery algorithm for it; (3) change of address when a node changes its attaching point, and; (4) non-optimal routes. To solve those problems, a new type of tree called adaptive robust tree (ART) and its meshed form, meshed ART (MART), are proposed in this chapter. The basis of ART/MART is an adaptive block addressing (ABA) scheme, which is used for logic address assignment as well as network autoconfiguration purpose. ABA takes into account the actual network topology and thus is fully topology-adaptive.

7.1 Background

A logic tree structure can be formed along with the setup of a wireless network. The first node in the network will designate itself as the root and begin to accept association requests from other nodes. Any node already in the network can determine whether to allow other nodes to join it, that is, whether to act as a router, depending on the availability of its resources such

as memory and energy. Various tree algorithms have been proposed for routing purpose in both wired and wireless networks [29,96–101]. Of our particular interest here is the cluster tree algorithm [29], which has been adopted by ZigBee [30]. Two network-wide parameters are defined in cluster tree: the maximum number of children a node can have, C_m, and the maximum level of the tree (i.e., the maximum hops from the root to a leaf node), L_m. A cluster tree can be formed through a top-down procedure. First, the root (normally a manually designated node) distributes the whole network address space evenly among its maximum allowed C_m children[1]; then each of the children further distributes the allocated address block evenly among its C_m children; and this procedure continues until the maximum level, L_m, is reached. The address assignment is solely based on C_m and L_m, irrespective of the number of children a node actually has. By knowing C_m, L_m, and its own tree level, a node is able to calculate the next hop by looking at the destination address in a packet.

Cluster tree, however, lacks flexibility and robustness. The inflexible address assignment often results in huge waste of addresses (our simulation shows that a network with a 16-bit address space can only accommodate several hundred nodes). Cluster tree also suffers for single point of failure (SPF). Among other problems are non-optimal routes and the need for address reassignment when a node changes its attaching point.

[1]Each child still gets $1/C_m$ of the total available addresses even if the actual number of children is less than C_m.

To solve the aforementioned cluster tree problems, we propose a new type of tree called adaptive robust tree (ART), which has the following features:

- Addresses are assigned adaptively according to the network topology. This means asymmetric topology will lead to asymmetric address assignment. Different nodes can have different numbers of children, and different branches below a node can have different address block sizes.
- The concept of C_m and L_m no longer exists. A node can accept as many children as it wishes, only if its physical capacity allows. And a tree can have as many levels as it needs. The limitation only comes from the address space.
- SPF problems can be efficiently fixed.
- If, due to link failure or node failure, a branch is detached from its original attaching point and re-attached to another point in the tree, the assigned addresses of nodes along this branch remain unchanged. This feature is highly desirable for routing and service discovery.
- If needed, a meshed ART (MART) can be formed. By using a MART, it is possible to route a packet through a shorter path. Some SPFs can also be removed by using a MART.

The simulation results of ART/MART will be given in chapters 8 and 9, together with those of topology-guided distributed link state (TDLS) and

Table 7.1: Adaptive Robust Tree Table (ARTT)

$type_1$	beg_addr_1	end_addr_1	$priority_1$	$next\_hop_1$
$type_2$	beg_addr_2	end_addr_2	$priority_2$	$next\_hop_2$
.....				

$type_i$:	type of branch i (*desIn*, *desOut*, *srcIn*, *srcOut*)
beg_addr_i:	beginning address of branch i
end_addr_i:	ending address of branch i
$priority_i$:	priority of branch i (*high, normal*, *broken*)
$next\_hop_i$:	next hop via which the whole branch i is routed

AODV.

7.2 Adaptive Robust Tree (ART)

7.2.1 ART Table

In adaptive robust tree (ART), each node keeps an ART table (ARTT) to track its branches. The structure of the table is given in Table 7.1. Each branch is assigned a block of consecutive addresses. But there is no need to assign consecutive blocks to those branches. This allows adding new branches later as well as repairing the tree efficiently (see subsection 7.2.2.3). Field $type_i$ determines under what condition a branch is selected for relaying a packet. For example, if $type_i$ =*desIn*, then the branch will be selected for relaying a packet if the destination address of the packet falls in the address block [beg_addr_i, end_addr_i]. In case that the destination address (or source address) of a packet falls

in (or out of) multiple address blocks, priority is applied in the order *desIn*→*desOut*→*srcIn*→*srcOut*; and field $priority_i$ further distinguishes branches with the same $type_i$ value. The parent of a node is treated as an implicit branch (i.e., not recorded in the ARTT of the node), which has the lowest priority. The function of ARTT will be discussed in detail in the following subsection.

7.2.2 Three Phases of ART

Functionally, three phases are defined in ART: initialization (or configuration) phase, operation phase, and recovery phase.

- During the initialization phase, nodes join the network and an ART is formed.
- After initialization, the network enters the operation phase and normal data communications start. During the operation phase, new nodes are still allowed to join the network.
- If a tree link is broken, then the recovery phase is triggered. Notice that the recovery phase is different from the other two phases in that, not the whole network, but only the part affected by the link failure needs to enter the recovery phase. And only failures (either node failures or link failures) happened after initialization will trigger the recovery phase. Failures during initialization should be handled by the initialization phase itself.

7.2.2.1 Initialization phase

An ART is formed during the initialization phase. The ART formation can be broken down into two stages: association and address assigning.

During the association stage, beginning from the root (a node that is normally manually designated, for example, by pressing a button), nodes gradually join the network and a tree is formed. But this tree is not an ART yet, since no node has been assigned an address. There is no limitation on the number of children a node can accept. A node can determine by itself how many nodes (therefore, how many branches below it) it will accept according to its capability and other factors. It is possible that a joining node can not find any other node to associate if the number of children a node can have is not flexible. Therefore, instead of simply accepting or rejecting an association request, a node uses an acceptance degree (AD) to indicate the willingness of acceptance. For example, a four-level AD scheme could be:

3 – accept without reservation

2 – accept with reservation

1 – accept with reluctance

0 – reject (a node should try to avoid this AD unless absolutely necessary)

When a joining node receives multiple association responses, it should choose the node with the highest AD for association, unless there are other high priority factors indicating not to do so. Using AD increases the chance with which a node successfully joins the network without severely overload-

ing an individual node.

After the tree reaches its bottom, that is, no more nodes are waiting for joining the network (a proper timer can be used for this purpose), a down-top procedure is used to calculate the number of nodes along each branch, as shown in Figure 7.1. The numbers in brackets indicate the numbers of nodes along branches below a certain node.

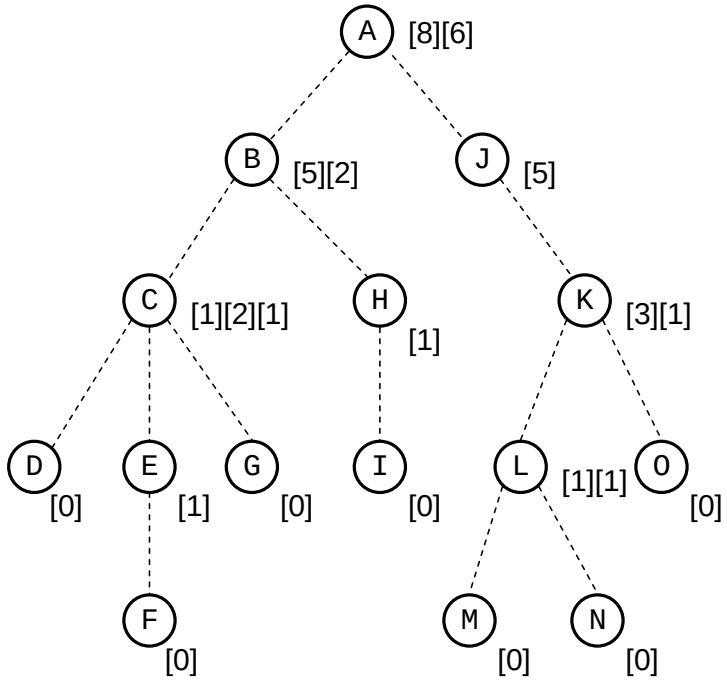


Figure 7.1: Calculation of Number of Nodes along Each Branch

When numbers of nodes are reported from bottom to top, each node can also indicate a desirable number of addresses. For example, node *F* can indicate it wishes to get 3 addresses (2 for possible future extension), though currently only one address is enough (for itself); node *E* can indicate it wishes to get 5 addresses (3 for node *F*, 1 for itself, and 1 for future exten-

sion); and so on. After the root, node *A* here, receives the information from all the branches, it begins to assign addresses.

During the address assigning stage, a top-down procedure is used. First, the root checks if the total number of nodes in the network is less than the total number of addresses available. If not, address assignment fails. One possible solution in case of address overflow is to separate the network into smaller ones or augment the address space. Here we do not handle address overflow, and will assume no address overflow happens hereafter. Next, the root assigns a block of consecutive addresses to each branch below it, taking into account the actual number of nodes and the wished number of addresses. The actual number of addresses assigned could be less or more than the wished one, depending on the availability of addresses, but no less than the actual number of nodes. This procedure continues until the bottom of the tree is reached. After address assigning, an ART is formed and each node has an ARTT for tracking its branches. For example, node *C* could have an ARTT as follows:

[*desIn*][6][8][*normal*][6]

[*desIn*][9][13][*normal*][9]

[*desIn*][14][14][*normal*][14]

The above ARTT indicates that node *C* has total 3 branches. Branch 1 owns address block [6, 7, 8], and any packet whose destination falls in this address block should be routed through branch 1 itself (as $next\_hop_i = 6$ points

to branch 1 itself); branch 2 owns address block [9, 10, 11, 12, 13], and likewise packets destined for this branch should be routed through branch 2 itself; branch 3 only owns one address 14.

The above addressing scheme, called adaptive block addressing (ABA) scheme here, takes into account the actual network topology and thus is fully topology-adaptive.

7.2.2.2 Operation phase

After an ART is built, the network enters the operation phase. Using the ARTT (see Table 7.1), a packet can be easily routed. As an example, when node *C* receives or generates a packet, it checks if the destination belongs to one of its branches. If the destination belongs to branch *i*, the packet will be relayed to the node with address $next\_hop_i$. Note that $next\_hop_i$ is not necessarily equal to beg_addr_i. The use of $next\_hop_i$ is for handling tree repair (details given in subsection 7.2.2.3). If the destination falls out of any branch below node *C*, the packet will be routed to the parent of node *C*.

Although no significant change of the node number or network topology is expected during operation phase, more nodes (therefore, more branches) are still allowed to be added at any level of the tree, only if additional addresses (reserved during initialization phase) are available. Address assignment can be locally adjusted within a branch if a node runs out of addresses. For instance, a node can request more addresses from its parent. If the par-

ent does not have enough addresses, it can try to either request additional addresses from its parent or adjust address assignment among its children. If there is a substantial change of the node number or network topology, which can not be handled locally during operation phase, the network is allowed to go through the initialization phase again.

7.2.2.3 Recovery phase

During the operation phase, link failures or routing node failures will trigger the recovery phase. We propose two different approaches for tree link repair. But before giving the details of tree repair and recovery, let us first take a look at the significance of the ARTT. First, there is no specific relationship between two different branches of a node. The address blocks of different branches can even be overlapped due to tree repair, where a priority value will be used to determine which branch will be actually used. This means one is free to move a branch from a place to another place (some details need to be addressed, see below) without changing any address within the branch. Second, a packet is either routed through one of the branches below a node or through the parent. Notice that branch i is not necessarily routed through branch i itself. It can be routed through another branch.

Approach 1

Now suppose the link between nodes J and K is broken (Figure 7.2).

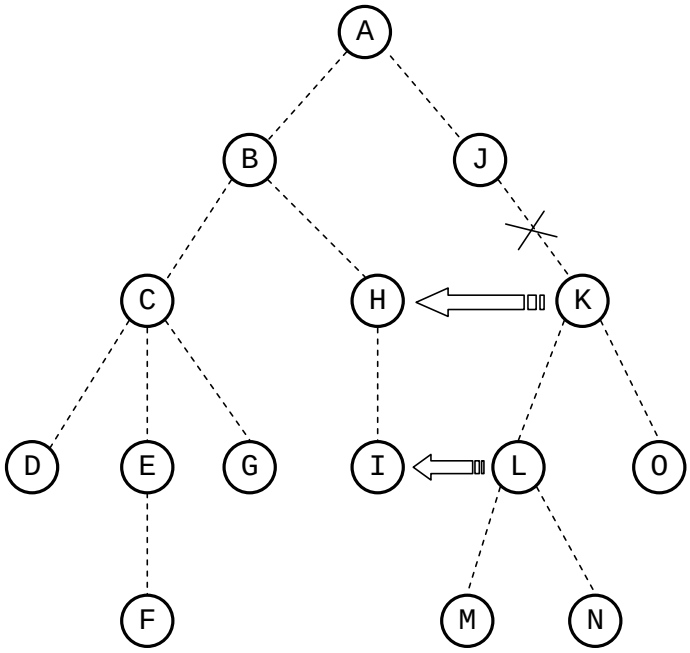


Figure 7.2: Tree Repair (Approach 1)

If node *K* is able to connect to node *H*[2], then the whole branch starting from node *K* (referred to as branch *K* for short here) can be moved to node *H*. During this procedure, no address change is needed for the whole branch *K*. But modifications will be made to ARTTs of all the nodes from node *K* to the nearest common ancestor of nodes *K* and *H* (node *A* here), and from the common ancestor to the new attaching point node *H*. Assume the original address block of node *K* is $[addr_0, \cdots, addr_m]$, then the modifications are as follows (each branch record is in the form of $[type_i][beg\_addr_i][end\_addr_i][priority_i][next\_hop_i]$):

- Node *H* adds a branch:

[2]In the case that node *J* detects the failure, it will notify its children by broadcasting a message with a limited time to live (TTL). The top node of the broken branch, here node *K* (or each of nodes *L* and *O* if node *K* fails), is in charge of the repair.

[*desIn*][$addr_0$] [$addr_m$][*normal*][$addr_0$]

- Node *B* adds a virtual branch:

  [*desIn*][$addr_0$] [$addr_m$][*normal*][$H's\ address$]

- Node *A* adds a virtual branch:

  [*desIn*][$addr_0$] [$addr_m$][*high*][$B's\ address$]

- Node *J* adds a virtual branch:

  [*desIn*][$addr_0$] [$addr_m$][*high*][$A's\ address$]

The modifications begin from node *H* and follow the order of nodes *H*, *B*, *A* and *J*[3]. Now take a look at how a packet destined for a node below node *K* (including node *K*) can be routed (a packet destined for a node outside the branch, no matter where it is originated, can be routed as normal).

1. Any node not within the branch beginning from the common ancestor (node *A* here) will first relay the packet to the common ancestor.
2. If the packet is originated from within branch *K*, it will be routed as normal within the branch.
3. Besides cases 1 and 2, the only remaining case is that the packet is received or generated by node *J*, *A*, *B,* or *H*. If node *J* receives or generates the packet, the packet will be relayed to node *A* according to the new branch with ($priority_i$ = *high*). If node *A* receives or generates the packet, the packet will be relayed to node *B* according to the

[3]It indicates that the broken branch is attached to another branch that is also broken if the nearest common ancestor of the two branches is unreachable. In this case, another attaching point should be tried.

new branch with ($priority_i$ = *high*). From node *B*, the packet will be forwarded to node *H*, and then to branch *K*.

If node *K* is unable to attach to node *H* and instead node *L* is able to attach to node *I*, tree repair can be similarly handled. In this case, we need to take care of packets both *to* and *from* branch *K*. To route packets to branch *K*, ARTTs of nodes *J*, *A*, *B*, *H* and *I* need to be modified. And node *K*'s address block (rather than node *L*'s address block) should be used in modifications of ARTTs. It is clear, if a packet reaches node *L*, then it will be able to reach any node within branch *K*. To route packets from branch *K*, following modifications are needed:

- Node *L* adds a virtual branch:

  [*desOut*][$addr_0$][$addr_m$][*normal*][$I's\ address$]

- Node *K* adds a virtual branch:

  [*desOut*][$addr_0$][$addr_m$][*normal*][$L's\ address$]

Field ($type_i$ = *desOut*) tells that a packet whose destination address is *outside* the address block [$addr_0$, $addr_m$] should be routed via the corresponding $next\_hop_i$.

We give priority to node *K* over node *L* in the above tree repair. Thereby, if node *K* is able to attach to another node not within its branch, then there is no need for node *L* to do anything. If node *K* fails in attempting to attach to another node, it will inform its children to do the repair. If node *L* also

fails, however, it should not continue to ask its children to do the repair immediately. Instead, it should report the failure to node *K*. Node *K* may further ask node *L* to continue the repair, in which case node *L* will in turn ask its children to do the repair. But node *K* may also ignore the failure report from node *L* (for example, if node *O* succeeds in the repair). This procedure continues until the tree is successfully repaired or the repair fails (i.e., no node can attach to another branch). Either way, the recovery phase ends there.

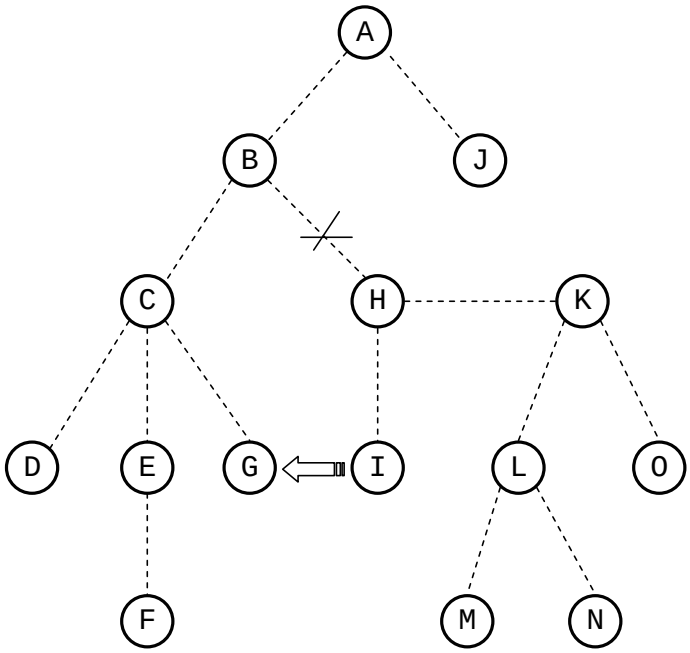


Figure 7.3: Multi-level Tree Repair (Approach 1)

Sometimes multi-level tree repair can happen. For example, in Figure 7.3, if the link between nodes *H* and *B* also breaks, the repair will have to fix not only branch *H*, but also branch *K*, which is under the care of node

H. The repair for branch *H* is a first-level repair. But the repair for branch *K* is a second-level repair, since the branch is already under the care of node *H*. It is not difficult to do a multi-level repair in ART. In the above example, assume node *I* successfully attaches to node *G*. From node *H*, node *I* can know that, besides the address block of the original branch *H*, there is another under-care address block beginning from node *K*. So node *I* can inform nodes *G*, *C*, and *B* to modify their ARTTs and add these two address blocks. Since there is no other node between the common ancestor node *B* and node *H*, no ARTT modification is needed for that branch. But if there are some nodes along the branch from node *B* to node *H*, then two types of modifications are needed: for first-level repair, an entry should be added to the ARTTs of those nodes; for second-level (or any higher-level) repair, all the old entries added for first-level (or any lower-level) repair should be removed. Removing the old entries causes a packet to be forwarded in the direction from node *H* to node *B* (originally in the opposite direction, from node *B* to node *H*).

If too many repairs happen, the network is allowed to be re-configured, that is, go through the initialization phase again.

Approach 2

In approach 2, whenever a link is broken, the node having detected the

problem will perform a local repair by broadcasting a route request (RREQ) message with a time to live (TTL) equal to *rpr_min_TTL* and an *up-down* flag indicating whether the upward link or downward link is broken. Only those nodes below (or above) the source of the RREQ can reply if the *up-down* flag is *down* (or *up*). If no route reply (RREP) is received within a certain period, the node increases the TTL by *rpr_inc_TTL* and tries again. This procedure continues until at least one RREP is received or the TTL reaches *rpr_max_TTL*. If multiple local routes are found, the node will select the best one based on the costs recorded in the corresponding RREPs. A route confirmation (RCFM) message is then unicast to activate the selected local route. Thus a bi-directional routing path will be built between the two nodes originally connected via tree link (they could be multiple tree levels away). The specific values of *rpr_min_TTL*, *rpr_inc_TTL*, and *rpr_max_TTL* are determined by applications.

Figure 7.4 shows an example of tree repair using approach 2. This time we assume that node *K*, which is along the routing path from node *C* to node *M*, fails. Node *J* successfully finds another local route *J-H-L* by broadcasting an RREQ with a TTL of 3 (not all broadcast propagation is shown in the figure) and an *up-down* flag of *down*.

To relay packets from node *J* to node *L* (see Figure 7.5), following modifications are made (assume the address block of node *K* is [A_0, A_m] and that of node *L* is [a_0, a_m]):

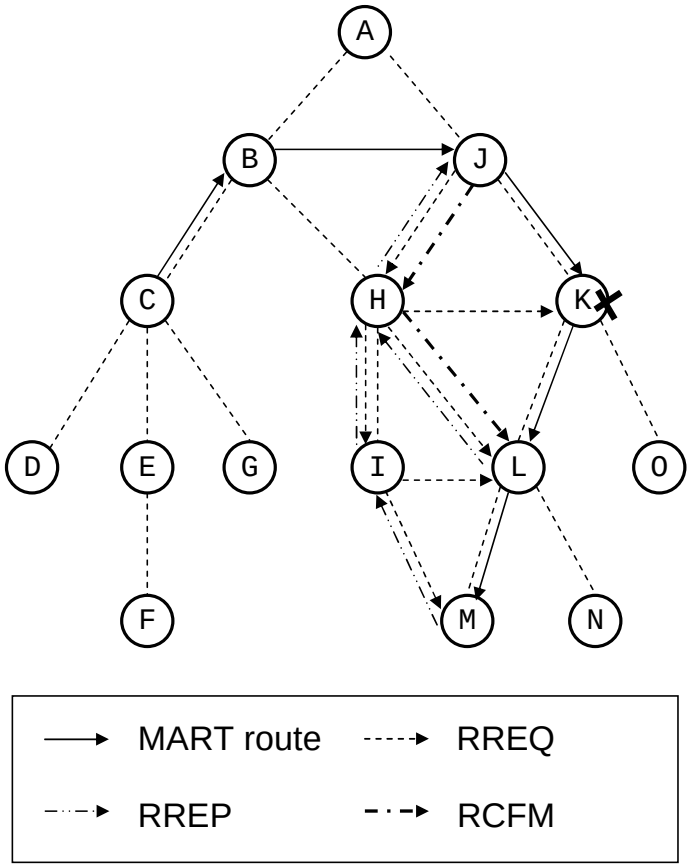


Figure 7.4: Tree Repair (Approach 2)

- Node *J* marks branch *K* as *down*:

  [*desIn*][A_0][A_m][*down*][$K's\ address$]

- Node *J* adds a virtual branch:

  [*desIn*][a_0][a_m][*normal*][$H's\ address$]

- Node *H* adds a branch[4]:

  [*desIn*][a_0][a_m][*normal*][$L's\ address$]

To relay packets from node *L* to node *J* (see Figure 7.5), following modifications are made:

[4]This branch could have already been added if MART is used (see subsection 7.3).

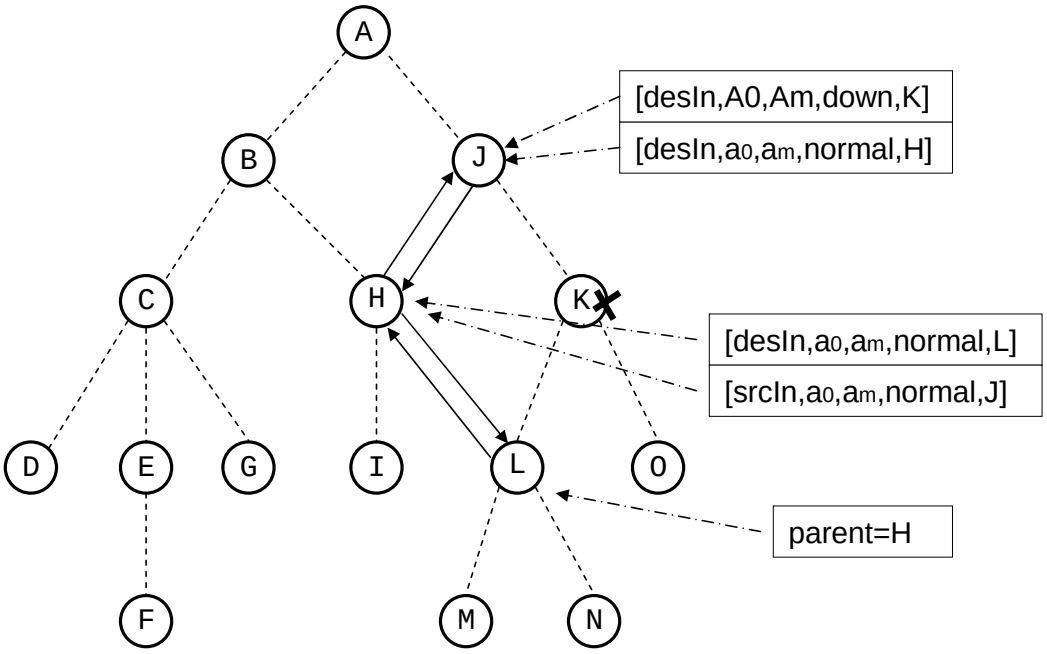


Figure 7.5: Data Forwarding after Tree Repair (Approach 2)

- Node *L* changes its parent from node *K* to node *H*.
- Node *H* adds a virtual branch:

  [*srcIn*][a_0][a_m][*normal*][$J's\ address$]

Note that not all data packets that leave branch *L* will be routed through the path *L-H-J*. For example, a data packet from node *M* to node *E* will actually follow the path *M-L-H-C-E*.

The above example only shows the repair for branch *L*. But actually one local repair will try to repair all detached branches.

Approach 2 is a local repair scheme and therefore enjoys better scalability than approach 1. The drawback of approach 2 is that, to broadcast a message with a TTL larger than 1 makes it inappropriate for beacon-enabled

networks [5]. It is not infeasible to do broadcasting in a beacon-enabled network, but the delay incurred by sleeping nodes (if those nodes are also required to receive the broadcast message) is not acceptable for an on-demand repair scheme. To avoid service disruption, a routing protocol should be able to switch to another route if the current one breaks.

7.3 Meshed ART (MART)

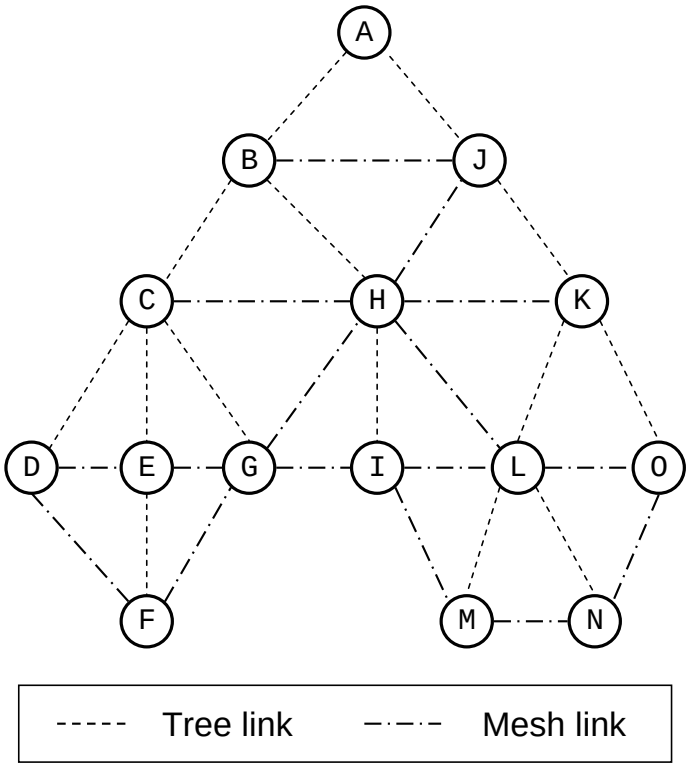


Figure 7.6: Meshed ART

A meshed ART (MART) can be formed on top of an ART. In Figure 7.6, besides the lines that represent tree links, additional lines (referred to as mesh lines here) are added so that the network now looks more like a mesh

than a tree. But from each individual node's point of view, the network is still a tree. Any two nodes connected through a mesh line treats each other as a child and adds an ARTT entry for each other. For example, node *K* treats node *H* as a child, and vise versa. Note that ancestors and descendents, no matter they are one level or multiple levels away from the node, are not meshed (i.e., not connected to the node through mesh lines).

By forming a MART, it is possible to route a packet through a shorter path. For instance, a packet from node *M* to node *I* can be transmitted directly (Figure 7.6), since from node *M*'s point of view, node *I* is its child. On the other hand, the original path in the corresponding ART is *M*-*L*-*K*-*J*-*A*-*B*-*H*-*I*. While this is an extreme example, there is a high probability that a shorter path can be found in MART than in ART. For example, from node *E* to node *H*, the path will be *E*-*C*-*H* instead of *E*-*C*-*B*-*H*. Note that the path from node *K* to node *G* is not *K*-*H*-*G*, but *K*-*J*-*B*-*C*-*G*. This is because that, although node *H* is treated as a descendent of node *K* in MART, node *G* is not. For simplicity and to avoid loops, node *H* only gives node *K* the address block of itself, and optionally those address blocks added due to tree repair, but not including the address block of branch 0 and address blocks added due to meshing behavior. So node *K* will not forward a packet destined for *G* directly to node *H*. Children added due to meshing behavior are likely at different tree levels. For example, nodes *J*, *K*, and *L* are children of node *H*, but at different tree levels. If node *H* has a packet for node *N*, it will find

that all the three nodes *J*, *K*, and *L* can relay the packet. In this case, node *H* will choose the node sitting at the largest tree level (that is, farthest from the root), namely, node *L*. Notice that node *H* can easily find out the relative tree levels of nodes *J*, *K*, and *L* through the address blocks allocated to them.

Another advantage of MART is that some SPFs are removed. For example, if the link between nodes *J* and *K* is broken, packets from node *K* to node *H* or *I* can still be routed. But packets from node *K* to other nodes such as node *B* can not be routed before tree repair is accomplished. MART also facilitates tree repair, since existing connections such as *K*–*H* can be readily used for tree repair purpose.

Chapter 8

Dual Routings

Chapter 8

Dual Routings

Logic tree routings such as cluster tree routing and meshed adaptive robust tree (MART) routing favor memory-constrained devices and are very suitable for short communication sessions. With a logic tree routing, a device can immediately begin to transmit packets to other devices once it joins the network, without going through the route discovery procedure. However, most tree routes are not optimal in terms of hop count. Logic tree routings also result in uneven traffic distribution. That is, a node at a smaller tree level normally needs to handle more traffic than a node at a larger tree level. As such, a node at a smaller tree level dies more quickly than other nodes due to quick battery depletion. Without other mechanisms, single point of failure (SPF) and network partition could easily happen in such a network. On-demand routings, on the other hand, are capable of finding optimal or near-optimal routes, and thus help reduce the message delivery latency. Nevertheless, compared with logic tree routings, they require more memory to store routing entries and also incur much control overhead. As most routes are formed on demand, the initial latency caused by route discovery is high. In general, on-demand routings are suitable for devices with sufficient memories, and favor long communication sessions. The *dual routings* (DR) approach proposed in this chapter combines MART with another on-demand routing and makes tradeoff between them according to the network conditions and application requirements.

8.1 Introduction

Tree routes are often non-optimal. This problem has motivated the practice of combining tree routing and non-tree (NT) routing. For example, ZigBee routing combines the cluster tree [29] routing with another on-demand routing, which is mainly based on the ad hoc on-demand distance vector (AODV) junior (AODVjr) [31] routing. Meshed adaptive robust tree (MART) routing can provide better routes than cluster tree, nonetheless routes in MART are still non-optimal in most cases. To optimize routes, another on-demand non-tree routing can be further put on top of MART. While packets can always be transmitted via the available MART route, the source can optionally trigger a route discovery procedure to find an optimal non-tree route to the destination. Tree routes along a single tree branch are generally optimal, if the tree has been optimized with respect to the routing cost(s) under consideration (e.g., hop count, link quality, and power level). One may has noticed that tree optimization can hardly be done when link quality is used as a routing cost. Generally there is little data traffic during the tree formation period, and therefore the link quality measured during this period does not reflect the real link status. In fact, it is not the link quality that determines the tree structure; rather it is the tree structure that significantly affects the link quality. However, considering that, along a tree branch, a node closer to the root normally has a worse tree link and that tree

Table 8.1: Non-Tree Table (NTT)

num_entry							
beg_addr_1	end_addr_1	$next\_hop_1$	$initiator_1$	$hops_1$	$cost_1$	$tree\_cost_1$	$tstamp_1$
beg_addr_2	end_addr_2	$next\_hop_2$	$initiator_2$	$hops_2$	$cost_2$	$tree\_cost_2$	$tstamp_2$
.....							

num_entry:	number of NTT entries
beg_addr_i:	beginning address of entry i (it is also the destination address)
end_addr_i:	ending address of entry i
$next\_hop_i$:	next hop via which entry i can be routed
$initiator_i$:	route discovery initiator of entry i
$hops_i$:	hops to the beginning address of entry i
$cost_i$:	cost to the beginning address of entry i (not used if hop count is the only cost)
$tree\_cost_i$:	cost from tree root to the beginning address of entry i
$tstamp_i$:	time when entry i is created or refreshed

links of sibling nodes tend to have similar link quality due to the interference between them, it still holds that tree routes along a single tree branch are generally optimal. Based on the above, optimal non-tree routes are expected to be orthogonal with high probability to tree routes in the sense that they mainly connect different tree branches. As a result, tree routes and non-tree routes interconnect all nodes and form a mesh.

8.2 Routing Table

Besides ART table (ARTT), another type of table called non-tree table (NTT) is defined in dual routings (DR) (Table 8.1). Each NTT entry provides not only an optimal route to address beg_addr_i, but also an auxiliary route to the whole address block [$beg\_addr_i + 1$, end_addr_i], if $end\_addr_i > beg\_addr_i$. An auxiliary route may or may not be optimal,

but it is in general better than the corresponding non-optimal MART route. In terms of cost, roughly we have:

$$\begin{aligned} ART\_route &\geq MART\_route \\ &\geq NT\_auxiliary\_route \\ &\geq NT\_optimal\_route \end{aligned}$$

Whenever a route is optimal, equal sign applies from there on.

8.3 Data Forwarding

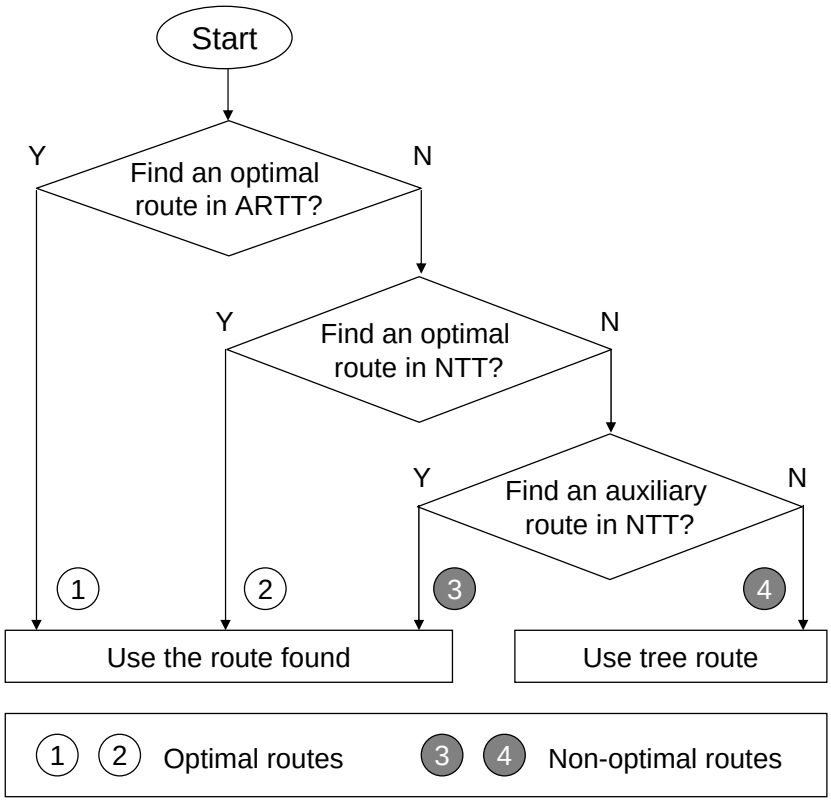


Figure 8.1: Data Forwarding

In DR, a local repair scheme is used for repairing broken links (see section 8.5). For simplicity and practically, a locally repaired tree route is still deemed optimal if the original one is optimal. This implies that any tree route recorded in an ARTT is deemed optimal. Therefore, a data packet can be forwarded using a route selected according to the flow depicted in Figure 8.1.

8.4 Route Discovery and Route Reuse

8.4.1 Route Discovery

When no optimal route is available to a destination, the source may trigger a route discovery procedure to find one, or just use the best available non-optimal route to relay packets when traffic duration is short or the available route is near-optimal. Route discovery proceeds as follows[1]:

1. The source initiates the route discovery procedure by unicasting a route request (RREQ) packet, whose format is given in Table 8.2, to the destination along the best available non-optimal route. The possible values of $route\_type$ are *optimal_NT*, *optimal_ART*, *near-optimal*, *non-optimal*, or *unknown*. However, for an RREQ, $route\_type$ does not take the value *optimal_NT* or *optimal_ART*. If a node does not have an optimal route, it can determine whether a near-optimal route exists

[1]Here all wireless links are assumed to be bi-directional.

Table 8.2: Route Request (RREQ) and Route Reply (RREP) Packet Formats

RREQ				
$route\_type$	dst_beg_addr	src_beg_addr	src_end_addr	src_tree_cost
$turning\_tree\_cost$	max_link_cost	$hops\_traveled$	$cost\_accumed$	TTL
(optional NTT fields included in a unicast RREQ)				
rec_fr_NTT				
beg_addr_i	$initiator_i$	$hops_i$	$cost_i$	$tree\_cost_i$
beg_addr_j	$initiator_j$	$hops_j$	$cost_j$	$tree\_cost_j$
.....				

RREP				
$route\_type$	dst_beg_addr	dst_end_addr	dst_tree_cost	src_beg_addr
src_end_addr	src_tree_cost	$hops\_traveled$	$hops\_total$	$cost\_accumed$
$cost\_total$				

$route\_type$:	route type (*optimal_NT*, *optimal_ART*, *near-optimal*, *non-optimal*, *unknown*)
dst_beg_addr:	beginning address of destination tree branch
dst_end_addr:	ending address of destination tree branch
dst_tree_cost:	cost from tree root to the destination
src_beg_addr:	beginning address of source tree branch
src_end_addr:	ending address of source tree branch
src_tree_cost:	cost from tree root to the source
$turning\_tree\_cost$:	cost from tree root to the node where two tree branches of a non-optimal tree route intersect
max_link_cost:	maximum link cost along the propagation path of an RREQ
$hops\_traveled$:	hops an RREQ or RREP has traveled
$hops\_total$:	total hops between the source and the destination (not larger than the $hops\_traveled$ in the RREQ unicast from the source)
$cost\_accumed$:	cost an RREQ or RREP has accumulated
$cost\_total$:	total cost between the source and the destination (not larger than the $cost\_accumed$ in the RREQ unicast from the source; if asymmetric links are assumed, $cost\_total$ is not needed and optimal routes for different directions should be discovered separately)
TTL:	Time to live (only in a broadcast RREQ)
rec_fr_NTT:	number of optional records included (only in a unicast RREQ), each comprising part of the fields from an NTT entry

according to the following:

- If either the node itself or its communication peer is close (here in terms of routing cost; the same below) to the tree root, the node has a near-optimal route to its communication peer.
- If the *turning_tree_cost* of a non-optimal tree route is close to either *src_tree_cost* or *dst_tree_cost*, this non-optimal tree route is near-optimal. Note that tree routes are symmetric.
- If the *tree_cost* recorded in an auxiliary NTT route is close to that of its communication peer, the node has a near-optimal route to its communication peer.

After receiving the unicast RREQ, the destination will first check if it has an optimal route to the source according to the flow shown in Figure 8.1. The destination may have an optimal route to the source, even if the source does not have one to the destination[2]. If no optimal route is found in NTT and ARTT, the destination further tries to find one using the route reuse scheme described in subsection 8.4.2. If still no optimal route is found, the destination will try to determine if it

[2]There are several cases that the destination has an optimal route to the source, while the source does not have one to the destination.

- The source is a descendent of the destination, but multiple tree levels away from the destination.
- The source is an end point of an optimal NTT route, but the destination is not.
- Both the source and the destination are along an optimal NTT route, but neither is an end point. In this case, the destination may find out that it has an optimal route to the source when receiving an RREQ unicast from the source (see subsection 8.4.2).

has a near-optimal route to the source. If an optimal or near-optimal route is found, the destination will unicast a route reply (RREP) packet, whose format is given in Table 8.2, to the source along this route. If $route\_type = optimal\_NT$ (that is, an optimal non-tree route to the source is available), as the RREP travels from the destination to the source, each node along the path adds or refreshes two NTT entries, one for the source and one for the destination, unless an optimal tree route already exists[3].

2. If neither an optimal route nor a near-optimal route to the source is found, the destination will then try to find an optimal route. If the $hops\_traveled$ in the RREQ unicast from the source is small, the destination will broadcast an RREQ with a time to live (TTL) equal to $hops2src\_min = min(hops\_traveled, hops2src)$, where $hops2src$ is the hop count to the source, derived from the NTT entry pointing to the source and deemed infinite if such an NTT entry does not exist; otherwise the destination will first unicast an RREP to the source as in previous step and then broadcast an RREQ with a TTL equal to $\lceil hops2src\_min/2 \rceil$.

3. After receiving a unicast RREP indicating that no optimal or near-optimal route has been found, the source first computes the minimum hop count, $hops2dst\_min = min(hops\_total, hops2dst)$, where

[3] An optimal tree route may be part of an optimal NTT route and, in this case, no NTT entries need to be created along this optimal tree route.

$hops2dst$ is the hop count to the destination, derived from the NTT entry pointing to the destination and deemed infinite if such an NTT entry does not exist. Then the source tries to find an optimal route to the destination if the destination has broadcast an RREQ (this can be determined by the value $route\_type = non\_optimal$ in the RREP unicast from the destination) or if it concludes there is no near-optimal route to the destination. In this case, the source will broadcast an RREQ with a TTL equal to $\lfloor hops2dst\_min/2 \rfloor$ if the destination has broadcast an RREQ, or otherwise with a TTL equal to $hops2dst\_min$. If neither a unicast RREP nor a broadcast RREQ is received from the destination within a certain timeout period after unicasting an RREQ, the source will broadcast an RREQ with a TTL equal to $min(network\_diameter, hops2dst)$ to find an optimal route to the destination.

As described above, RREQs can be broadcast by the source, the destination, or both. The rationale behind broadcasting RREQs from both the source and the destination is that the two broadcasts only cover roughly half of the nodes that would otherwise be covered by a single broadcast with a TTL equal to twice of that used by each of the two broadcasts. Figure 8.2 illustrates an example of broadcasting RREQs from both the source and the destination. The solid green arrows and blue arrows show the propagations of the two broadcasts from the source node *C* and the destination node *M*

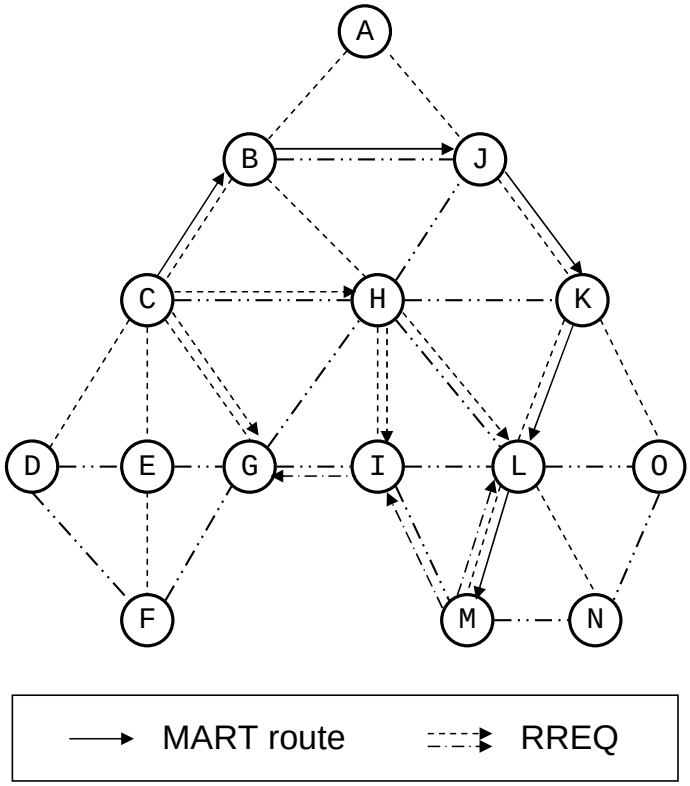


Figure 8.2: Route Discovery

respectively (not all transmissions are shown in the figure). Whenever an RREQ from the source meets an RREQ from the destination, a route is found and both RREQs will no longer be rebroadcast by the node where they meet. Note that in Figure 8.2, the transmission of RREQ from node *I* to node *G* happens before the transmission of RREQ from node *H* to node *I*; the former can not be stopped by the latter. An RREQ also stops when it reaches a tree branch to which the destination recorded in the RREQ belongs. The node where two RREQs meet (or where the RREQ is stopped by a tree branch) needs to unicast a gratuitous RREP packet to each of the

source and the destination to activate the routes in both directions. Multiple paths are likely to be found during the route discovery. How to utilize multiple paths is left to specific applications.

8.4.2 Route Reuse

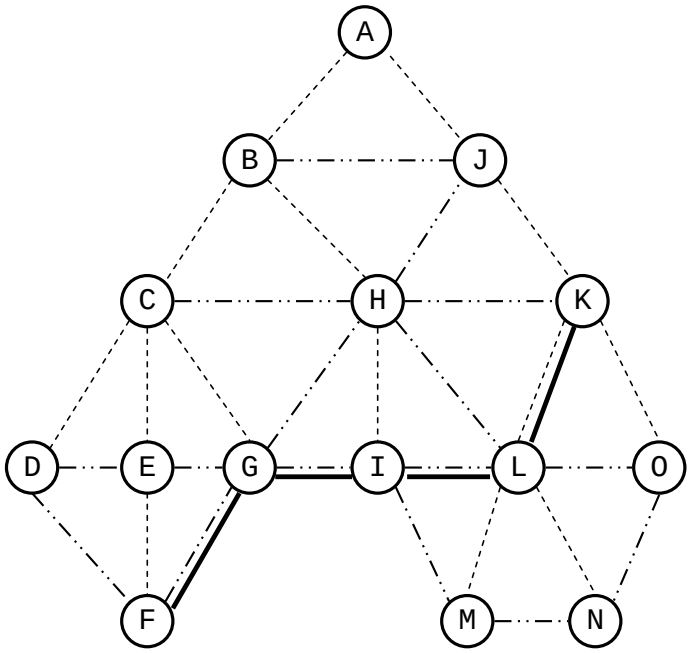


Figure 8.3: Route Reuse

Route reuse scheme makes full use of existing NTT routes. In Figure 8.3, there is an optimal NTT route between node *F* and node *K*. Any node along this NTT route will have an optimal route to each of node *F* and node *K*. But no node along this NTT route knows that it also has an optimal route to the intermediate nodes *G*, *I*, *L*, unless the destination happens to be its one-hop neighbor. The optional NTT fields included in a unicast RREQ,

however, will enable a node that has received this RREQ to derive an optimal route. This is simply done by checking if two nodes have a common pair of $(beg\_addr_i, initiator_i)$. Consider the example that node *L* needs to find an optimal route to node *G*; node *L* first unicasts an RREP to node *G* along the tree route *L*-*K*-*J*-*B*-*C*-*G*; node *G* may then find that it shares with node *L* an optimal NTT route going from node *F* to node *K*, thus concluding that it has an optimal route to node *L*, which is the same as that to node *K*. Note that, although node *G* also shares with node *L* the optimal NTT route going from node *K* to node *F*, it can easily find out by checking hop count information that node *F* and node *L* are in different directions.

8.5 Route Repair

Whenever a link is broken, the node having detected the problem will perform a local repair by broadcasting an RREQ with a TTL equal to *rpr_min_TTL*. If no RREP is received within a certain timeout period, the node increases the TTL by *rpr_inc_TTL* and tries again. This procedure continues until at least one RREP is received or the TTL reaches *rpr_max_TTL*. If multiple local routes are found, the node will select the best one based on the costs recorded in the corresponding RREPs. If the local repair is for a non-tree link, the node will unicast a gratuitous RREP along this local route so that all related nodes can also build a non-tree

routing entry to the original source. If the local repair is for a tree link, a bi-directional ARTT routing path will be built between the two nodes originally connected via tree link (they could be multiple tree levels away). If the local repair fails, a route error (RERR) packet will be unicast to the source and, upon receiving this RERR, the source will try to rediscover an route to the destination. The suggested default values for *rpr_min_TTL*, *rpr_inc_TTL*, and *rpr_max_TTL* are *3*, *1*, and $2\times$*rpr_min_TTL*.

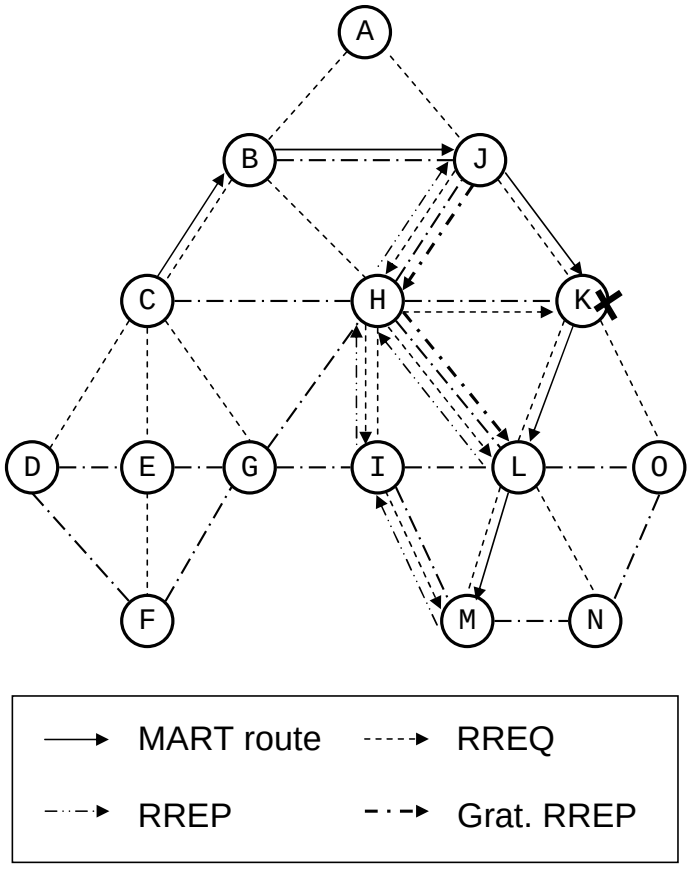


Figure 8.4: Route Repair

Figure 8.4 shows a tree link repair example. Node *K*, which is along the

routing path from node *C* to node *M*, fails. Node *J* successfully finds another local route *J-H-L* by broadcasting an RREQ with a TTL equal to *3* (not all broadcast propagation is shown in the figure). For relaying packets from node *J* to node *L*, following modifications are made (assume the address block of node *L* is [$addr_0$, $addr_m$]):

Node *J* adds a branch:

[*desIn*][$addr_0$][$addr_m$][*high*][$H's\ address$]

Node *H* adds a branch:

[*desIn*][$addr_0$][$addr_m$][*normal*][$L's\ address$]

For relaying packets from node *L* to node *J*, following modifications are made:

Node *L* changes its parent from node *K* to node *H*.

Node *H* adds a branch:

[*srcIn*][$addr_0$][$addr_m$][*normal*][$J's\ address$]

Note that not all data packets that leave branch *L* will be routed through the path *L-H-J*. For example, a data packet from node *M* to node *E* will actually follow the path *M-L-H-C-E*.

The above example only shows the repair for branch *L*. But actually one local repair will try to repair all detached branches.

8.6 Performance Evaluations

In this section, we evaluate the performance of DR and compare it with those of ART, MART, and AODV.

8.6.1 Performance Metrics and Experimental Setup

8.6.1.1 Performance metrics

To quantify the performance of DR, we define the following performance metrics.

- *Packet delivery ratio (pkt_d.r)*: The ratio of data packets successfully received at the destination to data packets sent at the source.
- *End-to-end hop count (e2e_h.c)*: The hop count a data packet traveling from the source to the destination, averaged over all successful end-to-end data transmissions within a simulation run.
- *End-to-end delay (e2e_dly)*: The transaction time of passing a data packet from the source to the destination, including time of all necessary processing, backoff as well as transmission, and averaged over all successful end-to-end data transmissions within a simulation run.
- *End-to-end communication efficiency (e2e_c.e)*: The total data bits successfully delivered divided by the total transmission time for both data packets and control packets.

8.6.1.2 Experimental setup

Simulations are carried out using NS2 [22]. All routings run on top of IEEE 802.15.4. The over air data rate is 250 Kbps (in the 2.4 GHz ISM band). There are total 100 nodes, which form a 10×10 grid. The distance between two horizontal or vertical neighbors is 10 meters and the transmission range is 12 meters. The tree root locates at the center of the network. The radio propagation model is two ray ground. Constant bit rate (CBR) is used in all experiments; and the packet rate is 1 packet per second as suggested in [112], with a packet size of 127 bytes at the PHY layer.

We apply two types of traffic: (1) peer-to-peer traffic, and (2) multiple-to-one traffic. The first type of traffic is used to study the general peer-to-peer behavior of DR. The second type of traffic targets wireless sensor networks, where traffic is typically between multiple source nodes and a sink. We refer to the second type of traffic as sink-type traffic hereinafter. Each 10 seconds a traffic flow is set up between two randomly selected nodes (for peer-to-peer traffic) or from a randomly selected node to the tree root (for sink-type traffic), and the duration of each traffic flow is determined in such a way that 5% of nodes in the network are transmitting data at any instant except the beginning and ending period of the simulation run. The total simulation duration is 2000 seconds, and the application traffic runs from 100 to 1900 second, leaving enough time for the experiment to shut down gracefully. Each experiment runs 10 times with different random seeds.

Table 8.3: Simulation Results for Peer-to-peer Traffic

	pkt_d.r (%)	*e2e_h.c (hop)*	*e2e_dly (sec)*	*e2e_c.e (bit/sec)*
ART	90.04	7.56	0.1107	15552
MART	91.68	6.94	0.0978	17603
AODV	97.08	6.34	0.0851	21159
DR	95.31	6.61	0.0907	21075

8.6.2 Simulation Results

Table 8.3 shows that, for peer-to-peer traffic, AODV performs better than MART, who in turn outperforms ART. This result is of no surprise. The main reason is that ART, like other tree routing protocols, suffers for non-optimal routes. By meshing ART, the situation is improved. For example, the end-to-end hop count decreases from 7.56 for ART to 6.94 for MART. Differences in other performance metrics are basically resulted from the difference of hop count. When compared with AODV, MART has a $97.08\% - 91.68\% = 5.4\%$ drop in throughput, a $(6.94 - 6.34)/6.34 \approx 9.4\%$ increase in end-to-end hop count, a $(0.0978 - 0.0851)/0.0851 \approx 14.9\%$ increase in end-to-end delay, and a $(21159 - 17603)/21159 \approx 16.8\%$ decrease in end-to-end communication efficiency. The performance of DR ranks the second. It is not as good as, but quite close to, that of AODV. The reason is that DR uses both optimal and near-optimal routes, which results in a less memory consumption and less control overhead. The slightly difference in end-to-end hop count ($6.61 - 6.34 = 0.27$) accounts for the performance

Table 8.4: Simulation Results for Sink-type Traffic

	pkt_d.r (%)	*e2e_h.c (hop)*	*e2e_dly (sec)*	*e2e_c.e (bit/sec)*
ART	91.44	4.26	0.0776	21841
MART	91.44	4.26	0.0776	21841
AODV	89.36	7.98	0.7035	13363
DR	91.12	4.26	0.0781	21720

differences between DR and AODV.

For sink-type traffic, ART and MART has exactly the same performance (Table 8.4). This is because the ART routes are already optimal and MART can not do better in this case. As can be seen from Table 8.4, sink-type traffic favors ART and MART. Compared with AODV, ART and MART have a $91.44\% - 89.36\% = 2.08\%$ gain in throughput, a $(7.98 - 4.26)/7.98 \approx 44.6\%$ decrease in end-to-end hop count, a $(0.7035 - 0.0776)/0.7035 \approx 89.0\%$ decrease in end-to-end delay, and a $(21841 - 13363)/13363 \approx 63.4\%$ increase in end-to-end communication efficiency. Notice that the AODV routes are far from optimal in this case, notwithstanding that AODV is claimed to be able to provide shortest routing paths. By further delving into our simulation results, we find that flooding, which is used in AODV for route discovery and route repair, is to be blamed. Although some reliable broadcast schemes have been proposed [30, 119], most broadcast schemes are unreliable. As a consequence, broadcast packets are often dropped due to collisions. As a consequence, AODV often

misses the shortest path when the network traffic load is not light or not evenly distributed. The performance of DR is almost the same as that of ART/MART. From Table 8.4, we can see DR has the same end-to-end hop count, indicating the routing path of DR is the same as that of ART/MART. However, according to DR, each source node tries to discover an optimal or near optimal route to the sink by sending a unicast RREQ, notwithstanding it already has such an optimal route. Note that a node can not find out that it has an optimal route towards its ancestors except for the one-hop parent.

In summary, although the performance of DR is not the best in any of the two simulation scenarios, it is quite close to the best performance. Not like ART/MART or AODV, which performs well in one scenario but not in another, DR maintains good performance in both scenarios.

8.7 Conclusions

Tree structures have long been exploited for routing purpose, especially for multicast, in both wired and wireless networks. The cluster tree algorithm proposed by Hester *et al*. [29] has been recommended and supported by IEEE in its new standard 802.15.4 [5]. And a slightly different version of cluster tree is used in ZigBee networks [30].

While cluster tree possesses special features, it faces the common problems other shared tree algorithms face, such as single point of failure (SPF)

and non-optimal routes. The address management, which is the basis of cluster tree, is not flexible and has substantially limited the application of cluster tree. To solve those problems, we proposed a new type of tree called adaptive robust tree (ART) and its meshed form, meshed ART (MART) in chapter 7. Another routing protocol, called dual routings (DR), is proposed in this chapter to further improve the performance of ART/MART. By combining MART with another on-demand wireless routing, DR can find optimal or near-optimal routes more efficiently than other on-demand wireless routings such as AODV [11] and AODVjr [31], whose route discovery is based on blind flooding. Our simulation results show that DR maintains good performance for both peer-to-peer and sink-type communications.

Chapter 9

A Scalable Topology-guided Distributed Link State Wireless Mesh Routing

Chapter 9

A Scalable Topology-guided Distributed Link State Wireless Mesh Routing

A distributed link state (DLS) scheme is proposed and put on top of the adaptive block addressing (ABA) scheme (see chapter 7) in this chapter. The new routing, as a whole, is named as topology-guided distributed link state (TDLS) wireless mesh routing. The network topology reflected in logic addresses is used as a guideline to tell towards which direction (rather than next hop) a packet should be relayed. The next hop is derived from each relaying node's local link state table. While its complexity is similar to that of adaptive robust tree (ART) and meshed ART (MART), TDLS improves the quality of routes, in terms of hop count or other routing cost metrics used, robustness, and load balancing. TDLS scales well with regard to various performance metrics. The ability of TDLS to provide multiple paths also precludes the need for explicit route repair, which is the most complicated part in many wireless routing protocols.

9.1 Introduction

Wireless mesh networking is a promising wireless technology for numerous applications [103], and especially appeals to those applications that can not be directly supported by other wireless technologies [104]. While considerable research efforts are still needed to fully materialize its po-

tential, it is conceivable that wireless mesh networking will play a pivotal role on the next wireless frontier. Besides large companies (such as Intel, Microsoft, Samsung, Nokia, to name but a few), industry alliances (such as ZigBee, Wi-Mesh, and recently-formed SEEMesh) that are keenly involved in the research on wireless mesh networks, several IEEE standards groups [105–107] have also established sub-working groups, including IEEE 802.11s, IEEE 802.15.5, and IEEE 802.16a/d/e/f, to work on new standards for wireless mesh networks.

Among the research issues to be addressed about wireless mesh networks are capacity and range enhancement, privacy and security, scalable multi-hop routing, energy conservation, low cost, self-configuration, zero or minimal maintenance, and coexistence with other wireless networks. In this chapter, we propose a scalable wireless mesh routing protocol, called topology-guided distributed link state (TDLS), for wireless mesh personal area networks (WMPANs), which is at the low end on the continuum of wireless mesh networks in terms of data rate. The IEEE 802.15.4 standard [5] specifies the physical (PHY) layer and the medium access control (MAC) sublayer for low rate wireless personal area networks (LR-WPANs), and ZigBee Alliance [102] has been developing other upper layers. WMPANs, as an enabling technology, possess some unique design features and show promise to bring ubiquitous networking into our lives [111]. Routing protocols designed for wireless mobile ad hoc networks (MANETs) such

as ad-hoc on-demand distance vector (AODV) routing [11] or its simplified version AODV junior (AODVjr) routing [31], dynamic source routing (DSR) [12], optimized link state routing (OLSR) [13], and topology broadcast based reverse-path forwarding (TBRPF) routing [14], can be directly applied to some mesh networks. Standard Internet routing protocols such as open shortest path first (OSPF) [15] protocol and routing information protocol (RIP) [16] can also be used if the network topology is relatively stable. However, those routing protocols may be too cumbersome for WMPANs, given the constraints on storage, power supply, and computational capacity of devices in those networks. TDLS, on the other hand, is a lightweight scalable routing protocol that well caters to the requirements of resource-constrained networks such as WMPANs and wireless mesh sensor networks (WMSNs). In TDLS, the network topology reflected in logic addresses assigned during the setup of a network is used as a guideline to tell towards which direction (rather than next hop) a packet should be relayed. The next hop is derived from each relaying node's local link state table. This approach makes TDLS scale well. A plethora of wireless routing protocols exploit hierarchical routing structure to achieve scalability [95, 113–117], whereas TDLS achieves the same goal by using a simple flat routing structure. Another feature of TDLS is that, by providing multiple paths, it precludes the need for explicit route repair, which is the most complicated part in many wireless routing protocols.

The remainder of this chapter is organized as follows. Details of the distributed link state scheme are covered in section 9.2, including the basic link state scheme, the extended link state scheme, link state generation, data forwarding, sanity/consistency checking, and link state maintenance. Simulation results are presented and analyzed in section 9.3. Finally, a summary is given in section 9.4.

9.2 Distributed Link State

The most distinct feature of TDLS is that it shuts out the tree repair, which is the most complicated part in tree routings. Maintaining simplicity is fundamentally critical for the success of WMPANs and WMSNs that comprise devices with constrained resource like small RAM size. For instance, Atmel [118] 8-bit processor has only a RAM of the order of 10 KB, which is even too small to house the AODV routing table in case of several hundreds of nodes. By contrast, TDLS only needs a tiny link state table, e.g., about 300 bytes for 30 neighbors, regardless of the network size. Notice that the above storage saving has not yet accounted for the saving on code size of the routing protocol.

TDLS is a fully-distributed scheme, as its name tells, which scales well. TDLS also maintains many advantages a proactive approach can have, for example, there is no initial route discovery latency and thus is suitable for

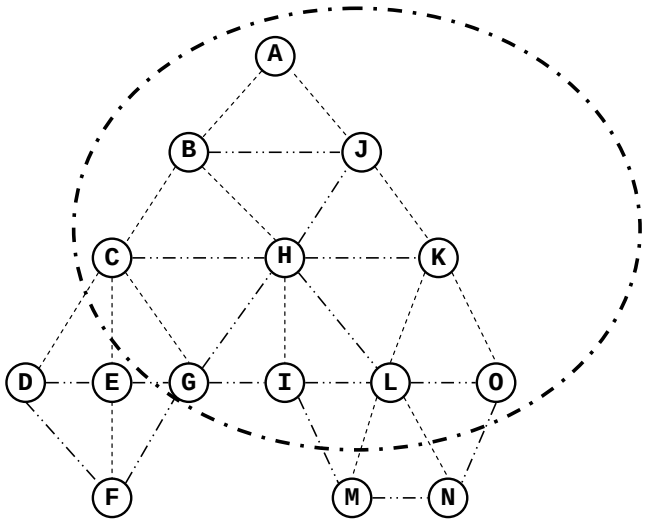


Figure 9.1: An Example of the Basic Link State Scheme

time critical sensor applications like light control within ZigBee [102]. It is also more efficient than most other proactive approaches, since no periodical update is needed after the link state generation stage (see subsection 9.2.3 and 9.2.5). Among other advantages are shorter paths (or better paths when cost metrics other than hop count are used in route selection), multiple paths, and robustness. In what follows, we present the details of TDLS, including the basic link state scheme, the optional extended link state scheme, link state generation, data forwarding, sanity/consistency checking, and link state maintenance.

9.2.1 The Basic Link State

In the basic link state (BLS) scheme, each node maintains a link state table (LST) for all its neighbors within *maxHops* (a parameter determined by specific applications) hops. Each neighbor's block address is logged in the LST so that the whole branch below it is routable. To forward a data packet, the LST is queried and the best path (when multiple paths are found) is used (see subsection 9.2.4 for details). Figure 9.1 illustrates a 3-hop BLS view of node *J*. In this example, nodes *D*, *E*, *F*, *M*, and *N* are not within 3 hops of node *J*, but they are still directly routable since they are the descendents of those nodes within 3 hops of node *J*.

9.2.2 The Extended Link State Scheme

The extended link state (ELS) scheme is optional in TDLS. In the ELS scheme, a node is not forced to use the predetermined *maxHops* parameter to build its LST. Although nodes are still required to exchange *maxHops*-hop neighbor information, a node can build an LST for *linkHops*-hop neighbors ($linkHops \leq maxHops$), depending on the node density around it. In general, a node will use a large *linkHops* if the node density is small around it, and vice versa. One of the goals of using DLS is to bypass broken tree routes. With high node density, a node is likely to find more paths that can be used to bypass a broken tree route. This improved reliability enables a node to build a relatively small LST.

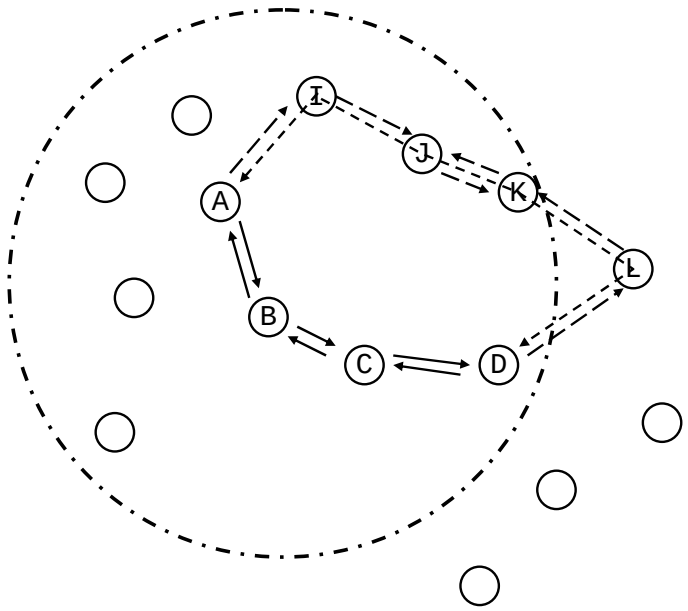


Figure 9.2: An Example of the Extended Link State Scheme

If the node density is very low around an area, even a *linkHops* as large as *maxHops* may not be enough. One solution is to allow nodes in this area to use a *linkHops* that is larger than *maxHops*. But since only *maxHops*-hop information is exchanged by default, all nodes in this area need to exchange *linkHops*-hop (*linkHops* $\geq$ *maxHops* in this case) information again. Here we propose to use another more efficient approach. After the basic LST is built, each node checks if it has multiple paths to each of the neighbors within *linkHops* hops. If not, it will unicast its complete LST to those neighbors. Upon receiving the LST, a node will unicast back its LST to the source. This in effect creates a (2 $\times$*linkHops*)-hop LST between related nodes. Figure 9.2 shows an example of the ELS scheme. After the

Table 9.1: Format of Hello Message

$begAddr$	$endAddr$	$tree\_level$	oh_nb_1, oh_nb_2, ..., oh_nb_k

$begAddr$:	beginning address of the address block owned by the node (this is also the address assigned to the node itself)
$endAddr$:	ending address of the address block owned by the node
$tree\_level$:	tree level of the node
oh_nb_i:	one-hop neighbor of the node

basic LST is built, node *A* finds that it only has one path to node *D* within *linkHops* (here 3) hops. So it will exchange LST information with node *D*. And both node *A* and node *D* will find out that there is another path between them, that is, *A-I-J-K-L-D*. Then each of them will record this path in the LST and also notify nodes *I*, *J*, *K*, and *L* so that those nodes can also update their LSTs. Finally, a second path is set up between node *A* and node *D*. Among all the neighbors within *linkHops* hops, the ancestors and descendents of a node (according to the tree structure) are more important than other nodes. So optionally a node can handle its neighbors differently, e.g., only guarantees that multiple paths are available to ancestors and descendents, or keeps more hop link state of ancestors and descendents than that of other nodes.

In summary, the ELS scheme is more reliable, efficient and adaptive than the BLS scheme, at the additional cost of complexity.

Table 9.2: Neighbor List

$begAddr_1$	$endAddr_1$	$tree\_level_1$	$hops_1$
$begAddr_2$	$endAddr_2$	$tree\_level_2$	$hops_2$
...	...	...	...
$begAddr_n$	$endAddr_n$	$tree\_level_n$	$hops_n$

9.2.3 Link State Generation

After the tree formation, nodes exchange link state information by periodically broadcasting several Hello messages with a TTL of *maxHops*. The Hello message format is given in Table 9.1.

The LST of a node, which consists of a *maxHops*-hop neighbor list (see subsection 9.2.3.1) and a connectivity matrix (see subsection 9.2.3.2), is updated upon the reception of each Hello message.

9.2.3.1 Neighbor list

Each node updates its neighbor list (Table 9.2) upon the reception of each Hello message. Not only the source of the Hello message is added into the neighbor list, but also the one-hop neighbors of the source[1], with the *endAddr* and *tree_level* marked as *unknown* temporarily. The *unknown endAddr* and *tree_level* will be replaced with actual values when a Hello message is received from the corresponding neighbor. If no Hello message is received from some neighbors during the whole Hello message exchange

[1]The one-hop neighbor list included in an incoming Hello message with a TTL of 1 (i.e., this is the last hop of the Hello message) is not used for any purpose, since some of the nodes in the one-hop neighbor list may be (*maxHops* + 1) hops away from the receiver.

procedure, a node can solicit for *endAddr* and *tree_level* information by broadcasting a message to its one-hop neighbors, including all the neighbors whose *endAddr* and *tree_level* are missing. Each one-hop neighbor received the message will reply if it can provide the *endAddr* and *tree_level* information of one or more neighbors included in the message. The field *hops* is computed according to the connectivity matrix described in subsection 9.2.3.2.

9.2.3.2 Connectivity matrix

From the one-hop neighbor information included in each Hello message[1], a node can construct a connectivity matrix for neighbors recorded in the neighbor list given in subsection 9.2.3.1. Table 9.3 illustrates one example.

The field *hops* of each node in the neighbor list can be calculated using the connectivity matrix. First the field *hops* of each node is set to *infinity*. Then, all nodes directly connected to *me* are one-hop neighbors (nb_2, nb_{n-1}, ... in above example). Next, all nodes directly connected to one-hop neighbors (and having a *hops* of *infinity*) are two-hop neighbors (nb_1, nb_3, ... in above example). This procedure continues until hop numbers of all neighbors are populated.

Table 9.3: An Example of Connectivity Matrix

	me	nb_1	nb_2	nb_3	...	nb_{n-2}	nb_{n-1}	nb_n
me	-	-	+	-	...	-	+	-
nb_1		-	+	-	...	+	-	-
nb_2			-	+	...	-	-	-
nb_3				-	...	+	-	-
...					...	...	...	...
nb_{n-2}						-	-	+
nb_{n-1}							-	-
nb_n								-

Note:

(1) The plus or minus sign ("+" or "-") at the cross cell of two nodes indicates they are or are not directly connected (i.e., they are or are not one-hop neighbors);

(2) For bi-directional links, the matrix is symmetric, so only half of the matrix is needed as shown here.

(3) Hop information can be calculated using the connectivity matrix. Here we have:
1-hop neighbors: nb_2, nb_{n-1}, ...
2-hop neighbors: nb_1, nb_3, ...
3-hop neighbors: nb_{n-2}, ...
4-hop neighbors: nb_n, ...

9.2.4 Data Forwarding

The pseudo code given in Table 9.4 elaborates on the procedure of selecting the next hop for data forwarding. When multiple neighbors are available for selection (see line 11 and line 32 of the pseudo code given in Table 9.4) and there are no other cost metrics indicating one neighbor is preferred over another, we can randomly select one neighbor for load balancing purpose. However, to mitigate "out of order" problems, it may be better to stick to

Table 9.4: Pseudo Code for Data Forwarding

```
1:  func_nextHop(dst)
2:      nb_found = search nb list for the lowest (i.e., with the largest tree level)
                   neighbor who is the ancestor of dst but is not my ancestor;
3:      if nb_found    //going down
4:          next_hop = getOneHopNb(nb_found);
5:          return next_hop;
6:      else if dst is not my descendent    //going up
7:          found = is there a neighbor who has a tree level less than mine?
8:          if found
9:              hops2root = the min (hops + tree_level) found among nb's that have
                            a tree level less than mine;
10:             minHops = the minimum hops found among neighbors that have
                          a (hops + tree_level) of hops2root;
11:             nb_found = select one of the neighbors that have a (hops + tree_level)
                           of hops2root and a hops of minHops;
12:             next_hop = getOneHopNeighbor(nb_found);
13:             return next_hop;
14:         else    //should go up, but can't
15:             return no_next_hop;
16:         end if
17:     else    //should go down, but can't
18:         return no_next_hop;
19:     end if
20: end func

21: func_getOneHopNeighbor(nb_found)
22:     mark the hop_number of each neighbor as infinity;
23      current_hops = hop number of the nb_found;
24:     while current_hops > 1
25:         for each neighbor nb_i with a hop_number of current_hops
26:             for each neighbor nb_j directly connected to nb_i
27:                     hop_number of nb_j = current_hops - 1;
28:             end for
29:         end for
30:         current_hops = current_hops - 1;
31:     end while
32:     return one of the neighbors with hop_number of 1;
33: end func
```

one neighbor for a while once it is selected rather than randomly select one neighbor each time. If no next hop can be found, a ring search should be performed. Ring search can be done by exchanging Hello messages as in link state generation stage, but with an incremental TTL.

9.2.5 Sanity/Consistency Checking

To reduce communication overhead and interference, no periodic Hello messages are broadcast after the link state generation stage. During the data transmission stage, Hello messages are only broadcast upon the detection of link failures, link recoveries, or new neighbors. If a node misses some Hello messages, its link state is likely to be inaccurate. Inaccurate link state can result in not only the selection of detour routes but, more seriously, routing loops.

One way to promptly detect inaccurate link state without using periodic Hello messages is to include one-bit *up-down* flag and the so-called *virtual tree level* of the relaying node in each message being relayed. The *up-down* flag indicates whether the previous hop is forwarding the message up or down in terms of tree level. The *virtual tree level* is defined as follows:

$$vTL = \left\{ \begin{array}{ll} nbL - h2Nb & (if\ going\ down) \\ nbL + h2Nb & (if\ going\ up) \end{array} \right\} \tag{9.1}$$

where,

nbL is the tree level of *nb_found* given in line 21 of the pseudo code in Table 9.4;

h2NB is hops to *nb_found*.

Let

$flag_1 =$ the *up-down* flag included in an incoming message;

$vTL_1 =$ the *virtual tree level* included in an incoming message;

$flag_2 =$ the *up-down* flag of the receiver of the message;

$vTL_2 =$ the *virtual tree level* of the receiver of the message.

then it follows that:

$$\left\{ \begin{array}{ll} vTL_2 - vTL_1 \geq 1 & (if\ flag_1 = down) \\ vTL_1 - vTL_2 \geq 1 & (if\ flag_1 = up) \\ or\ flag_2 = down & \end{array} \right\} \tag{9.2}$$

If the receiver calculates $flag_2$ and vTL_2 using only (*maxHops*-1)-hop link state information (i.e., one hop less than that used by previous hop), then only the equal sign "=" should be applied in equation (9.2).

A more efficient way is to only include the *up-down* flag and the *h2Nb* value in the message[2]. Similarly we define:

$flag_1$ = the *up-down* flag included in an incoming message;

$h2Nb_1$ = the *h2Nb* value included in an incoming message;

$flag_2$ = the *up-down* flag of the receiver of the message;

[2]The *virtual tree level* has the same bit size as an assigned address, but *h2Nb* only needs several bits (e.g., 3 bits).

$h2Nb_2$ = the *h2Nb* value of the receiver of the message.

If all the above values are calculated using *maxHops*-hop link state information, then there is no relationship between $h2Nb_1$ and $h2Nb_2$. This can be seen from equations (9.1) and (9.2). Equation (9.2) gives the relationship between two *vTL*s. There is no relationship between two *nbL*s if they are both calculated using *maxHops*-hop link state information (the previous hop and the receiver have different *maxHops*-hop link state information). Thereby, according to equation (9.1), there is no relationship between $h2Nb_1$ and $h2Nb_2$ either. As such, we use *maxHops*-hop link state information to calculate ($flag_1$, $h2Nb_1$), but only (*maxHops*-1)-hop link state information to calculate ($flag_2$, $h2Nb_2$). In this case, we should have:

$$\left\{\begin{array}{ll} h2Nb_1 - h2Nb_2 = 1 & (if\, h2Nb_1 > 1) \\ the\, destination\, is & (if\, (h2Nb_1 = 1) \\ \quad my\, descendent & and\, (flag_1 = dn)) \\ myTL - preTL \geq 1 & (if\, (h2Nb_1 = 1) \\ & and\, (flag_1 = up)) \end{array}\right\} \tag{9.3}$$

where $myTL$ and $preTL$ denote the tree levels of the current node and the previous hop respectively.

If equation (9.2) or (9.3) does not hold, the link state information of the previous hop or/and the receiver is inaccurate[3]. In this case, the receiver

[3]However, that equation (9.2) or (9.3) holds is not a sufficient condition to conclude that the previous hop and the receiver have accurate link state information. But it does tell that the packet is approaching the destination and thus the routing path is loop-free.

sends its ($maxHops-1$)-hop connectivity matrix to previous hop, who then compares the received connectivity matrix with its own and notifies all its neighbors that are related to any mismatch record of the two connectivity matrices. Any node being notified will broadcast several Hello messages to update the link state of its neighbors.

9.2.6 Link State Maintenance

A node should broadcast several Hello messages with a TTL of *maxHops* if it detects its one-hop connectivity has changed due to link failures, link recoveries, or the detection of new neighbors. Transmission failures may result from link failures (including node failures), collisions, or background interference. So bringing down a route each time a transmission fails is not a proper practice. In TDLS, a neighbor to which a transmission has failed is first put in a probe list. Each neighbor in the probe list can have a state of either *unknown* or *down*. A neighbor with an *unknown* state is probed each *probeInterval* seconds after the last probe using a timer (timer-driven) or probed immediately each time it is selected as the next hop of a data transmission (data-driven). Although the neighbor with an *unknown* state can still be selected as the next hop like a normal neighbor, it is not actually used for transmitting any data packet. All data packets having this neighbor as the next hop are buffered or dropped if there is no enough memory. The probe continues until the link to the neighbor is recovered or the total probe

number, including both timer-driven probes and data-driven probes, reaches *max_probe_num*.

If a link is recovered, the corresponding neighbor is removed from the probe list and all packets buffered for this neighbor, if any, are forwarded to this neighbor. A link is considered recovered if a MAC acknowledgment (ACK) of a probe (if the routing layer has access to the status of MAC ACK) is received or any packet including the reply of a probe is received from that neighbor by the routing layer (or overheard by the MAC if overhearing is supported).

If the probe number reaches *max_probe_num* before the link is recovered, the state of the corresponding neighbor is changed to *down*. The connectivity matrix is updated accordingly and Hello messages are broadcast with a TTL of *maxHops*. After the broadcast of the first Hello message, all packets buffered for the neighbor, if any, will be routed via other routes. It is worth noting that data packets should not be routed via other routes before the original next hop is determined *down* and at least one Hello message has been broadcast to all *maxHops*-hop neighbors. The rationale is that, it is possible, though the probability is not high, data packets will be forwarded back to the current node since other nodes do not know the original next hop is unreachable.

A neighbor remains in the probe list if the link to it has been determined *down*, but it will be probed only by a timer (it will not be used as

the next hop for any data transmission) and the probe interval is increased after each probe, up to a maximum value *max_probe_interval*. For example, a neighbor with a state of *down* can be probed using intervals 2, 4, 6, ..., *max_probe_interval*, ..., *max_probe_interval* seconds. This guarantees that, if the link recovers, it will be detected within no more than *max_probe_interval* seconds.

9.3 Simulations

For performance evaluation and comparison, we simulate ART, MART, TDLS, and AODV [11] in this section.

9.3.1 Performance Metrics and Experimental Setup

9.3.1.1 Performance metrics

To quantify the performance of TDLS, we define the following performance metrics.

- *Packet delivery ratio:* The ratio of data packets successfully received at the destination to data packets sent at the source.
- *End-to-end hop count*: The hop count a data packet traveling from the source to the destination, averaged over all successful end-to-end data transmissions within a simulation run.

- *End-to-end delay:* The transaction time of passing a data packet from the source to the destination, including time of all necessary processing, backoff as well as transmission, and averaged over all successful end-to-end data transmissions within a simulation run.
- *Hop delay:* The transaction time of passing a data packet to a one-hop neighbor, including time of all necessary processing, backoff as well as transmission, and averaged over all successful end-to-end data transmissions within a simulation run.
- *End-to-end communication efficiency*: The total data bits successfully delivered divided by the total transmission time for both data packets and control packets.
- *Hop communication efficiency*: The sum of products of data bits successfully delivered and the corresponding hops the data having traveled divided by the total transmission time for both data packets and control packets. This can be equivalently defined as the inverse of the average time needed for one data bit to travel one hop.

Except *packet delivery ratio*, the above performance metrics can be classified into two classes: end-to-end based metrics and hop based metrics. The end-to-end based metrics are suitable for comparing different routing approaches, for example, ART, MART, TDLS, and AODV in a network. However, when different network scales are used for scalability study purpose, the end-to-end based metrics can no longer capture the scalability

feature. In general, the average hop count of traffic flows in a large scale network is larger than that in a small scale network. As a result, the end-to-end based metrics are not comparable in these two types of networks. In this case, the hop based metrics are used.

9.3.1.2 Experimental setup

Simulations are carried out using NS2 [22]. All routings run on top of IEEE 802.15.4. In order to compare the performance of ART, MART, and TDLS with that of AODV [11], which can not work properly in IEEE 802.15.4 beacon enabled mode, we choose non-beacon enabled mode for all simulations. The over air data rate is 250 Kbps (in the 2.4 GHz ISM band).

To address the scalability problem, five sets of scenarios are defined, all in a grid form:

- 49 nodes (7×7 grid)
- 100 nodes (10×10 grid)
- 196 nodes (14×14 grid)
- 400 nodes (20×20 grid)
- 784 nodes (28×28 grid)

The distance between two horizontal or vertical neighbors is 10 meters. The PAN coordinator (i.e., the tree root) locates at the center of the network and starts a PAN at time 0.0 second. Every other node joins the PAN at a time

randomly chosen between 0.0 and 5.0 second. The parameter *maxHops* takes the value 3. The radio propagation model adopted in all experiments is two-ray ground reflection[4]. And the transmission range is 12 meters. Constant bit rate (CBR) traffic is used and the packet rate is 1 packet per second as suggested in [112], with a packet size of 127 bytes at the PHY layer. Each 10 seconds a traffic flow is set up between two randomly selected nodes, and the duration of each traffic flow is determined in such a way that 5% of nodes in the network are transmitting data and another 5% are receiving data at any instant except the beginning and ending period of the simulation run. The total simulation duration is 2000 seconds, and the application traffic runs from 100 to 1900 second, leaving enough time for the experiment to shut down gracefully. Each experiment runs 10 times with different random seeds.

9.3.1.3 Hidden terminal issue

IEEE 802.15.4 suffers from hidden terminal problem as a result of lacking request-to-send (RTS) and clear-to-send (CTS) control messages [112]. Hidden terminal problem leads to the dropping of data packets and, more seriously, the tearing down of routes. A partial remedy is to apply a link failure threshold at an upper layer [30] and do not bring down a route before

[4]While a more realistic radio propagation model such as log-normal shadowing or Railey can be used, using a deterministic radio propagation model like two-ray ground reflection provides a common non-time-varying condition for comparing different routing approaches, thus more accurately capturing the differences among them.

the failure threshold is reached.

In excess of the link failure threshold, we also use the receiver oriented TDMA (ROT) (see section 5.2) to improve the performance of IEEE 802.1.5.4. While normally medium access control is not a topic of routing protocols, we would like to take the cross-layer design notion and honor this approach.

In TDLS, the setup of ROT does not incur additional communication overhead except for the inclusion of clock information in the Hello message, since it can be done together with the link state generation (see subsection 9.2.3). To avoid unnecessary delay when traffic load is light, ROT is only used for retransmissions. ROT is adopted in all routing protocols used in our simulations, including AODV. Our simulation shows that the performance improvement by utilizing ROT is significant, notwithstanding that this overly simple scheme can only prevent primary collisions.

9.3.2 Numerical Results

In terms of packet delivery ratio (Figure 9.3), AODV outperforms all other three routings, namely, ART, MART, and TDLS. When the number of nodes is larger than 400, AODV has a gain of about 8% in packet delivery ratio, when compared with TDLS. When the number of nodes is less than 200, both AODV and TDLS have a packet delivery ratio larger than 90%. For all scenarios, TDLS performs better than ART and MART. And MART

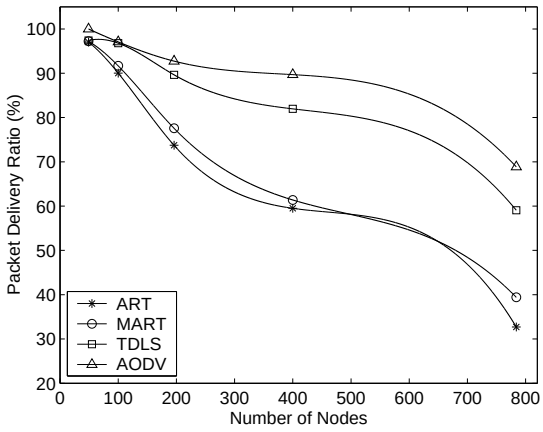


Figure 9.3: Packet Delivery Ratio

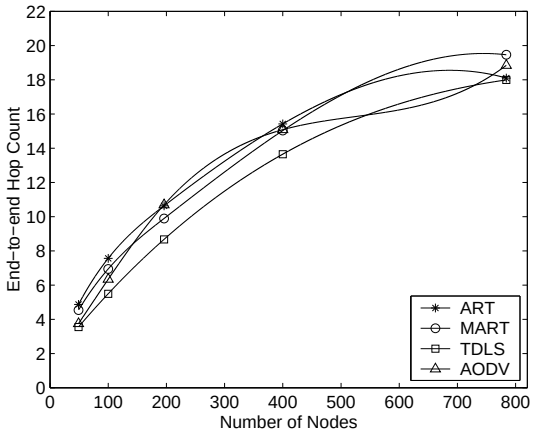


Figure 9.4: End-to-end Hop Count

in turn enjoys a slightly higher packet delivery ratio compared with ART. While the number of nodes roughly doubles each time, the packet delivery ratio drops smoothly, showing all schemes scale well in this regard.

The simulation results for end-to-end hop count are given in Figure 9.4. Surprisingly, AODV loses the battle to TDLS in this case. In fact, the performance of AODV is quite disappointing. Except for the 49 and 100 node scenarios, the end-to-end hop count of AODV is almost the same as

that of ART or MART. It is well known that tree routing such as cluster tree [29], ART, and MART suffers from non-optimal routing paths. In contrast, AODV is claimed to be able to provide shortest routing paths in most cases. Yet AODV falls short of achieving this goal in our simulations. By further delving into our simulation results, we find that flooding, which is used in AODV for route discovery and route repair, is to be blamed. Flooding is very detrimental in a wireless network, particularly one that lacks efficient means for coping with hidden terminal problems, as the case of WMPANs [112]. Although some reliable broadcast schemes have been proposed [30, 119], most broadcast schemes are unreliable. As a consequence, broadcast packets are often dropped due to collisions. To this end, it is of no surprise that AODV often misses the shortest path when the network traffic load is not light. As a matter of fact, sometimes AODV can not find a path at all. All our proposed schemes do not rely on flooding, though limited broadcast is sometimes used, mostly in a proactive way.

For both end-to-end delay and hop delay (Figure 9.5 and Figure 9.6), TDLS excels again. Followed is MART, who performs better than ART and AODV. As one can see from Figure 9.5, the end-to-end delay of AODV begins to shot up as the number of nodes reaches about 800. In terms of Hop delay, AODV peers ART, but gives in as the number of nodes exceeds about 700.

For end-to-end communication efficiency and hop communication effi-

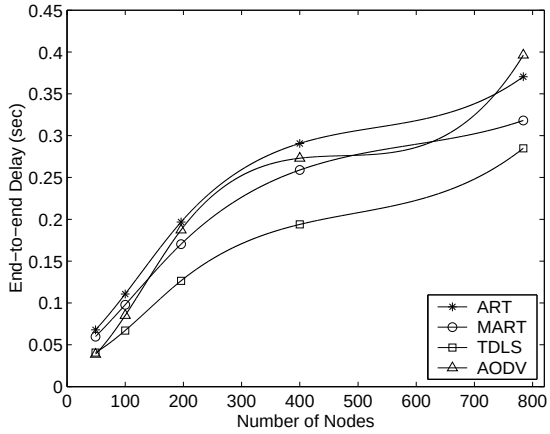


Figure 9.5: End-to-end Delay

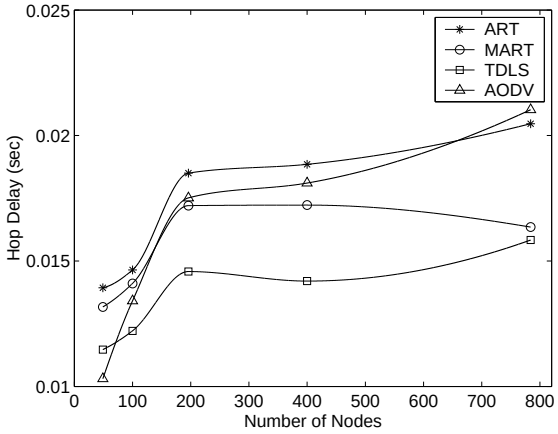


Figure 9.6: Hop Delay

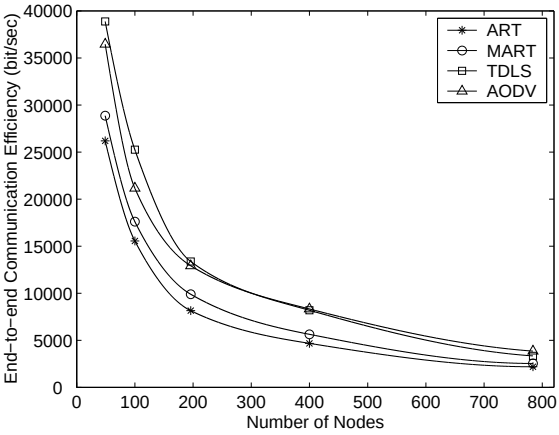


Figure 9.7: End-to-end Communication Efficiency

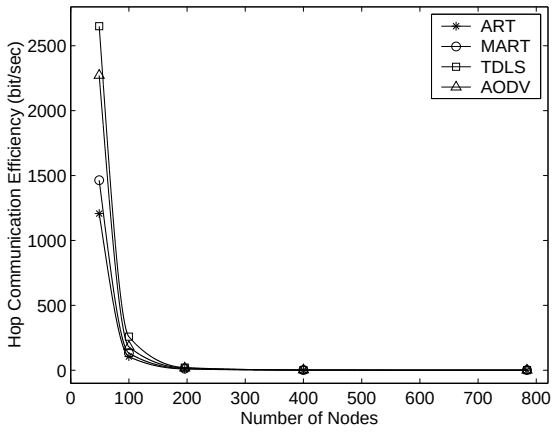


Figure 9.8: Hop Communication Efficiency

ciency (Figure 9.7 and Figure 9.8). The four routings rank in the order TDLS, AODV, MART, and ART. For end-to-end communication efficiency (Figure 9.7), TDLS beats AODV when the number of nodes is less than 200, but has a similar performance as the number of nodes continues to climb. This is also true with the hop communication efficiency (Figure 9.8). It is noteworthy that the hop communication efficiency for all routings plumbs as the number of nodes increases from 49 to 100, or equivalently, from Figure 9.4, the end-to-end hop count increases from about 4.3 to 6.3. The results indicate that, for the sake of communication efficiency, the end-to-end hop count is better to be limited to around 4.

9.4 Summary

An efficient scalable wireless mesh routing protocol, called topology-guided distributed link state (TDLS), is proposed in this chapter. TDLS

comprises two basic schemes, namely, the adaptive block addressing (ABA) scheme and the distributed link state (DLS) scheme. The ABA scheme is in charge of network autoconfiguration and logic address assignment. The DLS scheme utilizes the address block information provided by the ABA scheme as a guideline to extract the next hop for relaying a data packet. Compared with adaptive robust tree (ART) and meshed ART (MART), it is able to render better performance in terms of hop count or other routing cost metrics used, robustness, and load balancing. Our simulation results also show that the TDLS scheme outperforms AODV in almost every respect under the scenarios used in simulation, despite its simplicity. One exception is that, when the number of nodes is more than 400, AODV has a gain of about 8% in packet delivery ratio compared with TDLS.

One critical point not addressed by simulations, but mentioned in section 9.2, is that TDLS is far more memory-efficient than AODV, which makes it suitable for WMPANs and WMSNs that comprise devices with small RAM size. Although we evaluated AODV in networks with a node number up to 784 in section 9.3, in reality, AODV, as it is, would not have a chance in those networks, given the mere 10KB RAM owned by many 8-bit processors. Another point is that, as a proactive routing protocol, TDLS is suitable for time critical sensor applications, where again there is no room for reactive routing protocols such as AODV.

Part IV

Security Issues – Threats and Countermeasures

Chapter 10

A Systematic Analysis of the Threats Faced by Wireless Mesh PANs

Chapter 10

A Systematic Analysis of the Threats Faced by Wireless Mesh PANs

Wireless mesh personal area networks (WMPANs) offer device level wireless connectivity. They bring to light a host of new applications as well as enhance existing applications. Due to their low cost, low power consumption, and self-organization features, WMPANs are ideal for applications such as public security, battle field monitoring, inventory tracking, and home and office automation. Nevertheless, one critical issue, security, needs to be solved before WMPANs are commonly accepted. Pursuing security in WMPANs is a challenging task. On one hand, wireless communications are inherently susceptible to interception and interference. On the other hand, most devices in WMPANs are resource-constrained and lack physical safeguards. This chapter presents a systematic analysis of the threats faced by WMPANs with respect to the protocol stack defined by IEEE 802.15.4 and the ZigBee Alliance. Attacks are modeled and their impacts are evaluated. Some security problems within the current WMPAN security architecture are identified and remedies are suggested. Countermeasures of various attacks are also given.

10.1 Background and Motivation

With the release of IEEE 802.15.4 as well as advances in other fields such as embedded processors, micro-electromechanical systems (MEMS), and radio technologies, wireless mesh personal area networks (WMPANs)

(including wireless mesh sensor networks) are expected to thrive and affect our lives more than ever. Nevertheless, one critical issue, security, remains to be solved. It is more a matter of when than if that attacks can happen as WMPANs get more and more popular. Security could be one of the major obstacles in many applications.

Pursuing security in WMPANs is a challenging task. On one hand, wireless communications are inherently susceptible to interception and interference. On the other hand, most devices in WMPANs are resource-constrained and lack physical safeguards. Some security research has been done for wireless local area networks (WLANs) and wireless sensor networks (WSNs), but mainly having been carried out with respect to some specific attacks and only provide partial security for the system, which is far from enough. Since no standardized framework is available, different approaches have been proposed independently, which makes it either impossible to combine those approaches or difficult to obtain an optimal integrated solution. While some problems have been studied repeatedly and probably at different layers, some other problems are left untouched. Another problem is the circular dependency in some research. Because no systematic research has been done and no general guideline is present, some assumptions upon which the research is based are not true or practical. For example, a research of secure routing may assume that the lower MAC sublayer has a mechanism to securely distribute keys, whereas another research of key dis-

tribution may also assume, explicitly or implicitly, that a secure and reliable routing is available.

A well-structured systematic approach is essential for system level security. In this chapter, we study the security problems in WMPANs, from the view of different network layers. The work serves as a basis for building a system security framework. As 802.15.4 is the first and the only global standard for low rate wireless mesh personal area networks, we will mainly base our work upon this new standard.

The rest of this chapter is structured as follows. In section 10.2, we present a survey of the threats faced by WMPANs. We first outline our security objectives, and then classify the attacks and give out some specific examples which are separated into several groups based on different network layers. Then, in section 10.3, we introduce the NS2 simulator developed by us for WMPANs and provide the attack modeling results with discussions. In section 10.4, we focus on the design of secure WMPANs; remedies for problems within the current WMPAN security architecture are suggested and countermeasures of various attacks are given. Finally, in section 10.5, we wrap up this chapter with a summary.

10.2 Threats Faced by WMPANS

With the proliferation of WMPANs, the availability of security services for those networks will become a key issue. WMPANs play a vital role in many applications, including emergency applications. Yet, WMPANs, like any other wireless networks, are vulnerable to security attacks. Wireless communications are inherently not secure. Because they are broadcast in nature, and an adversary can eavesdrop on traffic, modify the messages, inject new messages, and replay old messages. The effectiveness of most existing network security gear is based on the availability of secure infrastructure, and, in many cases, relies on an accessible well-trusted third-party for bootstrapping and relaying security. Another fundamental assumption in those infrastructure-based networks is that the physical layer is at least marginally secure. Nevertheless, those conditions and assumptions no longer hold in WMPANs, and physical access control becomes impractical. In such an environment, a node could be compromised. A compromised node may exhibit Byzantine behavior, that is, deviate arbitrarily from its protocols. Thereby, a WMPAN may face attacks both from outside and within itself.

What makes the task more difficult is the stringent resource constraints on devices. Devices in a WMPAN have limited memory, processing power, bandwidth, and are normally battery-powered. To see the difficulties caused

by those limitations, let us take a look at the key establishment procedure. On one hand, the limited memory makes it infeasible to preload large number of keys into a device before deployment, which means devices may have to obtain the necessary keys in an on-demand fashion. Those keys obtained may not be permanently saved, not only because of memory limitation, but also because of security concerns. Due to the large number of devices that could be deployed in a WMPAN (e.g., a large scale sensor network) and the strong demand for low cost, a device is unlikely to get sufficient physical safeguards, and is susceptible to physical tampering or capture. In such a situation, a static key is like calling for troubles. On the other hand, the approach of obtaining keys on-demand and recycling them periodically can also prove to be expensive, as it consumes other precious resources like bandwidth and energy. While those resource limitations prevent WMPANs from using strong, complicated, and expensive protection measures, an adversary will by no means restrain from employing powerful attacks. Here is an envisioned scenario of such attacks. An adversary moves his powerful laptop (equipped with 802.15.4 compliant RF device) into a WMPAN and performs a brute-force attack by eavesdropping or capturing one of the devices. Or he may just simply keep on transmitting interference signals using large power, which could throttle a large number of devices in the network.

Wireless communications also pose another challenge for WMPANs. All the devices in wireless environments are required to cooperate with each

other for medium access. This is vital, as a node needs to get exclusive access to the shared channel to transmit its messages. Furthermore, due to the limited transmission range, a node may also need to rely on other nodes for relaying its messages destined for a distant node. The problem is that most protocols so far defined for wireless networks, including 802.15.4, do not include mechanisms to enforce a node to follow the protocols. A malicious or selfish node can easily take advantage of this. By deviating from the protocols, a node can get more access to the channel (by not following the medium access rules) or save its own resources (by not relaying messages). While the problem is more related to cooperation and fairness, it is equally, if not worse, harmful to other nodes and the whole network. The enforcement of such cooperation and fairness very much depends on proper security mechanisms.

In the following subsections, we first present the general security objectives we want to pursue. And then we identify some specific types of attacks, some of which are common to all wireless networks and some others only aim at WMPANS.

10.2.1 Security Objectives

- *Confidentiality:* There are two main attack classes, passive and active. A passive attack does not involve actions that affect the functions of the system being attacked. The typical passive attack is eavesdrop-

ping. An active attack, on the other hand, will normally try to disrupt the functions of the system, and often involves actions such as modifying or injecting messages. To launch effective active attacks, an adversary generally also eavesdrops. The main goal of confidentiality is to ensure that sensitive data are not disclosed to any entities other than the intended receivers. Confidentiality is the basic method to prevent passive attacks. The significance of preventing passive attacks is that passive attacks are often served as the bases of active attacks. Leakage of sensitive data could result in serious active attacks and devastating consequences. Another goal of confidentiality is to prevent an attacker from dissecting the ongoing traffic and extracting traffic pattern information such as source and destination, frequency, and length.

- *Integrity:* It is a basic requirement that a message is received as it is transmitted at the sender side. However, because of malicious attacks (or due to benign failures such as transmission collisions and radio propagation impairment, which are not our concerns here), a message may be corrupted in transit. Integrity guarantees that a message is transferred as it is, without replacement, deletion, injection, re-sorting, or any other modifications. Inasmuch as integrity is related to active attacks and as we mentioned above, wireless communications lack sufficient physical safeguards, the research focus should be how to detect and recover from attacks rather than prevent them from happening.

- *Authentication:* Authentication is used by a node to verify the identity of the peer node it is communicating with (entity authentication) or the origin of a message (data origin authentication). Entity authentication is generally applied in some initialization or configuration procedures such as the set-up of a connection-oriented session or the association of nodes during a tree formation. Without entity authentication, an adversary could masquerade another node, thus gain unauthorized access to resources and interference with the network operations without being identified and punished. In a message-based communication where no connection set-up or association occurs, data origin authentication is used to ensure the received message is really from the claimed sender. Authentication is important in WMPANs, especially in administrative tasks such as association, orphaning, coordinator relocation, superframe set-up, and beaconing. As injection is easy in wireless environments, it is crucial to guarantee that data used in any decision-making process are from a trusted source. Authentication is a stronger property compared with integrity in the sense that the latter normally embraces the former.

- *Freshness:* Unlike most general purpose networks, WMPANs are normally task specific. Information flowing in a WMPAN is often time-sensitive. In such networks, it is not enough to guarantee confidentiality and authentication. Replaying stale (but secret and authentic) mes-

sages can substantially disrupt the network operations and even cause catastrophes. For example, replaying the messages from a fire detection and warning system will lead to serious chaos. And replaying the messages from a traffic monitoring and control system such as an airport navigation system could mean huge loss of property and lives. One definitely does not want anybody to open his garage door by replaying a message either. Freshness ensures that the received message is recent and valid in the context of the applications.

- *Non-repudiation:* The techniques underlying authentication and non-repudiation are closely related, but there exist some differences between the purposes of them. While authentication is used to guarantee that a message is from the claimed sender, non-repudiation normally is employed not only to verify the source of a message, but also to prevent the source from denying having sent the message. Note that authentication alone does not necessarily ensure non-repudiation, though it is the basis of non-repudiation. Another difference is that non-repudiation can be applied to both transmission and reception. In some applications, preventing the receiver from denying having received a message is of equal importance to preventing the sender from denying having sent a message. Non-repudiation is useful for detection and isolation of compromised nodes.

- *Availability:* The goal of availability is to ensure the survivability of

network services despite attacks (e.g., denial of service (DOS) attacks) and normal failures (e.g., node failure and link breakdown). But normal failures are not our concerns here. As WMPANs are highly resource-constrained networks, they can easily suffer from attacks based on resource consumption. It is necessary to control those operations that consume large amount of network resources. In case attacks happen, there should be some mechanisms to isolate the attacks and prevent them from spreading. Some auto recovery mechanisms can also enhance the survivability of network services.

- *Fairness:* Fairness ensures that the network resources are used in a fair and efficient way. As aforementioned, cheating and unfairness are likely to happen in wireless environments. According to their purposes, attacks generally fall into two types, malicious attacks and rational attacks. The purpose of malicious attacks is to corrupt and destroy a system, even if at a high cost. By contrast, rational attacks are those attacks that bring in greater benefit to the attackers. A rational attacker will only attempt to launch such attacks if the reward by doing so is more than that by acting honestly. By fairness problems, we mean unfairness caused by rational attacks only.

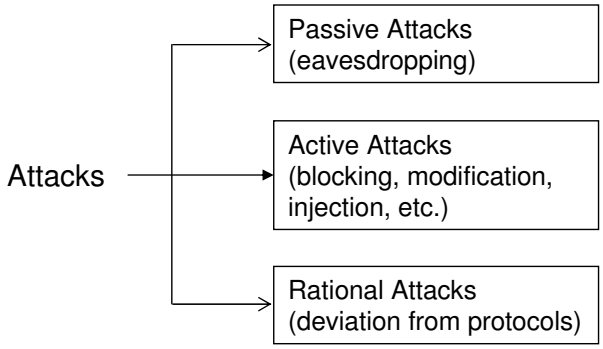
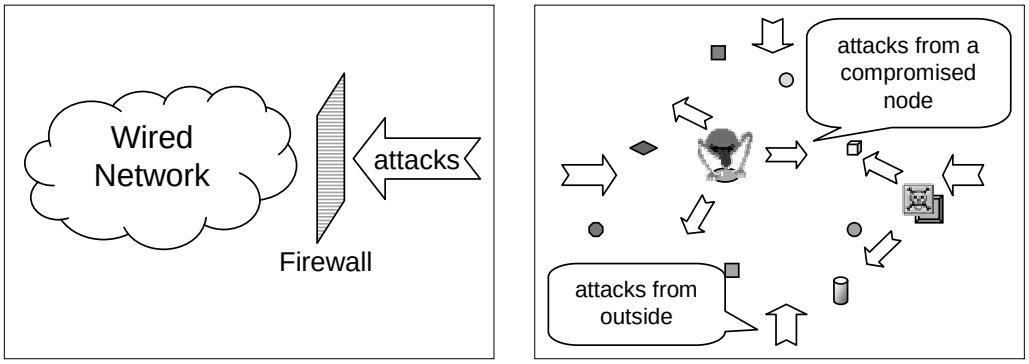


Figure 10.1: Classification of Attacks

10.2.2 Attacks

10.2.2.1 Classification and features of attacks in WMPANs

Besides passive attacks and active attacks, wireless networks also face fairness problems, referred to as rational attacks here (Figure 10.1). There is not always a clear cut-line between the active attacks and rational attacks. But by rational attacks, we emphasize that the primary attack goal is to maximize the attacker's benefit and minimize its cost, and the attacks are normally committed by deliberately deviating from protocols.



(a) A wired network with a firewall (b) A wireless network without a firewall

Figure 10.2: Attacks in Wired and Wireless Networks

Different from a wired network, in a wireless network it is often difficult for a node to distinguish outside attackers from inside attackers (Figure 10.2). As there are no clear physical margins in a wireless network, any central or static security control is unlikely to work. In a wireless network, there is no suitable single point where we can put the defense (like a firewall, which is often used in a wired network). Moreover, a node in a wireless network faces the risk of capture or hijack, indicating that no static trust should be assumed. In the following, we give some specific types of attacks that can be launched against different network layers in WMPANs.

10.2.2.2 PHY layer and MAC sublayer attacks

- *Jamming* (Figure 10.3 (a)) is the wireless equivalent of a denial of service (DOS) attack. It is simple and effective, especially in single frequency networks. It aims to weaken or zero-out the availability of system services. At PHY layer, a jamming attack can be easily carried out by continuously sending out radio signals using relatively high transmission power. All needed is a PHY compliant transmitter. The main frequency band of WMPANs is the ISM 2.4 GHz band. However, due to its global availability and license free feature, the 2.4 GHz is crowded with different devices such as wireless computers, high-end wireless phones, garage door openers, emergency radios, and Global Positioning Systems (GPS). This could mean that powerful jamming

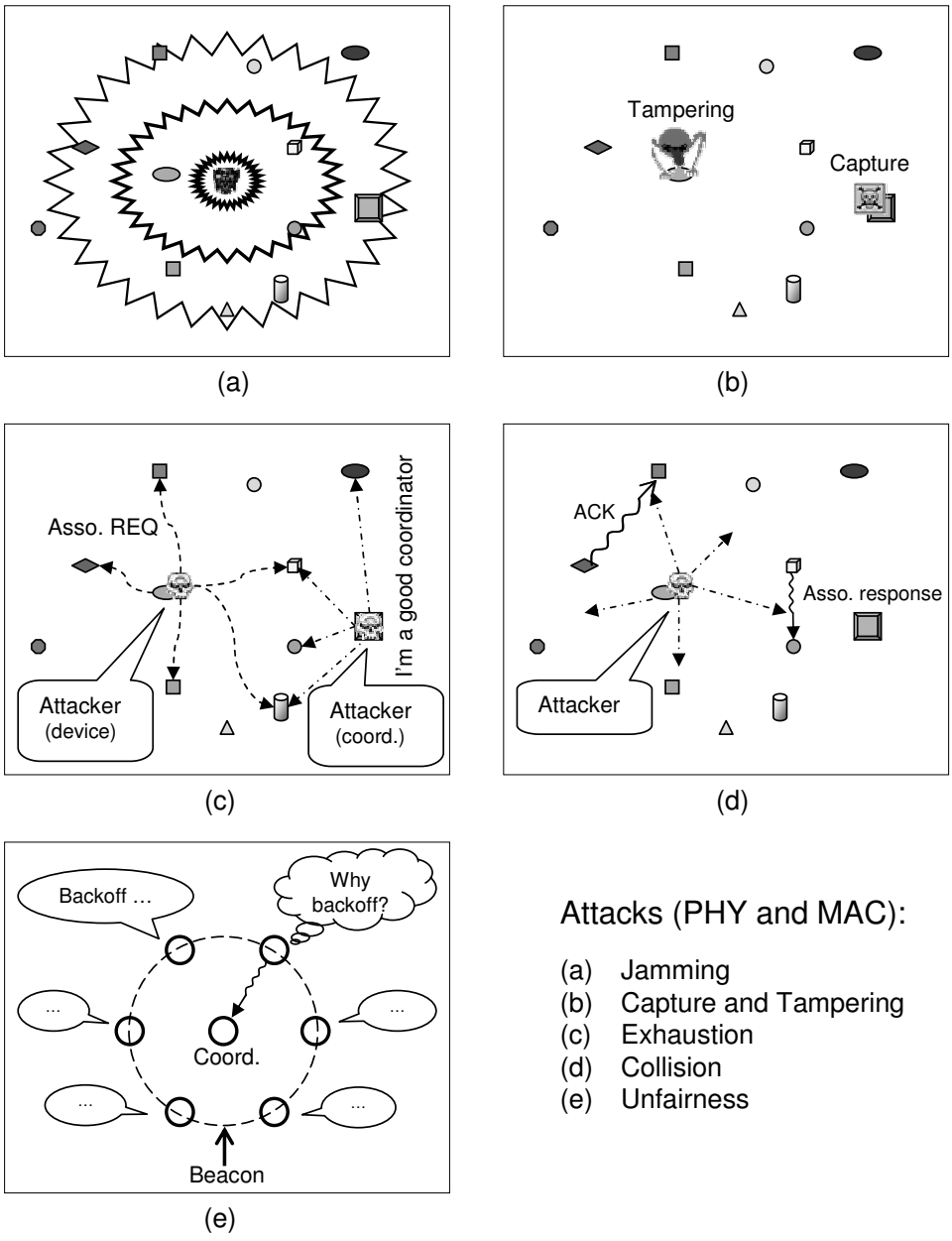


Figure 10.3: PHY Layer and MAC Sublayer Attacks

tools are readily at hand for adversaries. Jamming can also happen at upper layers if there are exploitable weaknesses in the protocol design. For example, at the MAC sublayer of 802.11, an adversary can launch denial of service attacks by periodically sending fake clear-to-

send (CTS) messages, with the duration field of each message greater than or equal to the interval between such messages. IEEE 802.15.4 does not employ the request-to-send (RTS) and clear-to-send (CTS) mechanism for medium access and therefore does not suffer from the CTS-based DOS attacks. However, DOS attacks are still possible at upper layers in WMPANs, for example, through the periodical transmission of "beacon-destroyer" messages in a beacon-enabled mode.

- *Capture and tampering* (Figure 10.3 (b)) are difficult to avoid in WMPANs, since the cost of sufficient physical safeguards defeats one of the important design goals, low-cost. While it is possible to provide strong physical safeguards for a few important devices such as the PAN coordinator or other devices holding sensitive information, most devices are left unprotected from such attacks. The power of this type of attack lies in that a captured or tampered device can be further used as an effective attack tool to continue compromising other devices, and the procedure can go on until the attacks flood through the whole network. Therefore, how to detect such attacks and minimize the damages upon the detection of them is the best thing we can do currently. One possible solution is to embed a small tampering-proof unit (like the SIM card in a cell phone) in each device and use it for storing some important data. Upon detection of tampering of the unit, a device will automatically trigger a procedure to erase all sensitive data in its memory and

maybe also send out a warning to its neighbors. But this approach is a double-side sword, as it could open the door to some new types of attacks. For example, an adversary could use the self-destruction feature of the device to destroy it (not necessarily physically) in a much easier way, or launch a DOS attack by impersonating the victim and sending out fake warning messages. Here design tradeoff is needed.

- *Exhaustion* (Figure 10.3 (c)) is also a type of DOS attacks from the point view of service availability. Since WMPANs are highly resource-constrained networks, exhaustion attacks in WMPANs are more destructive than those in wired networks. One common exhaustion attack is to exploit some initiation or connection procedures, like association procedures in WMPANs, that require both nodes involved to store some state values in their memory. In WMPANs, a device can try to associate with all the coordinators within its reach, notwithstanding the protocol demands that each device be associated only with one coordinator, and thus waste the storage of those coordinators or lead to association overflows at the coordinators (e.g., association overflows in the cluster-tree formation). A more powerful attack can be launched by a compromised coordinator, in which the compromised coordinator allures large number of nodes to associate with it by appearing to be a coordinator with high link quality (LQ) or low level in the tree. Such a coordinator will disregard the limitation of association num-

bers allowed by the protocol and will not concern about its memory constraint, as it does not really need to save all the association state data (but all the innocent devices will). After that, it can simply send out deliberately configured beacons to force all the devices to stay active for most of the time, resulting in quick battery depletions at those devices. Due to the one-hop characteristic of MAC sublayer, this kind of attacks are normally limited in neighborhood scope. Although an adversary can use a high power transmitter to reach nodes that are otherwise multi-hop away, the adversary will not be able to receive the messages from those far away nodes which is a necessary step for successful associations. However, a naive approach could still enable such attacks to devastate the network at the MAC sublayer. For example, if the selection of a coordinator for association is solely based on the quality of the coordinator, a device may keep trying to associate with a compromised coordinator even if the association has failed repeatedly. One solution would be to keep a copy of the association history, but this is not always easy due to the stringent storage memory. Furthermore, at a higher layer and especially through conspiracy among adversaries, such attacks can wreak destructions on large area of the network.

- *Collision* (Figure 10.3 (d)) is a difficult yet important challenge for wireless networks. Although much work has been done in the PHY

layer and the MAC sublayer to cope with this problem, it remains a hard point, even in a benign environment. In fact, jamming attacks are also based on collisions, but the attacks described here are more subtle and effective in terms of resource consumption. Such attacks are often launched by deviating from the protocols rather than blindly as in the jamming attacks. An attacker can selectively create collisions, especially to some sensitive control and management frames. For example, collision with an acknowledgment frame will cause the sender to back off exponentially, and collision with an association response frame will force the device to start the multi-step association procedure from the very beginning. Since collision at any part of a frame will corrupt the frame, one short transmission is enough to destroy a frame or even several frames. By creating collisions selectively, an adversary can make the attacks look like random collisions, thus gets away without being detected and punished. In a word, collision attacks are very effective and difficult to detect.

- *Unfairness* (Figure 10.3 (e)) is another problem most wireless networks face. It could substantially degrade the network performance, though normally not shut down the whole network. In wireless networks, one of the most possible places where unfairness could happen is the medium access part. Most wireless MAC sublayers use contention-based medium access schemes, in which all the nodes are

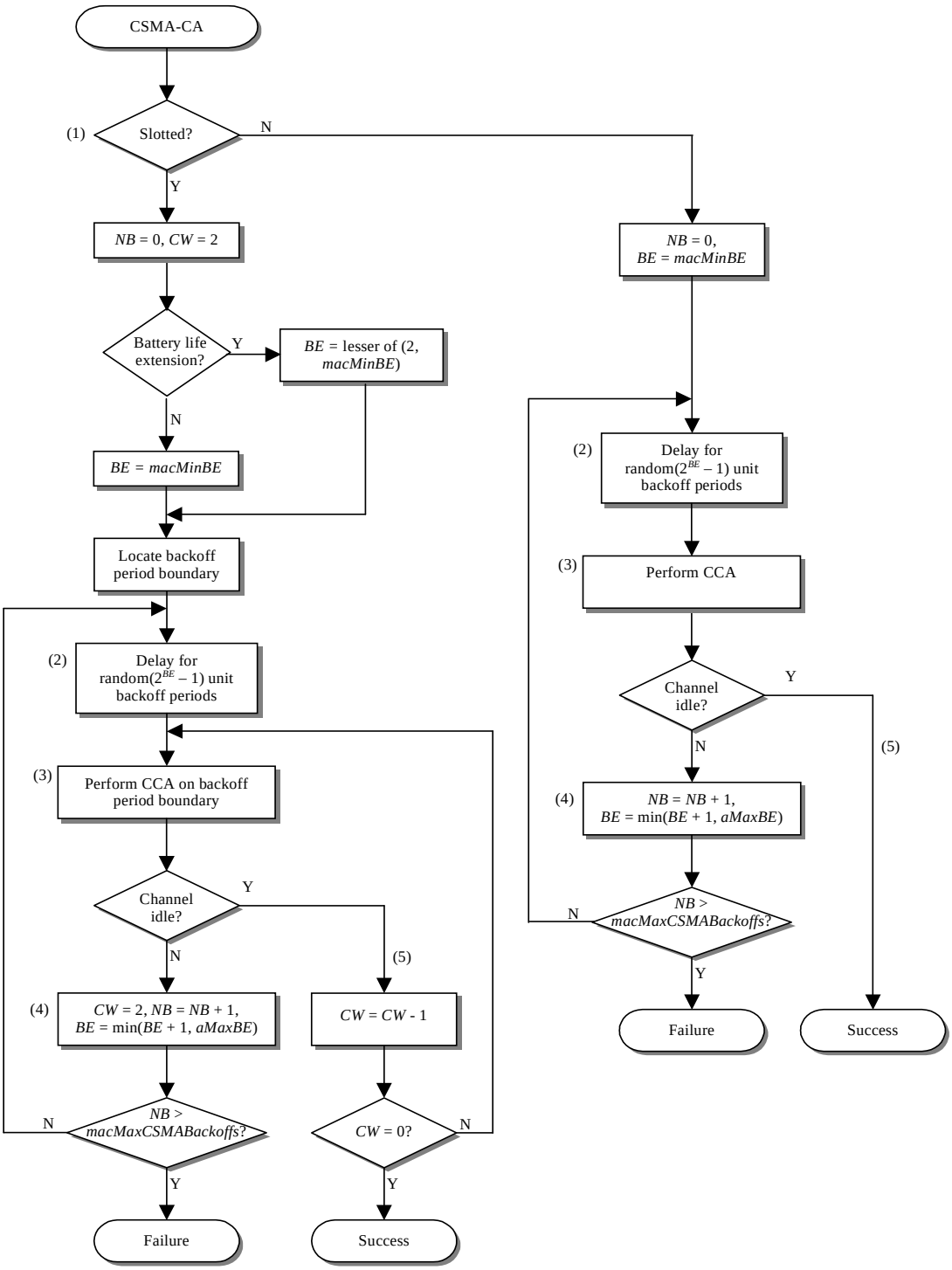


Figure 10.4: CSMA-CA Algorithm

assumed to compete for the control of transmission channels. Nodes must strictly follow the rules specified in those schemes to avoid col-

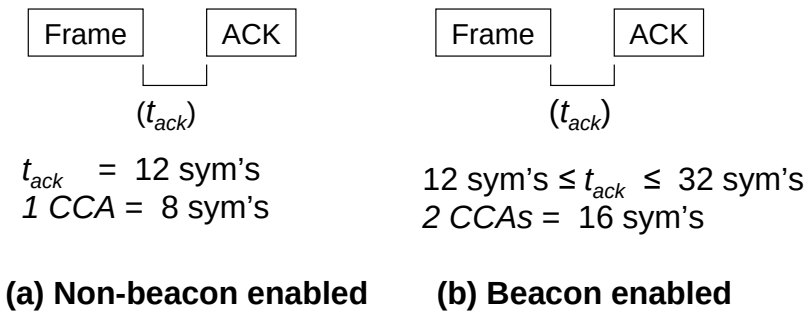


Figure 10.5: Acknowledgment of a Frame

lisions or to recover from them. To transmit a message, a node needs to seek a unanimous promise from all other nodes for an exclusive use of the channel. But almost all the available schemes have no effective ways to prevent a node from abusing those rules. In WMPANs, unslotted CSMA-CA is used for channel access in non-beacon enabled mode and slotted CSMA-CA for channel access during the CAP in beacon enabled mode, as illustrated in Figure 10.4 (from 802.15.4). Each node in a WMPAN needs to backoff $random(2^{BE} - 1)$ backoff periods (each period is 20 symbols) before it senses the channel. In beacon enabled mode, a node needs to perform two consecutive CCAs and the channel is considered idle only if both CCAs indicate so. In non-beacon enabled mode, only one CCA is needed. In such a scheme, any node will at least wait for $\{random(2^{BE} - 1)$ backoff periods + 2 CCAs $\geq$ 2 CCAs ≥ 16 symbols$\}$ at the beginning of the CAP in beacon enabled mode. Therefore a cheating node can capture the channel immediately after it receives a beacon, by sim-

ply skipping the backoff process as well as the CCA process. In non-beacon enabled mode, a cheating node can also get some priority in accessing the channel by using smaller backoff period and/or CCA duration. By transmitting messages one after another, a cheating node has a good chance to keep its control of the channel. Other nodes have little chance to transmit their messages before the cheating node finishes all its transmissions. Note that, although other nodes still have a small chance to sense the channel as idle if they by chance perform CCAs within the gap between a frame and its corresponding acknowledgment (when acknowledgment is required) as shown in Figure 10.5 or between an acknowledgment (or the frame itself if no acknowledgment is required) and the frame immediately following it [1], it helps nothing except creating collisions.

10.2.2.3 NWK layer attacks

The main function of NWK layer is routing. It is a big challenge for a routing protocol to function correctly and efficiently in the presence of Byzantine attacks which attempt to disrupt the routing service. Routing attacks can generally be characterized into the following types: routing disruption and resource consumption. A routing disruption attack tries to cause legitimate data packets to be routed in dysfunctional ways. A resource con-

[1] Unlike in 802.11, acknowledgment in 802.15.4 is not secured. In consideration of the low data rate and for simplicity, no strict mechanism is used to prevent other transmission from happening between a frame and its corresponding acknowledgment.

sumption attack, on the other hand, attempts to consume precious resources such as bandwidth, energy, memory, or computational power. In the following paragraphs, we give out some attack examples aimed at AODV/AODVjr routing and cluster-tree routing, both of which are used in ZigBee networks.

A. Attacks aimed at AODV and AODVjr

Routing disruption can be done by forging source and/or destination addresses, modifying the metrics in the RREQ and RREP packets, or creating a worm hole in the network [120]. We illustrate some of those attacks here.

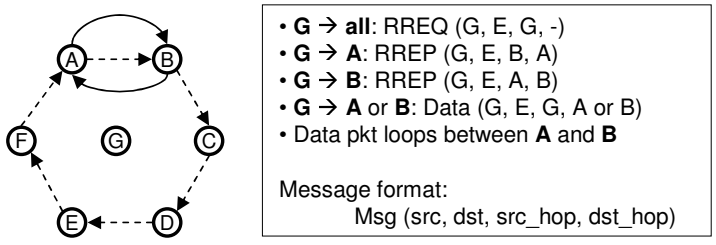


Figure 10.6: Loop in AODV

- *Loop in AODV*: By overhearing the transmission among its neighbors, a node can not only figure out how many neighbors it has, but also find out which two neighbors are adjacent to each other, that is, within one hop. With this information, a node will be able to create a loop among its neighbors, if no authentication is applied to the origin of a message. As illustrated in Figure 10.6, node *G* creates a loop between node *A* and node *B* by sending an RREP to node *A* designating that node *B*

is the next hop towards E, and then similarly tell node *B* that node *A* is the next hop towards node *E*. Whereas this small loop between two nodes is easy to detect, larger loop among three or more nodes will be more difficult to detect. Node *G* can also create loops among more neighbors, provided those neighbors form a close connection. It will also be possible to construct some large scale loops if more adversaries work in collusion. A loop is the most serious routing disruption, making a packet never reach its destination and making a node keep on transmitting. Obviously, loops are also a type of resource consumption attacks.

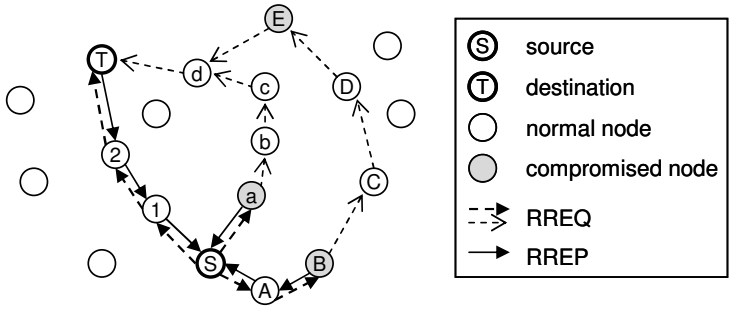


Figure 10.7: Detour in AODV

- *Detour in AODV*: A node can also attract traffic to itself by manipulating the route discovery procedure. For example, in Figure 10.7 (not all message propagation is depicted), node *a* can immediately send back a forged RREP (e.g., with one hop away from the destination node *T*) after receiving the RREQ. Although the normal path *S-1-2-T* is the

shortest one, but the source node *S* will regard *S-a-b-c-d-T* as the shortest path, ignoring the RREP from node *1*. To complete the path, node *a* will first initiate a route discovery from itself to node *T* and thus create a forward path from node *S* to node *T*. Then node *a* can send a gratuitous RREP to node *T* to set up the reverse path from *T* to node *S*. As a result, node *a* creates a detoured path between nodes *S* and *T* in both directions (node *B* can perform the same attack). Besides misleading the source node, a compromised node can do the same thing to the destination, if it is close to the destination. In Figure 10.7, node *E* can forge an RREQ and send it to node *T*, indicating that it is close to the source. Then node *T* will send an RREP to *E* (via node *d*) and ignore the RREQ from node *2*. After that, node *E* will build the path between nodes *S* and *T* in the same way that node *a* does. Once a compromise node succeeds in attracting traffic towards itself, it can simply drop all the packets received (like a black hole or sink hole), or more subtly, selectively drop some packets (like a gray hole).

- *Worm hole in AODV*: Figure 10.8 shows a worm[2] hole attack. By using a private network connection (e.g., via strong transceivers operating in a different channel or even directly wired together), two conspir-

[2]The term *worm* as applied to computers came from John Brunner's 1975 science fiction classic, *The Shockwave Rider*. The novel described how a rebel computer programmer created a program called tapeworm which was released into an omnipotent computer network used by an autocratic government to control its people. The government had to turn off the computer network, thus destroying its control, in order to eradicate the worm – from "Underground" by Suelette Dreyfus and Julian Assange.

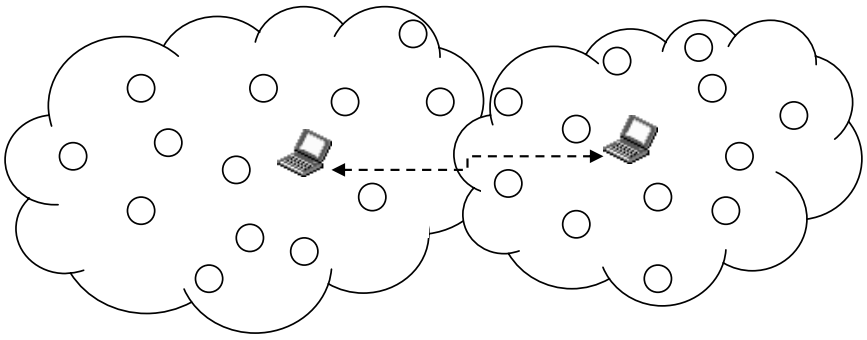


Figure 10.8: Worm Hole in AODV

ing nodes can short-circuit the normal flow of routing packets. This could cause all the traffic between two different parts of the network to be routed through this two nodes, as no other routes can survive when compete with this seemly excellent route. By creating such a virtual vertex cut, the adversaries can then launch various attacks. For example, with only routing packets forwarded, all the data packets forwarded all the way to the vertex cut merely get dropped. With many nodes sending their packets into oblivion, the attack is more than a network partition.

- *Rushing attack*: Another malicious attack is the rushing attack, which targets against on-demand routing protocols such as AODV and AODVjr that adopt duplicate suppression at each node [123]. By disseminating RREQs quickly (e.g., by using large transmission power) throughout the network, an attacker can cause other nodes to drop any later legitimate RREQs due to the duplicate suppression. The attack

will prevent any nearby node from finding a route to the destination.

Resource consumption is one of the attacks that an adversary usually launches. An attacker can keep on sending out RREQs (with arbitrary destination addresses) that flood through the network. Such flooding will be a disaster to the resource-constrained WMPANs. It will consume a lot of power as well as bandwidth, and greatly disrupt any ongoing communications. For existing routes, an adversary can also repeatedly send out gratuitous RREP messages with different source addresses. This will force all nodes along the route to store large number of route entries, which will never be used. Such attacks will surely result in storage overflows at the nodes being attacked and prevent normal routes from being formed.

B. Attacks aimed at cluster-tree

Cluster-tree routing is based on its tree formation algorithm. By manipulating the tree formation, an attacker can disrupt the routing. In principle, any association-related attacks that can be launched at the MAC sublayer can also be committed at the NWK layer. The difference is that the MAC sublayer attacks are a type of one-hop attacks, whereas the attacks described here are not limited to one-hop and can be more efficient by exploiting more information such as C_m and L_m which is not available at the MAC sublayer. Following are some examples.

- *Route disruption in the cluster-tree (by a compromised device)*: As

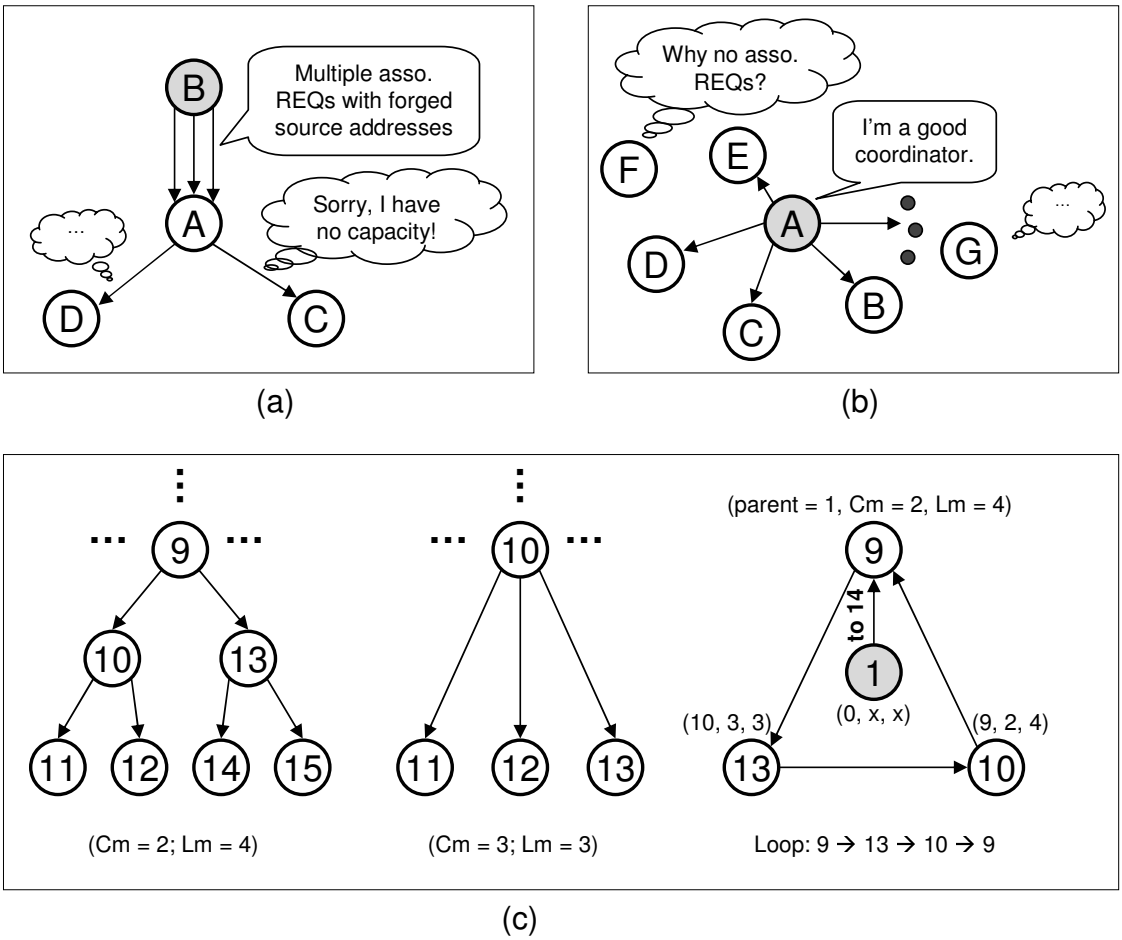


Figure 10.9: Attacks Aimed at Cluster-tree

shown in Figure 10.9 (a), an attacker can repeatedly send association requests to a coordinator, each time with a different forged IEEE device address. The coordinator under attack will soon reach its C_m capacity. Afterwards, any legitimate association request will be rejected by the coordinator. Such attacks are especially powerful when launched at a small level of the tree (i.e., close to the root of the tree).

- *Route disruption in the cluster-tree (by a compromised coordinator)*: A compromised coordinator can perform another type of attack. As

illustrated in Figure 10.9 (b), the coordinator *A* can capture the devices around it by announcing itself as an extremely good coordinator (e.g., at very small level in the tree). Note that the coordinator can accept large number of devices asking for associations, without following the limitation of C_m and/or L_m. To save its memory, the coordinator does not need to save all the association state data. After attracted large number of devices, the coordinator can then perform various attacks such as dropping or selectively dropping packets to and/or from its children. Another possibility is to broadcast specially configured beacons to keep all those devices from entering low energy consumption states. The coordinator will also be able to control the sub-tree formation. For example, the coordinator can indicate all its children that they have reached the bottom of the tree, and therefore disable any further associations thereafter. An opposite case is that, by indicating a small current level and a large L_m, the coordinator can encourage its children to support further associations. Although the coordinator will usually use a small level to attract devices around it, there is no difficulty for it to use a large level later for actual associations. The reason is that the association is a multi-step process, and the coordinator can first announce a small level and then use a different level for associations. A device under attack has little chance to notice this since it generally will not save the announced level information once it has chosen a

candidate coordinator for association.

- *Loop in the cluster-tree*: A tree is normally loop-free. However, we will show it is possible for a malicious coordinator to form a loop among its children. The problem lies in the fact that a coordinator has the power to assign a short address (together with other cluster-tree parameters such as C_m, L_m and current level L_i) to the device asking for association. This is necessary for forming a useful cluster-tree, but problems including routing loops can arise. Like in AODV and AODVjr, a loop between two nodes can be formed in cluster-tree routing, but such a trivial loop is easy to detect. A more subtle loop among more than two nodes can also be formed, as described in the following. In Figure 10.9 (c), the left part is the logical structure of a cluster-tree with $C_m = 2$ and $L_m = 4$ (not all the branches of the tree are depicted in the figure), the middle part is the logical structure of a cluster-tree with $C_m = 3$ and $L_m = 3$, and the right part is the physical structure of a certain network area. All the numbers within the circle are the short addresses assigned during associations. In the right part, the malicious coordinator (with short address *1*) can manipulate the associations and assign short addresses *9*, *10*, *13* to the three devices, and make them believe that they have the triplet (*parent*, C_m, L_m) values (*1*, *2*, *4*), (*9*, *2*, *4*) and (*10*, *3*, *3*) respectively. All the triplets are valid with respect to cluster-tree formation algorithm, so the de-

vices can not find any thing abnormal in terms of tree parameters. The C_m and L_m are values passed down from coordinator, and no authentication or verification is required. Hence, all the triplets are just good enough to all the devices. After the associations of all the three devices, the coordinator can trigger a loop by unicasting a packet to any of the three devices, indicating the destination short address is 14 or 15. According to cluster-tree routing, the packet will loop among the three devices.

Unlike in AODV and AODVjr, no network-wide flooding is needed in cluster-tree for route discovery or maintenance. Thereby, cluster-tree in general is not susceptible to flooding-based attacks (assuming data packets are not allowed to be broadcast[3]). Nonetheless, cluster-tree routing can still suffer from other *resource consumption* attacks. As shown above, by blocking at a certain level some or all branches of a tree, an attacker can rule out huge number of addresses from being used. Most cluster-tree routes are not optimal. A packet may need to travel from the bottom of one branch to the root and then from the root to the bottom of another branch, even if the source and destination are only a few hops away. In cluster tree, each node has the ability to calculate the complete path via which a packet will propagate. This means an attacker can deliberately transmit a packet to a node located at the other end of the tree, in an attempt to consume the precious

[3]In ZigBee, however, data packets are allowed to be broadcast.

power of each node along the path. What makes such attacks more powerful is that nodes close to the root of the tree normally have to route more packets than other nodes. For those nodes, the power can be depleted quickly. The above attacks will accelerate the power depletion of those nodes, which could eventually lead to network partitions. A more powerful attack is the "void address" attack. WMPANs use 16-bit short addresses, but cluster-tree often uses only part of the addresses (the lower address space). Thus, an attacker can send a packet to a non-existing short address that is beyond the address scope used by the cluster-tree. Albeit the destination does not really exist, the packet will still be forwarded up to the root, as no address scope verification is required in the current cluster-tree routing. When the packet arrives, the root may still fail to check the validity of the address and continue to forward the packet to a non-existing branch. Of course, the root will never receive an acknowledgment for the packet and will retransmit the packet. After all retransmissions fail, the root could try to start a costly repair to fix the seemly broken link. All this could help eat up the power of the root (if it is not mains powered) as well as that of nearby nodes.

10.2.2.4 Application layer attacks and other attacks

Layers above NWK are less standard in the sense that they are more dependent on the specific applications and services the network intends to provide. Unlike PHY, Data Link, and NWK layers, other upper layers can

also be left out or combined when needed. At the transport layer, an attacker can force a node to retransmit a batch of packets by sending a forged NACK message. However, transport layer is a less important layer for WMPANs, because the envisioned applications are low rate ones and the communication sessions are also short. So transport layer is likely to be omitted or combined with other layers in WMPANs, and we will not consider transport layer attacks here. One possible type of application layer attack is rational attack (unfairness problem). While a casual user normally lacks the ability to tamper with the lower layer functions, he may be able to control the applications to get some advantages over others. Another type of application layer attacks are those attacks targeting against key management service, an essential service for any security framework. Key management service can be provided at different layers depending on the specific security needs. Like those upper layers, the related attacks are not “standard" either and are difficult to summarize. In this subsection, we give out two attack examples, one is related to directed diffusion, and another related to key distribution.

Directed diffusion is a new communication paradigm for sensor networks [121]. It is data-centric and interest-based. A node in the network can diffuse an interest such as “white car" in an area of the network. Other nodes having received this interest begin to collect related data. When a white car is detected, the information will be returned via the reverse path to the node having diffused the interest, and the node is often referred to

as a sink for this reason. Intermediate nodes may aggregate the data, for example, give more accurate information about the car such as location and speed by combining data from several nodes. In directed diffusion, a node dynamically maintains a table of interest entries, each of which consist of a timestamp field and a duration field. Each interest entry is associated with a set of gradients which are used for sending data back to the senders of the interest (a gradient specifies both a data rate and a direction in which to send data matching the interest). When an interest is received by a node, it will create or update the interest entry and the corresponding gradient. In a sense, direct diffusion is related to on-demand routing algorithms. However, it has some special features. Initially, redundant paths generally exist and these paths are not free of loops. Later, direct diffusion employs two mechanisms to reduce path redundancy and to avoid path loops respectively: path reinforcement (positive or negative) and message cache.

Due to the robustness of flooding, it is difficult to break down the paths between the sink and other nodes. However, some other attacks are possible. An attacker can disrupt the paths by abusing the reinforcement. For example, by reinforcing the path via itself, an attacker can launch black hole or gray hole attacks. An attacker can also perform resource consumption attacks by keeping on sending out easy-to-match interest with large gradient values.

Some typical attacks against key management are depicted in Fig-

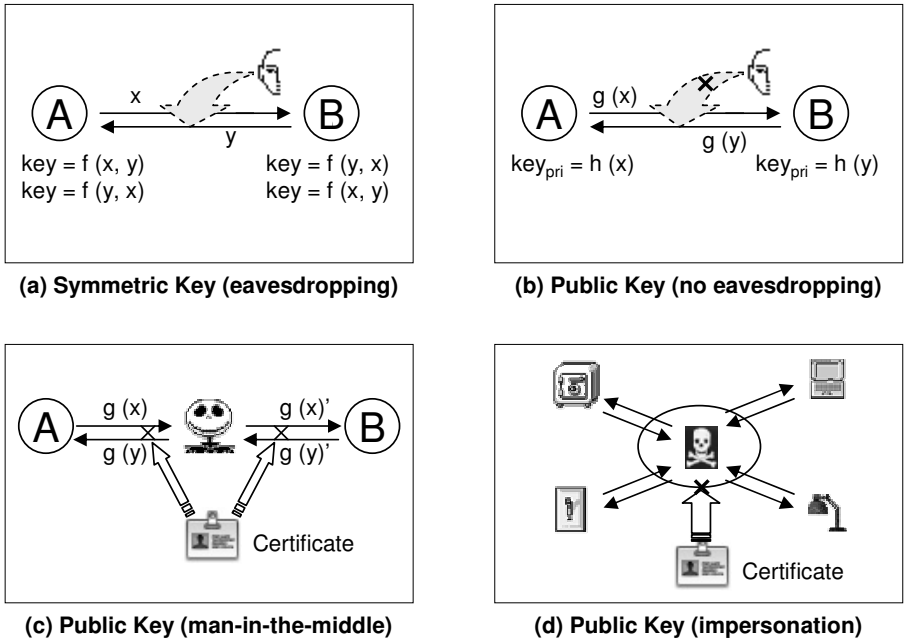


Figure 10.10: Attacks against Key Management

ure 10.10, Symmetric keys need to be exchanged between two communication entities, and therefore are susceptible to interceptions (Figure 10.10 (a)). In public key schemes, a pair of asymmetric keys (*public_key*, *private_key*) are used for each entity, and only the public key is published. While a message can be ciphered using the public key, it will be practically impossible for any adversary to decipher the encrypted message without knowing the corresponding private key (Figure 10.10 (b)). The one-way function used in a public key scheme prevents an attacker from performing reverse engineering notwithstanding he knows the ciphertext and the public key. But public key schemes themselves can not withstand the “man-in-the-middle" attacks (Figure 10.10 (c)) and the impersonation attacks (Figure 10.10 (d)). Authentication using a certificate is commonly used as a

countermeasure against such attacks. But authentication schemes generally require a public key infrastructure (PKI) which is in charge of issuing, verifying and revoking certificates.

10.3 Modeling Attacks in WMPANs

10.3.1 NS2 Simulator

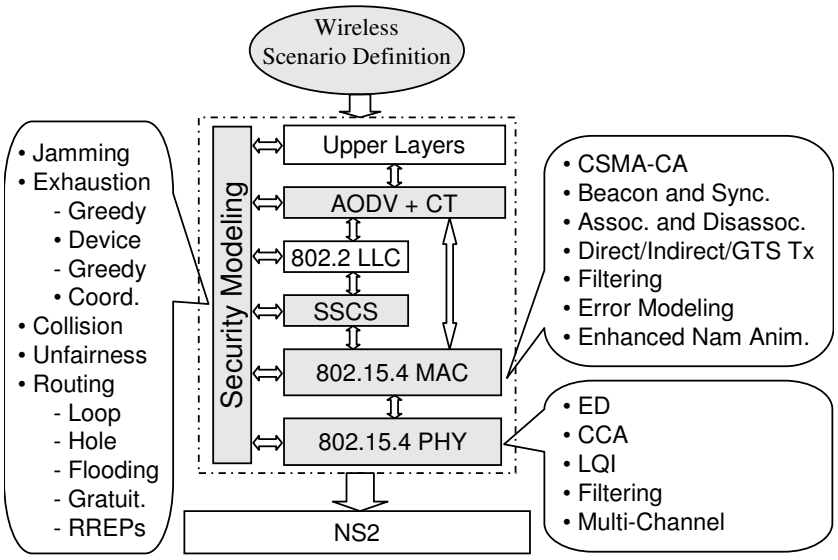


Figure 10.11: Modeling Attacks in WMPANs

We developed a WMPAN simulator based on NS2 [22] and modeled some attacks. Figure 10.11 outlines the function modules in the simulator, and a brief description is given below for each of the modules.

- *Wireless Scenario Definition:* It selects the routing protocol; defines the network topology; and schedules events such as initializations of

PAN coordinator, coordinators and devices, and starting (stopping) applications. It defines radio-propagation model, antenna model, interface queue, traffic pattern, link error model, link and node failures, superframe structure in beacon enabled mode, radio transmission range, and animation configuration.

- *Service Specific Convergence Sublayer (SSCS):* This is the interface between 802.15.4 MAC and upper layers. It provides a way to access all the MAC primitives, but it can also serve as a wrapper of those primitives for convenient operations. It is an implementation specific module and its function should be tailored to the requirements of specific applications.
- 8*02.15.4 PHY:* It implements all 14 PHY primitives and includes the following functions:
  - Energy detection (ED)
  - Clear channel assessment (CCA)
  - Link quality indication (LQI)
  - Filtering (carrier sense, interference and channel filtering)
  - Multi-channel support
- *802.15.4 MAC:* This is the main module. It implements all the 35 MAC sublayer primitives and includes the following functions:
  - Unslotted CSMA-CA and slotted CSMA-CA

  - Beacon enabled and non-beacon enabled modes
  - Beacon tracking and synchronization
  - Association and disassociation
  - Direct, indirect (data polling and extraction) and guaranteed time slot (GTS) data transmissions
  - Channel scan (ED, Active, Passive, and Orphan scan)
  - Filtering (beacon, duplication, and interference filtering)
  - Error modeling (link error, link failure, and node failure)
  - Enhanced Nam [35] Animation

- *Security modeling:* This is the module for studying the feasibility and consequences of attacks in WMPANs. Currently it models the following attacks:

  - *Jamming:* The duration of jamming and the transmission power used for jamming can be configured through Tcl scripts. When a jamming request is received, the device can immediately begin to transmit jamming signals if it has already joined a WMPAN (as the case of a compromised or hijacked device). Otherwise, the device will perform active channel scan to locate an existing WMPAN first.
  - *Exhaustion:* This part models the association-related exhaustion attacks. The attacks can be launched by either a device or a co-

ordinator. We refer to the corresponding device/coordinator as a greedy device/coordinator. A greedy device will first scan the channels and locate all the coordinators within its neighborhood, and then try to associate with all the coordinators it has detected. Impersonation is also allowed in this procedure, that is, the greedy device not only tries to associate with all the available coordinators, but also tries to associate with each coordinator repeatedly with forged device addresses. The number of associations for each coordinator can be defined through Tcl scripts. To save memory, a greedy device does not need to save all the association state data. Since large transmission power is not useful in an association process consisting of multi-steps, a greedy device uses the same transmission power as that used by a normal device. A greedy coordinator tries to attract devices hoping to join the network. When a normal coordinator receives a beacon request command frame which is used by a device to locate coordinators for association, it will immediately unicast a beacon to the device if it is in non-beacon enabled mode, or discard the frame silently if it is in beacon enabled mode. So a device can receive unicast beacons from coordinators in non-beacon enabled mode (referred to as non-beaconing coordinators) or broadcast beacons from coordinators in beacon enabled mode (referred to as beaconing co-

ordinators). In general, a non-beaconing coordinator has a better chance to be selected by a device for association, unless there are multiple non-beaconing coordinators that are hidden from one another [112]. A greedy coordinator will take advantage of this by immediately unicasting a beacon to the device soliciting beacons no matter it is in beacon enabled mode or non-beacon enabled mode. It will also manipulate the parameters in its beacon payload such as cluster-tree level to lure the device. Optionally, a greedy coordinator is allowed to use a large transmission power in order to attract more devices. However, the effect is limited. Although a remote device is able to receive the strong beacon signals, the association will not succeed as the greedy coordinator can not hear the correspondence from the remote device. The only way for a greedy coordinator to attract remote devices is to blindly and frequently transmit strong beacon signals, which could be too expensive. Like a greedy device, a greedy coordinator does not need to save all association state data.

– *Collision:* Collision attacks try to destroy as many messages as possible at a minimum cost. This is different from blind jamming attacks. Figure 10.12 illustrates the collision algorithm designed by us. Collisions are aimed at a broadcast frame or a unicast frame destined for one of the attacker's neighbors. If collision with a

frame is impossible, for example, the frame is transmitted to a node beyond the attacker's transmission range, the attacker will try to corrupt the corresponding acknowledgment frame if such an acknowledgment is required. Corrupting the acknowledgment will cause the sender to keep on retransmitting the frame, though it will not prevent the destination from receiving the frame. The attacker is allowed to use a large transmission power, but the efficiency of the attacks relies little on the transmission power. To make attacks appear like normal collisions, an adversary can apply a gray filter to randomly let some frames go through.

– *Unfairness:* This is also called rational attack. In our simulation, a cheating node tries to get advantages over other nodes for channel access by deviating from the CSMA-CA algorithm, but otherwise behaves normally. A cheating node will skip the backoff procedure upon receiving a beacon from its coordinator, and immediately begin to transmit a frame in its buffer. If the previous transmission is successful, it will continue to transmit other available frames without backoff. Otherwise, it will enter the normal backoff procedure. Although a cheating node can also jump to the channel upon detecting the end of a transaction from other nodes, the chance to succeed is lower than that upon receiving a beacon from the coordinator. The reason is that the successful channel

access of other nodes does not guarantee that the cheating node will capture the channel successfully by immediately following a transaction from those nodes, as they (the cheating node and other nodes) face different competitors. The cheating node in our simulation is relatively conservative and does not tail after transactions from other nodes.

– *Attacks against routing:* Routing is an important function in most networks. But it is more important in wireless networks due to the scarceness of resources and richness of communication interferences. A reliable and efficient routing is therefore essential for a wireless network to render satisfactory performance. The important function of routing makes it an attractive target of attacks. The attacks modeled in this part include: routing loops, black/gray holes, RREQ flooding and gratuitous RREPs. *Loop* attacks are performed by a malicious node by sending forged routing control messages to its neighbors. Such attacks are only possible when the neighbors of the attacker form a close connection. The attacker will first overhear the communications among its neighbors and then try to figure out the relationship among those neighbors, that is, whether a neighbor is within the transmission range of another neighbor. With the relationship information, the attacker will be able to find out if a loop attack is possible. If the condition for

forming a loop is satisfied, the attacker will then transmit deliberately counterfeited routing packets to those neighbors to form a loop among them. After a loop is formed, a triggering packet will be sent to one of the neighbors involved in the loop to put all the victims into endless drudgery. Forming a loop in cluster-tree is somewhat difficult, though possible as illustrated in Figure 10.9 (c). So only loop attacks against AODV are modeled in this part. For *black/gray hole* attacks, the adversary first alters the hop count in both the RREQ and RREP routing packets to make itself an excellent node for relaying packets. After the traffic is drawn towards itself, it then drops all the packets or selectively drops some of the packets. No matter whether a packet is dropped or not, the attacker will always send back an acknowledgment claiming it has successfully received the packet to prevent the upstream node from bringing down the route and finding an alternative one. Another simple way to launch attacks is to *flood* RREQs throughout the network from time to time. RREQ flooding attacks are very powerful, because they can cause network-wide congestions without using large transmission power like in the jamming attacks. Furthermore, flooding consumes the precious power of almost all the nodes in the network, whereas jamming normally only prevents other nodes from accessing the channel. Finally, flooding attacks

are difficult to detect. Improper countermeasures for flooding attacks could incur another common type of attacks, DOS attacks. In our simulation, the attack duration and the transmission frequency of RREQs (packets per second) can be defined through the Tcl scripts. Finally, the *gratuitous RREP* attacks deplete the memory storage of the nodes under attack. In this type of attacks, an attacker repeatedly sends gratuitous RREPs with feint source addresses to nodes for which it has route entries. The attacks will eventually cause memory overflow at each node along the path. The number of gratuitous RREPs for each route entry can be defined. To see the effect of this type of attacks, we also define the maximum memory capacity of each node.

10.3.2 Some Initial Experimental Results

In this subsection, we give out some attack modeling results. The focus has been given to those attacks that are more specific to WMPANs. Some critical attacks such as encryption key breaking can not be meaningfully simulated, and it is more suitable to study them in realistic environments.

10.3.2.1 Jamming and its orphaning effect

Figure 10.13 is a scenario snapshot from the jamming attack simulation. The experimental parameters are as follows:

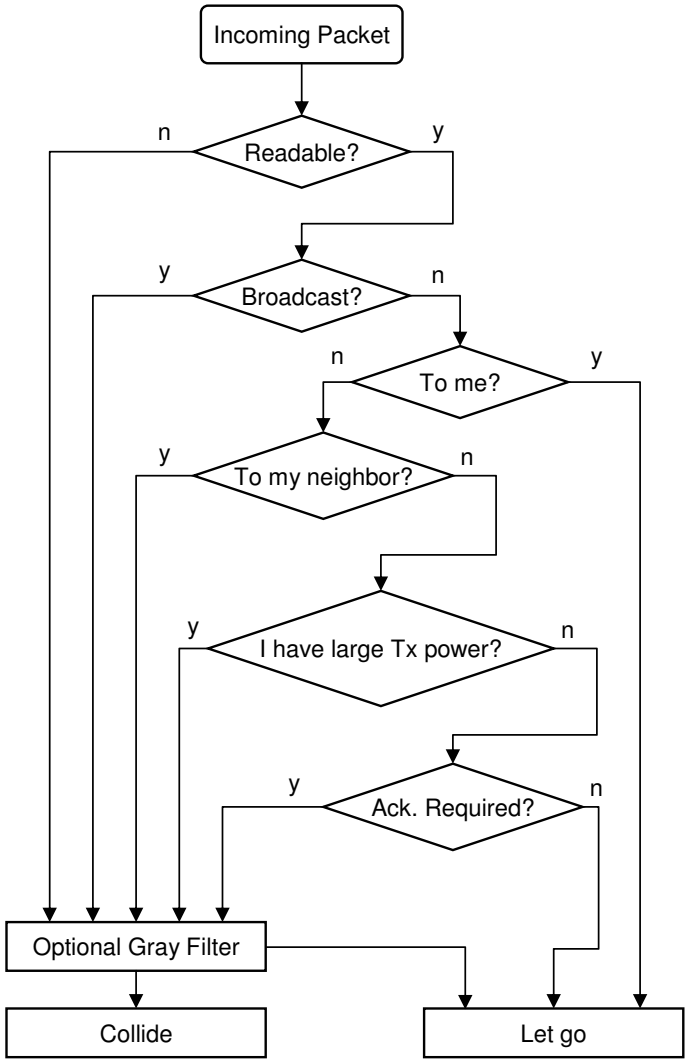


Figure 10.12: Collision Algorithm

- Number of nodes: 21
- Area: $50 \times 50\, m^2$
- Traffic: constant bit rate (CBR); 10 pkts/sec
- Traffic flows: node 9 $\rightarrow$ node 17 and node 3 $\rightarrow$ node 18
- Duration: $900\, sec$
- Closest neighbor distance: $10\, m$

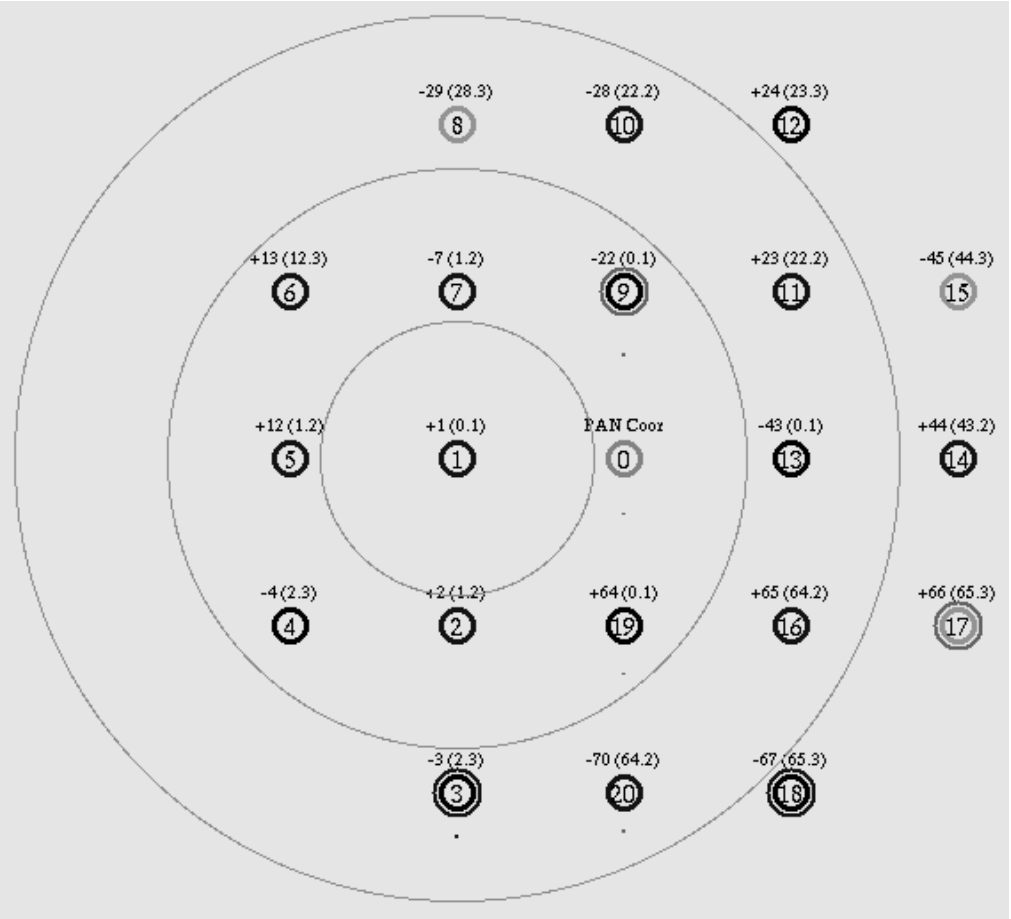


Figure 10.13: Jamming

- Transmission range: $12\,m$
- Beaconing parameters:
  - Beaconing nodes: 0, 2, 7, 11, 16
  - Beacon order: 3
  - Superframe order: 3
- Cluster-tree parameters:
  - Cm: 4
  - L_m: 3

  - Block Size: 85

- Jamming attack parameters:

  - Attacker: node 1
  - Transmission range: $30\,m$
  - Duration: $20\,sec$ to $23\,sec$

As expected, after the jamming begins, no packets can be successfully delivered, and the source nodes 9 and 3 start to drop packets. However, we have observed more serious consequences. Before jamming, all the nodes succeed in joining the network. But all the devices operating in beacon enabled mode (nodes 3, 4, 9, 12, 13, 18 and 19) get orphaned after a 3-second duration of jamming. In beacon enabled mode, a device will generally track the beacons from its coordinator. A device will conclude that it has been orphaned from its coordinator if it misses *aMaxLostBeacon* (default value 4) beacons in a row. An orphaned device will initiate an orphaning procedure in an attempt to relocate its coordinator. If the coordinator relocation fails, the orphaned device may try to associate with another available coordinator if the application requires it to do so. If all those attempts fail, then the device is cut off from the PAN and will not be able to communicate with any other nodes in the PAN. The orphaning mechanism is designed to help detect communication failures and recover from such failures. Nevertheless, a jamming attacker can use this to kick other devices out of the PAN.

Table 10.1: Superframe Duration (msec)

BO	0	1	2	3	4	5	...	14
2.4 GHz	3.84	7.68	15.36	30.72	61.44	122.88	...	62914.56
915 MHz	48	96	192	384	768	1536	...	786432
868 MHz	96	192	384	768	1536	3072	...	1572864

The significance is that the attacker no longer needs to keep on transmitting strong jamming signals, which is expensive. As shown in Table 10.1, for all three ISM frequency bands, the superframe duration is less than 3.1 seconds for beacon orders up to 5. Therefore many nodes could be left stranded when facing 10 second jamming. Note that devices associated with non-beaconing coordinators are not orphaned, since they do not track beacons and therefore orphaning mechanism is not applicable to them.

10.3.2.2 Exhaustion of association resources

Exhaustion experiments are performed using the topology shown in Figure 10.14 (a). Following are some experimental parameters (we leave out those parameters no closely related to the attacks):

- Number of nodes: 101
- Area: $80 \times 80\, m^2$
- Closest neighbor distance: about $7\, m$
- Transmission range: $20\, m$; for both normal nodes and attackers.
- PAN Coordinator: node 0; non-beaconing

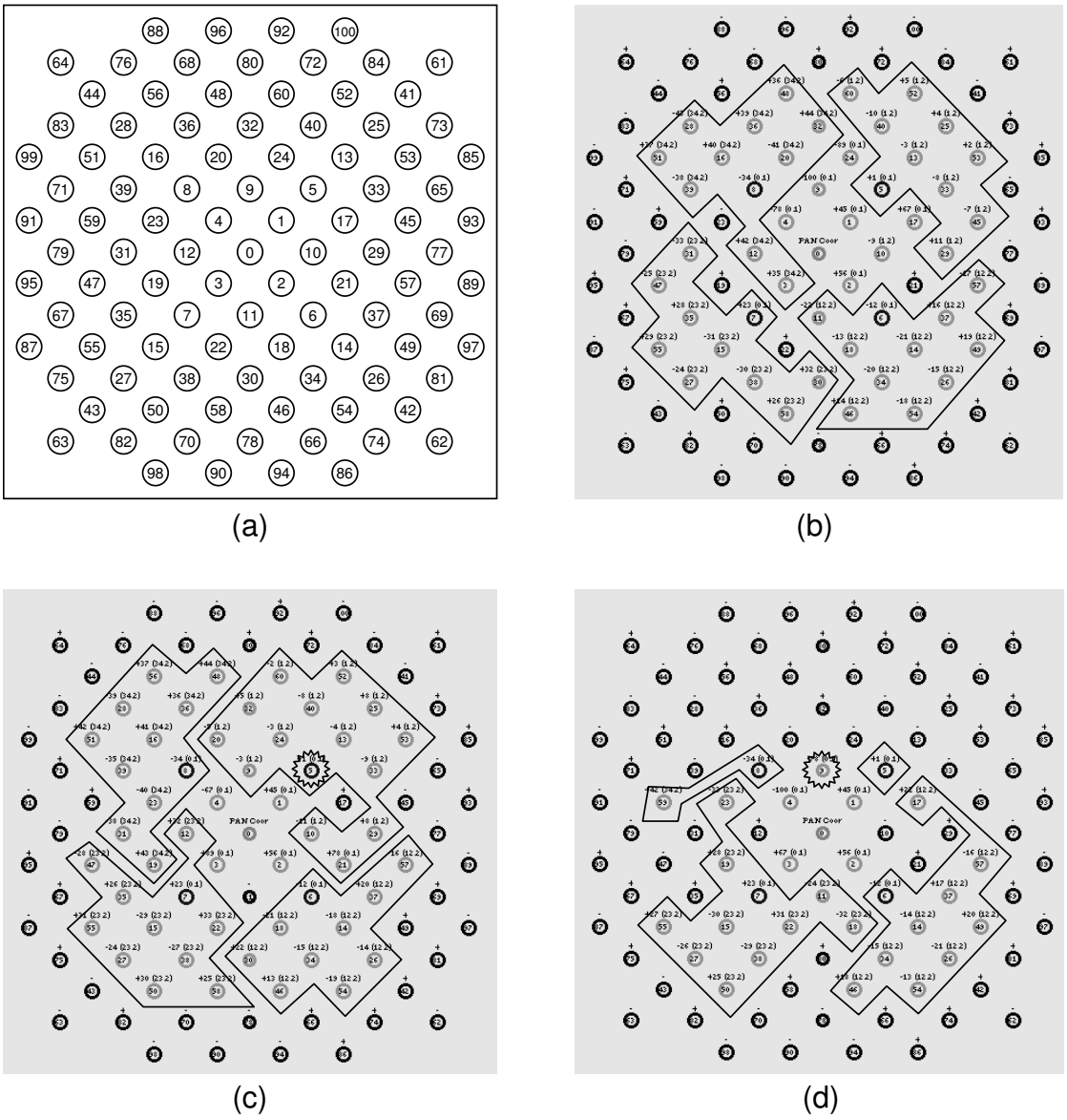


Figure 10.14: Greedy Coordinator and Device

- Coordinators: nodes 5, 6, 7, 8; all beaconing with beacon order 3
- Cluster-tree parameters:
  - Cm: 10
  - L_m: 2
- Attackers:

- Greedy coordinator: node 5
- Greedy device: node 9

Figure 10.14 (b) (c) (d) are simulation scenario snapshots. Figure 10.14 (b) is the normal experiment without attacks. The four areas embraced by lines represent the four coordinators and the devices associated with them. The middle part (not bounded by lines) is the PAN coordinator and the devices associated with it. We can see all the four coordinators have 9 or 10 children, while the PAN coordinator only has 7 children as it is circled by other coordinators.

Figure 10.14 (c) shows the experimental result with node 5 as a greedy coordinator. The greedy coordinator eats up more devices than a normal coordinator. Compared with the normal case, node 5 has increased the number of its children from 9 to 14. All the other coordinators (except the PAN coordinator) have the similar number of children as before. However, almost all the children are those devices out of the reach of the greedy coordinator. This means that other coordinators have very little chance to win over a device when competing with the greedy coordinator. Note that, to prevent the greedy coordinator from preempting the nearby devices, we schedule the attack for some time after all the coordinators have joined the PAN. The situation of the PAN coordinator is slightly different. The PAN coordinator has the following advantages compared with other coordinators when competing with the greedy coordinator:

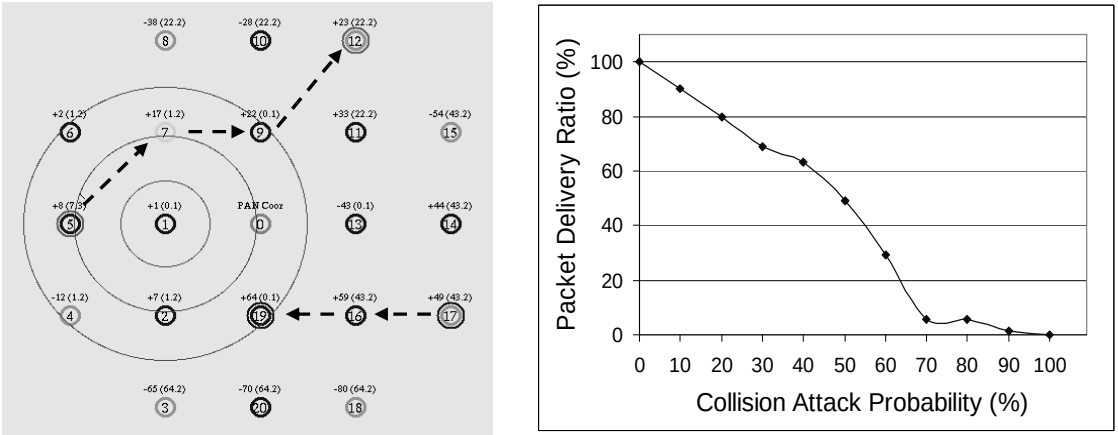
1. It is the first node in the PAN;
2. It has the lowest cluster-tree level.
3. In our simulation, the PAN coordinator operates in non-beacon enabled mode. This means it will immediately unicast a beacon to a device asking for beacons.

In our attack model, the greedy coordinator forges a cluster-tree level of 0, which is just as low as that of the PAN coordinator. No matter it operates in beacon enabled mode or non-beacon enabled mode, the greedy coordinator always immediately unicasts a beacon to a device soliciting beacons, just like the PAN coordinator. So the PAN coordinator only has the first advantage listed above compared with the greedy coordinator, though it is in a better position compared with other coordinators. By virtue of the first advantage listed above, the PAN coordinator does win over three devices which are also within the reach of the greedy coordinator. But the number of its children still drops from 7 to 4. It is clear that the greedy coordinator does not follow the cluster-tree algorithm – it has 14 children though the C_m used is only 10. Nevertheless, all the logical addresses allocated by the greedy coordinator are valid cluster-tree addresses. To accept more than C_m children, the greedy coordinator randomly re-uses those logical addresses, and duplication is inevitable.

Figure 10.14 (d) is the experimental result with node 9 as a greedy device. A coordinator handles association requests from devices based on

first-come-first-serve. So, unlike a greedy coordinator, a greedy device has no way to get any priority over other devices. Therefore, such attacks are only possible if they are launched before other devices begin to join the network through associations. In our simulation, we assume the attack condition is met. The greedy device in our simulation not only tries to associate with each coordinator it finds, but also tries to repeatedly associate with a coordinator by impersonating other devices. The simulation result shows that the nearby coordinators (nodes 0, 5 and 8) suffer a lot: the PAN coordinator only has 4 children, the left coordinator (node 8) only has one child, and the right coordinator (node 5) is left alone.

10.3.2.3 Collision



(a) (b)

Figure 10.15: Collision Attacks

Collision attacks are more efficient than jamming attacks. Figure 10.15 (a) is a collision attack scenario snapshot from our simulation. There are

two traffic flows: one is from node 5 to node 12, and another is from node 17 to node 19. Other related experimental parameters are:

- Traffic: Poisson traffic with a mean arrival rate of 10 pkts/sec
- Duration: $900\,sec$
- Closest neighbor distance: about $10\,m$
- Transmission range: $15\,m$; for both normal nodes and attackers.
- Attacker: node 1
- Collision probability: from 0% to 100%, increased by 10% each time. This determines the probability with which an attacker will try to corrupt a packet going-by.

According to our collision algorithm (see Figure 10.12), only the first traffic flow will be attacked. Figure 10.15 (b) shows that the packet delivery ratio for the first traffic flow decreases as the collision probability increases. Up to 50% collision probability, the packet delivery ratio decreases at about a constant rate as the collision probability increases. But as the collision probability continues to increase, the packet delivery ratio drops sharply and it touches 5.7% when the collision probability reaches 70%. The reason is that a higher collision probability will not only corrupt more packets, but also bring down involved routes.

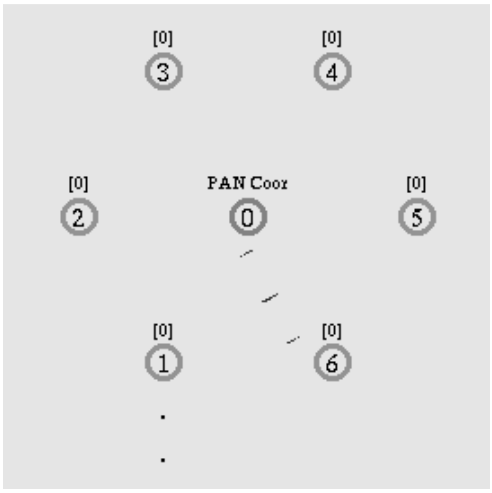
Like jamming attacks, collision attacks can also result in orphanings, but they are much more efficient. For example, by only corrupting beacons, an

attacker is able to orphan other devices at a very low cost. Although the collision attacks are normally limited in relatively small area, they could still be very powerful by selectively attacking some sensitive messages such as beacons, management commands and routing packets.

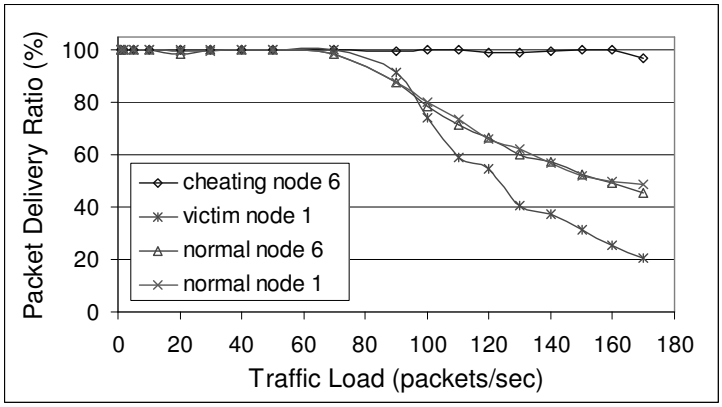
10.3.2.4 Cheating and unfairness

We use the star topology shown in Figure 10.16 (a) to simulate unfairness attacks. The star network operates in beacon enabled mode with beacon order 3. Node 0 is the PAN coordinator, and all other nodes are devices. Two CBR traffic flows are set up from node 1 to node 0 and from node 6 to node 0. Node 6 is the cheating node. The PHY protocol data unit (PPDU) of the CBR packet has a size of 97 bytes. The traffic load varies and takes the values from 1 packet per second to 170 packets per second. The data rate is 250 Kbps (in 2.4 GHz frequency band). The closest neighbor distance is about $10\,m$ and the transmission range is $15\,m$. The simulation duration is 900 seconds.

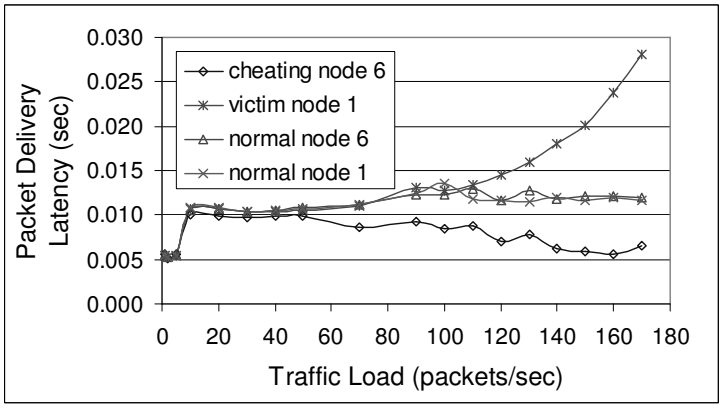
From Figure 10.16 (b) we can see the packet delivery ratio is not affected much by the unfairness attacks for traffic loads up to 70 packets per second. However, unfairness happens after the traffic load continues to increase. The packet delivery ratio of victim node 1 drops to 20.51% when traffic load reaches 170 packets per second, while the cheating node 6 maintains well its packet delivery ratio in spite of the increment of traffic load.



(a)



(b)



(c)

Figure 10.16: Cheating and Unfairness

In the normal case, the packet delivery ratios of both node 1 and node 6 decrease as the traffic load increases beyond 70 packets per second. No unfairness has been observed when attacks are not introduced, as can be seen from Figure 10.16 (b). As we mentioned before, unfairness attacks are a type of rational attacks, therefore it is understandable the unfairness problems arise only when the network is oppressed by limited resources. This is why the unfairness in packet delivery ratio does not appear when the traffic load is relatively small, or more specifically, when the traffic load is within the network capacity. Since WMPANs face low data rate applications, the unfairness in packet delivery ratio is not a big concern.

Unfairness in packet delivery latency is slightly different from that in packet delivery ratio. Although, for moderate traffic load, the packet delivery ratio is almost the same when facing attacks, the packet delivery latency is affected at the very beginning (Figure 10.16 (c)). That is, packets from the victim node 1 suffer additional delays although they are able to reach the destination successfully. The packet delivery latency is large when traffic load is high. Nonetheless, for the same aforementioned reason, we will not worry about unfairness problems in high data rate applications. For low traffic load, the seemingly small difference between the packet delivery delay of the cheating node 6 and that of the victim node 1 may not be that significant to data packets. An extra delay of 1 millisecond may not make much difference in many WMPAN applications. However, this

small difference could be critical to the network control and management. For example, when C_m is not large enough in the cluster-tree formation, a smaller delay will give a node a better chance to join the network. And in an AODV route discovery procedure, a smaller delay could help an adversary to launch "rushing attacks" [123] or attract traffic towards itself for "black hole" or "gray hole" attacks (see subsection 10.2.2.3). It is also helpful to other attacks where time is an important factor, like attacks from greedy coordinators or devices.

10.4 Securing WMPANs

10.4.1 Related Work

Security in wireless networks has become an active research area in recent years. In this subsection, we summarize some related research work that has been done in wireless networks, including wireless mobile ad hoc networks and wireless sensor networks. Not all the proposed approaches can be applied to WMPANs, but they are valuable references in designing secure WMPANs.

10.4.1.1 Key management and authentication

Zhou and Haas exploit inherent route redundancies in mobile ad hoc networks to enhance the network tolerance for Byzantine failures caused by

several compromised nodes or collusions [122]. The basic idea is to detect inconsistency using redundant information and to isolate compromised routers. The drawback is that maintaining redundant route information is expensive in resource-constrained wireless networks. They also employ threshold cryptography [124, 125] to distribute trust among nodes for reliable key management service. A key management prototype is also implemented.

Perrig *et al* come up with several mechanisms for key establishment using the framework of pre-distributing a random set of keys to each node [127]. The study is based on the probability by which two neighbor nodes share a key if they both contain a subset of keys randomly selected from a key pool. The idea is to maximize the chance of key sharing between neighbors with minimum number of keys pre-installed in each of them. They illustrate how to strengthen the security between two nodes by leveraging the security of other links, and how to preserve the secrecy of the rest of the network when any node is captured. Node-to-node authentication and quorum-based revocation are also discussed. Another key predistribution scheme based on probabilistic key sharing among the randomly distributed nodes is suggested in [42]. The approach maintains the similar security when compared to pair-wise private keysharing schemes without saving large number of keys.

A multicast and broadcast authentication protocol called Timed Efficient

Stream Loss-tolerant Authentication (TESLA) is proposed in [40, 126, 138, 139]. TESLA uses symmetric key to achieve asymmetry from clock synchronization and delayed key disclosure, which is different from traditional asymmetric protocols (e.g., RSA [140]) which rely on computationally expensive one-way trapdoor functions. For coding efficiency, all cryptographic primitives (encryption, message authentication code (MAC), hash, random number generator) are built out of a single block cipher. The approach scales to large number of devices, and tolerates packet loss. TESLA is based on loose time synchronization between the sender and the receivers.

A security framework is proposed in the context of military environments by Kong *et al* [144]. The framework supports a combination of infrastructure mode and infrastructureless mode. In infrastructure mode, security services, specifically authentications, are implemented on unmanned aerial vehicles (UAVs). When the UAVs fail or are destroyed, the system switches to infrastructureless mode, which maintains comparable security services among the surviving units.

A cryptographically secure representation of trust based on secure groups, *troups*, is presented in [145]. *Troups* are constructed in a distributed manner using collision-free one way accumulators [146, 147]. The *troup*-membership is verified using the zero-knowledge protocol of modular exponentiation. A prototype implementation of the troups based on authentication system is given in the paper.

Khalili *et al* propose using a threshold, ID-based cryptosystem to achieve security, efficiency, and resilience [148]. The keys are built on-the-fly, assuming no prior trust or authority relation among nodes before the network deployment. The master key is shared in a *t-out-of-n* threshold manner by the initial set of n nodes. To reduce computation, a node is not required to generate its own public key. Instead, the identity of the node serves the purpose.

A bootstrapping solution for the Dynamic Source Routing (DSR) protocol is presented in [149]. The solution employs the statistically unique and cryptographically verifiable (SUCV) identifiers [150] and public-secret key pairs generated by the nodes themselves to solve the address ownership problem [150, 152] and to counter the bidding-down attacks in return routability [151]. Public-key cryptography is used only in the first route discovery procedure towards a particular destination. A symmetric key is exchanged during this procedure and it is used thereafter to maintain the route.

Hubaux *et al* [153] propose a distributed service for establishing trust relations among network nodes from Pretty Good Privacy (PGP) [154] certificates, which does not rely on a trusted authority infrastructure.

10.4.1.2 Secure routing

Hu *et al* optimize the dynamic source routing (DSR) protocol and propose a new on-demand secure ad hoc network routing protocol, called *Ariadne* [123], that relies only on highly efficient symmetric cryptography. To prevent attackers or corrupted nodes from tampering with routes consisting of normal nodes, *Ariadne* authenticates routing messages using various schemes: shared secrets, broadcast authentication, a combination of shared secrets and broadcast authentication, or digital signatures.

Secure Efficient Ad hoc Distance vector routing protocol (SEAD), a secure ad hoc network routing protocol based on the Destination-Sequenced Distance-Vector routing protocol (DSDV) is presented in [41]. Efficient one-way hash functions are used to secure the routing operations.

A scheme for detecting a fault node along a routing path is given by Awerbuch *et al* [141]. The scheme requires the destination to return an acknowledgment to the source, for every successfully received data packet. If the number of missing acknowledgments reaches a threshold, the source starts a binary search on the path to locate the fault node.

The Security-aware Ad-hoc Routing (SAR) protocol by Yi, Naldburg *et al* [142] modifies AODV to include security metrics for path computation and selection. Different security levels are supported by using a shared key among nodes for each level. Only those nodes meet the required security level can participate in the routing.

10.4.1.3 Cooperation and unfairness

As a viable means to provide untethered communications, a multi-hop wireless network often requires the nodes in it to cooperate and relay traffic for other nodes [129, 130]. Nevertheless, such cooperation can not be always assumed, due to the limitation of power supply and other resources. Some research work regarding cooperation in wireless ad hoc networks has been done in [131, 132, 134]. In [131], the authors present two techniques that improve throughput in an ad hoc network in the presence of nodes that agree to forward packets but fail to do so. They use a *watchdog* to dynamically monitor the behavior of nodes and to identify misbehaving nodes. A *pathrater* is further used to help routing protocols get around those nodes. In [132], the authors propose a secure mechanism to stimulate nodes for cooperation and to prevent them from overloading the network. They introduce a mechanism in which nodes providing services are remunerated and nodes receiving services are charged. An algorithm is designed to help a node to decide whether to accept or reject a packet relay request based on the charging and remuneration mechanism. However, this per packet acceptance algorithm is rather inefficient. Other session-based algorithms for relaying traffic have been proposed in [134, 135]. Those per session approaches reduce the processing overhead in [132]. How to determine the optimal throughput that each node should receive based on the node lifetime constraints has been studied in [133]. A distributed and scalable acceptance

Table 10.2: IEEE 802.15.4 Security Suites

Security suite name	Access control	Data encryption	Frame integrity	Sequential freshness (optional)
None				
AES-CTR	x	x		x
AES-CCM-128	x	x	x	x
AES-CCM-64	x	x	x	x
AES-CCM-32	x	x	x	x
AES-CBC-MAC-128	x		x	
AES-CBC-MAC-64	x		x	
AES-CBC-MAC-32	x		x	

algorithm called Generous TIT-FOR-TAT (GTFT) is used by nodes to decide whether to accept or reject a relay request.

Felegyhazi *et al* identify the Nash Equilibrium [4] conditions for cooperation among nodes in a statically configured network [136]. A model is defined in a game theoretic framework and study results by using the model show that the proportion of nodes that need an incentive mechanism to cooperate is always around or above 50 percent.

10.4.2 Security Architecture Defined by IEEE 802.15.4 and ZigBee

10.4.2.1 Overview

IEEE 802.15.4 [5] provides link layer security for WMPANs, including access control, confidentiality, message integrity, and optional message

[4]The Nash Equilibrium, after Nobel Laureate (in economics) and mathematician John Nash, who contributed several key concepts to game theory around 1950. If there is a set of strategies with the property that no player can benefit by changing her strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute the Nash Equilibrium.

freshness, as outlined in Table 10.2. Access control is supported by all security suites except *None* and it provides the ability for a device to select the other devices with which it is willing to communicate. For security purposes, each device keeps an ACL in its MAC sublayer PAN Information Base (MPIB). The ACL contains up to 255 entries, one for each destination device. Each ACL entry consists of the destination address (IEEE address and optional logical short address), security suite identifier, and other security materials. By default, security is not enabled in 802.15.4. To enable security, upper layers should specify a security suite other than *None* in the ACL entry corresponding to the destination. However, acknowledgment frame is required to always use security suite *None*, and thus not protected.

The AES-CTR security suite provides confidentiality protection by encrypting the payload of a frame, using the AES [17] block cipher [32] with counter mode. The AES-CBC-MAC security suite, on the other hand, provides integrity protection, using the CBC-MAC [33]. And the AES-CCM security suite provides both confidentiality and integrity protection, using the CCM [34]. Both AES-CBC-MAC and AES-CCM have three variants (see Table 10.2), depending on the size of the MAC[5] used.

Upon 802.15.4, the ZigBee Alliance defines the NWK and application layer security services [30], based on CCM*, a minor modification of CCM [34]. Besides all the features of CCM, CCM* additionally offers

[5]Here MAC is short for Message Authentication Code, not for Media Access Control.

Table 10.3: Security Levels Available in ZigBee

Security level	Security attribute	Data encryption	Frame integrity
'000'	None		
'001'	MIC-32		x
'010'	MIC-64		x
'011'	MIC-128		x
'100'	ENC	x	
'101'	ENC-MIC-32	x	x
'110'	ENC-MIC-64	x	x
'111'	ENC-MIC-128	x	x

encryption-only and integrity-only capabilities, thus eliminates the need for CTR and CBC-MAC modes. Also, CCM* allows using a single key for all CCM* security levels (see Table 10.3). This is different from the MAC sublayer security modes, which require different keys for different security levels. As a result, different layers in ZigBee can reuse the same key. Another design feature of ZigBee is to use the so-called open trust mode, in which different layers of the communication stack and all applications running on a single device trust each other.

ZigBee uses a 128-bit link key (more actually its derivatives, see details below) to secure pairwise communications, probably multiple hops away, and a 128-bit Network key to secure broadcast communications. A device can acquire link keys and a Network key via key-transport or pre-installation. Link keys can also be obtained through key-establishment technique, based on a 'master' key, which itself can be obtained via key-

transport or pre-installation. The ultimate security between devices depends on the secure initialization and installation of these keys. To avoid security leaks due to unwanted interactions between different security services, ZigBee also uses a one-way function to derive various service-specific keys from the link key, including the key-load key, key-transport key, and data key. The key-load key, key-transport key, and data key are used to protect frames containing transported master keys, frames containing other transported keys, and all other frames that need to be secured respectively.

ZigBee performs centralized security control via a trust center. There is exactly one trust center in each secure network. The trust center is responsible for distributing and maintaining the Network key to devices as well as binding two applications and enabling end-to-end security between devices (e.g., by distributing master keys or link keys).

In both IEEE 802.15.4 and ZigBee, a frame counter, which is a monotonically increasing 4-octet sequence number bound to an encryption key, is used to prevent replay attacks.

10.4.2.2 Problems and remedies

There are some vulnerabilities in the current WMPAN security architecture. The prohibition of protecting MAC sublayer acknowledgment frame essentially destroys the whole security architecture of WMPANs. With the ability to forge an acknowledgment frame, an attacker can launch various

attacks. For example, an attacker can cripple the retransmission mechanism by forging an acknowledgment frame for a data or command frame corrupted due to collision, noise, or even intentional interference from the attacker. Through impersonation, an attacker can also make a device transmit frames to a non-existing device. One practical solution for this problem is to allow the source to determine whether a protected or non-protected acknowledgment frame is needed.

Frame counter overflow is another problem. Neither IEEE 802.15.4 nor ZigBee provides mechanisms to prevent an attacker from launching DOS attacks by exploiting frame counter overflow. Sastry *et al* showed that, a DOS attack can be easily carried out by forging an IEEE 802.15.4 frame and setting its frame counter to the maximum value $2^{32} - 1$, despite that the payload of the frame may not be a valid ciphertext under the key used by the destination [137]. A subtler attack is to intercept an AES-CTR [5] protected IEEE 802.15.4 frame, change the frame counter to $2^{32} - 1$, and then forward the frame to the destination. Note that, in AES-CTR, no integrity protection is applied and only the normal payload is encrypted, which means nothing prevents an adversary from modifying the frame counter. Similarly, in ZigBee, this can be done with an ENC [30] protected NWK frame. However, the Network key used in ZigBee makes the situation even worse. The network-wide shared Network key not only facilitates a device to impersonate another device, but also allows a device to broadcast forged messages

to the whole network. Such DOS attacks have been observed at both the MAC sublayer and the NWK layer using the NS2 simulator given in subsection 10.3.1. To solve this problem, the encryption-only mode should be avoided, and proper authentication should be provided for broadcast messages.

The Network key only provides limited protection for the network. Devices in WMPANs are resource-constrained and lack physical safeguards. A tampered or captured device could endanger the whole network. To reduce the risk, we propose a train multicast (TM) scheme. In TM, each coordinator generates a multicast key mk and distributes this key to all its children during the association procedure. This multicast key enables multicast communications among the coordinator and its children. A secure broadcast can be done through multiple multicasts. For example, the PAN coordinator can multicast a frame to all its children, and each child that acts as a router will decrypt the frame using the multicast key obtained from the PAN coordinator, encrypt the frame using its own multicast key, and then further multicast the frame to its children. This procedure continues until the frame reaches all the devices in the network. If a device has to broadcast a message, it can first unicast the message to the PAN coordinator and then broadcast from there. A better way is to only unicast the frame to its coordinator. Then the coordinator will multicast this frame down along its branch and also unicast the frame to its coordinator, who will continue the same procedure.

Symmetric keys are often used for coding data due to their high encryption/decryption efficiency. While they can also be used for authentications for pairwise communications, they are not suitable for authentications for multicast/broadcast communications. Public keys, on the other hand, are generally used for distributing symmetric keys and for authentications for both pairwise and multicast/broadcast communications. Nevertheless, it is too expensive to use public key algorithms such as RSA [140], ElGamal [156], DSA [155] and Diffie-Hellman key agreement [157] in WMPANs due to the limitation on both memory storage and computational capacity. This concern has led to the exclusion of public keys from the current WMPAN security architecture, notwithstanding the need for authentications for multicast/broadcast messages. This practice has a far-reaching effect on the security of WMPANs, since some important control messages and routing messages are often transmitted using broadcast. The lack of authentications for multicast and broadcast messages accounts for many attacks, such as impersonation, exhaustion based on flooding, route disruption, and loop. How to achieve asymmetry efficiently has been one of the important research topics for resource-constrained networks like WMPANs. Elliptic Curves Cryptography (ECC) [158], compared with the above public key algorithms, offers potential reductions in key size [169] as well as processing power and bandwidth due to the lack of a sub-exponential attack. Achieving asymmetry from clock synchronization and delayed key disclosure has also

been proposed in [40].

The key management based on the trust center is neither robust nor efficient. Wireless links are susceptible to environmental noise, interference, and lack of line of sight (LOS); communications between the trust center and a device can be lost, especially in a multi-hop and/or mobile environment. Without a proper backup scheme, the trust center could be a single point of failure (SPF). Heavily relying on the trust center reduces the robustness of the system. The trust center is responsible for key transport and update and, for security reason, all the keys are currently unicast. This puts heavy burden on the trust center as well as those devices near it, as they need to relay traffic between the trust center and other devices. This type of key transport and update operation is not efficient, and a bottleneck could be formed at the trust center or around it. So, for large scale networks, distributed or hierarchical key management schemes should be considered.

10.4.3 Improving the Security of WMPANs

Some of the attacks given in section 10.2 can be thwarted by enhancing the current WMPAN security architecture as suggested in subsection 10.4.2.2. Some others, however, call for the modification of certain service functions. In this subsection, we present some countermeasures of those attacks.

Attacks such as *jamming* and *collision* are closely related to PHY layer

and are difficult to cope with. Normally there is no way for a node under such attacks to fight back automatically. Securing the PHY layer in wireless environments is a challenging task due to the feature of open media. Some countermeasures such as spread spectrum have been studied [163–165]. Nonetheless, how to pair *pseudo-noise (PN) code* between two devices using spread spectrum is like sharing an encryption key between them, which is not an easy task due to the requirement for simplicity and low cost. For some applications like battlefield communications where high reliability and strong security are required, spread spectrum techniques will play an important role. For other applications, we can selectively equip some important devices (e.g., the PAN coordinator, coordinators, and other devices in charge of network management) with spread spectrum function module so that they can effectively reject interferences. Although the whole network is not protected, the small number of protected devices will be able to perform critical management functions, for example, monitor the behavior of the network and request for human intervention when needed. The availability of multiple frequency bands and channels also provides some protection against those attacks. And directional reception techniques could be a powerful curb on those attacks [166, 167].

Exhaustion attacks that are related to association can be prevented by authenticating sensitive information such as source address and tree level. This in general requires the use of some public key scheme and certificate

service.

The *unfairness* problem of channel access results from the false assumption that all the nodes will strictly follow the protocol. Contention-based channel access schemes are likely to fall for unfairness attacks. One countermeasure is to combine contention-based and contention-free schemes. In WMPANs, for example, a coordinator may allocate mini-slots at the beginning of each superframe, with each mini-slot allocated to a single device in a random order. The allocation information can be included in the beacon payload. Each device that wants to access the channel needs to wait until its mini-slot arrives. The mini-slots are used for channel access purpose rather than for data transmission as in the case of guaranteed time slots (GTSs) [5], and they should be short so as not to cause large delay. A device should begin to transmit its data within its mini-slot if it has data pending; otherwise it can just give up the mini-slot so that other following devices can access the channel. To prevent a cheating node from transmitting one packet after another without backoffs, we can similarly insert mini-slots into the contention access period (CAP) according to some schedule.

Flooding is often used for finding optimal routes. *Flooding* attacks in wireless networks are very harmful. Fortunately, they are preventable. The power of such attacks depends on how many nodes participate in the flooding for packet relay. If a broadcast packet is relayed only if it is from an authenticated source, then the broadcast can be well controlled. To prevent

those attacks, it would in general suffice to sign and authenticate routing entries and put a strict control over routing information update.

Route disruption attacks in the cluster-tree, no matter launched by a device or a coordinator, can be mitigated by performing authentication for source address, though it is difficult to completely rule out attacks launched by a coordinator. To prevent *loop* attacks in the cluster-tree, authentication for C_m, L_m, and the current tree level of the source should be provided. ***Resource consumption*** attacks in the cluster-tree stem from the fixed C_m and L_m. To counteract such attacks, we propose an adaptive block addressing scheme, in which short addresses can be adaptively assigned to reflect the actual network topology. This is achieved by splitting the tree formation procedure into three separate steps, namely, association, topology information collection, and short address assignment. After all devices join the network through association, network topology information can be collected and then used to guide short address assignment. "void address" attacks can be effectively blocked by checking the validity of short addresses. The node having detected that a packet is sent to a void address can unicast back a warning message to inform all the nodes along the path that the destination does not exist so that they can stop forwarding such packets.

10.5 Summary

As an enabling technology, WMPANs are expected to fill every aspect of our lives and play increasingly important roles. This chapter focuses on the security problems in those networks. We first present the security objectives that need to be achieved in WMPANs and, based on this, further provide a detailed analysis of the threats faced by WMPANs. A simulator is developed for modeling attacks and some experimental results are presented with discussions. A brief overview of the WMPAN security architecture defined by IEEE 802.15.4 and the ZigBee Alliance is also presented. Some problems are identified and remedies are suggested. Finally, countermeasures of various attacks are presented.

Chapter 11

A Lightweight Public Key Scheme: the derivable public key

Chapter 11

A Lightweight Public Key Scheme: the derivable public key

Public keys are generally used for distributing symmetric keys and for authentications for both pairwise and multicast/broadcast communications. Nevertheless, most existing public key schemes are too expensive for resource-constrained networks, as the case of most wireless networks. A lightweight public key scheme called Derivable Public Key (DPK) is proposed in this chapter. The DPK scheme distinguishes itself from other public key schemes by employing different key generation and management approach. It significantly simplifies the public key operations and reduces the related cost by eliminating the need for distributing and storing public keys other than the one from the trust center. This feature makes the DPK scheme very suitable for resource-constrained networks and for communications involving large number of time-varying communication entities.

11.1 Introduction

Symmetric keys are often used for coding data due to their high encryption/decryption efficiency. While they can also be used for authentications for pairwise communications, they are not suitable for authentications for multicast/broadcast communications. Public keys, on the other hand, are generally used for distributing symmetric keys and for authenti-

cations for both pairwise and multicast/broadcast communications. Nevertheless, it may be too expensive to use public key algorithms such as RSA [140], ElGamal [156], DSA [155], and Diffie-Hellman key agreement [157] in resource-constrained networks. This concern has led to the exclusion of public keys from most wireless networks, notwithstanding the need for key distributions and multicast/broadcast message authentications. As a result, some researchers have come up with other schemes for key distributions [42, 122, 125, 127] and multicast/broadcast message authentications [40, 42, 126, 139, 144, 152, 153] in those networks. Meanwhile, the effort to find more efficient public schemes is also under way. For example, Elliptic Curves Cryptography (ECC) [158], compared with the above public key algorithms, offers potential reductions in key size [169] as well as processing power and bandwidth due to the lack of a sub-exponential attack. In this chapter, we present a novel lightweight public key scheme called Derivable Public Key (DPK), including both the special version and the general version. The DPK scheme adopts some approaches used in ElGamal public key scheme [156], for example, prime number generation and digital signature algorithm. However, the DPK scheme employs quite different public key generation and management approach, which distinguishes itself from other public key schemes.

The DPK scheme significantly simplifies the public key operations and reduces the related cost by eliminating the need for distributing and stor-

ing public keys other than the one from the trust center. This unique feature makes the DPK scheme very suitable for resource-constrained networks and for communications involving large number of time-varying communication entities.

The remainder of this chapter covers the details of the DPK scheme. The special DPK scheme is presented in section 11.2; a simple example of the special DPK scheme is also provided. The general DPK scheme is described in section 11.3. And how to apply the DPK scheme to WMPANs is addressed in section 11.4. Finally conclusions are given in section 11.5.

11.2 The Special DPK Scheme

11.2.1 Setup

To set up the special DPK, the trust center proceeds as follows:

1. Choose a large prime p, such that $(p-1)$ has a big prime factor q and a primitive root $g \in \mathbb{Z}_p^*$, where $\mathbb{Z}_p^*$ is the prime residue class group modulo p.

2. Choose a hash function H:

   $$H: H_{in} = \{0, 1\}^* \rightarrow H_{out} = \{0, 1\}^k \in \mathbb{Z}_p^* \backslash \{p-1\}, \text{ but } \notin \mathbb{Z}_{p-1}^*.$$

   H is not necessary to be collision-resistant, but H_{out} should be an odd number and the greatest common divisor between H_{out} and $(p-1)$, $gcd(H_{out}, p-1)$, should be sufficiently large (see discussions below).

3. Optionally choose another hash function h:

$$h : h_{in} = \{0, 1\}^* \rightarrow h_{out} = \{0, 1\}^k \in \mathbb{Z}_p^*$$

4. Randomly choose two distinct large primes x_1 and x_2 within the range $1 \leq x_i \leq p - 2$ $(i = 1, 2)$.

5. Set the secret key of the trust center, sk_{tc}, to (x_1, x_2).

6. Compute $y_1 = g^{x_1}$ and $y_2 = g^{x_2}$.

7. For each device, generate the secret key

$$sk_i \;=\; (2a_i x_1 + x_2) H(a_i) \, mod \, (p - 1)$$

where $i = 1, 2, 3, ...$, and a_i is the address (or any other unique identifier) of the ith device.

8. Set the public key of the trust center, pk_{tc}, to (H, h, p, g, y_1, y_2).

9. Publish the public key of the trust center, pk_{tc}, to all the devices in the network. Distribute the secret keys of devices via some secure in-band key transport schemes or out-band methods.

11.2.2 Encryption

- To encrypt a message $m \in \{0, 1\}^k$ for the ith device, the sender selects a random integer r, $1 \leq r \leq p-2$, and compute the ciphertext c, which comprises a pair of elements in the residue class ring modulo p, $\mathbb{Z}_p$, as

follows[1]:

$$c = (g^r, m \oplus h(((y_1)^{2a_i} y_2)^{rH(a_i)})) \tag{11.1}$$

- The receiver can recover the message m from the ciphertext $c = (U, V)$, using its secret key sk_i, as follows:

$$\begin{aligned} m' &= V \oplus h(U^{sk_i}) \\ &= (m \oplus h(((y_1)^{2a_i} y_2)^{rH(a_i)})) \\ &\quad \oplus h(U^{sk_i}) \\ &= (m \oplus h(g^{(2a_i x_1 + x_2) rH(a_i)})) \\ &\quad \oplus h(g^{r(((2a_i x_1 + x_2) H(a_i)) \, mod \, (p-1))}) \\ &= (m \oplus h(g^{(2a_i x_1 + x_2) rH(a_i)})) \\ &\quad \oplus h(g^{r(2a_i x_1 + x_2) H(a_i)}) \\ &= m \end{aligned} \tag{11.2}$$

Here notice that $g^{p-1} = g^{\phi(p)} = 1$, where ϕ is *the Euler phi function*.

11.2.3 Digital Signature

- To sign a message $m \in \{0, 1\}^k$ (in practice, the hashed value of m rather than m itself is signed), the sender (assume the ith device here) proceeds as follows:

[1] All computation is carried out in $\mathbb{Z}_p$, and $(mod\, p)$ is assumed whenever necessary.

1. Select a random integer r within the range $1 \leq r \leq p-2$, with the greatest common divisor between r and $(p-1)$, $gcd(r,\, p-1) = 1$ (that is, r and $(p-1)$ are prime to each other).
2. Compute

$$\begin{aligned} \rho &= g^r \\ \sigma &= r^{-1}(m - (2\rho + 1)(sk_i))\, mod\, (p-1) \end{aligned}$$

3. The triple $(m,\, \rho,\, \sigma)$ is the signed message.

- The receiver can verify the signed message $(m,\, \rho,\, \sigma)$ as follows:

1. If ρ does not satisfy $1 \leq \rho \leq p-1$, reject the signature.
2. Compute

$$\begin{aligned} w &= (((y_1)^{2a_i} y_2)^{2\rho+1})^{H(a_i)} \rho^{\sigma} \\ v &= g^m \end{aligned}$$

3. Accept the signature if $w = v$; otherwise reject the signature. It can be proved that, if the ith device signed the message, then $w = v$ holds:

$$\begin{aligned} w &= \\ &= (((y_1)^{2a_i} y_2)^{2\rho+1})^{H(a_i)} \rho^{\sigma} \end{aligned}$$

$$
\begin{aligned}
&= ((g^{(2a_ix_1+x_2)(2\rho+1)})^{H(a_i)} \\
&\quad \rho^{r^{-1}(m-(2\rho+1)(sk_i))\, mod\, (p-1)} \\
&= g^{(sk_i)(2\rho+1)} g^{rr^{-1}(m-(2\rho+1)(sk_i))} \\
&= g^m \\
&= v
\end{aligned}
$$

Notice that

$$
\begin{aligned}
&g^{(2a_ix_1+x_2)H(a_i)} \\
&= g^{(2a_ix_1+x_2)H(a_i)\, mod\, (p-1)} \\
&= g^{sk_i}
\end{aligned}
$$

11.2.4 Discussions

1. The security of the special DPK scheme depends on the discrete logarithm assumption, that is, there is no efficient algorithm to calculate x from g^x. It also depends on the one-way feature of $(x_1,\, x_2) \rightarrow sk_i = (2a_ix_1 + x_2)H(a_i)\, mod\, (p-1)$, which makes it practically impossible to compute $(x_1,\, x_2)$ from sk_i.
2. It is not difficult to find the value of $(2a_ix_1+x_2)$ if $d_i = gcd(H(a_i),\, p-1)$ is not sufficiently large. Since $H(a_i) \notin \mathbb{Z}^*_{p-1}$, the inverse of $H(a_i)$

is not available in $\mathbb{Z}_{p-1}$ and the division $sk_i/H(a_i)$ is thus not defined in $\mathbb{Z}_{p-1}$. In other words, there is no unique solution for $(2a_ix_1 + x_2)$ such that $sk_i = (2a_ix_1+x_2)H(a_i)\,mod\,(p-1)$. However, one can find $f_{i.d} = (2a_ix_1 + x_2)\,mod\,((p-1)/d_i)$ using *the extended Euclidean algorithm*. And then $f_i = (2a_ix_1 + x_2)\,mod\,(p-1)$ must be in the form of $f_{i.d} + j * ((p-1)/d_i)$, where $j = 0, 1, 2, ..., d_i - 1$. If d_i is small, then finding $f_{i.d}$ is just as good as finding f_i. By knowing f_i, attackers can work in conspiracy to find the values of x_1 and x_2, and thus break the special DPK scheme. So it is important to ensure that d_i is sufficiently large.

3. One simple way to ensure that $d_i = gcd(H(a_i),\, p-1)$ is sufficiently large is to design the hash function *H* in such a way that the big prime factor of $(p-1)$, q, divides each $H(a_i)$. Namely, $H(a_i) = sq$, where s is some positive odd integer. Since $(p-1)$ is an even number and can be expressed as $2tq$, where t is some positive integer, the secret key can be expressed as

$$sk_i = (2a_ix_1 + x_2)sq\,mod\,(2tq) \tag{11.3}$$

$$= [(2a_ix_1 + x_2)s\,mod\,(2t)]q \tag{11.4}$$

This guarantees that sk_i is not zero, since $(2a_ix_1+x_2)s$ is an odd number and $2t$ is an even number. Equation (11.4) also reveals one short-

coming of the DPK scheme, that is, the maximum allowed number of secret keys is $t = (p-1)/(2q)$.

4. To decrypt a message, a device can use equation (11.2) if it has the secret key sk_i. However, it is also possible to decrypt the message by using equation (11.1) if the device knows the random number r. In normal case, only the destination has sk_i and only the source knows r, which prevents any third party (except the trust center) from decrypting the message.

5. It is not necessary for the source to generate a new r and transmit g^r to the destination along with each encrypted message. The source can use a single r to transmit a batch of messages. The frequency with which the random number r is refreshed depends on the nature of an application. However, different random number $r's$ should be used for encrypting messages for different destinations.

6. Since sk_i is an odd number, so is $(2\rho+1)(sk_i)$. Therefore

$$\begin{aligned} \sigma &= r^{-1}(m-(2\rho+1)(sk_i))\, mod\, (p-1) \\ &\neq r^{-1}m\, mod\, (p-1) \end{aligned}$$

That is, the secret key sk_i will not be left out when computing σ. But since it is just as difficult to find a random integer r to bypass the secret key sk_i as to find the secret key itself, we can replace $(2\rho+1)(sk_i)$ with

$\rho(sk_i)$ when computing σ (and modify the digital signature verification part accordingly).

7. For digital signature, it is important to select different random number $r's$ for different messages. If the same r is used for two different messages m_1 and m_2, then it is possible to compute r from $\sigma_1 - \sigma_2 = (m_1 - m_2)r^{-1} \, mod \, (p-1)$. And there is a high probability that the secret key sk_i can be computed by knowing r [170].

11.2.5 A Simple Example of the Special DPK Scheme

A simple example is given in this subsection for verifying the special DPK scheme (but not for testing the security of the special DPK scheme due to its simplicity).

11.2.5.1 Setup

1. Choose a large prime $p = 233$ and a primitive root $g = 17$.
2. Choose a hash function H:

   $H : \; H(a) = ((3a^3 - 5a^2 + 7) \, mod \, 8) * 29$, where $a = 1, \, 2, \, 3$ (3 nodes)
3. Optionally choose another hash function h:

   $h : \; h(b) = b$ (that is, we do not actually use a hash function here)
4. Randomly choose two distinct large primes $x_1 = 7$ and $x_2 = 13$ within the range $1 \leq x_i \leq p - 2$ $(i = 1, \, 2)$.

5. Set the secret key of the trust center, sk_{tc}, to $(7,\ 13)$.

6. Compute $y_1 = g^{x_1} = 17^7 = 43$ and $y_2 = g^{x_2} = 17^{13} = 191$.

7. For each device, generate the secret key $sk_i = (2a_i x_1 + x_2)H(a_i)\, mod\, (p-1)$:

$$\begin{aligned} sk_1 &= (2 \times 1 \times 7 + 13) \times H(1)\, mod\, 232 \\ &= 27 \times 5 \times 29\, mod\, 232 = 203 \\ sk_2 &= (2 \times 2 \times 7 + 13) \times H(2)\, mod\, 232 \\ &= 41 \times 3 \times 29\, mod\, 232 = 87 \\ sk_3 &= (2 \times 3 \times 7 + 13) \times H(3)\, mod\, 232 \\ &= 55 \times 3 \times 29\, mod\, 232 = 145 \end{aligned}$$

8. Set the public key of the trust center, pk_{tc}, to $(H,\ h,\ p,\ g,\ y_1,\ y_2) = (H,\ h,\ 233,\ 17,\ 43,\ 191)$.

9. Publish the public key of the trust center, pk_{tc}, to all the devices in the network. Distribute the secret keys of devices via some secure in-band key transport schemes or out-band methods.

11.2.5.2 Encryption

- Assume a node needs to send a message $m = 61$ to node 3 and it selects $r = 37$, then

$$\begin{aligned} c &= (g^r,\, m \oplus h(((y_1)^{2a_i} y_2)^{rH(a_i)})) \\ &= (17^{37},\, 61 \oplus h((43^6 \times 191)^{37\times 3\times 29})) \\ &= (75,\, 61 \oplus h((43^6 \times 191)^{13\times 232+203})) \\ &= (75,\, 61 \oplus (43^6 \times 191)^{203}) \\ &= (75,\, 61 \oplus 97) \\ &= (75,\, 92) \end{aligned}$$

- Then node 3 will recover the message m from $c = (U,\, V) = (75,\, 92)$, using its secret key $sk_3 = 145$, as follows:

$$\begin{aligned} m' &= V \oplus h(U^{sk_i}) \\ &= 92 \oplus h(75^{145}) \\ &= 92 \oplus (75^{145}) \\ &= 92 \oplus 97 \\ &= 61 \\ &= m \end{aligned}$$

11.2.5.3 Digital signature

- Assume node 3 needs to sign a message $m = 61$. It selects $r = 37$ and computes ρ and σ as follows:

$$\begin{aligned}
\rho &= g^r = 17^{37} = 75 \\
\sigma &= r^{-1}(m - (2\rho + 1)(sk_i)) \, mod \, (p - 1) \\
&= 37^{-1}((61 - 151 \times 145) \, mod \, 232 \\
&= 69 \times 206 \, mod \, 232 \\
&= 62
\end{aligned}$$

Note that $37^{-1} = 69 \, (mod \, 232)$.

So the signed message is $(m, \, \rho, \, \sigma) = (61, \, 75, \, 62)$.

- The receiver verifies the signed message as follows:

1. Check if ρ satisfies $1 \leq \rho \leq p - 1$: yes, it has $1 \leq 75 \leq 232$
2. Compute

$$\begin{aligned}
w &= (((y_1)^{2a_i} y_2)^{2\rho+1})^{H(a_i)} \rho^{\sigma} \\
&= (43^6 \times 191)^{151 \times 3 \times 29} \times 75^{62} \\
&= (43^6 \times 191)^{56 \times 232 + 145} \times 75^{62} \\
&= (43^6 \times 191)^{145} \times 75^{62} \\
&= 12 \times 226
\end{aligned}$$

$$
\begin{aligned}
&= 149 \\
v &= g^m \\
&= 17^{61} \\
&= 149
\end{aligned}
$$

3. Accept the signature since $w = v$.

11.3 The General DPK Scheme

11.3.1 Setup

To set up the general DPK, the trust center proceeds as follows:

1. Choose a large prime p, such that $(p-1)$ has a big prime factor q and a primitive root $g \in \mathbb{Z}_p^*$, where $\mathbb{Z}_p^*$ is the prime residue class group modulo p.

2. Choose a hash function H:

   $H : H_{in} = \{0, 1\}^* \rightarrow H_{out} = \{0, 1\}^k \in \mathbb{Z}_p^* \backslash \{p-1\}$, but $\notin \mathbb{Z}_{p-1}^*$.

   H is not necessary to be collision-resistant, but H_{out} should be an odd number and the greatest common divisor between H_{out} and $(p-1)$, $gcd(H_{out}, p-1)$, should be sufficiently large.

3. Optionally choose another hash function h:

   $h : h_{in} = \{0, 1\}^* \rightarrow h_{out} = \{0, 1\}^k \in \mathbb{Z}_p^*$

4. Randomly choose n distinct large primes $x_1, x_2, ..., x_n$ within the range $1 \leq x_i \leq p-2$ ($i = 1, 2, ..., n$).

5. Set the secret key of the trust center, sk_{tc}, to $(x_1, x_2, ..., x_n)$.

6. Randomly choose n distinct primes $c_1, c_2, ..., c_n$ and compute $y_i = g^{c_i x_i}$ ($i = 1, 2, ..., n$).

7. Randomly choose $(n-1)$ distinct positive integers $s_1, s_2, ..., s_{n-1}$, and, for each device, generate the secret key

$$sk_i \;=\; (2\sum_{j=1}^{n-1}(c_j a_i^{s_j} x_j) + c_n x_n) H(a_i) \, mod \, (p-1)$$

where $i = 1, 2, 3, ...$, and a_i is the address (or any other unique identifier) of the ith device.

8. Set the public key of the trust center, pk_{tc}, to $(H, h, p, g, y_1, y_2, ..., y_n, s_1, s_2, ..., s_{n-1})$.

9. Publish the public key of the trust center, pk_{tc}, to all the devices in the network. Distribute the secret keys of devices via some secure in-band key transport schemes or out-band methods.

11.3.2 Encryption

- To encrypt a message $m \in \{0, 1\}^k$ for the ith device, the sender selects a random integer r, $1 \leq r \leq p-2$, and compute the ciphertext c, which comprises a pair of elements in the residue class ring modulo p, $\mathbb{Z}_p$, as

follows:

$$c \;=\; (g^r,\, m \oplus h(((\prod_{j=1}^{n-1}(y_j)^{a_i^{s_j}})^2 y_n)^{rH(a_i)})) \tag{11.5}$$

- The receiver can recover the message m from the ciphertext $c = (U,\, V)$, using its secret key sk_i, as follows:

$$\begin{aligned} m^{'} &= V \oplus h(U^{sk_i}) \\ &= (m \oplus h(((\prod_{j=1}^{n-1}(y_j)^{a_i^{s_j}})^2 y_n)^{rH(a_i)})) \\ &\quad \oplus h(U^{sk_i}) \\ &= (m \oplus h(g^{(2\sum_{j=1}^{n-1}(c_j a_i^{s_j} x_j)+c_n x_n) rH(a_i)})) \\ &\quad \oplus h(g^{r(((2\sum_{l=1}^{n-1}(c_l a_i^{s_l} x_l)+c_n x_n)H(a_i))\, mod\, (p-1))}) \\ &= (m \oplus h(g^{(2\sum_{j=1}^{n-1}(c_j a_i^{s_j} x_j)+c_n x_n) rH(a_i)})) \\ &\quad \oplus h(g^{r(2\sum_{l=1}^{n-1}(c_l a_i^{s_l} x_l)+c_n x_n)H(a_i)}) \\ &\quad = m \end{aligned} \tag{11.6}$$

11.3.3 Digital Signature

- To sign a message $m \in \{0,\, 1\}^k$, the sender (assume the ith device here) proceeds as follows:

  1. Select a random integer r within the range $1 \leq r \leq p-2$, with the greatest common divisor between r and $(p-1)$, $gcd(r,\, p-1) = 1$

(that is, r and $(p-1)$ are prime to each other).

2. Compute

$$\begin{aligned} \rho &= g^r \\ \sigma &= r^{-1}(m-(2\rho+1)(sk_i))\, mod\, (p-1) \end{aligned}$$

3. The triple $(m,\ \rho,\ \sigma)$ is the signed message.

- The receiver can verify the signed message $(m,\ \rho,\ \sigma)$ as follows:

1. If ρ does not satisfy $1 \leq \rho \leq p-1$, reject the signature.
2. Compute

$$\begin{aligned} w &= ((\prod_{j=1}^{n-1}(y_j)^{a_i^{s_j}})^2 y_n)^{(2\rho+1)H(a_i)}\rho^{\sigma} \\ v &= g^m \end{aligned}$$

3. Accept the signature if $w = v$; otherwise reject the signature. It can be proved that, if the ith device signed the message, then $w = v$ holds:

$$\begin{aligned} w &= \\ &= ((\prod_{j=1}^{n-1}(y_j)^{a_i^{s_j}})^2 y_n)^{(2\rho+1)H(a_i)}\rho^{\sigma} \\ &= \left(g^{2\sum_{j=1}^{n-1}(c_j a_i^{s_j} x_j)+c_n x_n}\right)^{(2\rho+1)H(a_i)} \\ &\quad \rho^{r^{-1}(m-(2\rho+1)(sk_i))\, mod\, (p-1)} \end{aligned}$$

$$
\begin{aligned}
&= g^{(2\rho+1)(sk_i)} g^{rr^{-1}(m-(2\rho+1)(sk_i))} \\
&= g^m \\
&= v
\end{aligned}
$$

11.3.4 Discussions

1. To break the general DPK scheme, attackers need to find the values of c_1x_1, c_2x_2, ..., and c_nx_n. This needs three steps. The first step is to find

$$f_j = (2\sum_{l=1}^{n-1}(c_l a_{t_j}^{s_l} x_l) + c_n x_n) \, mod \, (p-1)$$

for $j = 1, 2, , ..., n$. For each j, a different t_j is used; so total n devices need to cooperate to provide the same number of $(f_j)'s$. The second step is to solve the following linear equations (denote $c_l x_l$ as X_l):

$$
\begin{cases}
2\sum_{l=1}^{n-1}(a_{t_1}^{s_l} X_l) + X_n &= f_1 \\
2\sum_{l=1}^{n}(a_{t_2}^{s_l} X_l) + X_n &= f_2 \\
\ldots & \ldots \ \ldots \\
2\sum_{l=1}^{n}(a_{t_n}^{s_l} X_l) + X_n &= f_n
\end{cases}
\tag{11.7}
$$

As we discussed in subsection 11.2.4, for each f_j, there are total d_{t_j} possible values, where $d_{t_j} = gcd(H(a_{t_j}), p-1)$. If we design the hash function H in such a way that the big prime factor of $(p-1)$,

q, divides each $H(a_{t_j})$, then each $d_{t_j} = \Theta(q)$. As a result, attackers must solve equations (11.7) $\Theta(q^n)$ times, each time using a different set of values ($f_1,\ f_2, ...,\ f_n$). After attackers find all the $\Theta(q^n)$ possible sets of values ($X_1,\ X_2,\ ...,\ X_n$), the third step is to determine which set of ($X_1,\ X_2,\ ...,\ X_n$) is the right one. This would require attackers to substitute each set of ($X_1,\ X_2,\ ...,\ X_n$) into the secret key formula

$$sk_i \ = \ (2\sum_{l=1}^{n-1}(c_l a_i^{s_l} x_l) + c_n x_n) H(a_i)\, mod\, (p-1)$$

for some devices not yet involved in the first two steps.

2. While a large n is preferred in theory, some tradeoff should be made between security and efficiency in reality. A practical strategy is to use a relatively small n and to refresh keys after a reasonable period.

3. It will be much more difficult to break the general DPK scheme if attackers are forced to solve non-linear equations instead of linear equations (11.7). This, however, can not be accomplished by simply replacing X_l with some term like $\varphi_l = c_l x_l^{t_l}$ or $\varphi_l = c_l \prod_z x_{u_z}^{t_z}$ in

$$sk_i \ = \ (2\sum_{l=1}^{n-1}(c_l a_i^{s_l} x_l) + c_n x_n) H(a_i)\, mod\, (p-1)$$

The reason is that equations in the form of

$$\begin{cases} 2\sum_{l=1}^{n-1}(a_{t_1}^{s_l}\varphi_l)+\varphi_n &= f_1 \\ 2\sum_{l=1}^{n-1}(a_{t_2}^{s_l}\varphi_l)+\varphi_n &= f_2 \\ \ldots & \ldots \ \ldots \\ 2\sum_{l=1}^{n-1}(a_{t_n}^{s_l}\varphi_l)+\varphi_n &= f_n \end{cases}$$

can be solved exactly as equations (11.7) by substituting φ_l for X_l. Notice that it suffices to find all $(X_l)'s$ or $(\varphi_l)'s$ (not all $(x_l)'s$) for breaking the general DPK scheme.

11.4 Applying the DPK Scheme to WMPANs

After the setup of DPK, pairwise communications can begin immediately by using the encryption and decryption procedures given in subsections 11.3.2and 11.3.3. However, for efficiency, two devices can negotiate to set up another link key and use that link key thereafter. They can determine how frequently the link key should be updated. Authentications for pairwise communications can also be done using the link key.

For broadcast, the public key of the trust center can serve as the Network key, but this makes the update of the Network key rather inflexible. Another way could be that the trust center generates a well-known broadcast key, for example, $bk = (2\sum_{l=1}^{n-1}(c_l x_l(\text{0xffff})^{s_l}) + c_n x_n)H(\text{0xffff})$, and publishes this key to all the devices in the network. And then any device knowing this key

can broadcast messages to the whole network using the same encryption algorithm given in subsection 11.3.2, but replacing $H(a_i)$ with $H(0xffff)$. Any device receiving the broadcast message can decrypt it using the well-known broadcast key bk plus the additional information g^r. Practically, we view g^r as the Network key, which can be easily distributed and updated. In this case, g^r should be distributed separately rather than included in each encrypted broadcast message.

It is important to notice that the Network key only provides limited protection for the network. Devices in WMPANs are resource-constrained and lack physical safeguards. A tampered or captured device could endanger the whole network. To reduce the risk, we propose a train multicast (TM) scheme using the DPK scheme. In TM, each coordinator (i.e., each router) generates a multicast key g^t just like generating the Network key g^r and distributes this key to all its children during the association procedure. This multicast key enables multicast communications among the coordinator and its children. A secure broadcast can be done through multiple multicasts. For example, the PAN coordinator can multicast a frame to all its children, and each child that acts as a router will decrypt the frame using the multicast key obtained from the PAN coordinator, encrypt the frame using its own multicast key, and then further multicast the frame to its children. This procedure continues until the frame reaches all the devices in the network. If a device has to broadcast a message, it can first unicast the message to

the PAN coordinator and then broadcast from there. A better way is to only unicast the frame to its coordinator. Then the coordinator will multicast this frame down along its branch and also unicast the frame to its coordinator, who will continue the same procedure. Besides data broadcast, TM can also be used to update the Network key, which has to be done by unicast before. A coordinator will multicast the new Network key to its children if none of its children has left the network since last key update; otherwise the new Network key will be unicast to those children still in the network, thus excluding those having left.

Like other public key schemes, the DPK scheme supports authentications for multicast and broadcast communications using the procedures given in subsection 11.3.3, but at a much lower cost. Authentications for multicast and broadcast communications are crucial for preventing many attacks.

11.5 Conclusions

A novel public key scheme, called derivable public key (DPK), is proposed in this chapter. In DPK, each device only needs to obtain the public key of the trust center and its own secret key. These two keys enable a device to communicate securely with any other device. By contrast, in other public key schemes, a device has to get the public key of each communication peer. In a network with N devices, this would mean that, if public keys

are not transmitted along with messages, each device needs to broadcast its public key to other devices and each device needs to store (N-1) public keys. For a network with large number of tiny devices, those old-good public key schemes consume too much network resources. As a result, public key schemes have so far been excluded from resource-constrained networks such as WMPANs. But now the DPK scheme holds the promise to change this situation and resource-constrained networks can expect to be better secured.

Part V

Conclusions and Future Work

Chapter 12

Conclusions and Future Work

Chapter 12

Conclusions and Future Work

12.1 Summary of the Dissertation

Chapter 1 sets forth the background, motivation, and contributions of this dissertation.

An overview of wireless mesh networking is given in chapter 2. The features of wireless mesh networks are summarized. The opportunities and challenges faced by this emerging technology are described. The ongoing wireless mesh standards activities around IEEE and ZigBee are also briefed.

Chapter 3 describes the features, applications, functions, and feasibility of IEEE 802.15.4 standard with respect to ubiquitous networking. The related ongoing mesh standards activities of IEEE 802.15.5 and ZigBee are also presented.

Chapter 4 evaluates the general performance of the IEEE 802.15.4 standard. An NS2 simulator, which covers all the 802.15.4 PHY and MAC primitives, is developed and five sets of experiments are designed for evaluating the performance of 802.15.4 under various conditions. The simulation

results show that 802.15.4 meets most of its design goals and is an energy-efficient standard favoring low data rate and low power consumption applications. However, the simulation work also identifies some problems of 802.15.4 and indicates that there is still space for improvement.

Two transmission scheduling schemes are presented in chapter 5. The simple receiver oriented TDMA (ROT) can be used to remove all primary collisions at a minimal addition of code size and control overhead. The medium access scheduling midware (MASM) is a contention free TDMA scheme and is able to eliminate all collisions, that is, both primary and secondary collisions. Some special design features such as using large time slot cycle (TSC), multi-level slot assignment, optimal time slot assignment, and self-correcting ability have made this scheduling scheme more robust and efficient than other TDMA schemes. As a midware, it also goes with different MAC and routing protocols.

A new MAC scheme based on CDMA, called PAR-CDMA, is proposed in chapter 6. The scheme supports both parallel receptions and accumulative receptions without the need for code assignment. Performance evaluation results show that, in terms of throughput, packet delivery ratio, hop delay, and transmission efficiency, PAR-CDMA towers above IEEE 802.11 in all scenarios.

A memory-efficient self-routing protocol, called adaptive robust tree (ART) routing, and its meshed form, meshed ART (MART) routing, are

introduced in chapter 7. The basis of ART/MART is an adaptive block addressing (ABA) scheme, which is used for logic address assignment as well as network autoconfiguration purpose. ABA takes into account the actual network topology and thus is fully topology-adaptive. By binding logic addresses to the network topology, routing can be carried out without going through route discovery. This eliminates the initial route discovery latency, saves storage space otherwise needed for routing table, and reduces the communication overhead and energy consumption.

Chapter 8 describes the dual routings (DR) approach. By combining MART with another on-demand wireless routing, DR can find optimal or near-optimal routes more efficiently than other on-demand wireless routings such as AODV [11] and AODVjr [31], whose route discovery is based on blind flooding. Our simulation results show that DR maintains good performance for both peer-to-peer and sink-type communications.

An efficient scalable wireless mesh routing protocol, called topology-guided distributed link state (TDLS), is proposed in chapter 9. TDLS comprises two basic schemes, namely, the adaptive block addressing (ABA) scheme and the distributed link state (DLS) scheme. The ABA scheme is in charge of network autoconfiguration and logic address assignment. The DLS scheme utilizes the address block information provided by the ABA scheme as a guideline to extract the next hop for relaying a data packet. Compared with ART and MART, it is able to render better performance

in terms of hop count or other routing cost metrics used, robustness, and load balancing. Our simulation results show that the TDLS scheme outperforms AODV in almost every respect under the scenarios used in simulation, notwithstanding its simplicity.

Chapter 10 focuses on the security problems in WMPANs. A detailed analysis of the threats faced by WMPANs is provided. A simulator is developed for modeling attacks and some experimental results are presented with discussions. A brief overview of the WMPAN security architecture defined by IEEE 802.15.4 and ZigBee is presented. Some problems are identified and remedies are suggested. Countermeasures of various attacks are also presented.

Finally, a novel public key scheme, called derivable public key (DPK), is proposed in chapter 11. In DPK, each device only needs to obtain the public key of the trust center and its own secret key. These two keys enable a device to communicate securely with any other device. The lack of authentications for multicast and broadcast communications in resource-constrained wireless networks accounts for many attacks. The DPK scheme holds the promise to change this situation and resource-constrained networks can expect to be better secured.

12.2 Open Issues and Future Research Directions

This subsection describes some open issues and possible future research directions for WMPANs, including enhancing ART/MART and TDLS, support of portable and mobile devices, testbed design, IPv6 over WMPANs, and coexistence. Another very attractive and feasible research area is wireless mesh LANs.

12.2.1 Enhancing ART/MART and TDLS

Our simulation results have shown ART/MART and TDLS perform well despite their simplicity. Particular, TDLS outperforms AODV in most cases. The initialization phase of ABA, the basis of ART/MART and TDLS, is an important phase. How to properly determine the time a network has finished the initialization is till an open issue. So far we simply use "average" scheme for address assignment. There must be better ways to do this. For example, if a hall is connected to several corridors, then those devices near corridors have a much higher chance to connect other devices along the corridors. So they should have a bigger address quota. In short, some work can be done with network commission tools.

12.2.2 Support of Portable and Mobile Devices

How to support portable and mobile devices has not been addressed in the context of ART/MART and TDLS. A quick thinking, however, shows that portable and mobile devices can be well supported by ART/MART and TDLS. Following are some initial thoughts:

- To avoid the problem of logic address change, portable and mobile devices are allowed to use their IEEE addresses only. Those devices will still associate with fixed devices, but no logic addresses will be assigned to them.
- To reach a portable or mobile device, the logic address of its parent will be used. When a packet reaches the parent, the parent will further deliver the packet to the device by checking a very short device sequence number (e.g., 4 bits for up to 16 under-care devices) included in the payload.
- Now the problem is how to know which fixed device a portable or mobile device is attached to. We can collect portable or mobile device information much like we collect children number information during the initialization phase of ABA. The information could be number of portable or mobile devices below each node, or simply a Boolean value indicating whether there is any portable or mobile device below a fixed device.

- When a portable node changes its attaching point, it will send update information, via unicast, to all those devices to which it has been communicating.
- If a portable or mobile device fails to update its communication peers and packets still arrive at the old parent, the old parent has a good chance to relay those packets to its new parent by using the link state table (LST) in TDLS. This can be done by proactively updating all neighbors within the TDLS scope when a fixed device finds any portable or mobile device has left it, or by reactively searching within the TDLS scope for the new parent when the old parent receives a packet.
- To initialize communications to a portable or mobile device, a node will do a multicast which will cover only those branches having portable or mobile devices attached. If there are only a few portable or mobile devices in the network, the overhead of the multicast is expected to be much less than that of a broadcast.

12.2.3 Testbed Design

At the current research stage, the importance of a testbed is obvious. A testbed can be used to verify the performance of ART/MART and TDLS, and to investigate practical issues not revealed by our simulations. While the simulation results for PAR-CDMA are very encouraging, it is yet to see the "reality" of this new MAC scheme by means of implementation.

Testbed is also an important tool, via which we can study the possibility of implementing IPv6 over WMPANs (see subsection 12.2.4). In what follows, we outline the objectives and possible approaches.

12.2.3.1 Objectives

The objectives of testbed design include:

- Functional components testing and performance analysis;
- Proposal prototyping and product testing;
  - Priori analysis and verification;
  - Proposal prototyping and analysis;
  - Product testing (initial functional testing, benchmark testing, etc.)
- Midware and system integration
  - Midware to glue different hardware and software (in the context of WMPANs)
  - Gateway/interface to existing wireless and wireline service delivery platforms

12.2.3.2 Possible approaches

An ideal testbed should

- be able to reflect/approximate the behavior of a real system;

- be flexible and powerful as a virtual system (e.g., software-based network simulators).

The key feature of our approach, therefore, can be described by the key word "duality".

- Programmable device and non-programmable device;
  - Programmable device;
    - Flexible and configurable;
    - Probe/snooper (can be interfaced to powerful computers for testing and analysis).
  - Non-programmable device;
    - Real product;
    - Proven (can serve as a golden device);
    - Time-efficient and cost-efficient for system integration.
- Hardware testbed and simulation/emulation testbed
  - Hardware testbed;
    - Real performance;
    - Difficult to set up a large scale network for testing.
  - Simulation/emulation testbed;
    - Not guaranteed to completely reflect the behavior of a real system;

    * Easy to simulate a large scale network.

  – Combining hardware testbed and simulation/emulation testbed;

    * Traffic from hardware testbed can be ported to simulation/emulation testbed, or vise versa;
    * By matching the behavior of a virtual device in a simulation/emulation testbed with that of a real device in a hardware testbed, a virtual device could act in good approximation as a real device.

  – Wireless link and wired link.

    * Wireless link;
      · normal wireless network operations;
    * Wired link
      · Monitoring and testing without affecting normal wireless network operations. For instance, all devices can be wired to a control center, which will log all the transmission and reception behavior of all the devices.

12.2.4 IPv6 over WMPANs

12.2.4.1 Background and motivation

With large amount WMPAN devices pervasively deployed, we will be provided the chance to closely “feel” and control the environments where

we live to an extent we can hardly imagine before. However, the success of WMPANs is not solely determined by how those networks have been designed, but also relies on how they are related to other technologies, especially those commercially proven technologies. In other words, whether a technology can well fit into the big picture of fait accompli technologies and how it makes use of existing technologies often, if not always, determines its eventual fate. With all the research work having been done in WMPANs, it is a good time to study the relationship between WMPANs and other related technologies, and zoom out research focus from WMPANs to their surroundings. The importance of such research comes from the fact that WMPANs target a unique type of applications and they are mainly designed as a complementary technology to other existing wireless and wired technologies. As such, it is imperative that we study and be able to answer the question how WMPANs will be interconnected to or even integrated into other network systems.

The great success of IP technologies urges us to take a close look at the issue how a WMPAN can be connected to the largest network in the world, Internet.

12.2.4.2 Possible approaches

One immediate approach is to use gateway technologies, as of the case of ZigBee ongoing research. Gateways make communication possible be-

tween different architecture and protocols. They repackage and convert data going from one network to another network. This approach puts the burden on gateway devices and hides the technical details of WMPANs from other networks (and vice versa). While gateways are good for data relaying, and perform well at the data link layer, they can not properly handle other upper layer functions such as network management and security. Two networks interconnected using gateways will be down-graded in terms of those upper layer features; in some cases, gateways could also be bottle-necks of the whole system.

An alternative approach is to make WMPANs IP-compatible. Due to the enormous number of WMPAN devices that could be deployed, IPv4 is obviously not a choice. IPv6, on the other hand, has a much larger address space (296 times of IPv4 address space and 264 times of IEEE address space) and has no difficulty to support the large number of WMPAN devices. Some of the advantages of an IP-compatible WMPAN are:

- routable to/from IP world;
- no network address translation (NAT) needed;
- making use of existing infrastructure;
- making use of simple network management protocol (SNMP) and other IP-oriented protocols;
- stateless address auto configuration;

- support of large scale and high density networks.

12.2.4.3 Open issues and possible solutions

WMPANs are based on a relatively simple protocol, IEEE 802.15.4, which is designed to support simple low-cost devices. Therefore, it is not practical to require a WMPAN to fully support IPv6, which is a complex protocol addressing various problems. We need to define a minimal IPv6-based stack on IEEE 802.15.4 to address the basic packet/header formats, efficient security model, and interface between layer 2 and layer 3. However in the case that full compatibility is required, some WMPAN nodes can be designated as gateways between WMPANs and IP networks. These nodes will be fully IP-compatible.

One immediate issue for designing an IP-compatible WMPAN is the different maximum transmission units (MTUs) used in IPv6 and IEEE 802.15.4. While IEEE 802.15.4 supports an MTU of 127 bytes, IPv6 requires a minimum MTU of 1280 bytes. A shim layer between IP and IEEE 802.15.4 will be needed to perform fragmentation and reassembly task. In order to save the scarce resource of WMPANs, this shim layer may need to be tightly bound to layer 2 and layer 3 and cross-layer optimization may be applied. For example, when a large IPv6 frame is fragmented into smaller IEEE 802.15.4 frames and transmitted over IEEE wireless links, it is desirable that this shim layer can terminate subsequent transmissions of IEEE

802.15.4 frames if the transmission of the previous IEEE 802.15.4 frame(s) from the same IPv6 frame fails, without waiting until the reassembly is done. This shim layer also needs to know the security requirement and the current security setting of the IPv6 frame and, based on this, decide whether to switch on or off the security processing at IEEE 802.15.4 MAC layer.

Header compression is another important issue to be addressed. Here we distinguish between two different level header compressions: one at IP level and one at MAC level. There are quite some literatures having addressed the IP header compression issue and therefore it is not our focus here. However, as we pointed out above, it may require that the shim layer be aware of the IP header compression processing status and be able to take proper actions when needed so that the network can operate more efficiently. For example, if an invalid context ID (CID) used in IP header compression is received, the shim layer may perform an early termination, that is, it will not go through all the fragment/reassembly process if the current IP frame is invalid.

MAC header compression plays an important role for an IP-compatible WMPAN. IEEE 802.15.4 uses an MTU of 127 bytes, out of which is the maximum 102 byte payload (or 81 byte payload if security is switched on). Without MAC header compression, a minimum 1280 byte IPv6 MTU will be fragmented into $1280/102 = 13$ (or $1280/81 = 16$ in the worst case) IEEE 802.15.4 frames, which results in an overhead of $13 * 25/1280 = 25.4\%$. This is a heavy tax on the resource-constrained WMPANs. The

situation will be totally different if MAC header compression is used. As an example, we can proceed as follows to compress the MAC header. First we assume batch transmission scheme is applied to the IP-compatible WMPAN, that is, all the IEEE 802.15.4 frames coming from a single IPv6 frame via fragmentation are transmitted in a batch. Under this assumption, we can construct two different types of IEEE 802.15.4 frames:

- A frame containing full header, which is the first IEEE 802.15.4 frame in a batch and is used to update or refresh the context for a batch of transmissions;
- A frame not containing full header, but carrying CID and its transmission order (i.e., sequence number) in the batch represented by the CID.

In most cases, one byte (at most two bytes) will be enough for the CID plus the transmission order in a batch, for example, 3 bits for CID, 4 bits for transmission order, and 1 bit as ending flag of a batch of transmissions. By using this scheme, the full frame header only needs to be transmitted once and the overhead in bytes now is: 25 (full header) + 1 (CID of first frame in the batch) + 10 (CIDs of subsequent frames, note that we only need 11 IEEE 802.15.4 frames now) = 36 bytes. This corresponds to an overhead in percentage of $36/1280 = 2.8\%$. One requirement in this scheme is that the IEEE 802.15.4 MAC layer needs to handle the special frames not containing full headers, or at least the IEEE 802.15.4 MAC layer can recognize the CID

contained in the frame and then pass the frame to shim layer rather than discard the frame as an invalid or corrupted frame.

Some other research issues also need to be addressed, including service discovery, potential changes at layer 3 and above, implementations, and best common practices of an IPv6 stack.

12.2.5 Coexistence

WMPANs can be deployed by anybody, at any time, at various locations, and with diverse coverage areas. With this versatility, another issue, coexistence, surfaces. It would be a disaster if wireless mesh networks can not coexist with other wireless networks or among themselves. Wireless devices need to share frequency bands, much like cars need to share lanes. If nobody follows rules, one goes nowhere. To develop a viable market, coexistence is one of the issues we need to look at in our future work.

12.2.6 Research for Wireless Mesh LANs

There is no doubt that wireless mesh LANs (WMLANs) will be one of most important mesh technologies. Some of our research work in WMPANs can be readily extended to WMLANs, including mesh topology learning, routing, and forwarding, mesh discovery and association, mesh medium access coordination, and mesh security.

Bibliography

[1] K. Sundaresan, R. Sivakumar, M. A. Ingram, and T.-Y. Chang, "A fair medium access control protocol for ad hoc networks with MIMO links," *IEEE Annual Conference on Computer Communications* (INFOCOM), 2004, pp. 2559-2570.

[2] W. Xiang, T. Pratt, and X. Wang, "A software radio testbed for two-transmitter two-receiver space time coding wireless LAN," *IEEE Communications Magazine* 42 (6), 2004, pp. S20-S28.

[3] R. Ramanathan, "On the performance of ad hoc networks with beamforming antennas," *ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MOBIHOC), Oct. 2001, pp. 95-105.

[4] A. Spyropoulos and C. S. Raghavendra, "Asymptotic capacity bounds for ad hoc networks revisited: the directional and smart antenna cases," *IEEE Global Telecommunications Conference* (GLOBECOM), 2003, pp. 1216-1220.

[5] IEEE P802.15.4/D18, Draft Standard: *Low Rate Wireless Personal Area Networks*, Feb. 2003.

[6] J. Hauser, D. Baker, and W. S. Conner, IEEE 802 11-04-0054-02, "Project authorization request (PAR) of IEEE 802.11 TGs," 2003.

[7] W. Steven Conner, IEEE 802 11-04/662r16, "IEEE 802.11 TGs usage models," 2005.

[8] N. Borisov, I. Goldberg, and D. Wagner, "Intercepting mobile communications: the insecurity of 802.11," *ACM Annual International Conference on Mobile Computing and Networking* (MOBICOM), Sept 2002, pp. 180-188.

[9] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," *IEEE Wireless Communications* 11 (1), 2004, 38-47.

[10] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks,* 2, pp. 351-367, 2004.

[11] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing," IETF RFC 3561, Jul. 2003.

[12] D. Johnson, D. A. Maltz, and J. Broch, "The dynamic source routing protocol for mobile ad hoc networks," IETF Internet draft, draft-ietf-manet-dsr-09.txt, Apr. 2003.

[13] T. Clausen and P. Jacquet, "Optimized link state routing protocol," IETF RFC 3626, Oct. 2003.

[14] R. G. Ogier, F. Templin, and M. Lewis, "Topology broadcast based on reverse-path forwarding (TBRPF)," IETF RFC 3684, Feb. 2004.

[15] J. Moy, "Open shortest path first routing protocol," IETF RFC 2328, Apr. 1998.

[16] G. Malkin, "The routing information protocol," IETF RFC 2453, Nov. 1998.

[17] FIPS Pub 197, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T, Springfield, Virginia, Nov. 2001.

[18] P. Kinney, IEEE 802 15-04-0042-01, "Project authorization request (PAR) of IEEE 802.15 TG5," 2004.

[19] H. Shao, IEEE 802 15-04-0655-00, "Technical requirements of IEEE 802.15 TG5," 2004.

[20] IEEE 802.11, Part 11: *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE, Aug. 1999.

[21] Bluetooth SIG, Bluetooth Specifications, V1.0, July 1999.

[22] USC Information Sciences Institute, Marina del Rey, CA. Network Simulator – NS2. (http://www.isi.edu/nsnam/ns).

[23] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," *Proc. MOBICOM*, 2001.

[24] A. Y. Wang, S.-H. Cho, C. G. Sodini, and A. Chandrakasan, "Energy efficient modulation and MAC for asymmetric RF microsensor systems," *IEEE Intl. Symp. Low Power Electronics and Design*, 2001.

[25] J. Heidemann, W. Ye, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM 2002), New York, NY, Jun. 2002.

[26] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, Vol. 19, No. 2, Mar. 2002.

[27] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," *UCLA/CSD-TR 02-0013*, UCLA Computer Science, 2002.

[28] http://www.ieee802.org/15/pub/05/15-05-0260-00-0005-802-15-5-mesh-networks.pdf.

[29] L. Hester, Y. Huang, A. Allen, O. Andric, and P. Chen, "neuRFon Netform: A self-organizing wireless sensor network," *Proceedings of the 11th IEEE ICCCN Conference*, Miami, Florida, Oct. 2002.

[30] ZigBee Specification, V 1.0, Jun. 2005.

[31] I. Chakeres and L. Klein-Berndt, "AODVjr, AODV simplified," *ACM SIGMOBILE Mobile Computing and Communications Review*, pp. 100-101, Jul. 2002.

[32] V. Rijmen and J. Daemen, "The block cipher," Rijndael. In J.-J. Quisquater and B. Schneier, editors, *Smart Card Research and Applications*, LNCS 1820, pages 288-296, Springer-Verlag, 2000.

[33] M. Bellare, J. Kilian, and P. Rogaway, "The security of the cipher block chaining message authentication code," *Journal of Computer and System Sciences*, 61(3):362-399, Dec. 2000.

[34] D. Whiting, R. Housley, and N. Ferguson, "Counter with CBC-MAC (CCM)," RFC 3610, Sept. 2003.

[35] USC Information Sciences Institute, Marina del Rey, CA. *Network Animator – Nam*. (http://www.isi.edu/nsnam/nam).

[36] Y. Liu, M. Lee, and T. Saadawi, "A bluetooth scatternet-route structure for multi-hop ad hoc networks," *IEEE Journal on Select Areas in Communications*, Vol. 21, No. 2, pp.229-239, Feb. 2003.

[37] A. Cerpa and D. Estrin, "Adaptive self-configuring sensor networks topologies," In *Proc. IEEE INFCOM*, New York, June 2002.

[38] A. Y. Wang, S. Cho, C. G. Sodini, and A. P. Chandrakasan, "Energy efficient modulation and MAC for asymmetric RF microsensor systems," *IEEE Intl. Symp. Low Power Electronics and Design*, 2001.

[39] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," In *First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[40] A. Perrig, R. Canetti, D. Song, and D. Tygar, "The TESLA broadcast authentication protocol," In *RSA Cryptobytes*, Summer 2002.

[41] Y. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications* (WMCSA 2002), pp. 3-13, IEEE, Calicoon, NY, June 2002.

[42] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," *Conference on Computer and Communications Security. Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, 2002.

[43] D. Ganesan, B. Krishnamachari, et al., "Complex behavior at scale: An experimental study of low-power wireless sensor networks," *UCLA/CSD-TR 02-0013, UCLA Computer Science*, 2002.

[44] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," In *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*, pages 151-162, ACM Press, 1999.

[45] http://www.ieee802.org/15/pub/04/15-04-0222-00-004b-zigbee-802-15-4-enhancement-database.xls.

[46] A. Woo and D. Culler, "A transmission control scheme for media access in sensor networks," In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking* (MOBICOM-01), pages 221-235, ACM Press, New York, 2001.

[47] R. D. Pietro, L. V. Mancini, and A. Mei, "Random key assignment for secure wireless sensor networks," In *Proceedings of the 2003 ACM Workshop on Security of Ad Hoc and Sensor Networks* (SASN '03), October 2003.

[48] M. Zorzi and R. R. Rao, "Multihop performance of energy-efficient forwarding for ad hoc and sensor networks in the presence of fading," *IEEE/ICC'04*, 20-24 June 2004, Paris, France.

[49] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *IEEE Computer Magazine*, October 2002, pp. 54-62.

[50] J. H. Schiller, *Mobile Communications*, Addison-Wesley, 2000.

[51] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad-hoc network research," *J. Wireless Commun. Mob. Comput.* (Mobile Ad-Hoc Networking–Research, Trends and Applications), vol. 2, pp. 483-502, 2002.

[52] G. R. Cooper and C.D. McGillem, *Modern Communications and Spread Spectrum*, McGraw Hill, New York, 1986.

[53] K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. W. III, "On the capacity of a cellular CDMA system," *IEEE Transactions on Vehicular Technology*, 40:303-312, May 1991.

[54] S. Gopalan, G. N. Karystinos, and D. A. Pados, "Capacity, throughput, and delay of slotted ALOHA DS-CDMA links with adaptive space-time auxiliary-vector receivers," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 79-92, Jan. 2005.

[55] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communications*. Reading, MA: Addison-Wesley, 1995.

[56] T. Ojanperä and R. Prasad, *Wideband CDMA for Third Generation Mobile Communications*, Artech House, Incorporated, 1998.

[57] J. G. Proakis, *Digital Communications*, McGraw-Hill, Inc., 2001.

[58] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Transactions on Networking*, 1:668-677, Dec. 1993.

[59] A. A. Bertossi and M. A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE Transactions on Communications*, 3(4):441-449, Aug. 1995.

[60] T. Makarrsi, "Transmitted-oriented code assignment for multihop packet radio," *IEEE Transactions on Communications*, vol. 35, Dec. 1987, pp. 1379-1382.

[61] R. Battiti, A. A. Bertossi, and M. A. Bonuccelli, "Assigning codes in wireless networks: bounds and scaling properties," *Wireless Networks*, vol. 5, issue 3, May 1999.

[62] J. Garcia-Luna-Aceves and J. Raju, "Distributed assignment of codes for multihop packet radio networks," in *Proc. of the IEEE MILCOM Conference*, vol. 1, 1997, pp. 450-454.

[63] J. E. Wieselthier, A. Ephremides, and J. A. B. Tarr, "A distributed reservation-based CDMA protocol that does not require feedback information," *IEEE Transactions on Communication*s, vol. 36, Aug. 1988, pp. 913-923.

[64] A. Banerjee, R. A. Iltis, and E. A. Varvarigos, "Performance evaluation for a quasi-synchronous packet radio network (QSPNET)," *IEEE/ACM Transactions on Networking*, vol. 9, no. 5, Oct. 2001.

[65] W.-T. Chen and N.-F. Huang, "The strongly connecting problem on multihop packet radio networks," *IEEE Transactions on Communication*s, vol. 37, Mar. 1989, pp. 293-295.

[66] J. Herdtner and E. Chong, "Analysis of a class of distributed asynchronous power control algorithms for cellular wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, Mar. 2000.

[67] D. Kim, "On the convergence of fixed-step power control algorithms with binary feedback for mobile communication systems," *IEEE Transactions on Communication*s, vol. 49, no. 2, Feb. 2001, pp. 249-252.

[68] C.-H. Liu and H. H. Asada, "A source coding and modulation method for power saving and interference reduction in DS-CDMA sensor network systems," in *Proc. American Control Conf.*, Anchorage, AK, May 2002, pp. 3003-3008.

[69] O. Dousse, F. Baccelli, and P. Thiran, "Impact of interference on connectivity in ad hoc networks," in *Proc. IEEE INFOCOM*, San Francisco, CA, Apr. 2003, pp. 1724-1733.

[70] S. De, C. Qiao, D. A. Pados, M. Chatterjee, and S. J. Philip, "An integrated cross-layer study of wireless CDMA sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, Sept. 2004, pp. 1271-1285.

[71] A. Muqattash and M. Krunz, "CDMA-based MAC protocol for wireless ad hoc networks," in *Proc. ACM MobiHoc*, Annapolis, MD, Jun. 2003, pp. 153-164.

[72] W. R. Heinzelman,W. R. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocols for wireless microsensor networks," in *Proc. Hawaiian Intl. Conf. Systems Science*, Maui, HI, Jan. 2000, pp. 3005-3014.

[73] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in

ad hoc wireless networks," in *Proc. ACM MOBICOM*, Rome, Italy, Jul. 2001, pp. 85-96.

[74] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large-scale wireless sensor networks," in *Proc. ACM MOBICOM*, Atlanta, GA, Sept. 2002, pp. 148-159.

[75] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 5, Jun. 1994, pp. 796-807.

[76] P. Frenger, P Orten, and T. Ottosson, "Code-spread CDMA with interference cancellation," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 12, Dec. 1999, pp. 2090-2095.

[77] M. Kobayashi, J. Boutros, and G. Caire, "Successive interference cancellation with SISO decoding and EM channel estimation," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 8, Aug. 2001, pp. 1450-1460.

[78] L. Fang and L. Milstein, "Performance of successive interference cancellation in convolutionally coded multicarrier DS/CDMA systems," *IEEE Transactions on Communication*s, vol. 49, Dec. 2001, pp. 2062-2067.

[79] J. G. Andrews and T. Meng, "Performance of MC-CDMA with successive interference cancellation in a multipath fading channel," *IEEE Transactions on Communication*s, vol. 52, no. 5, May 2004, pp. 811-822.

[80] M. K. Varanasi and B. Aazhang, "Multistage detection in asynchronous code-division multiple access communications," *IEEE Transactions on Communication*s, vol. 38, no. 4, Apr. 1990, pp. 509-519.

[81] T. R. Giallorenzi and S. G. Wilson, "Suboptimum multiuser receivers for convolutionally coded asynchronous DS-CDMA systems," *IEEE Transactions on Communication*s, vol. 44, no. 9, Sept. 1996, pp. 1183-1196.

[82] D. Divsalar, M. Simon, and D. Raphaeli, "Improved parallel interference cancellation for CDMA," *IEEE Transactions on Communication*s, vol. 46, no. 2, Feb. 1998, pp. 258-268.

[83] A. Nahler, R. Irmer, and G. Fettweis, "Reduced and differential parallel interference cancellation for CDMA systems," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 2, Feb. 2002, pp. 237-447.

[84] M. Sawahashi, K. Higuchi, H. Andoh, and F. Adachi, "Expirements on pilot-assisted coherence multistage interference canceller for DS-CDMA mobile radio," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 2, Feb. 2002, pp. 433-449.

[85] R. Schober and L. Lampe, "Differentially coherent parallel interference cancellation for DS-CDMA," *IEEE Transactions on Wireless Communication*s, vol. 4, no. 3, 2005, pp. 1030-1039.

[86] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, "Iterative multiuser detection for CDMA with FEC: near single user performance," *IEEE Transactions on Communication*s, vol. 46, no. 12, Dec. 1998, pp. 1693-1699.

[87] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C.B. Schlegel, "Iterative multiuser interference reduction: turbo CDMA," *IEEE Transactions on Communication*s, vol. 47, no. 7, Jul. 1999, pp. 1008-1014.

[88] X. Wang and H. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Transactions on Communication*s, vol. 47, no. 7, Jul. 1999, pp. 1046-1061.

[89] H. Schoeneich and P. A. Hoeher, "Single antenna interference cancellation: iterative semi-blind algorithm and performance bound for joint maximum-likelihood interference cancellation," *Proc. IEEE GLOBECOM*, San Francisco, CA, Dec. 2003, pp. 1716-1720.

[90] A. Nordio, M. A. Hernandez, and G. Caire, "Low-complexity turbo equalization and multiuser decoding for TD-CDMA," *IEEE Trans-*

*actions on Wireless Communication*s, vol. 3, no. 2, Mar. 2004, pp. 454-465.

[91] R. M. Buehrer and R. Mahajan, "On the usefulness of outer-loop power control for successive interference cancellation," *IEEE Transactions on Communication*s, vol. 51, no. 12, Dec. 2003, pp. 2091-2102.

[92] J. G. Andrews and T. H. Meng, "Optimum power control for successive interference cancellation with imperfect channel estimation," *IEEE Transactions on Wireless Communication*s, vol. 2, no. 2, Mar. 2003, pp. 375-383.

[93] Y. Wang, C.-Y. Tsui, R. S. Cheng, and W. H. Mow. "Power control of CDMA systems with successive interference cancellation using the knowledge of battery power capacity," *Asia and South Pacific Design Automation Conference*, 2004, pp. 125-130.

[94] A. Agrawal, J. G. Andrews, J. M. Cioffi, and T. Meng, "Iterative power control for imperfect successive interference cancellation," *IEEE Transactions on Wireless Communication*s, vol. 4, no. 3, 2005, pp. 878-884.

[95] Z. J. Haas, M. R. Pearlman, and P. Samar, "The bordercast resolution protocol for ad hoc networks," IETF: draft-ietf-manet-zone-brp-02.txt, Jul. 2002.

[96] K. Carlberg and J. Crowcroft, "Building shared trees using a one-to-many joining mechanism," *ACM SIGCOMM Computer Communication Review*, Vol. 27, Issue 1, pp. 5-11, 1997.

[97] J.-J. Pansiot and D. Grad, "On routes and multicast trees in the Internet," *ACM SIGCOMM Computer Communication Review*, Vol. 28, Issue 1, pp. 41-50, 1998.

[98] R. G. Ogier, "Efficient routing protocols for packet radio networks based on tree sharing," Proc. Sixth IEEE Intl. Workshop on Mobile Multimedia Communications (MOMUC'99), Nov. 1999.

[99] C.-C. Chiang, M. Gerla, and L. Zhang, “Adaptive shared tree multicast in mobile wireless networks," *IEEE Communications Magazine*, Vol. 3, pp. 1817-1822, Aug. 1998.

[100] N.-F. Tzeng and P. Alla, “Guided shared trees for efficient multicast in large networks," *Proc. IEEE Int’l Conference on Communications* (ICC 2003), May 2003.

[101] Sajama and Z. J. Haas, “Independent-tree ad hoc multicast routing (ITAMAR)," *Mobile Networks and Applications*, Vol. 8, Issue 5, pp. 551-566, 2003.

[102] ZigBee Alliance, http://www.zigbee.org.

[103] Mesh Networking Forum, Building the business case for implementation of wireless mesh networks, *Mesh Networking Forum* 2004, San Francisco, CA, Oct. 2004.

[104] I. F. Akyildiz, X. Wang, and W. Wang, “Wireless mesh networks: a survey,” *Computer Networks* 47(4): 445-487, 2005.

[105] IEEE 802.11 Standard Group. http://www.ieee802.org/11.

[106] IEEE 802.15 Standard Group. http://www.ieee802.org/15.

[107] IEEE 802.16 Standard Group. http://www.ieee802.org/16.

[108] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, “Capacity of ad hoc wireless networks,” *Proc. seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom), pp. 61-69, 2001.

[109] L. Huang, and T. Lai, “On the scalability of IEEE 802.11 ad hoc networks,” *ACM International Symposium on Mobile Ad Hoc Networking and Computing* (MOBIHOC), 2002, pp. 173-182.

[110] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, “Impact of interference on multi-hop wireless network performance,” *ACM Annual International Conference on Mobile Computing and Networking* (MOBICOM), Sept. 2003, pp. 66-80.

[111] J. Zheng and M. J. Lee, "Will IEEE 802.15.4 make ubiquitous networking a reality?: A discussion on a potential low power, low bit rate standard," *IEEE Communications Magazine*, Vol. 42, Issue 6, pp 140-146, Jun. 2004.

[112] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15.4," *Sensor Network Operations*, IEEE Press, 2006.

[113] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility," *IEEE WCNC*'99, New Orleans, LA, Sep. 1999.

[114] M. Joa-Ng and I-Tai Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, Aug. 1999.

[115] M. Jiang, J. Li, and Y. C. Tay, "Cluster based routing protocol," IETF draft: draft-ietf-manet-cbrp-spec-01.txt, Aug. 1999.

[116] K. Xu, X. Hong, and M. Gerla, "Landmark routing in ad hoc networks with mobile backbones," *Journal of Parallel and Distributed Computing*, Special Issue on Ad Hoc Networks, 63 (2), pp. 110-122, 2002.

[117] E. M. Belding-Royer, "Multi-level hierarchies for scalable ad hoc routing," *ACM/Kluwer Wireless Networks*, 9 (5), 2003, pp. 461-478.

[118] Amtel Corporation. http://www.atmel.com

[119] C. Zhu, M. Lee, and T. Saadawi, "A border-aware broadcast scheme for wireless ad hoc network," *IEEE Consumer Communications and Networking Conference*, 2004.

[120] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," In *First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[121] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks,"

In *Proceedings of the sixth annual international conference on Mobile computing and networking*, pp. 56-67, ACM Press, New York, NY, 2000.

[122] L. Zhou and Z. Haas, "Securing Ad Hoc Networks," *IEEE Networks Special Issue on Network Security*. Nov./Dec. 1999.

[123] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," *MobiCom 2002*, Sept. 2002, Atlanta, Georgia, USA.

[124] Y. Desmedt and Y. Frankel, *Threshold cryptosystems*, In G. Brassard, editor, Advances in Cryptology – Crypto '89 (Lecture Notes in Computer Science 435), pages 307-315, Santa Barbara, California, U.S.A., August 1990. Springer-Verlag.

[125] Y. Desmedt, *Threshold cryptography*, European Transactions on Telecommunications, 5(4):449-457, July-August 1994.

[126] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," In *Wireless Networks Journal* (WINE), Sept. 2002.

[127] A. Perrig, H. Chan, and D. Song, "Random key predistribution schemes for sensor networks," In *IEEE Symposium on Security and Privacy*, 2003.

[128] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, and R. Rao, "Cooperation in wireless ad hoc networks," In *Infocom*, 2003.

[129] J.-H. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," *Proc. of INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.

[130] V. Srinivasan, P. Nuggehalli, C-F. Chiasserini, and R. R. Rao, "Optimal rate allocation and traffic splits for energy efficient routing in ad hoc networks," *Proc. of Infocom 2001*, New York City, Jun. 2001.

[131] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," *Proc. of MobiCom 2000*, Boston, Aug. 2000.

[132] L. Blazevic, L. Buttyán, S. Capkun, S. Giordano, J. P. Hubaux, and J. Y. Le Boudec, "Self-organization in mobile ad-hoc networks: The approach of terminodes," IEEE Communications Magazine,Vol. 39, No. 6, Jun. 2001.

[133] L. Buttyán and J. P. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," In *ACM/Kluwer Mobile Networks and Applications* (MONET 2002).

[134] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao, "Energy efficiency of ad hoc wireless networks with selfish users," *European Wireless Conference 2002* (EW2002), Florence, Italy, Feb. 2002.

[135] N. Ben Salem, L. Buttyán, J. P. Hubaux, and M. Jakobsson, "A charging and rewarding scheme for packet forwarding in multi-hop cellular networks," In *MobiHoc* 2003.

[136] M. Felegyhazi, L. Buttyán and J. P. Hubaux, "Equilibrium analysis of packet forwarding strategies in wireless ad hoc networks – the static case," In *Proceedings of Personal Wireless Communications* (PWC '03), Venice, Italy, Sept. 2003.

[137] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," In *Proceedings of the 2004 ACM Workshop on Wireless Security*, Oct. 2004.

[138] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," In *IEEE Symposium on Security and Privacy*, May 2000.

[139] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," In *Network and Distributed SystemSecurity Symposium*, NDSS 01, Feb. 2001.

[140] R. L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM* 21(2) (1978) 120-126.

[141] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilent to byzantine failures," In

ACM Workshop on Wireless Security (WiSe), Atlanta, Georgia, Sept. 2002.

[142] S. Yi, P. Naldurg, and R. Kravets, "Security-aware ad-hoc routing for wireless networks," *MobiHoc* Poster Session, 2001.

[143] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," In *Proceedings of The Sixth International Conference on Mobile Computing and Networking* (MobiCom 2000), Boston, MA, Aug. 2000.

[144] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive security for multi-layer ad-hoc networks," In *John Wiley InterScience Press journa*l Special Issue of Wireless Communications and Mobile Computing, Aug. 2002.

[145] P. Dasgupta and S. Gokhale, "Distributed authentication for peer to peer networks," In *IEEE Workshop on Security and Assurance in Ad hoc Networks*, in conjunction with *the 2003 International Symposium on Applications and the Internet*, Orlando, FL, Jan. 2003.

[146] Benaloh, Josh, and M. de Mare, "One way accumulators: A decentralized alternative to digital signatures," *Advances in Cryptology Proceedings of Eurocrypt 93*, Springer-Verlag, 1993.

[147] Baric, Niko, and B. Pfitzmann, "Collision-free accumulators and fail-stop signature schemes without trees," *Advances in Cryptology Eurocrypt 97*, v. 1233 of LNCS, Springer-Verlag, 1997: 480-494.

[148] A. Khalili, J. Katz, W. A. Arbaugh, "Towards secure key distribution in truly ad-hoc networks," *IEEE Workshop on Security and Assurance in Ad hoc Networks*, in conjunction with *the 2003 International Symposium on Applications and the Internet*, Orlando, FL, Jan. 2003.

[149] R. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh, "Bootstrapping security associations for routing in mobile ad-hoc networks, Technical Research Report, Institute for Systems Research, University of Maryland. (available at *http://techreports.isr.umd.edu/reports/2002/TR_2002-44.pdf*)

[150] G. Montenegro and C. Castelluccia, "Statistically unique and cryptographically verifiabel (SUCV) identifiers and addresses," *Proceedings of the 2002 Network and Distributed System Security conference* (NDSS02), San Diego, Feb. 2002.

[151] G. Montenegro and P. Nikander, "Protecting against bidding down attacks," Internet Draft, draft-montenegro-mipv6sec-bit-method-00.txt April 2002.

[152] G. O'Shea and M. Roe, "Child-proof authentication for MIPv6 (CAM)," *ACM Computer Communication Review*, Apr. 2001.

[153] J.-P. Hubaux, L. Buttyán, and S. Capkun, "The quest for security in mobile ad hoc networks," In *Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking and computing 2001*, Long Beach, CA, USA.

[154] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly and Associates, 1995.

[155] ANSI X9.30-1, *The digital signature algorithm (DSA) (revised)*, American Bankers Association, working draft, July 1999.

[156] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, 31 (1985), 469-472.

[157] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, IT-22:644-654, Nov. 1976.

[158] I. Blake, G. Seroussiand, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.

[159] N. Koblitz, *Elliptic curve cryptosystems*, Mathematics of Computation, 48 (1987), 203-209.

[160] ANSI X9.62, *The elliptic curve digital signature algorithm (ECDSA)*, American Bankers Association, 1999.

[161] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread spectrum communications*, volume I-II. Computer Science Press, Rockville, MD, 1985.

[162] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, 75(1):56-73, Jan. 1987.

[163] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of spread spectrum communications – a tutorial," *IEEE Transactions on Communications*, 30(5):855-884, May 1982.

[164] M. B. Pursley and H. B. Russel, "Routing in frequency-hop packet radio networks with partial-band jamming," *IEEE Transactions on Communications*, 41(7):1117-1124, Jul. 1993.

[165] A. A. Hassan, W. E. Stark, and J. E. Hershey, "Frequency-hopped spread spectrum in the presence of a flower partial-band jammer," *Transactions on Communications*, 41(7):1125-1131, Jul. 1993.

[166] K. S. Kwak and J. W. Park, "Multiuser detection scheme using adaptive antenna array over Rayleigh fading channels," In *Vehicular Technology Conference Proceedings* (VTC), Vol. 3, pp. 2157-2161, Spring 2000.

[167] H. Ko, J. H. Lee, and B. Yu, "A switched beamforming system with multiuser detectors," In *Vehicular Technology Conference Proceedings* (VTC), Vol. 2, pp. 705-709, Spring 2000.

[168] S. Ray, J. B. Carruthers, and D. Starobinski, "RTS/CTS-included congestion in ad hoc wireless LANs," In *WCNC*, 2003.

[169] A. Lenstra and E. Verheul, "Selecting cryptographic key sizes," *Journal of Cryptology*, Vol. 14, No. 4, pp. 255-293, 2001.

[170] H. Delfs and H. Knebl, *Introduction to Cryptography*, Springer, 2002.