TECHNISCHE
UNIVERSITÄT
DRESDEN

Faculty of Computer Science Institute of Systems Architecture, Chair of Computer Networks

ENHANCING MOBILITY IN LOW POWER WIRELESS SENSOR NETWORKS

Jianjun Wen, M.Sc.
Born on: 23rd July 1982 in Taiyuan, China

DISSERTATION

to achieve the academic degree

DOKTOR-INGENIEUR (DR.-ING.)

First referee
Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill, TU Dresden
Second referee
Prof. Dr. Dieter Hogrefe, University of Göttingen
Advisor
Prof. Dr.-Ing. Eduard A. Jorswieck, TU Dresden

Submitted on: 13th July 2018
Defended on: 9th October 2018

SUPERVISOR

PD Dr.-Ing. habil. Waltenegus Dargie
Chair of Computer Networks
Faculty of Computer Science
Technical University of Dresden
Dresden, Germany

DOCTORAL COMMITTEE

Prof. Alexander Schill (first reviewer)
Chair of Computer Networks
Faculty of Computer Science
Technical University of Dresden
Dresden, Germany

Prof. Dieter Hogrefe (second reviewer)
Telematics Group
Institute of Computer Science
University of Göttingen
Göttingen, Germany

Prof. Sebastian Rudolph (chair)
International Center for Computational Logic
Faculty of Computer Science
Technical University of Dresden
Dresden, Germany

Prof. Eduard A. Jorswieck (subject expert)
Chair of Communications Theory
Faculty of Electrical and Computer Engineering
Technical University of Dresden
Dresden, Germany

Prof. Diana Göhringer (committee member)
Chair of Adaptive dynamic Systems
Faculty of Computer Science
Technical University of Dresden
Dresden, Germany

To my Parents

ABSTRACT

In the early stages of wireless sensor networks (WSNs), low data rate traffic patterns are assumed as applications have a single purpose with simple sensing task and data packets are generated at a rate of minutes or hours. As such, most of the proposed communication protocols focus on energy efficiency rather than high throughput. Emerging high data rate applications motivate bulk data transfer protocols to achieve high throughput. The basic idea is to enable nodes to transmit a sequence of packets in burst once they obtain a medium. However, due to the low-power, low-cost nature, the transceiver used in wireless sensor networks is prone to packet loss. Especially when the transmitters are mobile, packet loss becomes worse. To reduce the energy expenditure caused by packet loss and retransmission, a burst transmission scheme is required that can adapt to the link dynamics and estimate the number of packets to transmit in burst. As the mobile node is moving within the network, it cannot always maintain a stable link with one specific stationary node. When link deterioration is constantly detected, the mobile node has to initiate a handover process to seamlessly transfer the communication to a new relay node before the current link breaks. For this reason, it is vital for a mobile node to (1) determine whether a fluctuation in link quality eventually results in a disconnection, (2) foresee potential disconnection well ahead of time and establish an alternative link before the disconnection occurs, and (3) seamlessly transfer communication to the new link.

In this dissertation, we focus on dealing with burst transmission and handover issues in low power mobile wireless sensor networks. To this end, we begin with designing a novel mobility enabled testing framework as the evaluation testbed for all our remaining studies. We then perform an empirical study to investigate the link characteristics in mobile environments. Using these observations as guidelines, we propose three algorithms related to mobility that will improve network performance in terms of latency and throughput:

i) Mobility Enabled Testing Framework (MobiLab). Considering the high fluctuation of link quality during mobility, protocols supporting mobile wireless sensor nodes should be rigorously tested to ensure that they produce predictable outcomes before actual deployment. Furthermore, considering the typical size of wireless sensor networks and the number of

parameters that can be configured or tuned, conducting repeated and reproducible experiments can be both time consuming and costly. The conventional method for evaluating the performance of different protocols and algorithms under different network configurations is to change the source code and reprogram the testbed, which requires considerable effort. To this end, we present a mobility enabled testbed for carrying out repeated and reproducible experiments, independent of the application or protocol types which should be tested. The testbed consists of, among others, a server side control station and a client side traffic flow controller which coordinates inter- and intra-experiment activities.

ii) Adaptive Burst Transmission Scheme for Dynamic Environment. Emerging high data rate applications motivate bulk data transfer protocol to achieve high throughput. The basic idea is to enable nodes to transmit a sequence of packets in burst once they obtain a medium. Due to the low-power and low-cost nature, the transceiver used in wireless sensor networks is prone to packet loss. When the transmitter is mobile, packet loss becomes even worse. The existing bulk data transfer protocols are not energy efficient since they keep their radios on even while a large number of consecutive packet losses occur. To address this challenge, we propose an adaptive burst transmission scheme (ABTS). In the design of the ABTS, we estimate the expected duration in which the quality of a specific link remains stable using the conditional distribution function of the signal-to-noise ratio (SNR) of received acknowledgment packets. We exploit the expected duration to determine the number of packets to transmit in burst and the duration of the sleeping period.

iii) Kalman Filter Based Handover Triggering Algorithm (KMF). Maintaining a stable link in mobile wireless sensor network is challenging. In the design of the KMF, we utilized combined link quality metrics in physical and link layers, such as Received Signal Strength Indicator (RSSI) and packet success rate (PSR), to estimate link quality fluctuation online. Then Kalman filter is adopted to predict link dynamics ahead of time. If a predicted link quality fulfills handover trigger criterion, a handover process will be initiated to discover alternative relay nodes and establish a new link before the disconnection occurs.

iv) Mobile Sender Initiated MAC Protocol (MSI-MAC). In cellular networks, mobile stations are always associated with the nearest base station through intra- and inter-cellular handover. The underlying process is that the quality of an established link is continually evaluated and handover decisions are made by resource rich base stations. In wireless sensor networks, should a seamless handover be carried out, the task has to be accomplished by energy-constraint, resource-limited, and low-power wireless sensor nodes in a distributed manner. To this end, we present MSI-MAC, a mobile sender initiated MAC protocol to enable seamless handover.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor PD Dr.-Ing. habil. Waltenegus Dargie for his patient and instructive guidance through my entire Ph.D research life. I have worked with him since the early stage of my master study as a student assistant in his energy lab. He taught me a lot, not only how to do the research, but also his attitude of life. He is a person pursuing perfection, especially for research, which has a profound impact on me. He was always patient to listen to my immature ideas and arguments. I thank to him for all his effort of helping me to improve my scientific English, to construct my research ideas and to teach me how to be a thorough researcher.

I would also like to express my profound gratitude to Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill for providing me the opportunity to do research in his group and always giving me valuable feedback. Many thanks for his patience and kindness which gave me great power to conquer each challenge till this successful end. Furthermore, I would like greatly appreciate my advisor Prof. Dr.-Ing. Eduard A. Jorswieck and external reviewer Prof. Dr. Dieter Hogrefe for their kindness to review this dissertation and provide valuable feedback.

I want to express my next gratitude to my office mate and friends: Zeeshan who was always tolerant of my countless complaints about challenging research problems and always stood by and encouraged me. I do remember we often discussed with each other and wrote papers till late night together. Many thanks go to my best friend Taiba, who is considerate and sweet, for feeding me many gourmet food and encouraging me patiently when I felt depressed.

Additionally, many thanks go to my brilliant colleagues and students who were always supportive. I want to especially appreciate Chen Liang, Weiyi He for their hard work to conduct a large number of experiments with me. Sebastian Itting, who partially implemented and tested the robot control scripts for the testbed. Dr. Tenshi Hara, who was always enthusiastic to help me solve documentary problems and shared his research experience with me.

Another special acknowledgments go to my proof-readers. They are Dr. Taiba Zahid, Dr. Qian Dong, Dr. Lisa Lines and my supervisor. Without their careful and patient help, this dissertation would be more difficult to articulate.

I also want to express my gratitude to many of my friends in Germany and in China. Yi Zhang, Yunyi Fu, Ran Zhang, Hui Chen, Chao Li, Nikolaos Draganoudis. Especially, I would like to express my profound gratitude to Prof. Xianping Chen who is my best friend, brother and mentor, for encouraging and supporting me to study abroad.

Last but not the least, I want to express my most profound gratitude to my parents for their selfless love and support, for believing in me for no reason. To my father, thank you for teaching me to be a strong and honest man since I was in childhood. To my late mother, thank you for your hardwork and countless sacrifices for our family. You have always encouraged me to pursue my dreams even in the last days of struggling with the cancer.

Jianjun Wen
Dresden, October 2018

This work has been partially funded by the German Research Foundation (DFG) under project agreements: DA 1211/5-1 and SFB-921

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABTS	Adaptive Burst Transmission Scheme
ACK	Acknowledgment
ARR	Acknowledgment Reception Ratio
BAN	Body Area Networks
CCA	Clear Channel Assessment
CF	Consecutive Failure
CPDF	Conditional Probability Distribution Function
CS	Consecutive Success
CTI	Cross Technology Interference
CTS	Clear to Send
DC	Duty Cycle
IPI	Inter Packet Interval
LMS	Least Mean Square Filter
LPL	Low Power Listening
LPP	Low Power Probing
LQE	Link Quality Estimator
LQI	Link Quality Indicator
MAC	Medium Access Control
MWSN	Mobile Wireless Sensor Network
OSI	Open System Interconnection

PAN	Personal Area networks
PDF	Probability Distribution Function
PDR	Packet Delivery Ratio
PFR	Packet Failure Rate
PHY	Physical Layer
PRR	Packet Reception Ratio
PSR	Packet Success Rate
QoS	Quality of Service
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
TTL	Time to Live
WSN	Wireless Sensor Network

LIST OF PUBLICATIONS

1. **Jianjun Wen** and Waltenegus Dargie. A Handover Triggering Algorithm for Managing Mobility in WSNs. In Proceedings of the 21st International Conference on Information Fusion (FUSION), pages 1-7. IEEE, 2018.
2. **Jianjun Wen** and Waltenegus Dargie. A Mobility Management Protocol for Wireless Sensor Networks. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC), pages 1-7. IEEE, 2018.
3. **Jianjun Wen**, Zeeshan Ansar, and Waltenegus Dargie. Mobilab: A Testbed for Evaluating Mobility Management Protocols in WSN. In International Conference on Testbeds and Research Infrastructures, pages 49-58. Springer, 2016.
4. **Jianjun Wen**, Zeeshan Ansar, and Waltenegus Dargie. A System Architecture for Managing Complex Experiments in Wireless Sensor Networks. In Proceedings of the 25th IEEE International Conference on Computer Communication and Networks (ICCCN), pages 1-9. IEEE, 2016.
5. **Jianjun Wen**, Zeeshan Ansar, and Waltenegus Dargie. A Link Quality Estimation Model for Energy-Efficient Wireless Sensor Networks. In Proceedings of the IEEE International Conference on Communications (ICC), pages 6694-6700. IEEE, 2015.
6. Zeeshan Ansar, **Jianjun Wen**, and Waltenegus Dargie. Efficient Online Burst Transmission Scheme for Wireless Sensor Networks. In Proceedings of the 25th IEEE International Conference on Computer Communication and Networks (ICCCN), pages 1-9. IEEE, 2016.
7. Zeeshan Ansar, **Jianjun Wen**, Eyuel Debebe Ayele, and Waltenegus Dargie. An Efficient Burst Transmission Scheme for Wireless Sensor Networks. In Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pages 151-155. ACM, 2015.
8. Eyuel D Ayele, **Jianjun Wen**, Zeeshan Ansar, and Waltenegus Dargie. Adaptive Sleep--Time Management Model for WSNs. In Proceedings of the 24th IEEE International Conference on Computer Communication and Networks (ICCCN), pages 1-7. IEEE,

2015.

9. Waltenegus Dargie and **Jianjun Wen**. A MAC Protocol for Medical Applications. In Proceedings of the 13th international symposium on Information processing in sensor networks (IPSN), pages 287-288. IEEE Press, 2014.
10. Waltenegus Dargie and **Jianjun Wen**. Power-Latency Trade-Offs in Virtualized Environments. In Proceedings of the 28th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA), pages 398-403. IEEE, 2014.
11. Waltenegus Dargie and **Jianjun Wen**. A Seamless Handover for WSN Using LMS Filter. In Proceedings of the 39th IEEE Conference on Local Computer Networks (LCN), pages 442-445. IEEE, 2014.
12. Waltenegus Dargie and **Jianjun Wen**. A Probabilistic Model for Estimating the Power Consumption of Processors and Network Interface Cards. In Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 845-852. IEEE, 2013.

1 INTRODUCTION

Wireless sensor networks are useful for a large number of applications which require mobile nodes. For example, in the healthcare domain, they have been proposed to monitor patients with Parkinson Disease[SXFL15], gastroparesis [GOV+03], epilepsy [LEVG+13], and asthma [SGS+09]. As a result, there have been endeavors to integrate medical devices and make them interact with existing wireless sensor platforms. For instance, the wireless motility capsule integrating pH, pressure, and temperature sensors for the diagnosis of gastroparesis has officially been approved by the United States Food and Drug Administration (FDA) since 2006. It has produced promising results and may replace existing invasive and painful procedures such as endoscopy [RKM+09]. Most existing or proposed healthcare applications rely on Body Area Networks (BAN) and are often self-contained. In a BAN, nodes transfer sensed data to a mobile phone or a laptop computer, which is carried by the user or placed nearby. In some applications, individual nodes temporarily store the data they sensed locally, which are then offloaded, either manually or automatically, to a base station whenever the user happens to be at a close proximity. The advantage of the first strategy is that live and steady monitoring can be supported. One of its disadvantages is that the user is force to always carry an additional device (mobile phone or a laptop) with them or make do with restricted mobility. The advantage of the second is that the user can enjoy unrestricted mobility but the applications have to be delay tolerant. Moreover, individual nodes should have sufficient storage.

The scope and usefulness of the proposed healthcare applications can be significantly enhanced if the BAN they employ is augmented by Personal Area Networks (PAN). In places such as the patient's home or rehabilitation centers, additional and stationary nodes can be strategically placed so that the BAN can interact with them to transfer data to a remote base station where the information is available to an expert or for advanced data processing. However, there are two formidable challenges. First, most healthcare applications require high throughput, low latency, and low jitter to ensure that the sensed data is reliable (to determine whether measurements are correlated, for example). Second, it is difficult to maintain a reliable link between a mobile node and a stationary node, because the quality of a wireless link, in addition to distance, strongly depends on mobility.

To address the aforementioned challenges, this dissertation focuses on designing and implementing a set of novel communication protocols in mobile wireless sensor networks (MWSN) in which nodes with limited resources are static-mobile mixed and transmission is in burst.

1.1 CHALLENGES IN MOBILE WIRELESS SENSOR NETWORK COMMUNICATION

In wireless sensor networks, the typical device is usually equipped with a low-power microcontroller unit (MCU), a low-power low-data-rate radio transceiver, and limited memory resources (RAM and ROM). It is often powered by a battery with constraint capacity. For example, the most commonly used platform in WSNs, TelosB mote [PSC05], uses a 16-bits MSP430 MCU with 48 KB ROM and 10 KB RAM, works at a frequency of 8 MHz and integrates with an IEEE 802.15.4 compliant transceiver ChipCon CC2420. This mote is powered by two AAA batteries. These limitations in hardware resources pose several challenges when designing communication protocol in WSNs.

Energy efficiency. Energy efficiency is the most important concern when designing communication protocols. Since the motes are battery powered and usually deployed in harsh environments or in large quantities, changing or recharging the batteries is often impossible. Compared to other components in the mote, the radio transceiver consumes most of the energy. To achieve the long node and network lifetime requirement, duty cycling mechanism is commonly used in WSNs. As such, the nodes are periodically in deep sleep and active mode. The ratio of sleep and active period is called the duty cycle (DC). This parameter can be adjusted according to the traffic load in the network.

Link dynamics. Due to the low power feature, the radio transceiver used in WSNs is prone to packet loss. To ensure the quality of service requirement, retransmission mechanism is widely used at different protocol layers (such as MAC, routing etc.). However, retransmission can degrade the performance of the network in terms of throughput, as many packet retransmissions increase the collision probability and in turn aggravate the packet loss. Furthermore, inefficient retransmission causes energy wastage.

Besides the aforementioned challenges, dealing with mobility in energy constraint, resource limited, low-power wireless sensor networks poses more challenges.

1) Links in MWSNs experience more dynamic fluctuations compared to their stationary counterpart. Link dynamics have been studied exhaustively in the static environment by the WSNs research community. Prior works show that low power wireless links are affected by many factors [SKAL08][DÖD11][BZB$^+$14][LC12], such as temperature, interference and background noise, which result in link quality degradation in time span from hundreds of milliseconds to several seconds. In mobile environments, mobility is the main contributor to link dynamics. To highlight this challenge, a simple experiment is conducted. A static receiver

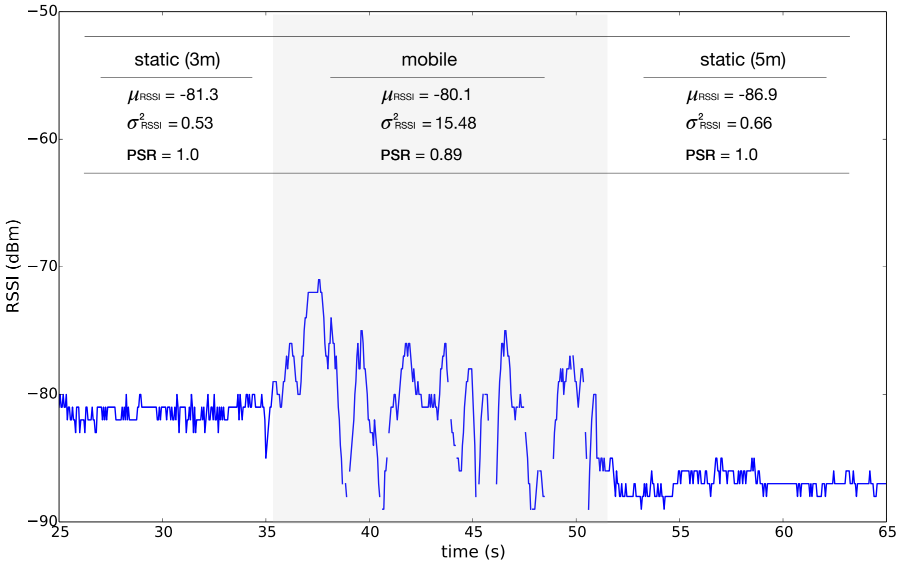


Figure 1.1: Link dynamics. The link is between a static receiver and a mobile transmitter which is carried by a robot. The mobile transmitter first sends packets at a location 3 meters away from the receiver for 10 seconds (static (3m) in the figure). It then moves toward a five-meter location at a speed of 0.13 m/s (mobile in the figure). Finally, it stays at the 5 meter location, transmitting packets for another 10 seconds (static (5m) in the figure). The data is collected at the transmitter side. The mean value (μ_{RSSI}), variance (σ^2_{RSSI}) of RSSI and PSR are calculated accordingly.

and a mobile transmitter are deployed. The transmitter first stays at 3 meter location away from the receiver for 10 seconds and then moves toward the 5 meter location at a speed of $0.13 m/s$ and stays there for another 10 seconds, while transmitting packets continuously. We collect the received signal strength indicators (RSSI) and the acknowledgments (ACK) at the transmitter side and calculate the link quality in terms of RSSI fluctuation (mean and variance) and packet success rate (PSR) at each section. As shown in Figure 1.1, we observe that: a) the movement of the mobile node introduces more packet losses than the static scenario; b) the RSSI experiences more fluctuation, which is as high as more than 10 dB. These observations require mobile communication protocol has the ability of being resilient to the transient link dynamics and agile to react to persistent link quality degradation.

2) In MWSNs, the devices are mobile, instead of static, resulting in more frequent topology changes. In traditional static WSNs, network dynamics is usually coped with at the network layer. The node has to access the quality of links and update their routing information periodically by using link quality estimators and specific routing metrics. Usually, this updating activity is at very low frequency, since the link quality changes slowly over time in a fixed environment and link quality estimation over the path is costly for both time and energy. However, due to the mobility nature, the link conditions of mobile nodes are changing more frequently, thus resulting in more frequent topology updating. This constant updating requirement exacerbates scarce bandwidth resources and increase energy expenditure. So leaving the mobile node out of routing construction is a better option. Furthermore, despite the link dynamics, the mobile node may move out of the communication range of its current receiver, which can lead to disconnection from the network and increase latency. To deal with this issue, a seamless handover mechanism is required so that a new link can be established before the current link breaks.

3) Although duty cycling significantly improves energy efficiency, it poses the challenge of discovering alternative relay nodes. As we know, to achieve the long node and network lifetime requirement, the duty cycling mechanism is commonly used. Before data transmission, the transmitter and the receiver have to be active and idle at the same time. Especially in an asynchronous schedule, the transmitter has to wait for a long time to be synchronous with the intended receiver, which increases latency. In static deployment, this delay can be shortened by learning the neighbors' schedule in advance, for example, the phase lock mechanism in ContikiMAC [Dun11]. In contrast, in mobile environment, the mobile node does not have enough information or is unaware of neighboring nodes before probing them. Hence, choosing an appropriate alternative receiver before the current link breaks is time consuming.

1.2 OBJECTIVES OF THE DISSERTATION

With the growing interest in involving mobility in WSNs applications, it is of paramount importance to enhance the low power communication protocol supporting mobility efficiently. Among all of the solutions, seamless handover (or handoff) should be the best option. The term handover is defined as a process a mobile node transfers communication from one relay node to the other. For more details, we assume a scenario illustrated as Figure 1.2. A mobile node has bulk data to transfer and moves freely within the network. According to the moving scope, mobility can be classified into two categories: intra-network mobility and inter-network mobility. Intra-network mobility refers to the mobile node roaming within a specific WSN domain where its address remains the same during handover. Inter-network mobility refers to the mobile node moving from one WSN domain to the other. In this case, during the handover process, the mobile node's address has to be re-assigned by the new network. In this dissertation, we only focus on intra-network mobility issues.

When the mobile node is ready to transfer bulk data, it first joins the network and then begins transmission in burst with a stationary node (relay node). Due to link dynamics, the link quality fluctuates over time, which leads to packet loss. As a result, transmitting data packet continuously without awareness of link quality is not energy efficient. To reduce the energy expenditure caused by packet loss and retransmission, a burst transmission scheme is required, which can adapt to the link dynamics and estimate the proper number of packets to transmit in burst. Moreover, as the mobile node is moving within the network, it cannot always maintain a stable link with one specific stationary node. When link deterioration is

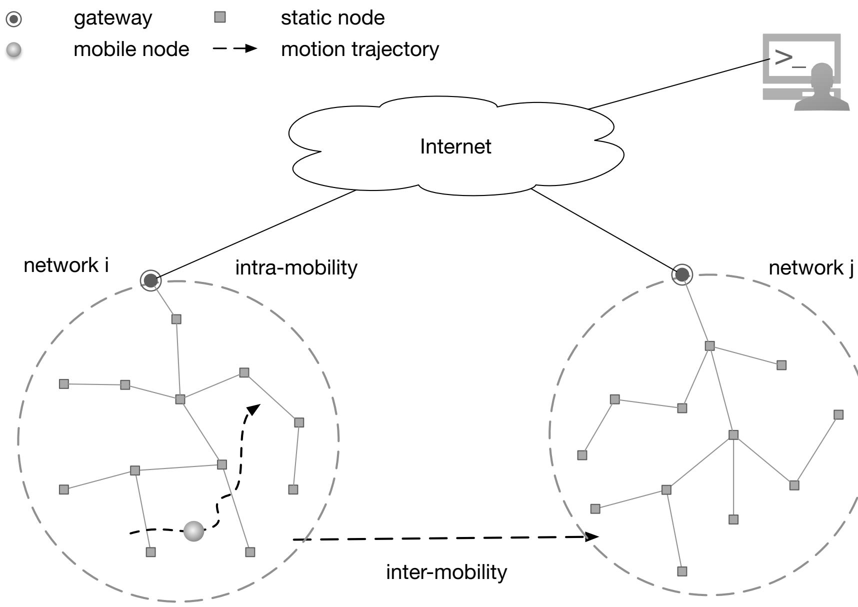


Figure 1.2: Mobility scope

constantly detected, the mobile node has to initiate a handover process to seamlessly transfer communication to a new relay node before the current link breaks. For this reason, it is vital for a mobile node to be able to (1) determine whether a fluctuation in link quality eventually results in a disconnection, (2) foresee potential disconnection well ahead of time and establish an alternative link before the disconnection occurs, and (3) seamlessly transfer communication to the new link.

In this dissertation, the following questions are addressed:

1. What is the optimal burst transmission size when the mobile node attains a stable link.
2. When is the most appropriate time to initiate the handover process when the mobile node detects constant link quality deterioration with the current relay node.
3. How to discover alternative receivers in the vicinity of the mobile node during the handover process.
4. Which neighbor node is the best option to seamlessly handover to before the current link breaks.

1.3 CONTRIBUTIONS OF THE DISSERTATION

In this dissertation, we focus on dealing with mobility issues in low power mobile wireless sensor networks. To this end, we begin with designing a novel mobility enabled testing framework as the evaluation testbed for our remaining studies. We then perform an empirical study to investigate the link characteristics in mobile environment. Using these observations as guidelines, we propose a set of algorithms and communication protocols that deal with

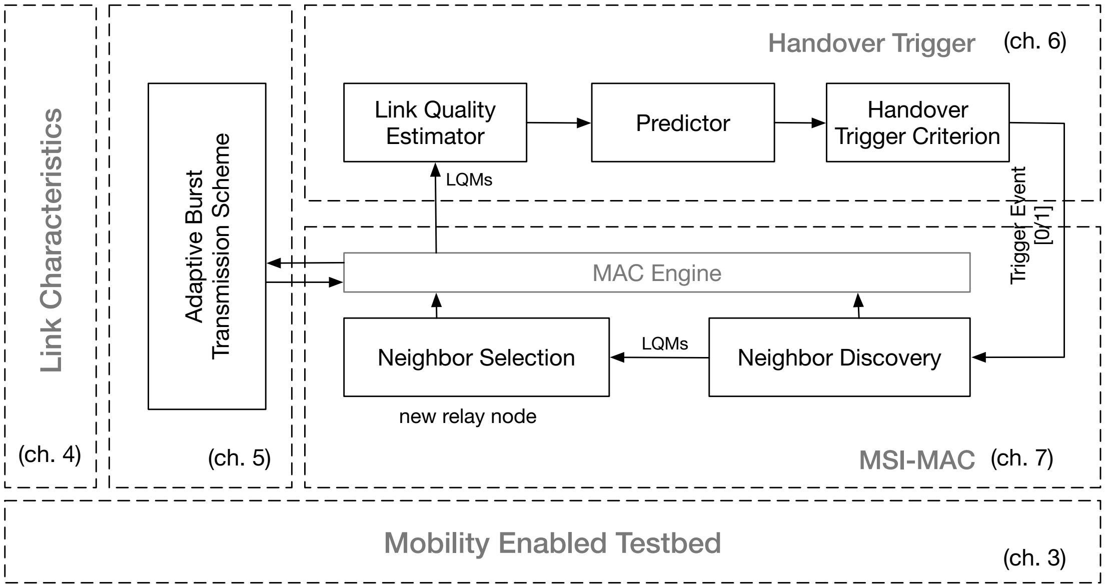


Figure 1.3: Overview and task description of this dissertation

mobility to improve network performance in terms of latency and throughput. Figure 1.3 illustrates the research components of this dissertation. The main contributions are highlighted as follows.

MOBILAB: MOBILITY ENABLED TESTING FRAMEWORK – CHAPTER 3

Wireless sensor networks that support the mobility of nodes are finding applications in different areas such as healthcare, elderly care, and rehabilitation from total knee and hip replacement. These application areas also require reliable and high throughput networks. Considering the high fluctuation of link quality during mobility, protocols supporting mobile wireless sensor nodes should be rigorously tested to ensure that they produce predictable outcomes before actual deployment. Furthermore, considering the typical size of wireless sensor networks and the number of parameters that can be configured or tuned, conducting repeated and reproducible experiments can be both time consuming and costly. The conventional method for evaluating the performance of different protocols and algorithms under different network configurations is to change the source code and reprogram the testbed, which requires considerable effort.

In this chapter, we present a wireless sensor network testbed for carrying out repeated and reproducible experiments, independent of the application or protocol types which should be tested. The testbed consists of, among others, a server side control station and a client side traffic flow controller which coordinates inter- and intra-experiment activities. The testbed is fully implemented for the TinyOS and TelosB platforms. Diddyborg robots are employed for emulating different types of movement in indoor and outdoor environments.

LINK CHARACTERISTICS OF MOBILE WIRELESS SENSOR NETWORKS – CHAPTER 4

Radio link quality of low-power WSNs is one of the essential factors that should be taken into consideration when designing Medium Access Control (MAC), or routing protocols. Due to low-cost, low-power features, the radio transceivers used in WSNs can be affected by background noise, multi-path fading, shadowing and environment changing. Furthermore, the imperfectness of the hardware production and design of the antenna usually cause the irregularity of the radio propagation in different directions. These factors may lead to link quality fluctuations, and may subsequently affect the performance of the protocols. In addition, when the mobile nodes are included in the network, other factors may dominantly affect the link quality, such as the body-effect and the angle of the antenna. In the last decades, a large number of studies have been conducted in order to investigate and model the characteristics of low power radios, and how they affect the performance of communication protocols. Most of these studies only focus on static scenarios. There has been less effort on studying the mobility impact on link quality and what factors should be taken into account when designing mobility-aware protocols. To better understand the challenges and to provide guidelines for designing mobility management protocol, we conduct a large number of experiments in different environments and settings.

ADAPTIVE BURST TRANSMISSION SCHEME FOR DYNAMIC ENVIRONMENT – CHAPTER 5

Understanding fluctuations of link quality in a wireless sensor network is useful for different reasons. For example, nodes can determine when and for how long they should transmit packets, so that they can reduce packet loss rate and the cost of retransmission (delay as well as power consumption). However, because the quality of a link depends on many factors, it cannot be accurately known except in a probabilistic sense. In this chapter, we estimate the expected duration in which the quality of a specific link remains stable using the conditional distribution function of the signal-to-noise ratio (SNR) of received acknowledgment packets. The expected duration is employed to determine how long nodes should transmit packets in burst and how long they should refrain from contention. To develop our model, Imote2 sensor platforms are deployed in indoor and outdoor environments. More than 70,000 packets are transmitted. Then additional 16,900 packets are transmitted to test our model. In 90% of the time, this approach results in high packet delivery in comparison to the case in which packets are transmitted without the knowledge of link quality fluctuations

A HANDOVER TRIGGERING ALGORITHM FOR MANAGING MOBILITY IN WSNS – CHAPTER 6

One of the reasons why a handover triggering threshold is required is that a handover entails a seamless neighbor discovery phase wherein a transmitting node searches for an alternative relay node. During this phase, it has to transmit packets in a multicast or broadcast mode, which is inefficient. Almost all proposed handover triggering algorithms rely on an empirically obtained RSSI threshold or the failure of a single packet is sufficient to trigger a handover. An empirical threshold is highly environment dependent. In this work, we propose a handover triggering algorithm which does not rely on a predefined threshold. Instead, it equates the cost of packet retransmission (which can be expressed in terms of packet delivery latency or energy) with the cost of a handover. If the former is higher than the latter, a handover is triggered. To compute these costs, the approach proposed in this work establishes the statistics of received acknowledgment packets and employs a Kalman filter to characterize and predict link quality fluctuation.

MSI-MAC – CHAPTER 7

In cellular networks, mobile stations are associated with the nearest base station through intra- and inter-cellular handover. The underlying process is that the quality of an established link is constantly evaluated and handover decisions are fully made by resource rich base stations. In wireless sensor networks, should a seamless handover be carried out, the task has to be accomplished by energy-constraint, resource-limited, and low-power wireless sensor nodes in a distributed manner. In this chapter, a sender initiated mobility management protocol is proposed to enable seamless handover. The protocol is fully implemented in TinyOS environment for the TelosB and Imote2 platforms. Experiment results show that our protocol achieves high reliability and triggers less handover requests (less than 50% to 80%) compared to three state-of-the-arts. Furthermore, our protocol reduces the signaling overhead by up to 95%.

Chapter 3, 5, 6 and 7 are based on following publications:

- Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. Mobilab: A Testbed for Evaluating Mobility Management Protocols in WSN. In International Conference on Testbeds and Research Infrastructures, pages 49-58. Springer, 2016.
- Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. A System Architecture for Managing Complex Experiments in Wireless Sensor Networks. In Proceedings of the 25th IEEE International Conference on Computer Communication and Networks (ICCCN), pages 1-9. IEEE, 2016.
- Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. A Link Quality Estimation Model for Energy-Efficient Wireless Sensor Networks. In Proceedings of the IEEE International Conference on Communications (ICC), pages 6694-6700. IEEE, 2015.
- Jianjun Wen and Waltenegus Dargie. A Handover Triggering Algorithm for Managing Mobility in WSNs. In Proceedings of the 21st International Conference on Information Fusion (FUSION), pages 1-7. IEEE, 2018.
- Jianjun Wen and Waltenegus Dargie. A Mobility Management Protocol for Wireless Sensor Networks. In Proceedings of the IEEE Symposium on Computers and Commu-

nications (ISCC), pages 1-7. IEEE, 2018.

1.4 STRUCTURE OF THE DISSERTATION

The remainder parts of this dissertation are structured as follows: In Chapter 2, we summarize the corresponding related work in recent decades. Chapter 3 presents MobiLab, a mobility enabled testing framework. All of the protocols, algorithms and results presented in this dissertation are verified and evaluated on this testbed. In Chapter 4, we empirically study the link quality characteristics of mobile wireless sensor networks. The observations and conclusions in this chapter are fundamental to the design of other algorithms and protocols in this dissertation. In Chapter 5, an adaptive burst transmission scheme (ABTS) is presented. This scheme is an adaptive approach to tune the burst transmission size which can maintain a high packet success rate, even when the link quality is not perfect. In Chapter 6, we present a Kalman Filter based handover trigger algorithm, namely KMF. In chapter 7, a mobile sender initiated mobility management protocol (MSI-MAC) is proposed. At last, we conclude this dissertation and discuss the future work in Chapter 8.

2 BACKGROUND AND LITERATURE REVIEW

In this chapter we give an overview of relevant state of the arts to this dissertation. First, we review bulk data transfer protocols in wireless sensor networks. Second, we describe the common features of Medium Access Control (MAC) protocols and give a brief discussion about them. Third, we assess mobility aware MAC protocols in detail and make comparison. Finally, we examine the mobility enabled testbeds with which the performance of the proposed protocols were tested.

2.1 BULK DATA TRANSFER IN WIRELESS SENSOR NETWORKS

Devices such as wireless electrocardiograms typically generate data at a rate of tens of kilobits per second. While this in itself may not be high, if other sensors such as 3D accelerometers and gyroscopes have to be sampled at comparatively the same rate, then the aggregate data rate from a single node can be high. This high data rate requirement motivates bulk data transfer protocol. In the early stages, energy efficiency is not the first concern when designing bulk data transfer protocol. They usually disable duty cycling mechanism to achieve high throughput. Additionally, they all assume that links are stable. However, in real deployment, links are not steady, instead, they are dynamic and prone to packet loss. Some researches reported that packet losses are not independent, but are temporally correlated with each other [SKAL08][DÖD11]. In other words, if one packet transmission failed, the probability that subsequent consecutive transmissions also failed is high. This failure periods could span typically up to 500 ms [SDTL10]. Since duty cycling mechanism is not enabled in these protocols, radios keep on all the time even when a large number of consecutive packet losses occur. In this case, network energy efficiency is significantly decreased as the number of retransmissions increases. By taking link dynamics and energy efficiency into consideration, new protocols with duty cycling enabled are proposed. The basic idea is that, if a consecutive packet loss period is detected, nodes can turn off their radios and halt data transmission for a while to avoid unnecessary energy wastage. In the following, we summarize the existing bulk data transfer protocols from two perspectives.

2.1.1 BULK DATA TRANSFER PROTOCOL WITHOUT DUTY CYCLING

Flush [KFD$^+$07] is a sink initiated multi-hop bulk data transfer protocol for wireless sensor networks. The pipelined transmission begins by sending a request from the sink node to a specific source node. Once this request is received at the source node, it begins to transfer bulk data at the maximum rate along a route to the sink node until all packets are sent. After bulk data transmission is finished, the sink node sends selective negative acknowledgments back to the source node, which is the retransmission request for the missing packets. This process is repeated until all the data packets are successfully received by the sink node. In addition, during data transmission, all nodes involved in communication continuously estimate the maximum transmission rate which avoids interference from neighbor nodes by overhearing control packets (request messages) and data traffic. Since Flush uses single channel for pipelining, the throughput is limited (below 1 kB/s).

To solve the bandwidth bottleneck caused by intra-path interference, a TDMA-based, multi-channel bulk data transfer protocol named PIP [RCBG10] is proposed by Raman et al.. In PIP, instead of using beacons for synchronization, it uses simple per-hop synchronization by data packets themselves along the route. The schedule and channels used in transmission and receiving are determined centrally by the sink node. When all these parameters are agreed along the path, the source node starts transferring data packets in burst until all packets are received at the sink node. Since multi-channels are used during data transmission, intra-path interference is significantly eliminated. Thus, high throughput can be achieved up to 7.5 kB/s.

Practical packet pipeline (P^3) [DC14] is a synchronized multi-path, multi-channel, bulk data transfer protocol which exploits both sender and receiver diversities to avoid single path failure. In P^3, data packets are concurrently transmitted by multiple senders to multiple receivers at every time slot. This design is based on the constructive interference. Constructive interference is a phenomenon that when multiple nodes transmit the same data packet with synchronization, the receiver can decode this packet correctly. Furthermore, at each transmission/receiving cycle, a different radio channel is assigned to mitigate intra-path interference like in PIP. By fully utilizing the radio transmission capacity, the throughput can reach up to 22.3 kB/s.

2.1.2 BULK DATA TRANSFER PROTOCOL WITH DUTY CYCLING

As links are dynamic over time, turning on the radio all the time is not energy efficient, especially when a large number of packet failures occur. To eliminate energy wastage, several protocols with awareness of link dynamics are proposed. By studying the short-term variation of wireless links, STLE [ALL$^+$09] is proposed to identify short-term stable periods in a dynamic link. They use the temporary stable links for multi-hop communication. Similarly, Rusak et al. [RL09] investigate the time varying characteristics of wireless channels at physical and link layers. They observe that packet reception rate (PRR) changes over time, suggesting that at different time scales the channels are most appropriately characterized as bursty rather than stable. They apply wavelet transformation on RSSI for analyzing and characterizing the burstiness of the channels. They observe that burst periods repeat themselves and have self-similar nature.

Srinivasan et al. [SKAL08] propose a β metric to measure the burstiness of a link. The β factor is a measure of how close a link is to an ideal link. It is calculated by using the Kantorovich-Wasserstein (KW) distance [CT05], which measures the distance between a conditional probability delivery function (CPDF) of a given link with an ideal link. The CPDF expresses the probability of receiving the next packet successfully after n consecutive successes or failures. The value of β determines the burstiness of the link. A $\beta = 1$ represents a perfect link and $\beta = 0$ represents an uncorrelated link. To explore the performance of the β metric, the authors propose a transmission control scheme which is intended to increase the packet reception ratio by sending packets in bursts until they encounter a failure. When a failure is detected, transmission is halted for 500 ms. The limitation of this approach is that the algorithm requires a large amount of data to predict the success of the next packet.

Munir et al. [MLH$^+$10], propose a scheduling algorithm which produces latency bound for real-time periodic streams for burst links. The authors define the burst period as a period of continuous packet loss and use a metric called Bmax to compute this. They perform an empirical study for 21 days and collect data from different links. For each link they transmit over 3 million packets and record the data trace of success and failure which is used to compute Bmax. The algorithm is used offline. Likewise, Brown et al. [BMR$^+$11] introduce BurstProbe, a mechanism to measure link burstiness online. Probing slots, embedded transmission schedule to access link burstiness online, are shared between neighbors. The probe mechanism is more reactive for capturing burst period due to online probe sharing, but it increases the energy consumption and duty cycle by 2%.

2.2 MEDIUM ACCESS CONTROL PROTOCOLS IN WIRELESS SENSOR NETWORKS

The medium access control (MAC) protocol belongs to the data link layer of the Open System Interconnection (OSI) model. It is responsible for coordinating multiple nodes to access a shared wireless medium. Since transmission is broadcast by nature in wireless communication, the shared medium can only be accessed by one transmitter at a time. Otherwise, interference or collision may happen, which results in packet loss. Hence, a MAC protocol has to define a mechanism to schedule transmissions between nodes to avoid or mitigate collision. According to the scheduling mechanism they used, the existing MAC protocols in WSNs

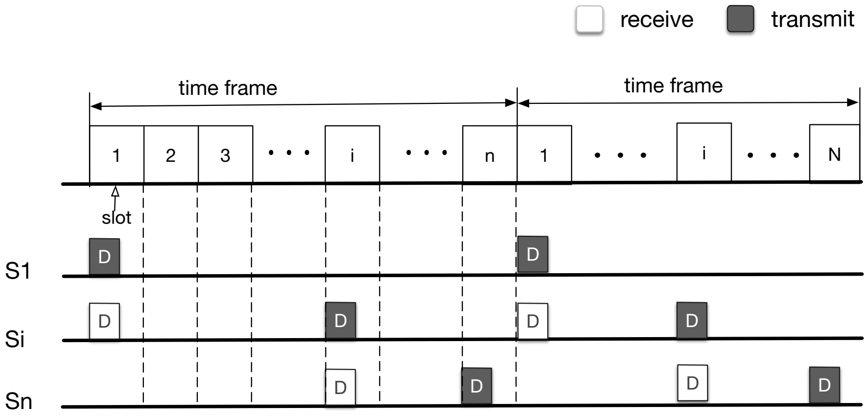


Figure 2.1: An example of TDMA based MAC protocol

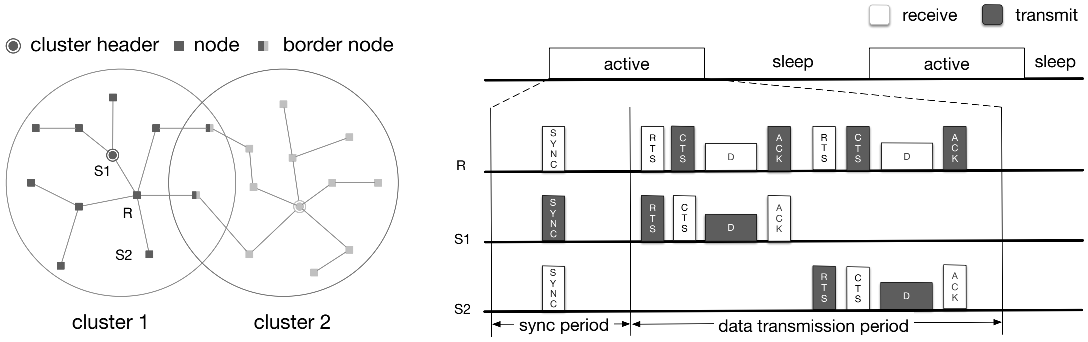
can be classified into three categories [HXS+13]: frame-slotted, synchronous, asynchronous.

2.2.1 FRAME-SLOTTED MAC PROTOCOLS

In frame-slotted MAC protocols, the wireless medium is divided into slots in time manner and each slot is allocated to a specific node. A node can only access the medium in its own slot, thus collisions are completely avoided. For example, in Time Division Multiple Access (TDMA) based protocols, time is divided into slots and each slot is assigned to one and only one node. A node can transmit packets only during its own time slot. Consequently, there is no transmission collision in the network. Figure 2.1 illustrates the basic mechanism of TDMA based protocol. Although TDMA-based protocols can prevent transmission from collision, they have some drawbacks. The time slots are preallocated to nodes, which means they are only applicable for fixed topology. Any change in topology may lead to inefficient medium utilization (i.e. a node leaves the network) or require time slot reassignment (i.e. a new node joins the network). Additionally, the clocks of each node will differ due to clock drift. Thus, periodical time synchronization is required.

2.2.2 SYNCHRONOUS MAC PROTOCOLS

In synchronous MAC protocols, nodes in a network are usually organized in clusters. Nodes within the same cluster are synchronized with each other. They have a common active/sleep schedule. The nodes that have more than one schedules are border nodes which can bridge two or multiple clusters. The cluster topology is shown in Figure 2.2a. The active period is divided into two parts, one for synchronization and the other for data transmission. When a node wakes up, it first listens to the medium for a period of time to receive the schedule from neighboring nodes. If no schedule is received from other nodes, it broadcasts its own schedule. Otherwise it follows the received schedule. After the synchronization period, nodes within the same cluster use request-to-send (RTS) and clear-to-send (CTS) control packets to



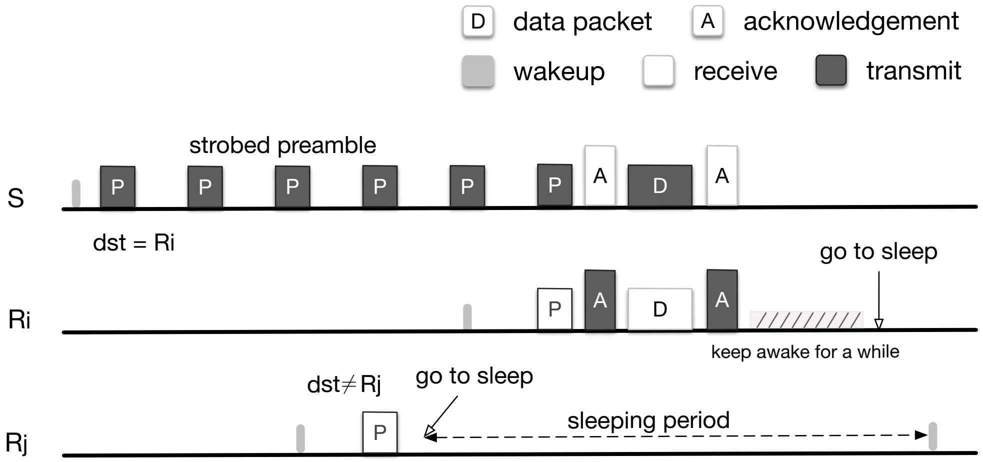
(a) clusters (b) S-MAC

Figure 2.2: An example of asynchronous MAC protocol: S-MAC

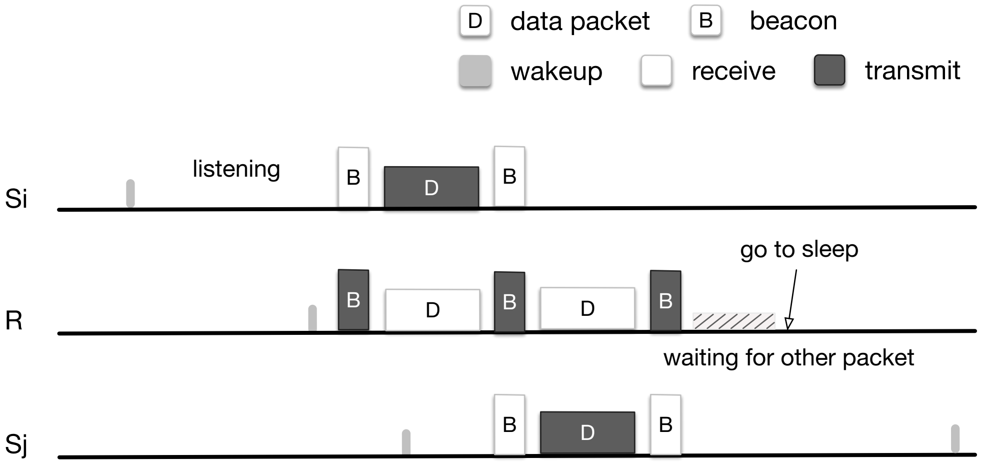
contend the medium. Once the medium is obtained, data exchange begins. Figure 2.2b illustrates the working mechanism of a representative synchronous MAC protocol in WSNs, namely S-MAC[YHE02].

2.2.3 ASYNCHRONOUS MAC PROTOCOLS

Unlike the synchronous protocols in which nodes within the same cluster have a common active/sleep schedule, nodes choose their own schedule independently in asynchronous protocols. The asynchronous schedule mechanism relieves the time synchronization requirement. However, preamble sampling or low power probing prior to data transmission is required. For example, X-MAC [BYAH06] is one of the most representative asynchronous protocols. In X-MAC, nodes periodically switch on and off their radios without knowing the schedules of other nodes. When a node wakes up, it first performs Clear Channel Assessment (CCA) to detect whether the medium is busy or not. If the medium is idle and it has data packets to transmit, it sends a sequence of strobed preambles which are short packets including destination address. Once the intended receiver wakes up and receives the preamble, it replies with an early acknowledgment. Upon receiving the early acknowledgment from the intended receiver, the transmitter sends its data packet subsequently. Other awaken nodes which are not the target receiver can go to sleep immediately after receiving a preamble packet to save energy. Figure 2.3a shows the working mechanism of X-MAC. The preamble sampling based protocol can significantly improve energy efficiency when the traffic load is light. However, the preamble transmission occupies the medium before a connection with the intended receiver is established, which prevents other nodes from transmitting their data packets. To avoid the medium occupation caused by preamble transmission, several receiver-initiated protocols are proposed, which rely on beacon messages to establish the connection, such as RI-MAC [SGJ08], RC-MAC [HWXC10] and A-MAC [DDHC+10]. Figure 2.3b shows the working mechanism of RI-MAC. In RI-MAC, nodes sleep and wake up asynchronously. When a node with data packet to transmit wakes up (S in the figure), it first listens to the medium and



(a) sender-initiated: X-MAC



(b) receiver-initiated: RI-MAC

Figure 2.3: Examples of asynchronous MAC protocols

waits for a beacon from intended receiver. A receiver (R) immediately broadcasts a beacon to announce that it is awake and ready to receive data packet after switching on its radio. Once the beacon is received, the transmitter can send its data packet immediately. Then the receiver replies with another beacon to acknowledge the successful reception of the data packet, and to announce readiness for another receiving. If there is no packet received after beacon transmission, the node will go to sleep.

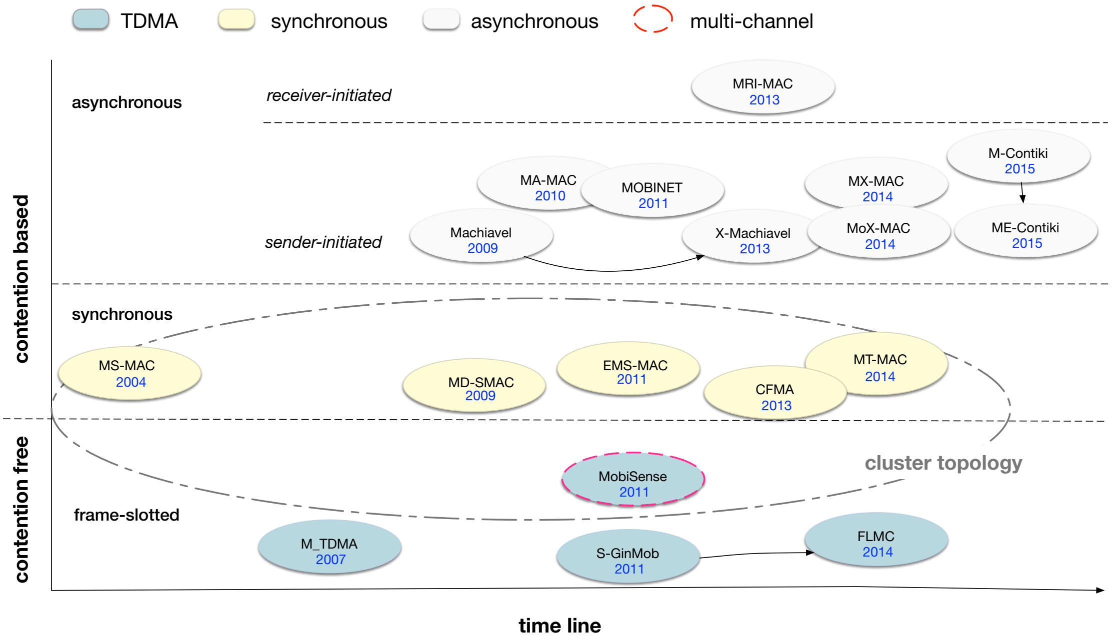


Figure 2.4: Taxonomy of mobility aware MAC protocols

2.3 MOBILITY AWARE MEDIUM ACCESS CONTROL PROTOCOLS IN WIRELESS SENSOR NETWORKS

Wireless sensor networks that support mobile nodes require mobility management protocols. The task of these protocols is to continuously evaluate link quality and to transfer communication to a more reliable link when the link quality or packet loss rate of the current link deteriorates beyond a certain level (this level is defined by the application according to the quality of service requirement, such as the acceptable packet loss rate). In this section, we provide an overview of the existing or proposed mobility aware MAC protocols. A taxonomy of mobility aware MAC protocols is illustrated in Figure.2.4, which follows the same classification as discussed above.

2.3.1 FRAME-SLOTTED PROTOCOLS

M_TDMA [JK07] is an extension of TDMA MAC protocol to support mobility in wireless sensor networks. In M_TDMA, the network is partitioned into several non-overlapping clusters at the start-up phase. The header node is responsible for managing the schedule and for assigning time slots to other nodes within the cluster. A time frame is divided into two parts: the control part and the data part. The control part consists of the first three slots in a time frame, which are used to exchange control messages and to deal with mobility. The header node broadcasts the cluster information in the first slot, such as cluster schedule, node ID and round number. If a new node is present in the cluster due to mobility, and hears the cluster information, it broadcasts its own node ID in the second slot. Upon receiving a new

ID, the head node assigns an unused time slot and sends it to the new node in the third slot. Afterwards, all the nodes in the cluster transmit their data packet subsequently according to the schedule.

S-GinMob [SZSV11] is a cross layer mechanism to support mobility in an industry environment. It extends GinMAC [SBR10], which is a TDMA-based MAC protocol. In S-GinMob, nodes are organized as a tree topology and it is managed by a component named Dynamic Topology Control (DTC). In addition to the topology management, the DTC module is responsible for managing the schedule and allocating different functions to the time slots, such as scanning neighbors, and transmitting and receiving data packets. A mobile node can overhear these control messages and estimate their RSSI in idle time slots to detect new attachments and make handover decisions. If the RSSI value of control messages from a new attachment point is higher than the current attachment node, the mobile node will send a join request to the new attachment point. When the join request is acknowledged, the mobile node will handover to the new attachment point in the next frame.

FLMC [ZCV14] is a variant of S-GinMob. Instead of using the RSSI value to trigger the handover process in S-GinMob, FLMC develops a new handover trigger algorithm based on fuzzy logic. It takes two metrics, RSSI and link loss, as the inputs of the fuzzy inference system. If the output, trigger decision probability, is below a predefined threshold, the handover procedure is initiated.

MobiSense [GLJ11] is a hybrid solution that integrates TDMA and multichannel communication schemes together to handle mobility in wireless sensor networks. In MobiSense, nodes are organized into clusters and each cluster has a header node. The header node is responsible for maintaining topology and for managing schedules within the cluster. Adjacent cluster header nodes operate on different channels, except that they send probe beacons on a common channel during the neighbor discovery phase. Similar to the TDMA protocol, communication in MobiSense is organized into time frames. Each frame is divided into downlink and uplink data transmission slots, admission mini slots, and discovery slots. The first two slots are scheduled slots which are used to exchange data packets. The discovery slots are used to broadcast beacons by header node. A mobile node can overhear these beacons to gather network information and to make handover decisions. For example, if a mobile node detects the RSSI value of packets from the current cluster header is below a predefined threshold, it initiates the handover process by overhearing the discovery beacon. If the RSSI value of the received beacon from a new header is higher than the handover threshold, it sends a join request to the new header in a randomly selected admission slot in the next frame. The handover process is complete if the new header node admits the handover request.

2.3.2 SYNCHRONOUS PROTOCOLS

As discussed in section 2.2.2, the common design principles of synchronous MAC protocols for wireless sensor networks are: 1) nodes in the network are organized into clusters; 2) the communication scheme is partitioned into two phases, one is used for the synchronization of the active time of all nodes within the cluster, and the other is used for exchange of data packets.

When mobile nodes accommodate in the network, the scenarios can be classified into two cases: 1) The mobile node roams within a cluster. It can be treated as a static node because it shares the same schedule as its neighbors. 2) The mobile node approaches the border nodes and attempts to cross the border of the two clusters. Communication may be disabled while crossing the border, because the mobile node may go out of the communication range of the current cluster and has not been yet synchronized with the new one. This disconnection, usually up to 2 minutes, is too long and consequently degrades the performance of the protocol in terms of energy[1] and latency. Hence, the synchronous mobility management protocol is to find a solution to 1) detect the mobile node that is attempting to cross the cluster border and 2) shorten the synchronization time when the handover occurs.

MS-MAC [PJ04] is the first MAC protocol which supports mobility in wireless sensor networks. It handles the handover by following two schedules when the mobile node crosses the border of two clusters. At each periodic synchronization phase, each node learns the mobility status of its neighbors by estimating the RSSI value of the received synchronization messages (SYNC). When the mobile node is crossing the cluster border, an active zone is created within the neighborhood of the mobile node to adapt the synchronization period according to the mobility information. The border nodes broadcast their schedules more frequently than in the stationary scenario, thus the mobile node can establish a connection with the new cluster before it disconnects from the old cluster. However, during the handover phase, the border nodes consume more energy due to frequent synchronization.

Inspired by MS-MAC and DS-MAC [YBS+08], the authors developed a mobility-aware, delay-sensitive MAC protocol named MD-SMAC [HSFG09]. In addition to the basic operations inherited from MS-MAC and DS-MAC, MD-SMAC makes several modifications to enhance mobility management and power consumption efficiency. In the mobility management aspect, the mobile node increases its listening frequency to receive the new schedules when it is crossing the cluster border. In the energy consumption aspect, MD-SMAC only allows the mobile node to listen to the schedules of new clusters, and to keep the other nodes in regular operations. After receiving the new schedule, the mobile node decreases the listening frequency to its default setup (from 30 seconds to 5 minutes.). At the same time, it withdraws the old schedule and adopts the new one immediately.

To deal with high energy consumption in MS-MAC in random movement and high mobility scenarios, the enhanced mobility-aware S-MAC (EMS-MAC) [ZTBW11] is proposed. The author argued that using RSSI alone to detect mobility is not accurate, due to the unstable nature of RSSI measurement. Even in a stationary network, the RSSI value can fluctuate drastically. These RSSI fluctuations can cause false positive mobility events. To reduce unnecessary han-

---

[1] the mobile node has to keep awake to listen to the synchronization messages of the new cluster. The active mode of the radio consumes more energy than the sleeping mode.

dover events, EMS-MAC combines RSSI with link quality indicator (LQI) to predict mobility. Instead of creating an active zone which involves all the border nodes in the neighborhood of the mobile node when the mobile node attempts to cross the border, EMS-MAC introduces a mobility zone which only includes the nodes that detected the mobility of the approaching mobile node. After mobility detection, the border node sends a SYNC message to introduce itself and the mobile node replies to the SYNC packet with the handover request. Then the border node broadcasts the new cluster schedule. Once the mobile node has learned the new schedule, it adopts this schedule as well as the old one to finish the handover process.

CFMA [KA13], collision free mobility adaptive MAC protocol, is another type of synchronous mobility management protocol. Nodes in CFMA are organized in a cluster-tree topology. Each cluster has a head node called coordinator. Other nodes in the cluster are synchronized with the coordinator by periodically transmitted beacons. The coordinators are not only functioned as bridges to the sink node, but also can communicate with each other directly to exchange network information, especially when the mobile node is moving from one cluster to another. Another contribution of CFMA is the idea of allocating predefined backoff value to the nodes which have buffered data to avoid collision at the beginning of each transmission round. The mobile node can overhear these backoff assignment packets. If it detects that the RSSI value of these packets from the neighboring coordinator is continuously increasing, the mobile node requests the current coordinator for a delay value of the neighboring cluster. Once the mobile node acquires the delay value from the new cluster coordinator, it can handover to the new cluster with a specific delay. Furthermore, the mobile nodes always have the highest priority when they enter a new cluster, which ensures that they can join the new cluster with minimum latency.

MT-MAC [ZAI+14] is the latest developed mobility management MAC protocol in the synchronous group, which is based on T-MAC [VDL03]. The mobility handling strategy is almost the same as EMS-MAC. The only difference between these two protocols is that MT-MAC inherits an adaptive duty cycling mechanism from T-MAC and supports burst transmission. Alternatively, EMS-MAC uses a fixed duty cycling strategy.

2.3.3 ASYNCHRONOUS PROTOCOLS

Another large group of mobility management protocols is extension of asynchronous MAC protocols. Unlike synchronous protocols in which synchronization, RTS and CTS are used to access the medium, asynchronous MAC protocols apply different schedules for each node in the network, and use low power listening (LPL) or low power probing (LPP) to establish communication between the sender and the receiver. According to which node, sender or receiver, initiates the communication, asynchronous MAC protocols can be categorized into two groups: sender-initiated and receiver-initiated.

Both sender-initiated and receiver-initiated MAC protocols work well in static networks, however, inherently they are not suitable for mobile wireless sensor networks. As the mobile node is not fixed to one place all the time, it may lose the current neighboring information before its next wakeup. The unawareness of the neighbor nodes makes the mobile node a formidable challenge to establish a connection with the intended receiver, especially for sender-initiated protocols. For receiver-initiated MAC protocols, although the mobile node can get new receiver information by overhearing beacon messages, the handover latency could be high [DD12]. To support mobility and to improve performance in terms of latency and power consumption in asynchronous MAC protocols, several efforts have been made in the WSN research society. Most of the proposed protocols extend low power listening based protocols, such as X-MAC [BYAH06] and Contiki-MAC [Dun11].

Inspired by B-MAC [PHC04], a mobility enabled MAC protocol named Machiavel [KN09] has been proposed to address the synchronization problem and mobility issues in dense wireless sensor networks. In the design of Machiavel, mobile nodes can preempt the data transmission of the current static transmitter and proceed its own data transmission thereafter. In more details, when a mobile node wakes up, it first samples the medium to detect if there is any preamble in progress. If no signal is detected, the mobile node then follows the standard procedure of B-MAC, such as transmitting a long preamble to synchronize with the intended receiver. Otherwise, the mobile node waits until the end of the synchronization process (preamble and SYNC message scheme), and then takes possession of the medium owned by the static transmitter which sent the preamble previously. To this end, the static node should delay for a specific time prior to its data transmission to allow the mobile node to take charge of the medium. This delay value is called MIFS (Machiavel inter-frame space), and is defined according to the time that a node should take to sample the medium.

By taking advantage of the strobed preamble, the author of Machiavel made some optimization and proposed an enhanced version of Machiavel, called X-Machiavel [KMN13]. X-Machiavel introduces several new features to cope with the rendezvous and contention issues. Instead of using only one type of preamble in the static network, X-Machiavel introduces another two types of preamble packet, which are used for the mobile node and the forwarder node to proceed data packet originated from the mobile source. Accordingly, another type of early acknowledgment packet is introduced to answer the two new preambles which indicates that the static node can take care of the data for the mobile node. The initiation of data transmission for the mobile node can fall into two cases. Like in Machiavel, if the mobile node detected the medium is free after waking up, it transmits the first type of preamble (specific to the mobile node) to discover the active receiver nearby. Any active static nodes in the vicinity can intercept this preamble and reply with early acknowledgment, regardless of whether they are the intended target or not. Then the data transmission follows the principle of X-MAC. If the static node receives the data packet from a mobile node that is not intended for it, it forwards this packet to the final destination via a routing algorithm before its own data packet[2]. In the other case, if the medium is occupied by one static node and its preamble is received by the mobile node, the mobile sender entrusts its data packet immediately to this

---

[2] the author assumes that the data packet in a mobile node has higher priority than that in other nodes

static node by taking possession of the medium.

MA-MAC [ZD10] is an extension of X-MAC which is optimized to support mobility in dense wireless sensor networks. It assumes a human pedestrian mobility pattern which is known prior. According to the subsequent RSSI values, the distance between the two nodes can be easily estimated via a modified theoretical path loss model. The authors argue that although the distance estimation is simple and inaccurate, it is sufficient to handle mobility. To initiate the handover process, two distance thresholds are predefined. The first threshold indicates the starting point where the mobile node should initiate the neighbor discovery process and the second represents the ending point where the handover process must be finished before it. After the link is established, the mobile node unicasts data packets in burst similarly to X-MAC, while it keeps learning the link quality in terms of RSSI of the ACK packet and estimates the relative distance. Once it detected the distance is below the first threshold, it initiates the neighbor discovery process by setting the handover request bit in the MAC header and turning to broadcast mode. If the other neighboring static receivers wake up and receive the handover request, they reply with a control packet to notify the mobile node of their availability. The mobile node will choose the next receiver on the first-come, first-serve principle. In the MA-MAC design, aside from the original MAC header defined in the IEEE-802.15.4 standard, an additional handover header which includes the handover request and redundant address information is defined and resides in the payload section.

Discovering neighboring receivers by preambles is both time and energy consuming. By taking advantage of LPL, MOBINET [RMN11] allows the mobile node to construct and maintain a neighborhood table by overhearing the ongoing traffic (e.g. preamble and data packets). Each time mobile node wakes up, instead of sampling the medium and transmitting preambles, it keeps listening to the communication channel until any messages are intercepted. After learning neighbor information from the header of the intercepted messages, it stores the neighbor node's address with a time-to-live (TTL) attribute in the neighborhood table. Any record in the table of which the TTL has expired will be deleted, and a new overhearing process will be initiated to update the table. If the mobile node has pending data packets to transmit, it will select one receiver from the table and proceed its data packets as in X-MAC. For the neighbor selection, MOBINET proposed two strategies, random and selective. The random strategy is adopted when the mobile node has less neighborhood information, for instance, when it has recently joined the network. When the mobile node has known its neighborhood for a long time, it then can select the best receiver depending on the different application criteria (e.g. routing metric).

Similarly, mobile nodes in MoX-MAC [BNG14] keep overhearing the preamble and early acknowledgment to learn the neighborhood information once they wake up ,instead of using data packets that are embedded with the handover request to discover neighbors. After learning of the intended receiver with the early ACK, the radio of the mobile node remains switched on and backs off for some time to let the ongoing data transmission finish. Then it delivers its data packet to the previous static transmitter. To this end, the static transmitter should remain awake for a while after each data transmission to receive data packets from any mobile nodes.

MX-MAC [DW14] is another extension of X-MAC which supports seamless handover in mobile wireless sensor networks. In this protocol, it assumes that the mobile node has a large number of packets to transmit in burst and most of the time, the traffic is one way from the mobile node to the sink node. The address space of the network is separated into two groups (mobile and static) by reserving the second most significant bit[3]. The process of data transmission is divided into five phases according to the finite state machine running in the mobile node. They are the initial phase, the network joining phase, the normal data transmission phase, the neighbor discovery phase and the handover phase.

When a mobile node awakes, due to the unawareness of neighboring nodes, it first performs Clear Channel Assessments (CCA) to check the medium is free or not. If the medium is free, it then continuously multicasts the first data packet in the queue as the preamble which includes the joining network request. If any receivers nearby hear this request, they reply to the mobile node with acknowledgments embedded with their address information. Only the receiver which replies with the acknowledgment first becomes the relay node. Upon knowing the receiver address, the mobile node continuously unicasts the remaining packets to the current receiver and estimates the link quality simultaneously by the corresponding acknowledgments. Unlike other protocols, the link quality estimation in MX-MAC does not access each single RSSI value directly. Instead, it processes them in a time slot manner. After the transmission of w packets (a predefined window size), it calculates the link quality metric scored from 1 to 5 according to a probabilistic method. Next, this link quality metric will be input to the Least-Mean-Square filter (LMS) to predict the link quality in the future. By combining the current link quality with the predicted one, a threshold-based handover triggering criterion is proposed. If the combined link quality metric is continuously below the triggering level for some time slots, the mobile node will initiate a handover process by embedding the handover request to the following data packets and switching its transmission mode from unicast to multicast. Similarly to MA-MAC, MX-MAC can keep connection with the current receiver while discovering the new potential relay nodes. After each handover request transmission, it is not only waiting for acknowledgment from the current receiver, but also backs off for an additional time to wait for the response from other receivers nearby. Any active receivers in the vicinity of the mobile node which heard the handover request can reply with a special acknowledgment to show their willingness to be the next receiver. In case of multiple receivers replying to the same handover request which may collide with each other, each potential receiver will perform a random backoff prior to responding to the handover request. Upon receiving the feedback from a new potential receiver, the mobile node will return to the unicast mode and handover to the new receiver immediately.

To reduce the latency and energy consumption caused by synchronization before data transmission, ContikiMAC [Dun11] uses data packets instead of strobed preambles to synchronize

---

[3] '1' represents the static receiver and '0' is the mobile sender.

the transmitter and the receiver. Additionally, a transmitter can learn the schedule of a receiver, and wakes up a little earlier than the receiver. This is known as a phase-lock mechanism. Additionally, ContikiMAC is optimized to support burst transmission by embedding a pending flag in the header. By taking advantage of ContikiMAC, M-ContikiMAC [PGN+14] is proposed to support mobility under bursty traffic. In M-ContikiMAC, anycast transmission is used to synchronize the transmitter and the receiver. When a mobile sender wakes up, it anycasts the first data packet repeatedly to discover any potential receivers in its neighborhood. Any static receivers in the communication range of the mobile transmitter can reply with an early acknowledgment upon receiving the anycasted data packet. Once an acknowledgment packet is received, the mobile sender transmits the remaining packets in burst to the receiver which replies with the acknowledgment. During burst transmission, if a packet failure occurs, the mobile sender will go to sleep and search for another receiver during the next wakeup. However, M-ContikiMAC has packet duplication problem because multiple awaken receivers can manipulate the same anycasted data packets. To overcome the duplication problem, ME-ContikiMAC [PKG+15] is proposed. Instead of anycasting data packets to discover neighboring node, in ME-ContikiMAC, a special control packet is introduced.

MRI-MAC [Don13] is a receiver-initiated MAC protocol which supports the mobile node to transmit data packets in burst. At the initial phase, it works in the same way as RI-MAC. The author assumes that the mobile node is the primary source of transmission and has a series of packets (assume n packets) to transmit at each time when it wakes up. After establishing a link with a specific receiver, the mobile node transmits the first w packets to estimate the relative distance between itself and the current receiver. However, the distance estimation algorithm is not given in their design and they only assume that there is a one-to-one map between the RSSI and the relative distance. If the mobile node detects that the relative distance is less than a predefined threshold which indicates that the remaining packets cannot be finished before the link breaks, the mobile node will initiate a handover process immediately by embedding a handover request in the following data packets. In addition to the handover request, a waiting period in terms of time slot is also embedded in the data packets in case there are multiple candidate receivers responding to the request at the same time. These data packets with the handover request are broadcasted to all receivers nearby. The current receiver processes these packets as normal data packets and simply replies with a beacon as an acknowledgment, while the other active receivers which satisfy the criterion[4] randomly select a time slot according to the waiting period embedded in the handover request to respond with a special beacon as a handover reply. The receiver which chooses the earliest time slot will become the winner. By finding the new receiver with the handover reply, the mobile node switches its next data transmission from broadcast to unicast mode and transmits the remaining packets to the new receiver.

---

[4] The relative distance is long enough to receive the remaining packets

2.3.4 COMPARISON

Table 2.1 gives a comparison summary of the existing mobility aware MAC protocols. They are compared from four perspectives: handover trigger method, neighbor discovery and selection, burst transmission and evaluation method.

HANDOVER TRIGGER METHOD

As far as handover triggering methods are concerned, different protocols use different approaches, the simplest metric being evaluating the RSSI values of the received packets, like in MobiSense, MX-MAC, MA-MAC and S-GinMob. They rely on an empirically obtained RSSI threshold to trigger a handover. An empirical threshold is highly environment dependent. For example, for the CC2420 radio which implements the IEEE 802.15.4 standard, different values are identified (−75 dBm in [GLJ11], −87 dBm in [Lev06] or −80 dBm in [SKAL08]) to achieve a 90 % packet delivery rate. Thus, it has to be calibrated before deployment, otherwise it may degrade the performance. In MX-MAC, it focuses on estimating the actual RSSI fluctuation using a LMS filter. By carefully studying the characteristics of link quality fluctuation in an industrial environment, FLMC employs a fuzzy logic to estimate the link quality. This approach takes the RSSI values of incoming packets and packet loss rate as its inputs. The output of the algorithm is a trigger decision probability, which, if it falls below a predefined threshold, is used to initiate a handover. M-ContikiMAC and ME-ContikiMAC assume that a single failed or lost packet is sufficient to trigger a handover. In MRI-MAC, the authors assume that there is a one-to-one mapping between the RSSI values and the distance separating the transmitter and the receiver. The mobile node first transmits n packets to estimate its relative distance from the receiver. If the relative distance is beyond the predefined threshold (which implies that subsequent packets may not be transmitted successfully), it triggers a handover immediately. All the synchronous mobility aware protocols use the mobility detection method to trigger the handover process. They continuously estimate the RSSI values of SYNC packets from the neighboring node. If they detect that the RSSI values are constantly changing (increasing or decreasing), they assume the node is mobile. If a node labeled as mobile presents in the communication range of border nodes, the border nodes adapt their synchronization frequency to expedite the neighbor discovery process. In EMS-MAC, the author argued that using RSSI value alone to detect mobility is not accurate, because of the unstable nature of RSSI measurement. Hence, they combine RSSI and LQI together to estimate the node's mobility status.

NEIGHBOR DISCOVERY AND SELECTION

According to the strategy that mobile node used to discover neighbors, neighbor discovery process can be classified as active or passive. In the active method, the mobile node actively broadcasts or anycasts data packets or control packets to discover neighboring nodes. In the passive method, the mobile node learns neighbor information by listening to periodically transmitted control packets from the neighbor node. Since in TDMA-based protocols periodical synchronization is a must, the mobile node can listen to synchronization messages to discover neighbor nodes. Thus, neighbor discovery in TDMA-based protocols is passive. The link with the current receiver may break as the mobile node can move out of the communication range before receiving synchronization messages from the neighboring node. Similarly, in synchronous protocols, the mobile node can listen to the SYNC messages from the border node to discover neighbors, thus they are passive. In contrast, in asynchronous protocols, there is no synchronization, thus the mobile node discovers neighboring nodes by actively transmitting control packets or data packets. MOBINET and MoX-MAC are two exceptions. In the former protocol, the mobile node overhears ongoing traffic to construct a neighbor table. In the latter, the mobile node overhears preambles and early acknowledgment to learn the neighborhood information. Among the protocols, only Machiavel, X-Machiavel, MX-MAC and MRI-MAC do not employ signaling packets to discover neighbor nodes.

BURST TRANSMISSION

Only MT-MAC, MA-MAC, MX-MAC, MRI-MAC, M-ContikiMAC and ME-ContikiMAC support burst transmission.

EVALUATION METHOD

As we observed, 67% of the proposed protocols are evaluated by simulation and only 28% are evaluated in real deployment. In M_TDMA, the authors only propose a communication scheme and do not provide any method to evaluate performance. The most popular simulator that is used to evaluate the protocol performance is Cooja [ODE+06], which is a java based Contiki simulator. M-ContikiMAC, ME-ContikiMAC and MobiSense are evaluated with it. The second most popular simulators are WSNet and NS2. Three MAC protocols, MS-MAC, MD-S-MAC and MRI-MAC, choose NS2 as the simulator, because the first two protocols are based on S-MAC which is implemented in NS2 and MRI-MAC is based on RI-MAC which also has NS2 implementation. In MOBINET, Machiavel and X-Machiavel, the WSNet simulator is chosen as simulation tool. In EMS-MAC and MT-MAC, Castalia is used to verify the design, which is an open source simulation tool developed over OMNET++. CFMA is the only one which uses Matlab as the simulation tool. The other five protocols, S-GinMob, FLMC, MA-MAC, MX-MAC and MobiSense, are tested in real deployment.

Table 2.1: Summary of mobility aware MAC protocols

protocol	handover trigger	neighbor discovery				make before break	burst	evaluation
		active	method	signaling	decision			
M_TDMA	-	x	listening	✓	-	x	x	no
S-GinMob	link quality	x	listening	✓	link quality	✓	x	testbed
FLMC	link quality	x	listening	✓	link quality & hop	✓	x	testbed
MobiSense	link quality	x	listening	✓	link quality cluster size	✓	x	testbed & simulation
MS-MAC	mobility detection	x	listening	✓	-	x	x	simulation
MD-SMAC	mobility detection	x	listening	✓	-	x	x	simulation
EMS-MAC	mobility detection	x	listening	✓	-	x	x	simulation
CFMA	mobility detection	x	broadcast/unicast	✓	-	x	x	simulation
MT-MAC	mobility detection	x	broadcast	✓	-	✓	✓	simulation
MA-MAC	link quality	✓	broadcast	✓	first serve	✓	✓	testbed
MOBINET		x	overhearing	-	random/selective	x	x	simulation
Machiavel X-Machiavel	-	hybrid	broadcast overhearing	x	first serve	x	x	simulation
MX-MAC	link quality	✓	multicast	x	link quality	✓	✓	testbed
MoX-MAC	-	x	overhearing	✓	-	x	x	simulation
M-ContikiMAC ME-ContikiMAC	packet loss	✓	anycast	✓	first serve	x	✓	simulation
MRI-MAC	distance estimation	✓	broadcast	x	distance	✓	✓	simulation

Notes: "-": unknown "✓": yes "x": no

2.4 EVALUATION METHODOLOGY FOR MOBILITY MANAGEMENT PROTOCOLS IN WIRELESS SENSOR NETWORKS

Testbeds are intended to efficiently test wireless sensor networks before actual deployments. Compared to the area or volume an actual deployment occupies, testbeds are considerably compact, so that they can be installed in labs or in areas which are easily accessible. This means, some communication parameters are intentionally scaled and events can be deliberately injected into the network to suit the test setting and to emulate actual events.

There are several testbeds, most of which are available for public use. Some of these are TWIST [HKWW06], WISEBED [CFK⁺10], MoteLab [WASW05], and TempLab [BZB⁺14]. These testbeds share similar design principles. As far as hardware is concerned they provide additional wired or wireless interface (USB, Ethernet or Wi-Fi) as backbone channels for stable programming, controlling and data logging. As far as software is concerned, they provide (a) web-based interfaces to remotely access the testbeds and to manage experiments; and (b) mechanisms to automatically program, configure, and run the testbeds according to specific requirements. Rakotoarivelo et al. [ROJS10] meaningfully separate the software services into three logical services: control, management, and measurement.

2.4.1 EXPERIMENT MANAGEMENT METHODS

The experiment services of existing or proposed testbeds can be classified into two broad categories, inter-experiment and intra-experiment management services.

INTER-EXPERIMENT MANAGEMENT

Almost all publicly available testbeds provide inter-experiment management services [EAR⁺06]. These testbeds provide web-interfaces to enable users to install their own program images on the testbeds and to specify experiment procedures remotely. Combined with different scheduling polices (e.g., priority-based [WASW05] or microeconomic processing [CBA⁺05]), physical resources can be reserved and experiments can be conducted automatically. During the experiment execution, the data collection service actively gathers in the background application-, protocol-, or network-specific data and stores them in a local or remote database. After the execution of the experiments, users can download the data and perform off-line analysis.

INTRA-EXPERIMENT MANAGEMENT

WISEBED [CFK⁺10] is the first collective effort of nine European universities to build a heterogeneous wireless sensor network testbed. It provides a group of command line scripts to help users to manage their experiments. Users can send arbitrary binary messages to individual nodes at runtime via a web interface or a script to interact with the experiment. This feature is useful and flexible both to retrieve and modify the state of execution, nevertheless, requires elaborate design and specification of experiment procedures (users are required to define and implement their own experiment control protocol).

In contrast, FlockLab [LFZ+13] uses a hardware input/output mechanism (GPIO) to interrupt and control the experiment execution at a node level (which is more efficient than software-based interruption). However, it requires a dedicated hardware platform with an interface board. RadiaLE [BKJ+11] is an application specific framework which aims to facilitate the design and implementation of link quality estimators (LQE) in wireless sensor networks. It consists of one control station (a PC) and 49 TelosB nodes which are connected via USB cables and hubs to form a radial topology. The control station has the ability to configure the network parameters and to initiate data transmission by sending commands to specific nodes, according to the desired traffic pattern. The control method, however, is basic and application-dependent (can be used for studying LQE only).

Minerva [SK13] is a distributed debugging testbed and provides python script interfaces to reset, halt and resume the execution of nodes which can be used for intra-experiment management. But, the testbed requires a special hardware support (a debugging board connected to a sensor node via JTAG interface) limiting its usefulness to carry out complex experiments in different testbeds.

2.4.2 MOBILITY ENABLED TESTBED

Emulab [FFL06] is perhaps the first publicly reachable mobility-enabled testbed for WSNs experimentation. The testbed is deployed in an L-shaped area and consists of (1) 25 Mica2 static nodes installed on the walls and ceiling of a building to form a grid-like topology, (2) 6 mobile nodes attached to robotic platforms, which can perform user-specific and accurate way-point walking models (according to the authors, the position of the robots can be determined within 1 cm error, the worst-case), (3) 6 cameras which are installed on the ceiling to track the robots, and (4) additional 3 web-cams to provide live-monitoring. One of the limitations of the testbed is the difficulty of influencing the movements of the robots during experiment execution, because their movement pattern is predetermined and is not accessible at runtime.

Kansei [EAR+06] is a testbed employing the same types of robots like Emulab to support mobility, but it does not provide any positioning system. The testbed uses five robots integrating TMote Sky nodes and Extreme Scale Motes (XSM). These robots are deployed on top of a Plexiglas plane in which 210 XSMs and TMote Sky nodes are arranged in a 15 × 14 grid bench-work. In addition to the common functionalities the previous testbed provides, Kansei provides a mechanism to inject events into individual nodes and gateways.

Sensei-UU [RHLG10] employs a Lego NXT robot as the mobile platform, on which a TelosB node and a smartphone are attached. Its unique feature is employing WL-500GP wireless access point as a control station to provide programming, experiment monitoring, and data logging functionality via a wireless channel. While it is relatively easy to reproduce and repeat

Table 2.2: Summary of mobility enabled testbeds

Testbed	mobile platform	relocatable	characteristics
Emulab	6 robots	no	- fixed deployment - camera for localization
Kansei	5 robots	no	- walking on top of Plexiglas plane - no positioning
Sensei-UU	1 Lego NXT robot	yes	- relocatable - tape based walking
SensLab	locomotive	no	- accurate motion control and positioning - no random walk model
TrainSense	locomotive	no	

experiments with this testbed, it has some drawbacks: (1) the robot requires the installation of tapes on the floor, which limits the types of movement that can be imitated by the mobile platform (i.e., undertaking different random movements is difficult); and 2) it is difficult to support multiple mobile nodes at the same time.

SensLAB [DRCF+11] and TrainSense [SSZ+13] are two recently proposed testbeds for mobile platforms. Both utilize toy trains as mobile platforms. Since the trains run on tracks, which physically limit their motion, the testbeds are difficult to extend. It is also difficult to introduce random walks into experiments. One of the merits of these testbeds is their ability to provide better accuracy of localization and control of mobility compared with the other mobile platforms. Table 2.2 shows summary of mobility enabled testbed we discussed above.

3 MOBILAB: MOBILITY ENABLED TESTING FRAMEWORK

In order to obtain predictable performance and reproducible results from wireless sensor networks, complex and repeated experiments should be conducted with testbeds before actual deployments take place. Since most applications have their unique characteristics and requirements, the testbeds should be flexible and effectively separate the experiment phase from the application development and network management phases. In the past decades, the research community has made a considerable progress in developing reliable and flexible testbeds. Some of these provide web interfaces so that application developers can install program images on remotely available networks and execute code. Then experiment results (sensed as well as performance related data) can be extracted from the networks and delivered to the developers for offline analysis and debugging. Some of these testbeds, besides providing common services, such as infrastructure management, experiment control (experiment scheduling and resource reservation), and data collection, enable also the inclusion of domain specific services, such as sensor data profiling [BZB⁺14], mobility management [EAR⁺06], and distributed and online tracing/debugging [SK13].

In most testbeds experiment procedures are embedded into the application logic and, hence, intra-experiment activities (such as an activation or deactivation of the collision avoidance functionality of a MAC protocol or the modification of network parameters) have to be carefully planned before a program is compiled and flashed to individual nodes. Arbitrary configurations cannot be carried out without affecting the execution of the application logic. Furthermore, the specification of complex procedures comes at the price of developing complex application layer services. If application developers wish to introduce new procedures unforeseen at the time of uploading their image the only option they have is modifying their image and reinstalling it, which is a tedious and time consuming process. Furthermore, embedding experiment procedures in the application logic have another side effect, namely, experiment execution times will be subject to timers' error due to drift. To alleviate this problem time synchronization should be necessarily a part of the application logic. Otherwise, experiments may not be reproducible. Finally, most existing testbeds provide experiment data management at the server side but not for individual nodes. There are no common interfaces or library files available for application developers to seamlessly gather data and performance indicators from individual nodes.

In this chapter we propose a testbed for evaluating the effect of mobility in wireless sensor networks, namely MobiLab. The testbed separates the concern of application development from the evaluation of the application in different mobile scenarios. By doing so, complex and reproducible experiments can be carried out to ensure that the behaviors of applications are both reproducible and predictable. We fully implemented the testbed for the TinyOS and TelosB platforms. Our mobile nodes are carried by Diddyborg robots [did], each of which is controlled by 6 powerful gear motors, so that the robots can be tasked to emulate different types of movements in indoor as well as outdoor environments.

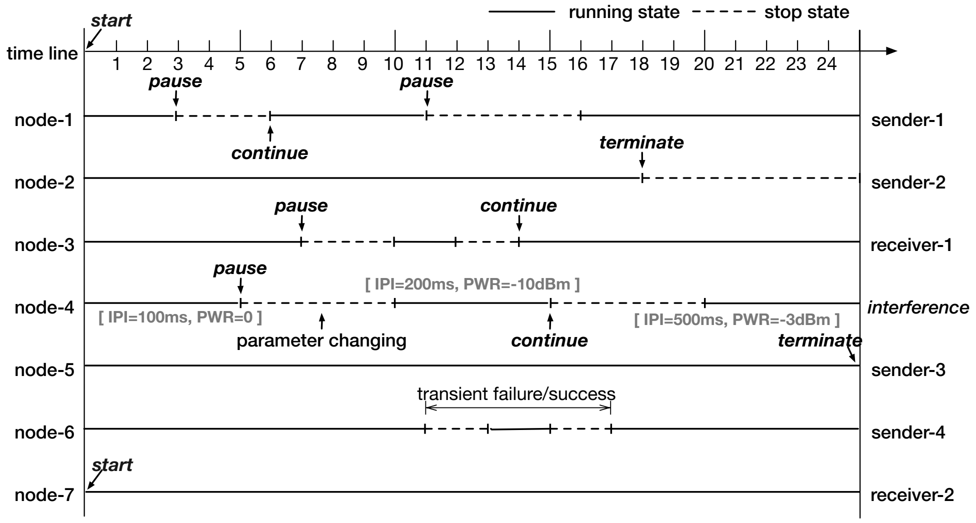
Furthermore, a comprehensive traffic flow control protocol is also proposed, which integrates a set of toolkits for seamlessly performing changes to and configure protocols and algorithms during experiments. The contributions of this chapter can be summarized as follows:

- We define a set of primitives to control experiments as they are being conducted. These primitives provide simple and enhanced controlling strategies.
- We propose a light-weight protocol to communicate commands that control experiments at runtime.
- We propose system architecture to integrate, process, and manage the traffic flow control commands.
- We design and implement the hardware and software architecture of the testbed.

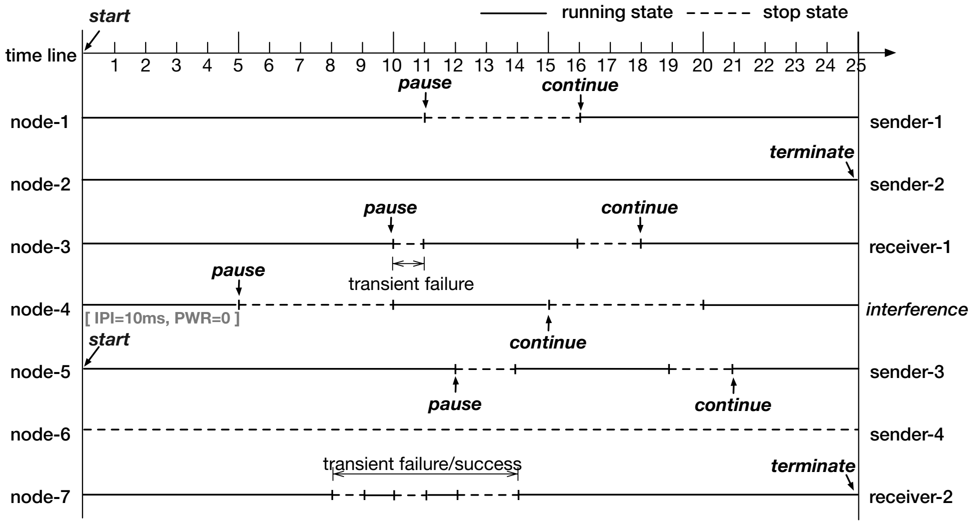
3.1 CHALLENGES AND REQUIREMENTS

In order to illustrate the difficulty of conducting moderately complex experiments with existing testbeds, we provide an example using seven wireless sensor nodes (Figure 3.1). The experiment is intended to investigate the effect of random interference on the reliability of the network (measured in terms of overall packet loss). The experiment schedules specify the beginning and end of transmission times; inter-packet intervals (IPI) with which packets should be transmitted; communications types (unicast or broadcast); and a transient failure of nodes. The schedules also define the transmission patterns of nodes (when and for how long they should transmit).

In the first schedule, node 1 and 2 communicate with node 3; node 4 broadcasts to all nodes; and node 5 and 6 communicate with node 7. The Table in Figure 3.1c displays the configuration of some of the physical-layer parameters for schedule-1. In the second schedule, node 6 is entirely terminated and transient failure is introduced to node 7 (which is a receiver). The Table in Figure 3.1d displays the configuration of parameters for schedule-2.



(a) schedule-1



(b) schedule-2

test case 1				
node	channel	tx-power (dBm)	IPI (ms)	receiver
node-1	26	0	20	node-3
node-2	26	0	50	node-3
node-3	26	0	100	-
node-4	26	0	100	broadcast
node-5	26	-10	100	node-7
node-6	26	-10	200	node-7
node-7	26	-10	250	-

(c) parameters for test case 1

test case 2				
node	channel	tx-power (dBm)	IPI (ms)	receiver
node-1	24	-10	20	node-3
node-2	24	-10	50	node-3
node-3	24	0	100	-
node-4	24	0	10	broadcast
node-5	24	-3	100	node-7
node-6	-	-	-	-
node-7	24	-3	250	-

(d) parameters for test case 2

Figure 3.1: An example experiment: (a) and (b) Scheduling the transmission time and duration of nodes – The activity state of each node is represented by a solid line whereas the inactivity state is represented by a dashed line. (c) and (d) Fixing protocol parameters for two test cases.

The simplest way to perform the above experiments is to embed the experiment flow (schedules) into the application logic using timers to control the experiment at runtime. This, however, introduces some challenges. Firstly, a time synchronization protocol has to be implemented at the application layer to synchronize the timers of all nodes. Otherwise the discrepancy in time drift in each node may lead to incongruity of schedule execution. Secondly, the integration of the time synchronization protocol at the application layer violates the principle of separation of concern, because the application developer is concerned not only with the development of the application logic but also with network management. Thirdly, suppose our initial plan was to run schedule-1 only but after having observed the experiment results, we decided to modify the first schedule to produce the second and rerun the experiment. In this case, the application logic has to be modified in the source code, recompiled, and flashed to all the nodes. Reprogramming nodes not only is time consuming but also decreases the lifetime of the hardware (the number of erase/reprogram cycles is limited in most existing flash memories; for example, for the MSP430 MCU (used in the TelosB platform), the operation is limited to 10,000 times [Tex06]). Additionally, the whole network needs to be reset manually for each round of the experiments resulting in unnecessary user intervention. In order to address these challenges, we propose a traffic flow control framework having the following features.

Integrability. Our framework is easily integrable with testbeds or application-dependent infrastructures. Researchers can use the framework to build and control their own experiment with little or no modification to their testbeds. Experiments can rerun multiple times to extract reproducible results without any manual involvement.

Scalability. As different infrastructures contain different number of nodes (from tens up to hundreds) [BKJ+11] [HKWW06] [DRCD+11], and different experiments require different combinations of nodes, our framework is both scalable and adaptive.

Reconfigurability. Most experiments are performed multiple times, not only under the same configuration but also with different parameter settings, to test the effect of different configurations on performance, network lifetime, and energy consumption. Some of the parameters that should be adjusted at runtime are (1) the duty cycle of MAC protocols [BYAH06] [DW14], (2) communication channels and transmission power levels to study link quality fluctuations [SDTL10] [WAD15]; and (3) entries in routing tables. Thus, the framework should enable the configuration of these parameters without the need to reprogram the network. Additionally, in some experiments, event injections such as mimicking temporary node failures is useful to support.

Automatic execution. In existing testbeds, experiments automatically begin as soon as the nodes are active. Manual intervention is required to stop and restart experiments. In some cases advanced devices are used to remotely control experiments, but this makes experimentation unnecessarily expensive. Our framework employs software-controlled mechanism to manage experiments automatically.

Seamless data collection. One important and imperative process during experiment execution is extracting data and performance indicator metrics (such as RSSI, SNR, and timestamps) from the network. This process, however, should not interfere with the normal operation of the network (for example, by taking away precious bandwidth or communication time). Therefore, the framework should be able to seamlessly gather these data, temporarily store them locally, and enable the efficient collection at a most convenient time for online as well as offline analysis and debugging.

3.2 TRAFFIC FLOW CONTROL PROTOCOL

In order to cleanly separate experiment execution from experiment management, we propose a server-client architecture. The server specifies experiment schedules and dispatches them; individual nodes execute experiments and provide feedback. This simple approach relieves individual nodes from the burden of managing and executing experiments at the same time. Our approach requires a system architecture to orchestrate experiment procedures and a communication protocol to communicate commands, feedbacks, and experiment data. In this section we introduce the protocol and in the next the system architecture.

We propose a traffic flow control protocol (TFCP) in order to facilitate the remote management of inter- and intra-experiment executions. We define a set of control primitives which can be exchanged by the control protocol.

3.2.1 TRAFFIC FLOW CONTROL PRIMITIVES

The traffic flow control primitives abstract a set of commands which control the execution of experiments in an application-independent manner. We classify our primitives into basic and enhanced primitives, according to their control granularity in an experiment execution. These primitives reside on top of any of the existing communication protocols and serve as agents for exchanging messages between experiment controller (server) and sensor nodes (client).

The basic primitives consist of *start* and *stop* commands, which are used to begin and stop an experiment. We keep them to two in order to limit the number of overhead messages that should be exchanged between the server and the nodes. The enhanced primitives, on the other hand, enable the execution of more complex experiments and provide fine-grained control. They consist of the following commands: *pause*, *continue*, *terminate*, *reset*, *clear*, and *read*. The combination of *pause* and *continue* can be used to suspend the execution of an experiment at an individual node for an arbitrary time, while the *terminate* command can be used to break an experiment entirely. The last three primitives are useful for managing logged data and local resources.

The enhanced primitives can also be used for event injection into a network. Suppose we wish to test how a routing protocol copes with the dynamic behavior of a network. The dynamic behavior of individual nodes, such as when they leave and join a network or become temporarily unavailable causes a topology change which in turn affects the performance of the routing protocol significantly. It is not unusual to observe in real wireless networks arbitrary appearance and disappearance (failure) of nodes. The enhanced primitives provide adequate mechanisms to emulate and test these types of failures. Using *pause* and *continue* transient failures can be introduced whereas *terminate* can be used to emulate permanent failures. Compared with mechanically turning on and off nodes, which is presently the most frequently used approach to emulate node availability and to introduce transient failures, our approach is more efficient because it enables nodes to retain runtime states even when they are no longer available as active nodes.

Table 3.1: TFCP message types and commands.

type	QoS	command	parameters	initiator	description
setup	reliable, once or more	*init*	application dependent	server	setup the parameters of test round
control	reliable, exactly once	*start*	none	server	initiate the test round
		stop	none	node	notify finish of test round
		pause	none	server	suspend execution
		continue	none	server	resume execution
		terminate	none	server	stop execution permanently
manage	reliable, once or more	*clear*	none	server	erase data storage
		reset	none	server	reset the node
		read	block id	server	retrieve data from the local storage
data	unreliable	-	-	node	data report from nodes to server

3.2.2 PROTOCOL DESIGN

Our protocol can be encapsulated inside the payload of the backbone communication network (such as Ethernet and IEEE 802.11). This way it can easily be ported to or integrated with different backbone networks. This is shown in Figure 3.2. The TFCP layout is composed of a *session ticket*, *type*, and *command + parameters* or *data units* which are of variable length. The session ticket is a 1-byte-length random number which is used to identify and separate different test rounds. We categorize the messages exchanged between the experiment controller (server) and nodes into four groups, based on their QoS requirement and functionality. Table. 3.1 summarizes how the primitives are categorized. The commands in the “setup”, “control” and “manage” groups require reliable communication (feedback is required to indicate failure or the successful execution of experiments). The commands in “control” group can be executed exactly once within an experiment, while the ones in the other two groups can be executed multiple times. Unreliable communication (no explicit feedback is required) is sufficient to extract data from individual nodes.

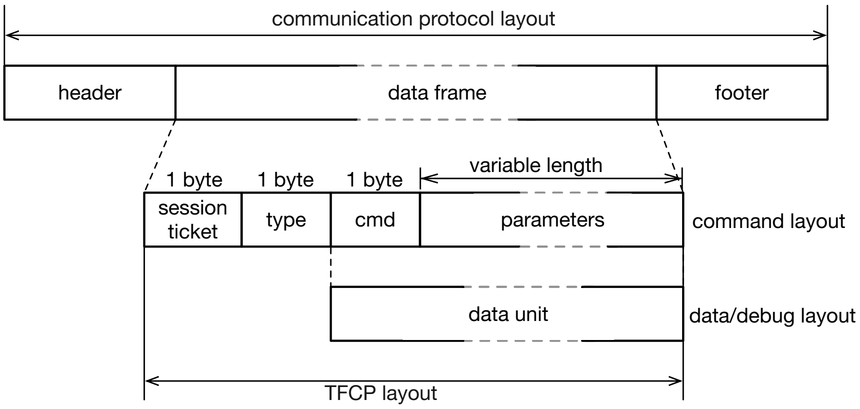


Figure 3.2: TFCP encapsulated as the payload of an existing communication packet.

3.3 ARCHITECTURE OF MOBILAB

The main purpose and, therefore, contribution of our testbed is the flexible but reproducible execution of complex experiments with wireless sensor networks in which some of the nodes are mobile. The testbed enables users to upload their own program image onto individual nodes and to specify experiment procedures and the movement pattern of mobile robots independent of the types of applications the wireless sensor networks are supporting. To achieve these goals, our testbed separates resource management into different concerns.

3.3.1 HARDWARE ARCHITECTURE

The hardware architecture (displayed in Figure 3.3) consists of four modules: a control station, a wireless sensor network, a node manager, and a backbone wireless channel. The control station serves as the main interface between the user and the testbed. A group of dedicated software services run in the control station to manage the testbed resources and to control experiments. In the next subsection we provide a detail description of the software architecture of the control station.

The wireless sensor network consists of three types of nodes: static relay nodes, mobile nodes, and sniffer nodes. The sniffer nodes are special stationary nodes that are not involved in any experiment, but are useful for monitoring the state of the wireless channel to obtain complementary information about experiment execution during debugging. By changing the firmware, the sniffer nodes can also produce interference into the network. A node manager interfaces a node with the control station. Each node manager (for our implementation we used a Raspberry Pi board) is connected to a wireless sensor node via a USB port. The node managers and the control station establish the backbone network to exchange management and experiment information at runtime. We use a WiFi *ad hoc* network as our backbone network because of its scalability and flexibility. A node manager enables to easily program and control a node as well as to collect useful performance related data from it. The node manager connected to a mobile node has the additional task of controlling the motion of the robot and collecting location information.

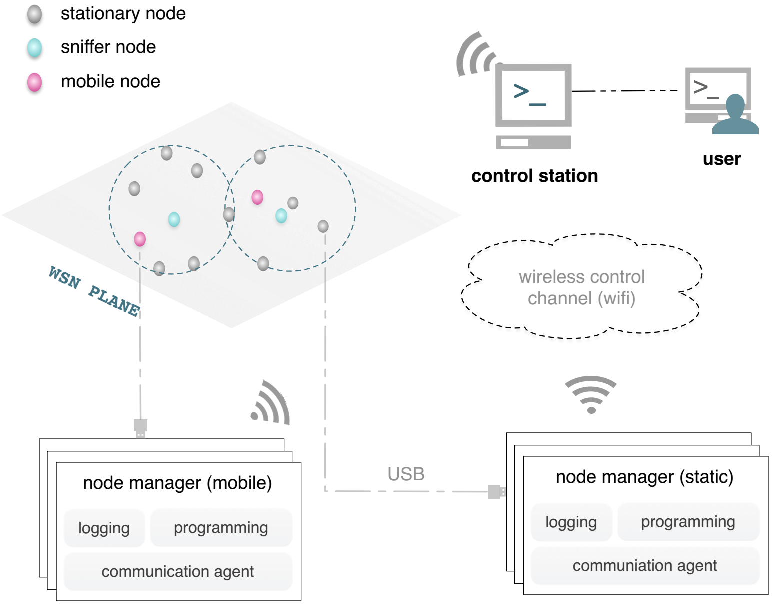


Figure 3.3: The hardware architecture of MobiLab.

3.3.2 SOFTWARE ARCHITECTURE

The control station is the most important module in MobiLab. It ensures that the testbed as a whole functions as a unified system. It is through the control station every program image or command is propagated to the wireless sensor network. Figure 3.4 displays its software architecture, which consists of a user interface, a resource management service, an experiment management service, a data management service, and a data analysis service.

USER INTERFACE

MobiLab provides both a web-based and a command-line-interface through which users can access the testbed and conduct experiments remotely. Users can browse active nodes and their status, upload program images into the wireless sensor network, and specify and manage experiment procedures using experiment execution primitives we defined (to be discussed below).

RESOURCE MANAGEMENT

MobiLab does not require a fixed infrastructure (a specific network size or topology) to run experiments. As to which specific pair of nodes should communicate with one another at any given time and for how long can be specified in experiment procedures to evaluate, for example, link quality fluctuation between them. The resource management service is responsible for authorizing nodes to join the network and users to access individual nodes; for managing binary images, and for ensuring proper program installation. Moreover, the resource management service uploads and deletes program images to and from nodes and controls versions. In it, a synchronization daemon runs in the background to ensure that program images in the control station and the node managers are consistent.

EXPERIMENT MANAGEMENT SERVICE

The experiment management service enables users to define and manage inter- and intra-experiment activities. As regards management, users can initiate, interrupt, suspend, modify, and end experiments at runtime by using experiment execution primitives (see Table 3.2). The primitives enable users to configure interaction (transmission power, channel, partner nodes) and to specify communication duration, among others. When an experiment procedure is submitted to it, the experiment management service validates the procedure to ensure that it is executable, parse the procedure to extract experiment parameters, translates the parameters into binary, creates a control flow (execution sequence), and passes the control flow to the execution manager. The execution manager is responsible for coordinating the execution of an experiment procedure until it terminates. A virtual node manager within the control station's architecture creates a virtual representation for each physical node. The aim is to hide differences in hardware architecture between nodes from users and to provide common interfaces for accessing and interacting with them.

DATA MANAGEMENT AND ANALYSIS

Data management or logging is one of the useful features of testbed frameworks. When an experiment is launched, MobiLab creates an instance of a data logging module which is then associated with the communication agents of the corresponding virtual nodes. During experiment execution, the physical nodes log the desired data locally and forward them to their virtual node managers at the control station, which then stores the data in a database. Alternatively performance indicators can be directly streamed to virtual node managers as they are generated. In MobiLab, we integrated different performance evaluation metrics for mobility management protocols.

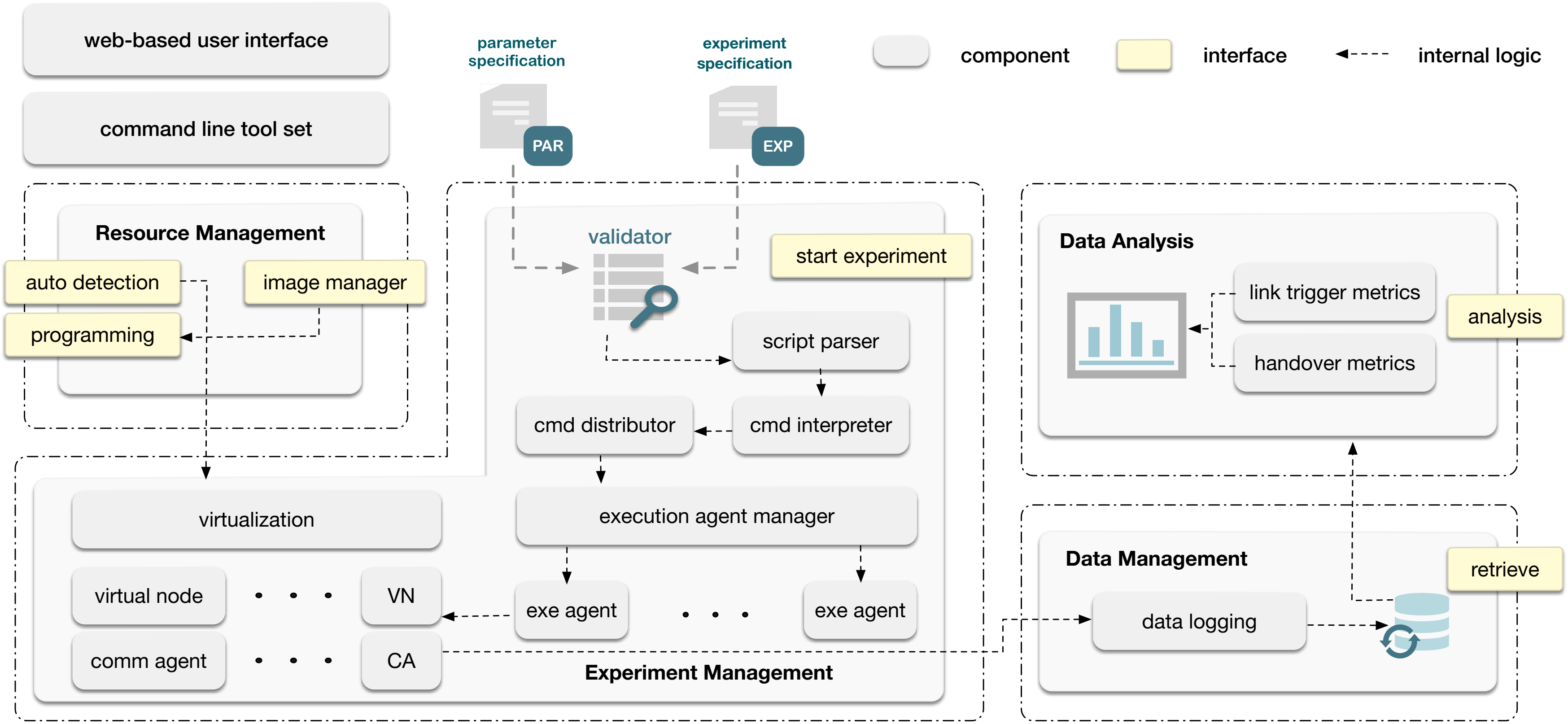


Figure 3.4: The software architecture of the control station.

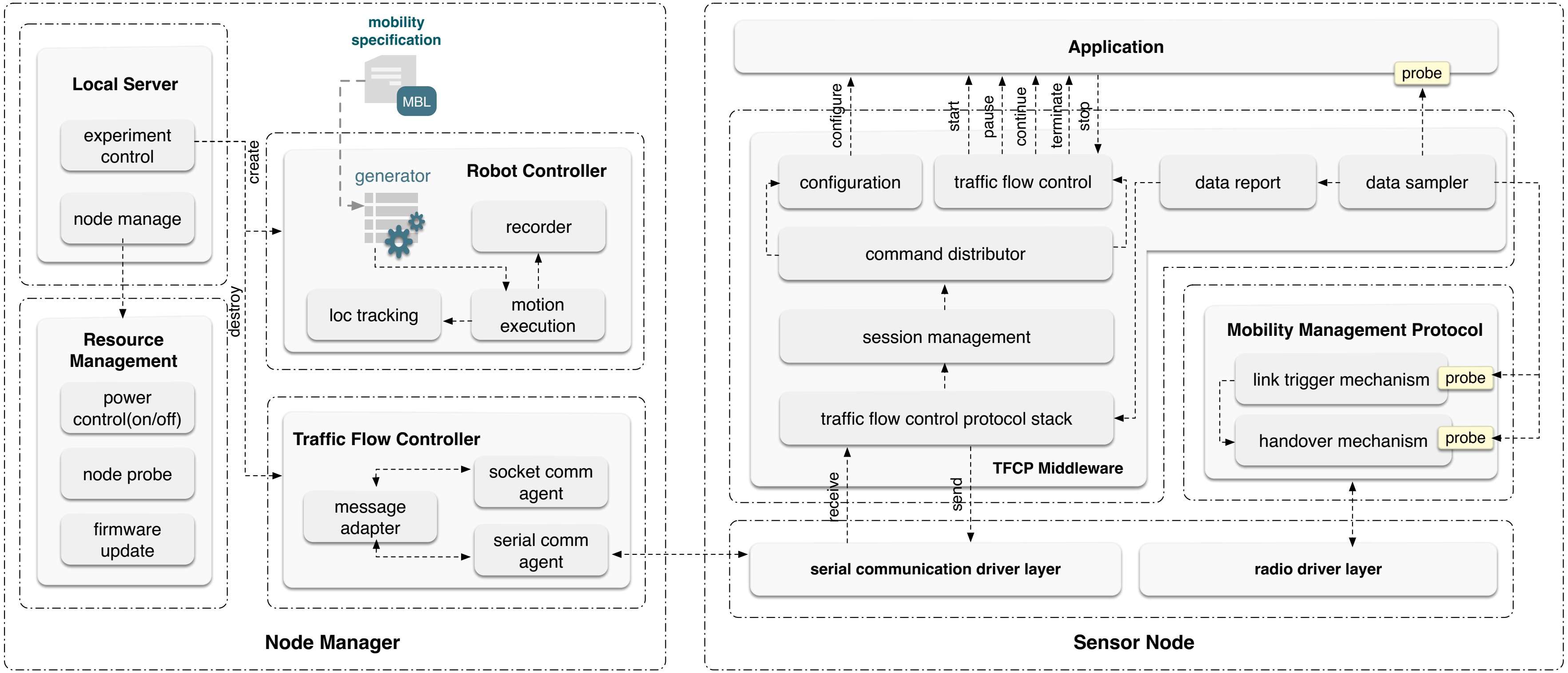


Figure 3.5: An overview of the software architecture of the testbed from a single node perspective: (Left) The software architecture of the node manager. (Right) The software architecture of a sensor node.

Table 3.2: Traffic flow control primitives.

primitive	description
configure	setup the application dependent parameters
start	initiate the test round
stop	notify finish of test round
pause	suspend execution
continue	resume execution
terminate	stop execution permanently

3.3.3 NODE MANAGER

A node manager is a physical device which is physically connected with a wireless sensor node via a USB interface. The idea is to facilitate the dynamic reprogramming of nodes, the replacement of modules, and the collection of relevant performance indicators during experiments. The software aspect of a node manager has three components, which are the local server, a resource manager, and a traffic flow controller. A robot node manager includes an extra module for managing mobility.

LOCAL SERVER

It is a socket-based server that receives commands and messages destined to the physical node from the control station. Its main responsibility is managing the physical node and controlling the proper execution of experiments. The server is logically connected with the resource management service at the control station, thus it is able to provide the functionalities for probing the sensor node, updating firmware and physically powering on and off the node; it is also responsible for coordinating experiment control flows and commands pertaining to the motion of a robot.

TRAFFIC FLOW CONTROLLER

The procedure of an experiment is first encoded using the traffic primitives we specified in Table 3.2. By the time it reaches the traffic flow controller at the node manager, it is translated into a sequence of commands and parameters. The traffic flow controller is responsible for creating a channel between the node manager and the physical node and for transmitting the commands and parameters in their sequence and appropriate delay to the physical node. It also channels the logged data from the physical node to the node manager. The node managers are time synchronized with the control station at the beginning of each run of an experiment.

ROBOT CONTROLLER

The motion of a robot is controlled by a robot controller. The controller is instantiated by its node manager before an experiment is launched and destroyed after the experiment is completed. Different mobility models can be implemented and integrated into the node manager a priori and an instance of a model can be loaded when the robot controller is first instantiated. The parameters of this model can be modified at runtime by using the experiment primitives in Table 3.2. Currently, we are experimenting with straight line walking and the random waypoint model [CBD02].

3.3.4 SENSOR NODE

Figure 3.5 (right side) illustrates the software architecture of a wireless sensor node. Most relevant to this paper is the traffic flow control protocol middleware (TFCP), which we shall discuss in some detail. The TFCP middleware is an application independent layer for managing inter and intra-experiment activities. It is loosely coupled with the OS layer, interacting with communication drivers by `send` and `receive` interfaces and exposing six interfaces to the higher layers (MAC, network and application layers), so that users can setup experiment and application specific parameters and control the execution steps of experiments. The data sampler and report module locally collects and aggregates performance indicator metrics from relevant layers and communicates them with the control station via the TFCP middleware.

The mobility management protocol does not belong to the MobiLab testbed. We integrated it to investigate the performance of different mobility management protocols under the same setting.

3.4 IMPLEMENTATION

We established a wireless sensor network with TelosB nodes; its size varied between 10 and 20 stationary nodes and three mobile nodes. We implemented the control station on a laptop computer, in a Linux environment. Each TelosB node is physically attached to a Raspberry Pi 2 model B board [ras], which serves as a node manager. The Raspberry Pi board has 4 USB ports, a 900 MHz quad-core ARM Cortex-A7 CPU, 1 GB RAM, and 8 GB micro SD card. The mobile TelosB nodes are carried by Diddyborg robots [did] and controlled by their own onboard Raspberry Pi boards. The Raspberry Pi boards established a background ad-hoc network using USB-WiFi adapters and the AP-Hotspot[1]. Several studies confirm that IEEE 802.11 and IEEE 802.15.4 radios can coexist with each other without obvious interference if non-overlapping channels can be selected carefully [SDTL10] [BKM+12], which we did.

All the software components running inside the control station and the node managers are developed with Python, which can easily be ported to other platforms. The web application

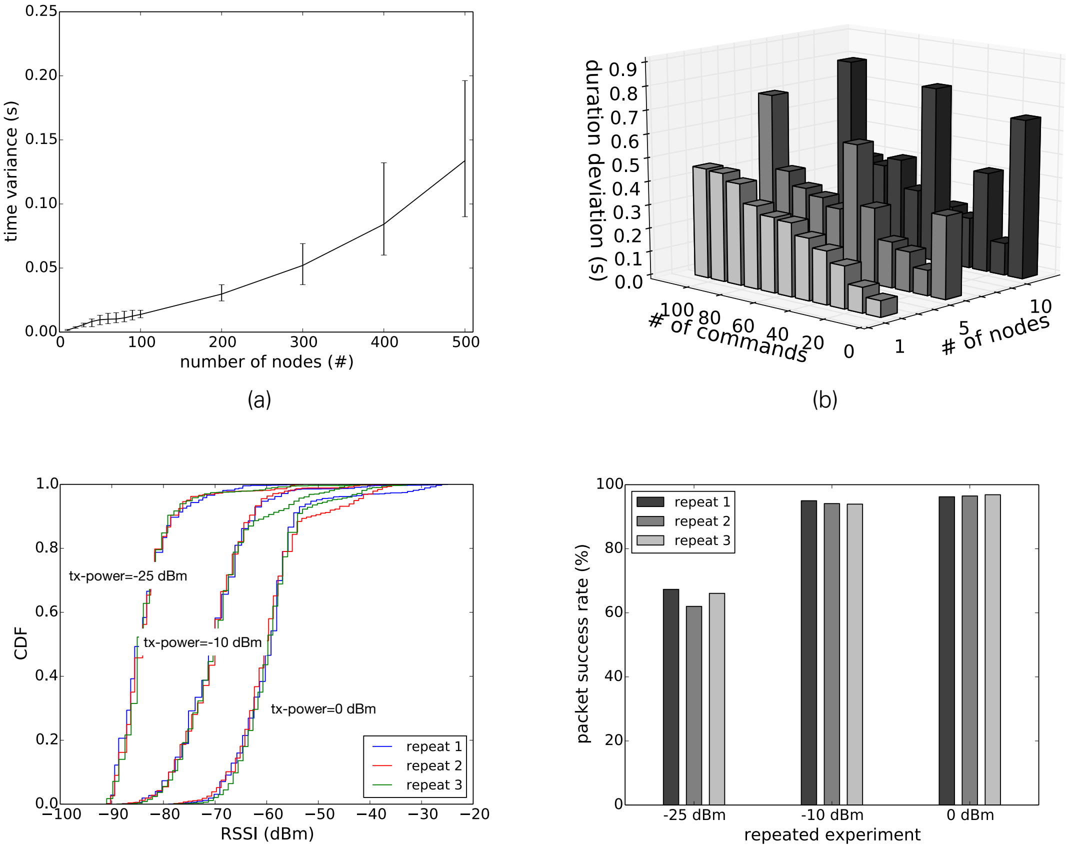
---

[1] https://github.com/hotice/AP-Hotspot

is based on Django Framework and Apache server. The TFCP middleware running on each sensor node is built on top of TinyOS and has a footprint of 1058 bytes of ROM and 84 bytes of RAM.

3.5 EVALUATION

In order to carry out reproducible experiments, the detail of the experiment procedures are scripted, i.e., the beginning, end, and duration of every activity is specified. When the control station dispatches experiment procedures, they may not be executed by the individual nodes at precisely the same time. Consequently, nodes may not begin and end the execution of experiments at the same time. This phenomenon is an aspect of both the size of the network and the complexity of the experiments.



(a)

(b)

(c)

(d)

Figure 3.6: Performance analysis of MobiLab as an experiment management tool: (a) The time variance of synchronized starting time for networks of different sizes; (b) The deviation in the duration of arbitrary control commands in a single experiment; (c) CDF of RSSI fluctuation of repeated experiments; (d) packet success rate of repeated experiments

To investigate this phenomenon, we launched a set of simple experiments with variable number of physical nodes (from 5 to 20) and emulated nodes (up to 500). We recorded the starting time of each node and calculated the maximum variance (time difference between the earliest starting node and the latest staring node). We observed that the maximum variance of experiment beginning time was 200 ms. Lastly, we inserted arbitrary number of control commands (*pause* and *continue* commands) in the experiments lasting up to 600 seconds, and varied the number of nodes from 1 to 10. We did not observe significant increments of experiment completion times when the number of commands increased (shown in Figure 3.6b). To show the repeatability of an experiment using MobiLab, we evaluated the RSSI fluctuation and packet reception rate under different configurations. We conducted a series of experiments and repeated each experiment three times. As Figure 3.6c and 3.6d show, the CDF of RSSI values are almost the same for each repetition of the same experiment and the packet reception rates are comparably the same.

3.6 CONCLUSION

In this chapter we introduced MobiLab, a testbed we developed to experiment with wireless sensor networks which support mobile nodes. MobiLab separates the concerns of application and protocol developments from their testing phase. Our main motivation is performing repeated and reproducible experiments independent of the types of network topology, communication protocols, communication parameters, and sensors involved in the experiments. We presented both the conceptual architecture and the implementation of our testbed. The hardware architecture of MobiLab consists of a control station, node managers, sensor nodes and mobile robots as mobile platforms. The node managers establish a WiFi backbone network to provide management and data collection functionalities during the execution of experiments. The wireless network enables flexible deployment and scalability. The hardware components of MobiLab are commercial off-the-shelf (COTS) products, which makes MobiLab affordable and easy to reproduce. From the software perspective, besides sharing the same design principles with existing testbeds, MobiLab provides several novel contributions such as supporting both inter- and intra-experiment management, TFCP middleware in a sensor node, and a robot motion management. Except for the sensor node architecture, which is implemented in nesC for the TelosB platform, all the remaining software components are implemented in python, which is relatively easy to port to other platforms. Our future goal is to use the testbed for testing different mobile applications and routing and MAC protocols, to enlarge the wireless sensor network, and to deploy the testbed in different environments.

4 LINK CHARACTERISTICS OF MOBILE WIRELESS SENSOR NETWORKS

Radio link quality of low-power wireless sensor networks (WSNs) is one of the essential factors which has to be taken into consideration, when designing media access control (MAC), or routing protocols [SDTL10]. Due to the low-cost, low-power features, the radio transceivers used in WSNs are quite prone to be affected by background noise, multi-path fading, shadowing and environment changing. Furthermore, the imperfectness of the hardware production and design of the antenna usually cause the irregularity of the radio propagation in different directions. All these factors may lead to link quality fluctuations, and can subsequently affect the performance of the protocols. Besides these factors, when the mobile nodes are included in the network, there could be some other factors which may dominantly affect the link quality, such as the body-effect and the angle of the antenna. In the last decades, a large number of studies have been conducted in order to investigate and model the characteristics of low-power radios, and how they affect the performance of the communication protocols. Most of these studies only focus on static scenarios. Less effort has been made to study the mobility impact on link quality and what factors have to be taken into account when designing mobility-aware protocols.

4.1 BACKGROUND

A large number of research efforts have been performed on empirically studying the characteristics of low power wireless links. Most of these studies focus on stationary networks. They argued that links are dynamic and affected by temporal and spacial factors [SDTL10] [LMZ+16], such as temperature, surrounding environment, relative distance between the transmitter and the receiver etc.. The differences in experimental environments, devices and parameter settings result in different observations and conclusions. However, they all use the same link quality metrics:

Received signal strength indicator (RSSI) is a measurement of received signal power in dBm which is estimated over 8 symbols period of time (128 μs) [Ins13]. In CC2420, it can be read directly from an 8 bits, signed 2's complement register.

Signal to noise ratio (SNR) is an enhanced metric used to measure the link quality taking background noise into consideration. It is defined as the ratio of the received signal strength to background noise. When there is no incoming signal, the value read from the RSSI register represents background noise. Hence, SNR can be simply calculated as:

$$SNR[dB] = RSSI - Noise \tag{4.1}$$

Link quality indicator (LQI) is a link quality metric defined in the IEEE-802.15.4 standard [C$^+$06] and is a characterization of the strength and quality of the received packet. It has different implementations for different radio vendors. For example, for CC2420, the LQI is an averaged correlation value which is measured by the first 8 symbols of each incoming packet, ranging from 50 to 110 [Ins13].

Acknowledgment (ACK) is a link layer metric, which indicates that the transmitted packet has been successfully received by the receiver. This metric can be extracted at the sender side whether the corresponding acknowledgment packet is successfully received within a specific time, after each data transmission.

Packet reception ratio (PRR) is a receiver side metric, which can be calculated as the ratio of the number of successfully received packets over the number of transmitted packets within a specific time window. This metric is usually equivalent to the acknowledgment reception ratio (ARR) which is a link layer metric at the sender side, if disregarding link asymmetry. ARR is sometimes referred to packet delivery ratio (PDR) or packet success rate (PSR). We will use ARR, PDR and PSR alternatively in this dissertation.

Commonly, a link can be classified into three regions according to PDR, namely connected, transitional and disconnected region. In the connected region where the PSR is usually above 90%, communications are quite reliable and symmetric, thus, packet losses occur occasionally. In the transitional region where the PSR varies from 10% to 90%, link quality fluctuates frequently and links are referred to as dynamic and busty. In the disconnected region where the PSR is below 10%, link quality is too poor and communications are hard to proceed.

4.2 EXPERIMENTAL METHODOLOGY

In order to study the mobility impact on the link quality of wireless sensor networks, we have conducted a variety of experiments in different environments and setups. A large amount of link quality information (e.g., RSSI, ACK, LQI etc.) has been collected during these experiments.

(a) nodes (b) foyer (c) corridor (d) pathway (e) lawn

Figure 4.1: Experiment environment for empirical studies on link quality in mobile wireless sensor networks: (a) an example of static and mobile node. Indoor: (b) foyer (c) corridor. Outdoor: (d) pathway (e) lawn

4.2.1 PLATFORMS

In all the setups, we employ TelosB motes [PSC05] and Imote2 [NHS+08] as the experimental hardware platforms. Despite hardware engineering differences, both platforms are integrated with a Chipcon CC2420 radio chip, which is a widely used IEEE 802.15.4 compliant transceiver. The CC2420 has 16 non-overlapping channels in the 2.4 GHz unlicensed ISM band, each channel occupies 2 MHz and has 5 MHz channel spacing [Ins13]. It shares the same wireless spectrum with other wireless technologies like WiFi and Bluetooth. Despite having the same transceiver, the two platforms have different RF engineering designs: the TelosB platform has an on-board printed inverted F-style antenna, while the Imote2 platform integrates a 2.4 GHz surface mounted antenna. Unless explicitly specified, the findings presented in this study are generated from both platforms. To avoid the hardware discrepancy, each experiment is repeated at least 10 times with different hardwares.

4.2.2 TESTBED AND ENVIRONMENTS

We utilize MobiLab testbed [WAD16a] to control all the experiments, regardless of whether they are static or mobile and up to 16 motes are involved accordingly. Figure 4.1a shows an example of a static and a mobile node used in the experiment. MobiLab is a scalable mobility enabled wireless sensor network testbed, which can be easily deployed to different environments and can conduct reproducible and repeated experiments automatically by the control script (as discussed in Chapter 3). Despite the different deployment topologies, the backbone channel of the testbed represents an ad-hoc network via WiFi communication. It is used to control the experiment and to collect link quality data.

The experiments are conducted in both indoor and outdoor environments. In the indoor environment, the nodes are deployed in a foyer (a large, empty space with statues in the center) and a corridor of the faculty building and are arranged in a linear topology with a clear line of sight (see Figure 4.1b and 4.1c). In the outdoor environment, the nodes are spread on a pathway beside a building or lawn in a garden (see Figure 4.1d and 4.1e). In all the setups, except the experiment conductors, there are other pedestrians moving around the test field occasionally.

4.2.3 COMMUNICATION PROTOCOL AND ITS PARAMETERS

We develop a specific link quality evaluation protocol in TinyOS 2.1, which is inherited from the default radio stack of CC2420 with some modifications. For each packet transmission, background noises (before and after each transmission or after each reception), RSSI, LQI, ACK, timestamps and packet sequence numbers are recorded. To avoid collision, a dedicated time schedule is implemented to ensure that only one pair of transmitters and receivers are active at one time. Unless specified, the mobile nodes attached on the robots are the transmitters and all other nodes in the network are receivers.

In most of the experiments, transmission power is set to -25 dBm which leads to a radio communication range up to 20 meters in both indoor and outdoor environments. The reason that this low transmission power is chosen is two-fold: 1) The speed of the mobile platform is approximately 0.13 m/s, so it takes a significantly longer time to travel to observe the link disconnection than if a larger transmission power is used, e.g., 0 dBm. By using -25 dBm, the communication range is limited to 30 meters. Thus, all the link characteristics (perfect, transitional, disconnected regions [SDTL10]) can be observed in a small area. 2) The maximum coverage of the backbone channel of MobiLab testbed in outdoor environment is approximately 50 meters. We can not use a more powerful WiFi access point because it requires AC power supply, or more access points which increase the management complexity.

In this chapter, we focus on the mobility impact on the link quality of wireless sensor networks. The cross technology interference (CTI) is not taken into consideration. To minimize the CTI impact, the transmission channel is set to 26, which is orthogonal to most widely used WiFi channels[1] (channel 1, 6, 11) [SDTL10][LPLT10][MGC16]. Figure 4.2 shows the average background noise with error bars in different environments when the experiments are conducted. The background noises are almost consistent (around -95 dBm to -92 dBm) with little variation which indicates that the CTI is negligible. Thus, in the following analysis, RSSI is used in stead of SNR as one of the link quality metrics.

4.3 STATIC VS MOBILE

There are several empirical studies on the characteristics of low power links in wireless sensor networks, which mostly focus on stationary scenarios [SDTL10][BKM$^+$12][DRR$^+$15]. These observations provide implications to protocol design in static networks, like MAC and routing protocols. These can not be applied to the design of mobility aware protocols directly,

---

[1] WiFi channel 1 is used as the testbed backbone channel and in our office building the WiFi channel 6 and 11 are used.

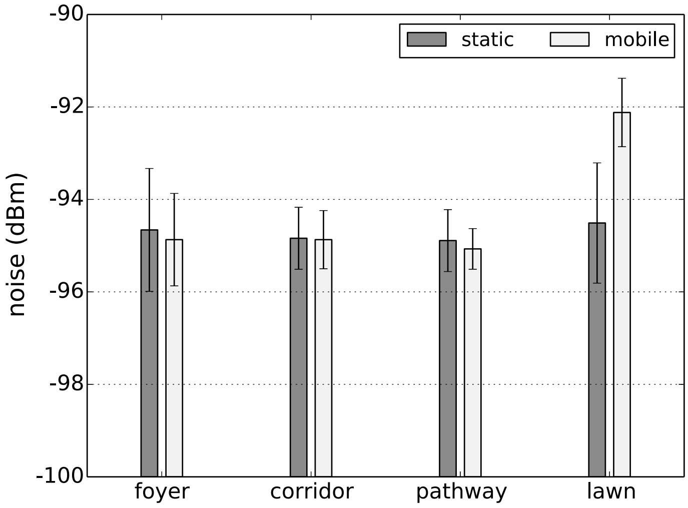


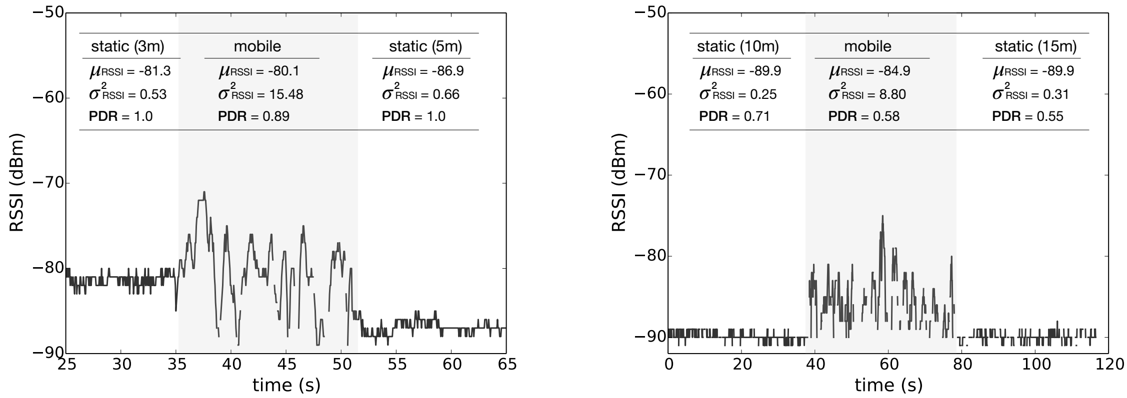
Figure 4.2: Background noise in different environments

because links in the mobile scenario are significantly different from their static counterparts. Understanding the characteristics of mobile wireless links and the differences with static scenarios can give us implications of designing efficient mobility aware protocols. To this end, we conduct several experiments to highlight the differences between static and mobile links in different environments.

4.3.1 TEMPORAL CHARACTERISTICS

Experiment setup. A mobile node carried by a robot is the transmitter. It continuously sends packets to a stationary node with 50 ms inter-packet interval (IPI). The mobile node is initially deployed at a location 3 meters away from the static receiver and sends packets for some time. It then moves 5 meters in a straight line and stays there to complete the transmission. The second experiment has the same settings as the first one. However, the mobile node begins at a different initial location (10 meters) and has a longer moving distance (5 meters). The experiments are conducted in an indoor environment without significant interference.

Observations. Figure 4.3 shows the temporal characteristics comparison of static and mobile scenarios. RSSI fluctuations are presented in real time and the mean and variance values of RSSI are calculated section by section. Figure 4.3a shows RSSI fluctuations in a connected region where links are stable and the packet delivery ratio is above 90%. When the transmitter is static (at 3 m and 5 m locations), link quality is quite stable that the packet delivery ratio is 100%, and RSSI slightly varies in time (the standard deviation is less than 1 dB). When the transmitter is mobile (moving from 3 m to 5 m), RSSI fluctuates drastically, such that the standard deviation is around 4 dB. The packet delivery ratio drops to 89%. Figure 4.3b shows the temporal behavior of link variations in a transitional region. It shows the similar variation



(a)

(b)

Figure 4.3: Comparison of temporal characteristics for static and mobile links: (a) link fluctuation in the connected region. (b) link fluctuation in the transitional region.

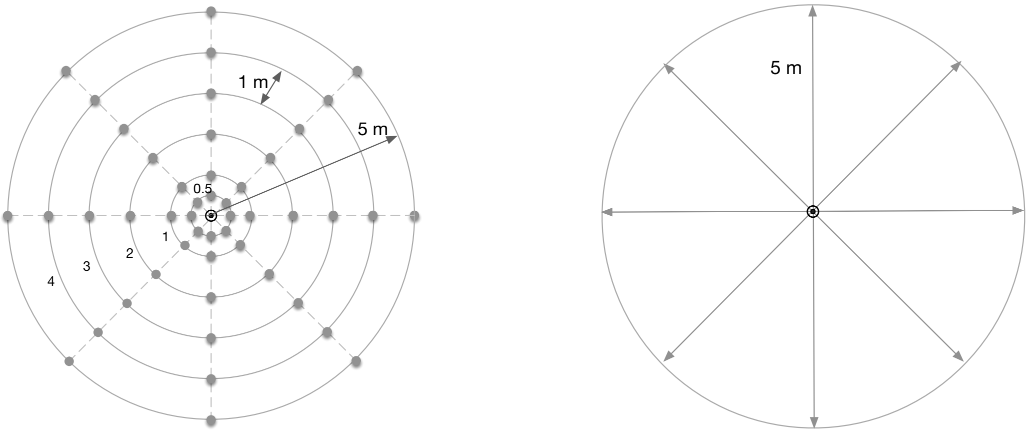
patterns as in the connected region. From the above observations, it can be concluded that:

- RSSI fluctuates drastically when the transmitter is mobile, regardless of its relative distance to the stationary receiver.
- These drastic fluctuations result in significant packet loss, even when the link is in the connected region.

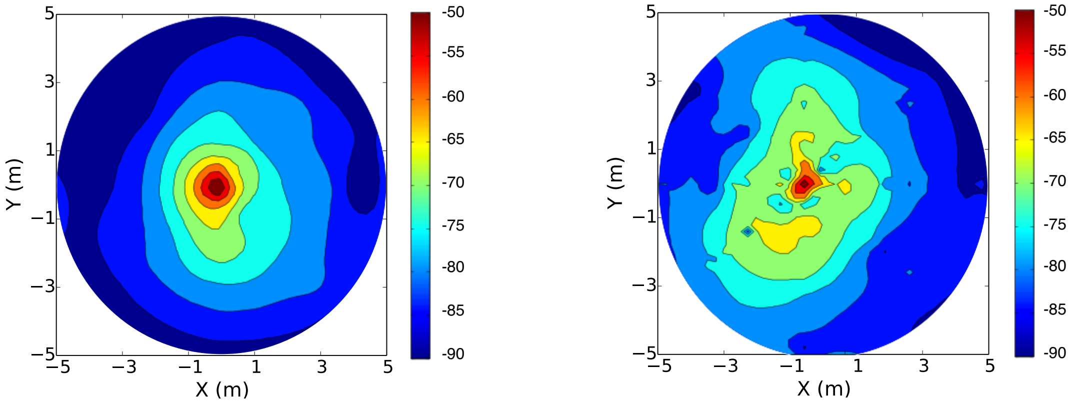
4.3.2 SPATIAL CHARACTERISTICS

Experiment setup. To investigate the differences in spatial characteristics of static and mobile links, we conduct two experiments in an outdoor environment (the lawn). For the static scenario, transmitter nodes are deployed in radial topology, with the receiver node at the center. For the mobile scenario, the transmitter node is carried by a robot and moves away from the stationary receiver in 8 directions along radius paths. Figure 4.4a and 4.4b show the deployment topology for the static experiment and the moving trajectory for the mobile scenario.

Observations. Figure 4.4c and 4.4d show the contour plots of RSSI distribution for static and mobile links respectively. It is obvious that RSSI distribution is spatially irregular, regardless of whether links are static or mobile. Signal propagation is anisotropic against direction changes which results in a non-uniform distribution of RSSI at different locations. The radio irregularity phenomenon is confirmed by previous empirical studies on static wireless links [ZHKS04] [BKM+12]. There are three potential causes for radio irregularity: (a) The omnidirectional antenna does not have the same gain in different directions. (b) Environments in different directions are not uniform, which may lead to different degrees of path loss. (c) Signal varies over time. We argue that the non-uniform antenna gain is the primary factor causing the radio irregularity in the static observation, because the experiment is conducted on a lawn where there are no obvious obstacles nearby and the sensor nodes are deployed in a small circular area (5 m radius). To confirm our argument, we deploy two nodes with a spacing of 1 m. One node is fixed while the other node spins at slow speed[2]. Figure 4.5 shows the RSSI variation against different antenna directions.



(a) static deployment (b) mobile trajectory



(c) RSSI distribution of static links (d) RSSI distribution of mobile links

Figure 4.4: Comparison of spatial characteristics for static and mobile links. (a) Static deployment topology, the node at the center is receiver and the others are transmitters. (b) Trajectory of mobile transmitter. The mobile sender moves from the center toward the outer circle along a radius. (c) Contour of RSSI distribution for static links. (d) Contour of RSSI distribution for mobile links.

From Figure 4.4c and 4.4d, we observe that the radio irregularity becomes worse when links are mobile. RSSI variations are more dynamic compared to static links which is also confirmed in the earlier discussion. Although the signal propagation is more irregular, the measured RSSI values are raised in all directions in the mobile links. This can be also observed in Figure 4.3, in which the average RSSI in the mobile links is higher than that in the static links.

---

[2] The node spins 360 degrees in 20 seconds. The rotation speed is controlled by a gear motor.

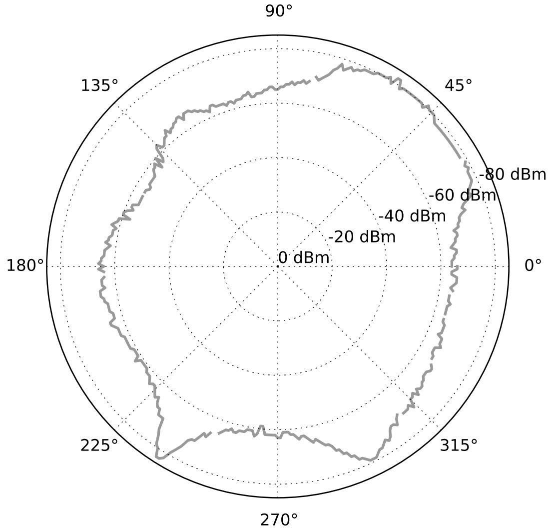


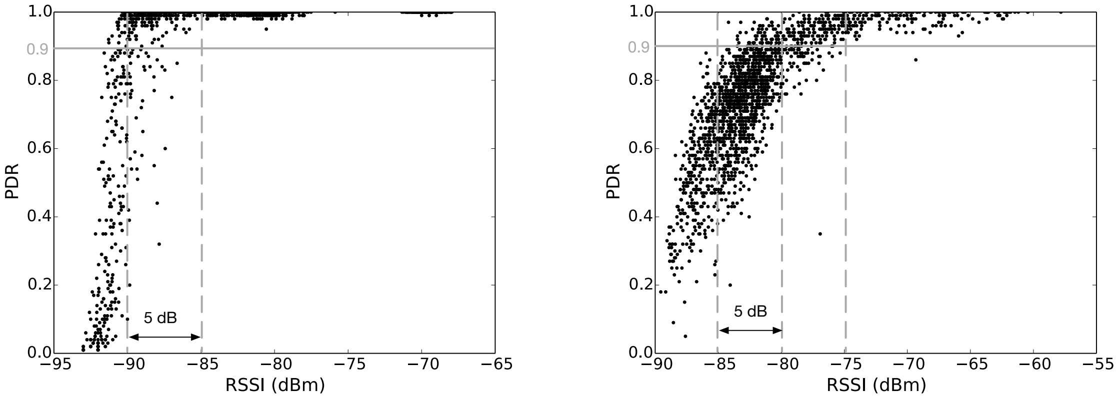
Figure 4.5: Radio irregularity. Two nodes are deployed in a line with 1 m spacing. One is fixed and the other spins at a speed of 0.05 round per second. Degree 0 is the position that the two antennas face to each other.

4.3.3 PDR AND RSSI RELATIONSHIP

Experiment setup. The data used in this section is from all the experiments we conducted for the comparison of static and mobile links. The packet delivery ratio is calculated every 100 packets.

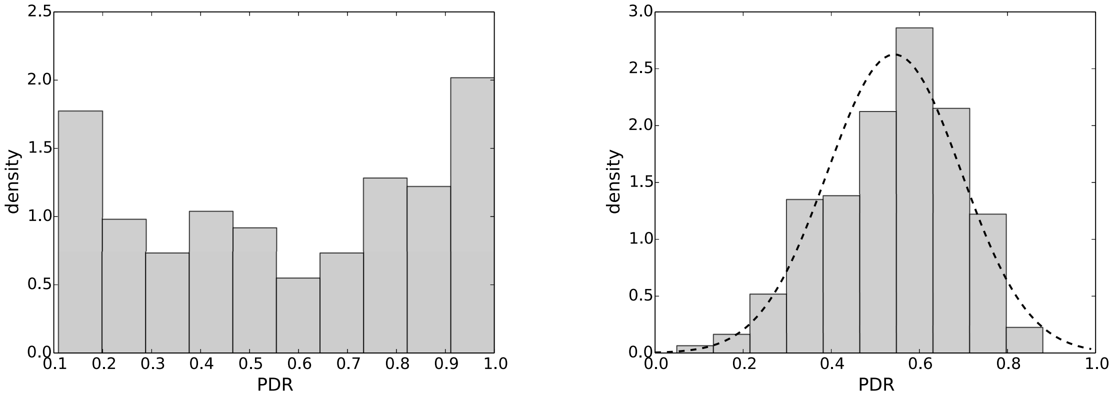
Observations. Figure 4.6a and 4.6b show PDR distributions against RSSI for static and mobile links respectively. The PDR is calculated every 100 packets and the RSSI value shown in the figures is averaged over successful packets. From the plots, it can be known that the PDR is above 90% when the average RSSI is greater than -85 dBm in static links. In mobile links, this threshold is -75 dBm. When the average RSSI value is below this threshold, PDR varies significantly from 100% to 0% (transitional region). This empirical threshold is highly environment dependent. For example, for the CC2420 radio which implements the IEEE 802.15.4 standard, different values are identified: -75 dBm in [GLJ11], -87 dBm in [Lev06] or -80 dBm in [SKAL08] to achieve a 90% PDR. In addition to the different RSSI thresholds to separate connected from transitional region in static and mobile links, the width of transitional region is different. In static links, this width is only 10 dB, while in mobile links it is 5 dB more than that in static links.

To further analyze the relationship between PDR and RSSI in the transitional region where PDR varies from 90% to 10%, we plot the density function of PDR given RSSI within a range of 5 dB. It can be observed that the PDR is almost uniformly distributed when the RSSI is between -95 to -90 dBm in static links, while in mobile links the distribution of PDR is more like a normal distribution, as shown in Figure 4.6c and 4.6d. Figure 4.6e and Figure 4.6f depict



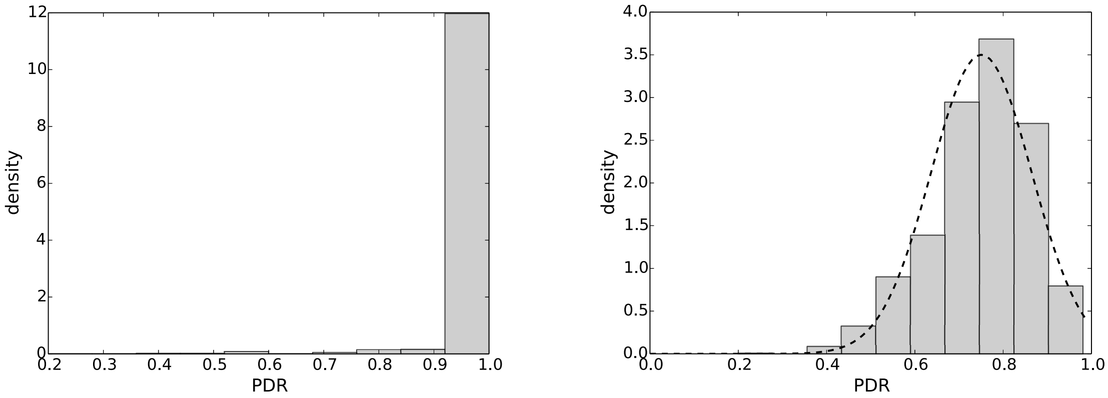
(a) static links

(b) mobile links



(c) static $RSSI \leq -90dBm$

(d) mobile $-90 < RSSI \leq -85dBm$



(e) static $-90 < RSSI \leq -85dBm$

(f) mobile $-85 < RSSI \leq -80dBm$

Figure 4.6: Relationship between RSSI and PDR. Packet delivery ratio is calculated every 100 packets. The density functions of PDR are generated given RSSI with 5 dB width.

the distribution of PDR in another 5 dB section of RSSI for static and mobile links. From Figure 4.6a, we can only observe that when the RSSI is between -90 and -85 dBm, PDR varies from 30% to 100%. However, in Figure 4.6e, it shows that the PDR is above 90% with a probability of 85.4%.

4.4 CHARACTERISTICS OF MOBILE LINKS

In section 4.3, we compared the differences of link quality fluctuations in static and mobile links. Although they have similar characteristics, such as radio irregularity and RSSI variations, the observations from static links cannot be directly utilized as the guideline of designing mobility aware protocols. We argue that links in the mobile environment are more dynamic than their static counterparts. In this section, we explore the characteristics of mobile links from multiple perspectives in depth.

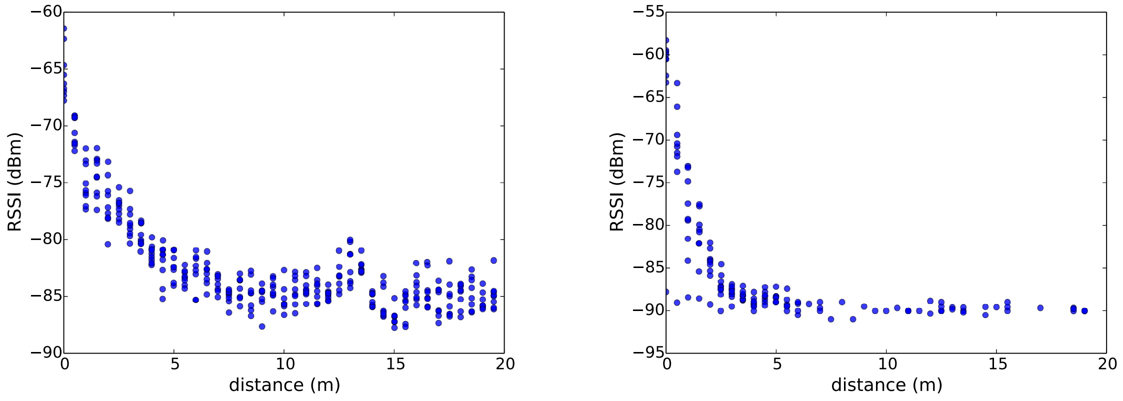
4.4.1 TEMPORAL-SPATIAL CHARACTERISTICS

Experiment setup. To investigate the temporal-spatial characteristics of the link quality in mobile links, we conduct experiments in indoor (foyer) and outdoor (pathway) environments. One stationary node is deployed as the receiver and a mobile node moves away from the receiver in a straight line. The moving path is 20 meters long. The mobile transmitter sends packets at an IPI of 50 ms. The experiments are repeated 10 times.

Observations. Figure 4.7 shows the temporal-spatial behavior of mobile links in terms of RSSI, LQI and PDR. All link quality metrics are collected at the sender side. The three figures on the left are for the indoor environment and the figures on the right are for the outdoor environment. The RSSI and LQI points depicted in the figures are averaged values over 80 packets. And PDRs are generated by accumulating ACKs. Figure 4.7a and 4.7b show RSSI variations along the distance. RSSI is decreasing while the relative distance between the sender and the receiver is increasing. This follows the log-normal path loss model [NH93], which is defined as:

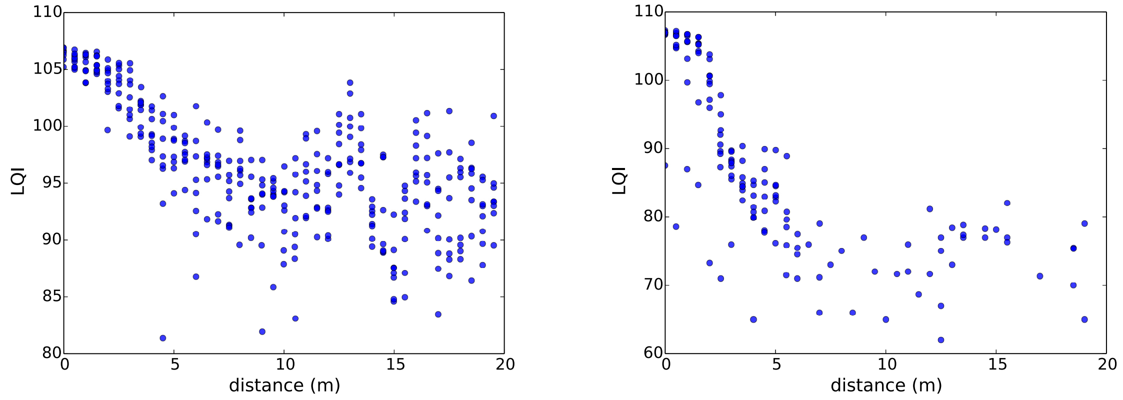
$$RSSI(d) = RSSI(d_0) - 10 * n * log_{10}(\frac{d}{d_0}) + N(0, \sigma^2) \tag{4.2}$$

where $RSSI(d)$ is the path loss at distance d, $RSSI(d_0)$ is the path loss measured at a reference distance d_0, n represents the path loss exponent and $N(0, \sigma^2)$ is the white noise. In different environments, the path loss exponents are different. By applying curve fitting, we can estimate the path loss exponents which are 0.89 in the indoor environment and 1.12 in the outdoor environment in our experiments.



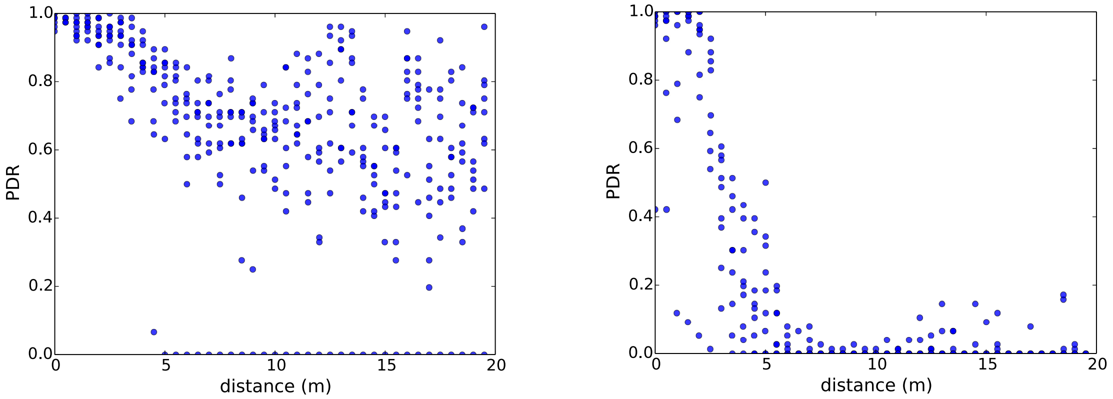
(a) RSSI indoor

(b) RSSI outdoor



(c) LQI indoor

(d) LQI outdoor



(e) PDR indoor

(f) PDR outdoor

Figure 4.7: Temporal-spatial characteristics of link quality in indoor and outdoor environments.

Table 4.1: Correlation coefficients of PDR and RSSI, PDR and LQI

	$\rho_{RSSI,PDR}$		$\rho_{LQI,PDR}$	
slotsize	mean	variance	mean	variance
10	0.56	0.002	0.73	0.002
50	0.67	0.003	0.80	0.005
100	0.73	0.005	0.85	0.011

From Figure 4.7e and 4.7f, we observe that link quality in terms of PDR is not strictly correlated with distance, although PDR generally decreases with distance increasing. In Figure 4.7e, the PDR in section 10 to 14 meters is higher than that in the section of 5 to 10 meters. From 14 to 15 meters, the PDR decreases, then after 16 meters it increases slightly. In Figure 4.7f, we can observe the same phenomenon that farther locations have an improved PDR. Additionally, in the outdoor experiment, the links are almost disconnected after 7 meters. But in the indoor environment, even after 20 meters, the packet success rate is relatively high. Because of the space limitation in our foyer, we cannot observe link behavior beyond 25 meters.

By simply observing the plot shape of the PDR, RSSI and LQI, we find that PDR is highly correlated with RSSI and LQI. They almost have the same variation tendency. To confirm this argument, we calculate the correlation coefficient of PDR and RSSI, PDR and LQI by using the following equation:

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y} \tag{4.3}$$

where $\rho_{x,y}$ is the correlation coefficient of x and y. $cov(x, y)$ represents covariance of x and y, and ρ_x, ρ_y denote standard deviations respectively. Table 4.1 shows the correlation coefficients using different slot sizes to calculate the PDR. When using a larger slot size to calculate the PDR, the correlation coefficient is increasing. In other words, the PDR is significantly correlated with RSSI and LQI.

Another conclusion we can draw from this experiment is that, in mobile links, the transitional region is much more larger than the connected region. As shown in Figure 4.7e, the connected region (PDR > 0.9) in the indoor environment is only 3 meters whereas the transitional region is more than 17 meters[3].

4.4.2 CONSECUTIVE FAILURE ANALYSIS

In section 4.4.1, we investigated the temporal-spatial behavior of mobile links in indoor and outdoor environments. We observed that in both environments, link quality is not strictly correlated with distance and the PDR is divergent in the transitional region. However, the efficiency of the PDR depends on the slot size. In [SDTL10], the authors argued that when using a smaller slot size to calculate the PDR, it results in more good or poor links; when using a larger slot size, the number of links in the transitional region increases and the number

---

[3] Due to space limitations, the disconnected region is not observed in the experiment.

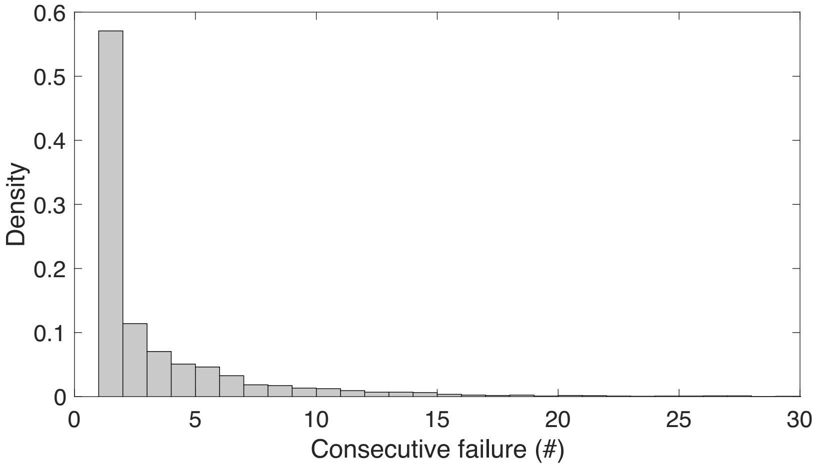
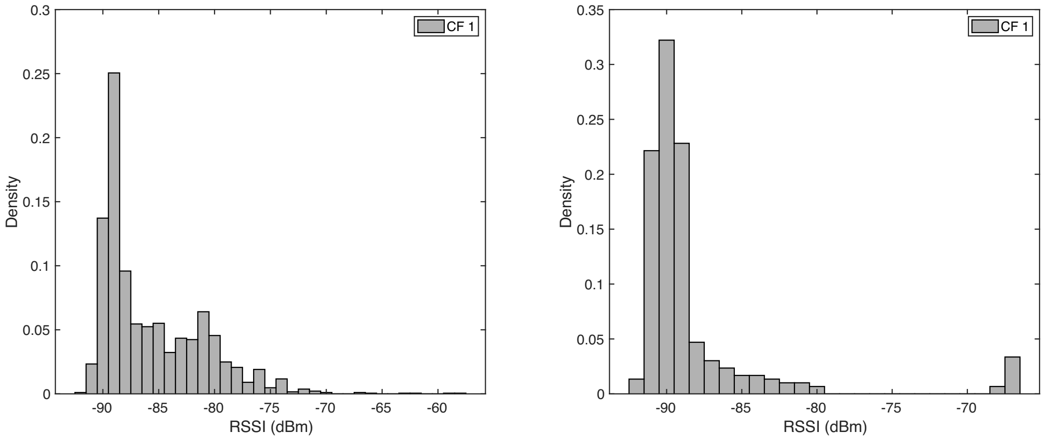


Figure 4.8: Density function of consecutive failure

of links in the connected and disconnected regions decreases. To mitigate the bias caused by slot size, in this section, we analyze the correlation of consecutive packet failure (CF) and RSSI variation.

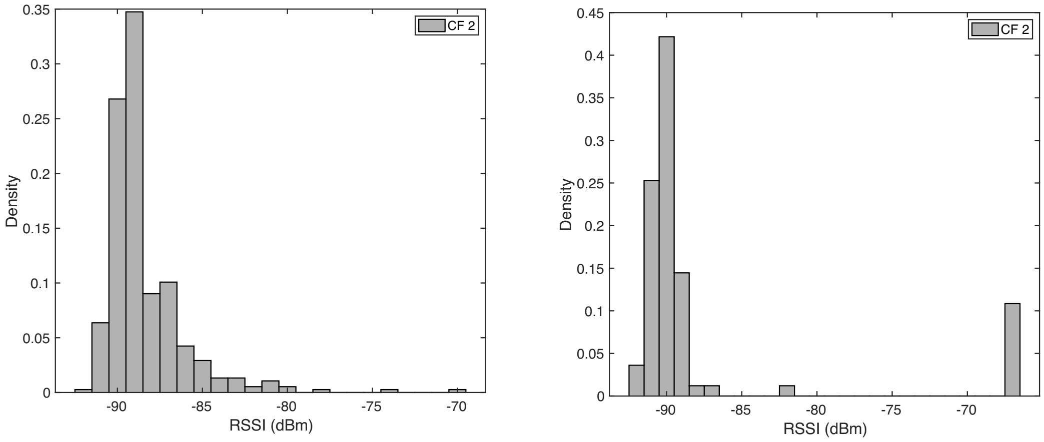
Experiment setup. The data analyzed in this section is the same as in the previous section. CF is accounted through ACK sequences and RSSI is estimated by averaging the RSSI values from successfully transmitted packets before and after the failure(s).

Observations. Figure 4.8 shows the density function of consecutive failure in indoor and outdoor environments. We observe that single packet failure occurs more often than consecutive failures. In 95% of the cases, the number of consecutive failures is less than 10. It is most likely that single packet failure is caused by link fluctuations due to mobility. To further analyze the RSSI variation impact on transmission failure, we plot the density function of the RSSI given specific CF, as shown in Figure 4.9. Figure 4.9a and 4.9b show the RSSI distribution when single packet failure occurs in indoor and outdoor environments respectively. In the indoor environment, single packet failure can be observed in a wide range of RSSI, which means that it is difficult to predict via RSSI values. In the outdoor environment, the result is almost the same, but within a narrow RSSI range which is 5 dB less. From the figures, it should be noted that the probability of single packet failure increases as RSSI decreases. This observation is applicable for the case where the consecutive failure is larger than 2 (as shown in Figures 4.9c to 4.9f).



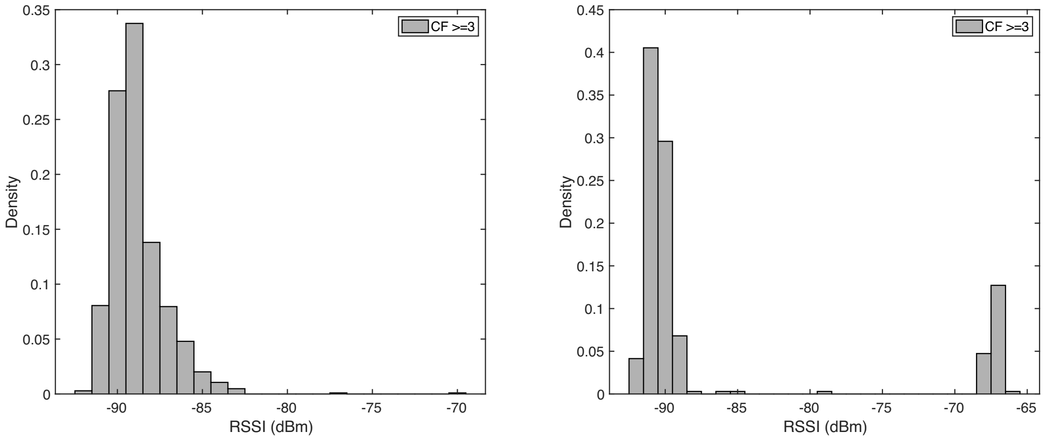
(a) CF=1 indoor

(b) CF=1 outdoor



(c) CF=2 indoor

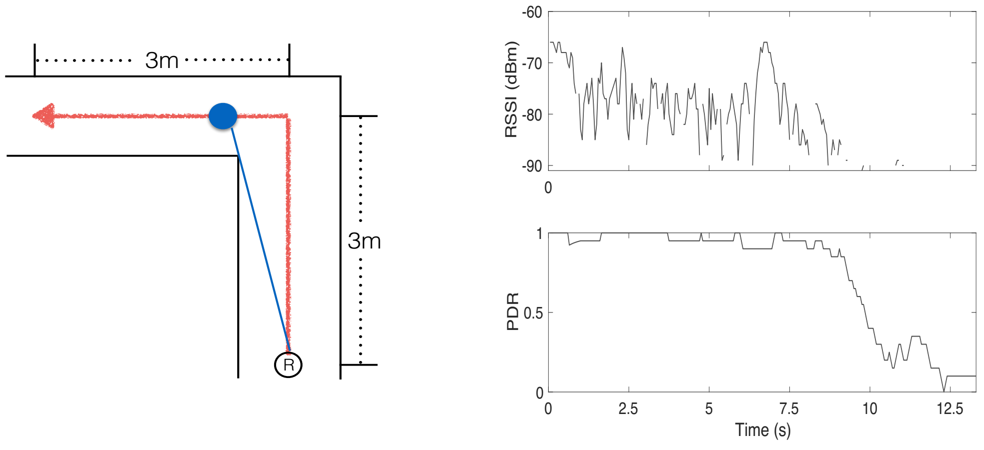
(d) CF=2 outdoor



(e) CF>=3 indoor

(f) CF>=3 outdoor

Figure 4.9: RSSI distribution upon specific consecutive failure (CF) in indoor and outdoor environments.



(a) corner deployment (b) link fluctuation

Figure 4.10: Corner impact on link quality. (a) deployment: R is the stationary receiver, the blue circle represents the mobile sender, and the red line with the arrow is the moving trajectory. (b) link quality fluctuations.

4.4.3 CORNER IMPACT

Experiment setup. Figure 4.10a shows the experiment deployment in a corridor. One receiver is placed on the ground where is 3 meters away from a corner, and one mobile node carried by a person moves away from the receiver toward the corner and then turns left to the other corridor until the link disconnects. The speed of the mobile node is approximately 0.5 m/s which is guided by a metronome.

Observations. Figure 4.10b shows link quality in terms of PDR and RSSI is affected by environment. We can observe that RSSI is slowly decreasing as the distance between the sender and the receiver is increasing and the PDR is above 95% during the first 8 seconds. Afterwards, PDR and RSSI sharply drop to a low level and the link is disconnected. Around 7 second, there is a spike in the RSSI. This is because the transmitter and the receiver is in the line of sight for a short period of time when the person turns to the other corridor. Afterwards, the link is blocked by the corner.

4.5 CONCLUSION

In this chapter, we empirically study the link characteristics in mobile wireless sensor networks. We first compare mobile links with their static counterparts from temporal and spacial perspectives. We observe that mobile links are more dynamic than static links. RSSI fluctuates more drastically in mobile links, which results in significant packet losses. The differences between static and mobile links imply that observations from the static experiments can not be directly applied to mobility-aware protocol design. To investigate the link characteristics of mobile links, we design and conduct a large number of experiments in indoor and outdoor environments. The observations from these experiments reveal the essential characteristics of mobile links and provide fundamental knowledge to assist with designing mobility aware protocols, which will be discussed in the remainder of this dissertation.

5 ADAPTIVE BURST TRANSMISSION SCHEME FOR DYNAMIC ENVIRONMENT

The scope and application of wireless sensor networks is significantly different from other types of wireless networks such as wireless local area networks and ad hoc networks. Whereas the latter are used by many users and many applications, wireless sensor networks are deployed mainly with a single application or even a single sensing task in mind. Moreover, the nodes associated with the latter networks can be charged almost on a daily basis while this is not the case for wireless sensor nodes. In fact these nodes should spend much of the time in a sleeping state to save energy, because they have to operate for a long time without charging or replacing their batteries.

One of the factors which considerably affect the performance as well as the lifetime of wireless sensor networks is link quality fluctuation. Link quality fluctuation can reduce throughput, increase packet delivery latency and jitter, and cost energy due to the retransmission of lost or corrupted packets. This is particularly true for wireless sensor networks which are deployed in harsh environments. The term "harsh" should be understood broadly, for many urban deployments (such as for traffic monitoring, pipeline monitoring, structural health monitoring) where human and car movements are frequented can experience a large packet loss rate [SDTL10].

Commercially available radio chips, such as CC2420, provide a summary of the link quality (RSSI and LQI) by evaluating incoming packets and make this information available to the MAC and higher-layer protocols. This knowledge can be useful in a variety of ways. For example,

- MAC layer protocols can take advantage of this knowledge to save energy, for example, by defining an optimal duty cycle.
- Applications can define a higher-level power management policy that takes the quality of a link into account, for example, whether packets should be transmitted in burst,

whether lost packets should be retransmitted, or whether packet loss can be tolerated to a certain extent.

- If packets should be transmitted in burst, then knowledge of the link quality can be useful for determining the size of a burst.
- In a multi-hop communication, MAC layer protocols can autonomously decide to which neighbor packets should be forwarded.

In most real-world deployments, the quality of a link cannot be known in a deterministic sense and should be modeled as a random process. Statistics pertaining to this process can be obtained directly from the link quality estimation metrics. Because the lifetime of typical wireless sensor networks should be long, sufficient statistics can be gathered from incoming data and acknowledgment packets. An interesting task would be to identify periodicity in the fluctuation of the link quality so that application can determine when to transmit packets and when to refrain from transmitting. To be sure periodicity in a strict sense is difficult to determine because the factors that affect the quality of a link are so diverse. Instead, one can define periodicity in the mean square sense.

For a time varying random process, $\mathbf{l}(t)$, the mean square periodicity can be expressed as [PP02]:

$$E\left\{\left(\mathbf{l}(t+T)-\mathbf{l}(t)\right)^2\right\}=0 \tag{5.1}$$

where T is the period. The autocorrelation of such a process must be doubly periodic:

$$R\left(t_1+mT, t_2+nT\right)=R\left(t_1, t_2\right) \tag{5.2}$$

where t_1 and t_2 are two arbitrary time instances and m and n are two arbitrary integers. It should be noted that periodicity in the mean square sense does not require that the process should be strictly periodic with period T and probability of 1.

The difficulty of this approach is its demand to determine both R and T. If, on the other hand, $\mathbf{l}(t)$ can be considered statistically stationary (at least in a wider sense), then it suffices to observe the process for a certain period of time to obtain the distribution or the density function and with it to determine T. In this chapter we propose a lightweight approach to determine the periodicity of a link quality fluctuation in the mean-square sense and experimentally demonstrate how it can be used to compute the number of packets that can be successfully transmitted in burst.

Whereas link quality estimation has been studied in the past in different contexts, to the best of our knowledge ours is the first to determine periodicity and to use the result for computing an optimal burst size. Our approach can also be useful for determining optimal duty cycles, though the focus of this work is not on duty cycle.

The rest of this chapter is organized as follows: In Section II, we describe related work on link quality estimation and on measurement and analysis of burstiness. In Section III, we present experimentally obtained data and analyze them to identify the relevant parameters that can help us identify periodicity in link quality fluctuation. In Section IV, we introduce our approach to determine periodicity. In Section V we provide quantitative results and evaluate their implication. Finally in Section VI, we provide concluding remarks and outline future work.

5.1 APPROACH

The contention-based MAC protocols in wireless sensor networks are designed by taking the uniqueness and limitations of the networks into account [DP10]. For example, most of them avoid the use of control packets (RTS and CTS) by assuming that collision is a rare occurrence, because packets are generated and transmitted infrequently (if collisions occur, then packets are retransmitted). Similarly, they define duty cycles for nodes to sleep much of the time. Nevertheless, these protocols also force nodes to contend for the medium for each packet they transmit. As long as the assumption concerning the packet generation and transmission rates holds, contention for each packet is acceptable, but when the assumption is no longer valid, the throughput of these protocols becomes a significant bottleneck.

More recently, a new batch of MAC protocols has been proposed to enable bulk data transfer, and, thereby, achieve high throughput [DÖD11][KFD+07]. The idea is to enable nodes transmit multiple packets in burst once they have won a medium. These protocols disregard fairness because they assume that a wireless sensor network belongs to a single application and a node that has interesting data should have priority. Even when data have to be gathered from each node with equal proportion, burst transmission avoids aimless contention and enables nodes to sleep longer.

One essential question that has not been sufficiently addressed concerning burst transmission is determining the size of a burst. Addressing this question is important because burst transmission cannot go on endlessly. Secondly, contending nodes should estimate how long a burst transmission lasts, so that they can attempt to win the medium at the right time. Thirdly, the efficiency of burst transmission depends on how the quality of a link fluctuates. The longer the transmission ends, the more likely the quality of a link fluctuates, which means the probability of unsuccessfully transmitting packets becomes high and, hence, the cost (both delay and energy) of retransmission becomes high as well.

In this chapter, we aim to determine the appropriate size of a burst by taking the statistics of link quality fluctuation into consideration. We identify stable regions during packet transmissions and describe the durations of these regions using a probability distribution function (PDF). Once the PDF of a given region is known, then it is possible to determine the expected duration of this region. The objective is to tailor the burst size to the duration of a region which most likely characterize a link.

In order to investigate how the quality of a link fluctuates and to identify the appropriate metrics that can describe the quality of a link, we deployed IMote2 sensor platforms (which integrate the CC2420 radio) in different locations (both outdoors and indoors) and transmit-

Table 5.1: Summary of the experiment set up for characterizing the fluctuation of link quality.

parameter	**value**
Environment	indoor, outdoors
Burst transmission	5000, 30,000
Overall packets	70,000
Inter-packet interval	20 ms, 100 ms
Transmission power	-10 dBm (outdoors), 0 dBm (indoors)
Packet size	28 Byte

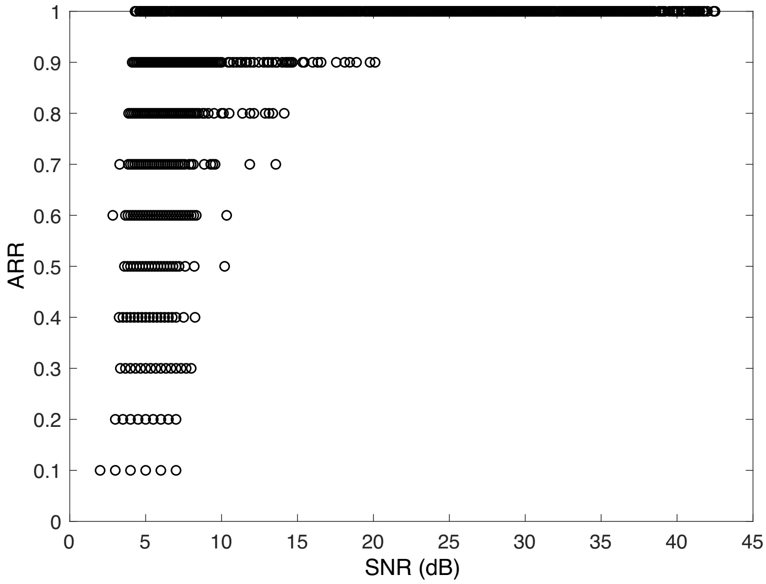


Figure 5.1: A summary of the relationship between the SNR and ARR of a wireless link.

ted packets continuously. We considered different distances between a transmitter and a receiver as well as different transmission power levels. Table 5.1 summarizes some of the parameters we included in our experiment set up. Altogether we transmitted 70,000 packets. For management reasons, we inserted a 20 ms (during the transmission of 30,000 packets) and a 100 ms (during the transmission of 5000 packets) inter-packet transmission interval during transmission.

For a 0 dBm transmission power, we varied the distance between the transmitter and the receiver in 2 m interval from 1 m to 35 m, until the link was totally disconnected. For a -10 dBm transmission power, we varied the separation distance from 5 to 17 m in intervals of 2 and 5 m. No lost packet was retransmitted. A packet transmission was considered successful when the transmitter received an ACK packet. Otherwise it was marked as failed. From the successfully received ACK packets, we estimated the Acknowledgment Reception Rate (AAR) [SDTL10].

We selected ARR for characterizing the quality of a link and signal-to-noise ratio (SNR) for characterizing the quality of received packets. Unlike the RSSI, the SNR contains information pertaining both the received signal's strength and the background noise. Then we evaluated how ARR and SNR are related.

Regardless of the location of the nodes and the distance of separation between them, packets were always received ($ARR \approx 1$) when the SNR was greater than 21 dBm. We characterized this link as a good link, in agreement with previous observations made by other researchers. On the other hand, when the SNR was less than 2 dBm, the ARR was less than 0.1, describing a bad link where 90% of the packets were lost. The region between the good and the bad links describe an intermediate region in which ARR varies uniformly between 0.1 and 0.9. The links in this region are bursty in nature. Our experimental observations are similar to previous findings [BKM+12] [SDTL10], even though they used differet platforms (TelosB and Micaz). Figure 5.1 summarizes the relationship between ARR and SNR for our experiment. Figure 5.2 displays the three regions we identified to describe a bad, an intermediate, and a good link and how the SNR and ARR fluctuate in these regions.

5.2 LINK QUALITY ESTIMATION

We use the conditional probability distribution function to describe the duration for which the quality of a link can be considered stable, i.e., all packets transmitted within this duration most likely experience a similar link quality. If this PDF is available to the MAC protocol or the application, it can determine the number of packets it should transmit successively, how often it should contend to seize the medium, or how long on average it should spend in sleep mode.

5.2.1 THEORETICAL CONDITIONAL PDF

Suppose the fluctuation of SNR of received ACK packets for a particular link is expressed as a random variable $\mathbf{s}$ with a PDF $F(s) = P\{\mathbf{s} \leq s\}$, where s is a real number. The conditional PDF of the duration in which the link can be considered stable[1] can be expressed as:

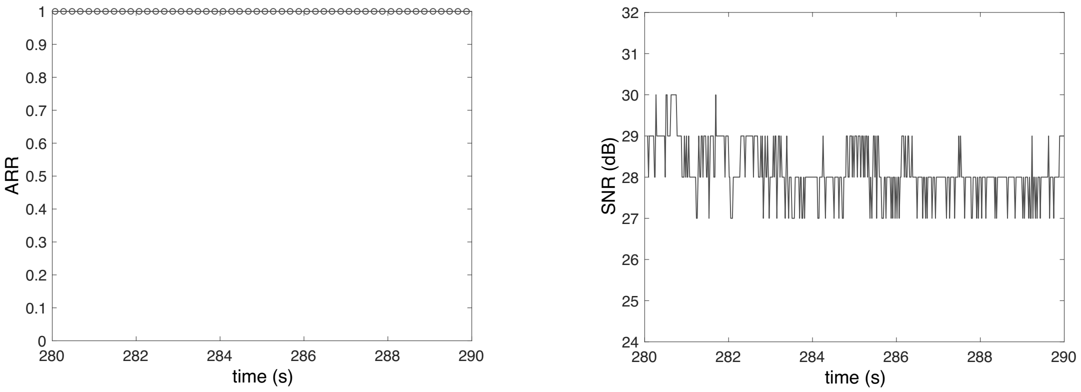
$$F\left(t|s_{th}\right) = P\left\{\mathbf{t} \leq t|\mathbf{s} \geq s_{th}\right\} \tag{5.3}$$

Which can also be expressed as,

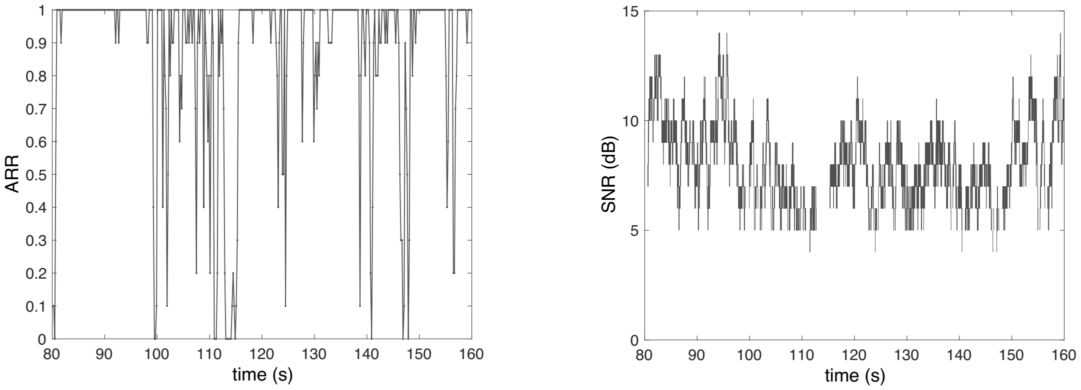
$$F\left(t|s_{th}\right) = \frac{P\left\{\mathbf{s} \geq s_{th}|t\right\}}{\int_0^\infty f(s|t)ds} \tag{5.4}$$

---

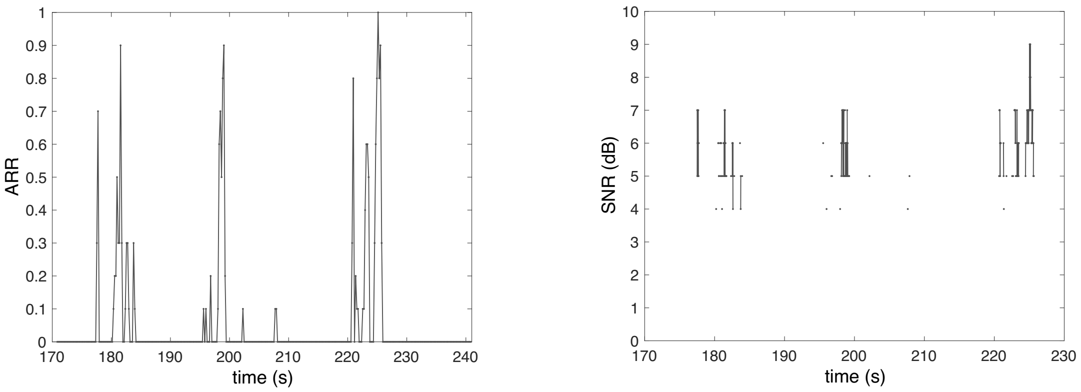
[1] It should be noted that stable does not imply good. It simply mean that the quality of the link in this duration can be considered unchanging.



(a) good link



(b) intermediate link



(c) bad link

Figure 5.2: An illustration of the three link types. In the good link *ARR* $\approx$ 1 all the time. The intermediate link is characterized as 0.1 $\leq$ ARR $\leq$ 0.9. In the bad link, ARR < 0.1. ARR = 1 means all packets were received successfully whereas *ARR* $\approx$ 0 means nearly all packets were lost.

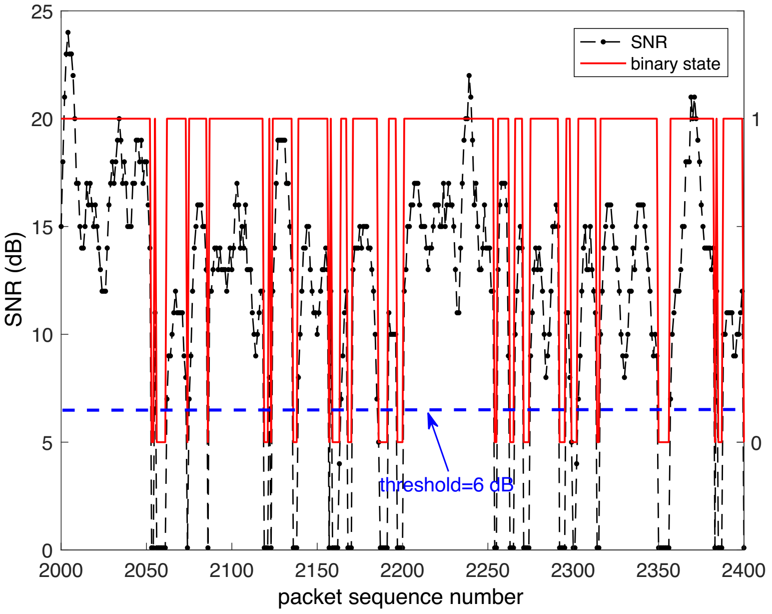


Figure 5.3: The fluctuation of SNR in received ACK packets and the transformation of the continuous function to a discrete function to estimate the conditional duration of a stable condition.

where $f(s|t)$ is the conditional probability density function of **s** given t. The expected duration in which the link quality is above the specified threshold can be expressed as:

$$E\left[\mathrm{t}|s_{th}\right] = \int_0^{\infty} \left[1 - F\left(t|s_{th}\right)\right] dt \qquad (5.5)$$

The number of packet which should be transmitted in burst can be determined by taking Equation 5.5 along with the packet size (which is 28 Bytes in a TinyOS environment), the transceiver's data rate (250 Kbps for CC2420), and the MAC protocol primitives (for IEEE 802.15.4 compliant MAC protocols these are CCA, exponential random back-off, and SIFS) into consideration.

5.2.2 EMPIRICAL CONDITIONAL PDF

Equation 5.5 can be determined empirically for each link within a network. We shall demonstrate this approach by example. The CC2420 transceiver can decode a packet correctly only when the packet error rate (PER) is less than one percent. According to the IEEE 802.15.4 specification, a typical low-cost detector implementation is expected to meet the 1% PER requirement for SNR values of 5-6 dB [80203]. Therefore, we choose 6 dB as our first threshold. However, because most real world links are in the intermediate region, we also considered a threshold of $SNR > 10\ dB$.

Figure 5.3 displays a snapshot of the fluctuation of the SNR of acknowledgment packets

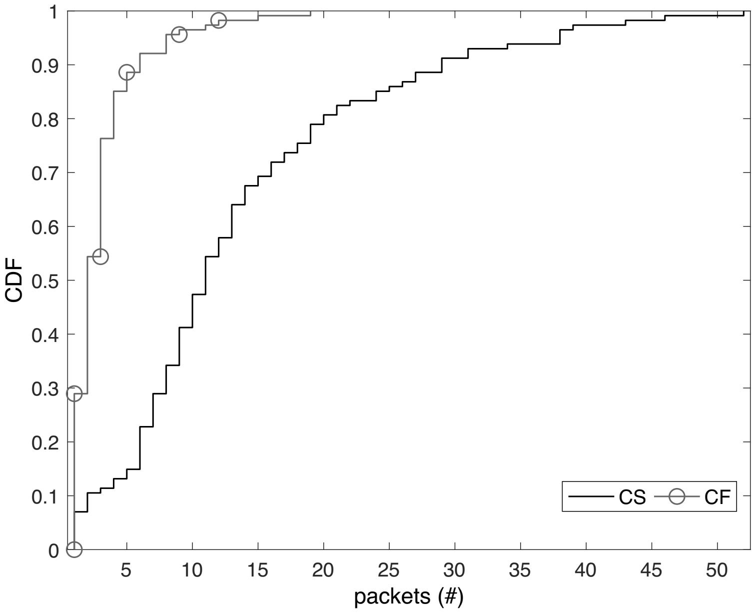


Figure 5.4: The empirical conditional distribution function of consecutive success and failure of a link.

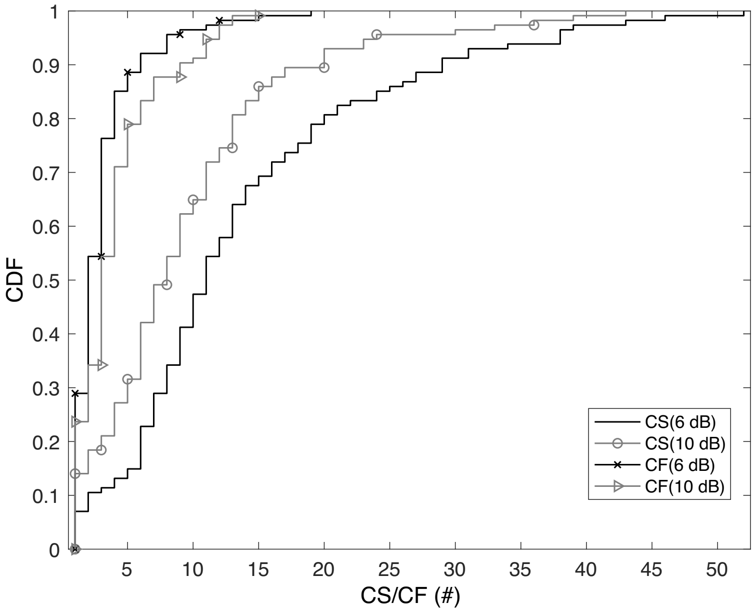


Figure 5.5: Empirical conditional PDF of consecutive success (CS) and failure (CF) for different SNR thresholds.

during the continuous transmission of 30,000 packets in an outdoor location. The distance between the two communicating nodes was 5 m and the transmission power of both nodes was -10 dBm. In order to determine the durations in which the link quality stays above 6 dB

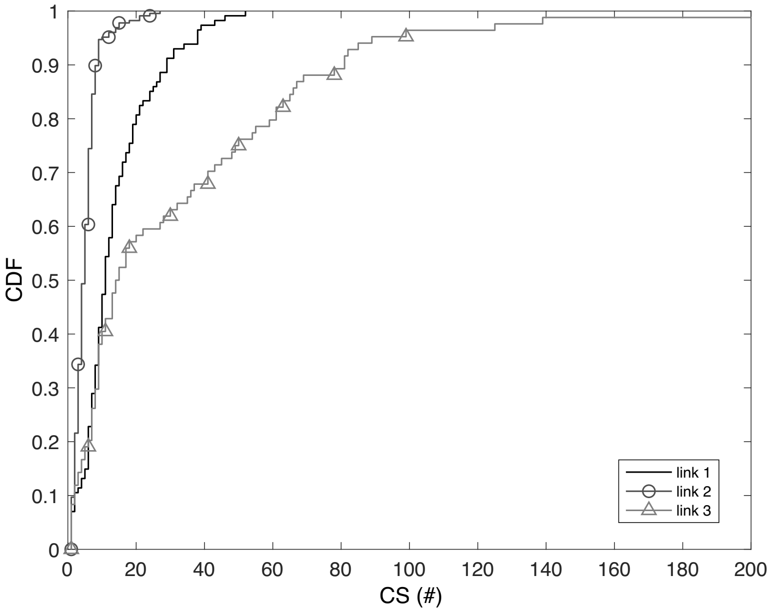


Figure 5.6: The conditional PDF of consecutive success for different links.

continuously, we transformed the continuous function to a discrete function by setting 6 dB as the threshold:

$$f(t) = \begin{cases} 1 & \text{if SNR} \geq 6\ dB \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

The discrete function can be understood as a function of time since the packets are transmitted consecutively and the width of each pulse in the function can be understood as the time duration in which the channel behavior can be considered as stable because all the packets transmitted within this duration are either received or lost with the same probability. By measuring the width of each pulse which are above the threshold, the conditional probability distribution function of the time duration for successfully transmitting packets in succession (in other words Equation 5.4) can be obtained. Conversely, the conditional PDF of the duration in which successive packets fail can be obtained by measuring the width of each pulse below the threshold.

Figure 5.4 displays the conditional PDFs of continuous success and continuous failure of the link described above. The SNR threshold was 6 dB. Figure 5.5 shows how the conditional PDF of continuous success changes for different SNR thresholds. In general, as the SNR threshold increases, the probability of receiving packets successfully increases, but the probability of getting a stable link decreases. Figure 5.6 compares the conditional PDF of continuous success for different links with the same SNR threshold.

Table 5.2: A summary of the parameters used to transmit packets in burst in different links for $SNR = 6\ dB$ as a threshold).

	link1	link2	link3	link4	link5
CS	29	2	2	8	6
CF	12	13	6	7	3
location	9 m	27 m	35 m	13 m	3 m
power	0 dBm	0 dBm	0 dBm	-3 dBm	-10 dBm

5.3 EVALUATION

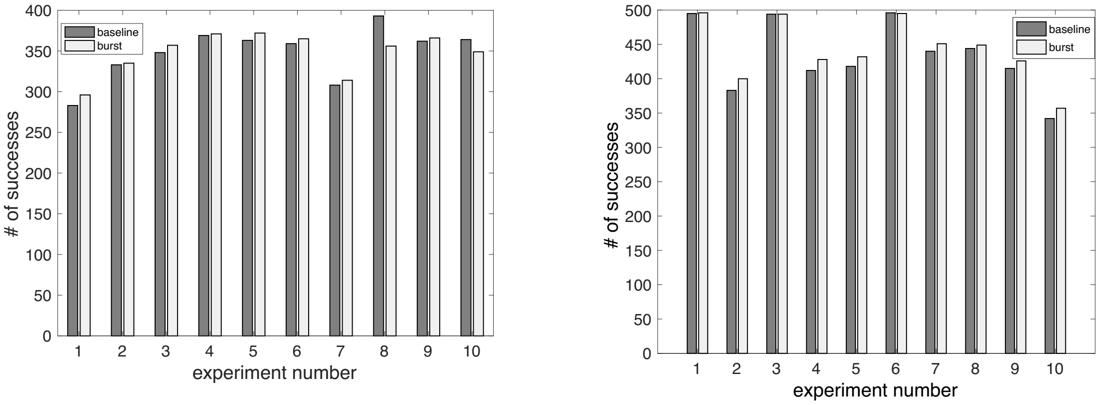
To evaluate the usefulness of Equation 5.5, we first transmitted 30,000 packets continuously in each link in order to obtain statistics pertaining to the SNR fluctuation of received ACK packets. There was a 20 ms interval between transmissions to annotate the received packets and to store the metrics we needed to characterize the packets (RSSI, LQI, and SNR). After the transmission was completed, we obtained the empirical PDF of the durations for continuous success (CS) and continuous failure (CF). We fixed the SNR threshold at 6 dB.

During the test phase, we transmitted 400, 500, 1000, 2000, 3000, and 10000 packets, but this time the packets were transmitted with and without intermission. For the case of with intermission, we used the expected duration of continuous success to transmit the packets in burst and the expected duration of continuous failure to make the nodes refrain from transmitting. In all the experiments, lost packets were not retransmitted. For each test case, we repeated the experiment ten times. Figure 5.7 compares the number of successfully transmitted packets when packet transmission was made without intermission and when packet transmission was made with the knowledge of the conditional PDF of CS and CF. As can be seen in the Figure 5.7, our approach yields (90% of the time) the highest number of successfully transmitted packets for most of the test cases. This is particularly the case as the number of transmitted packets increased. When the number of transmitted packets increased, so did the transmission time, in which case the link characteristic was better represented by the statistics we obtained by transmitting the 30,000 packets.

Figure 5.8 compares the average number of successfully transmitted packets for five different indoor and outdoor links. This time we transmitted 1000 packets for testing. Table 5.2 summarizes the parameters we computed or fixed for the experiment.

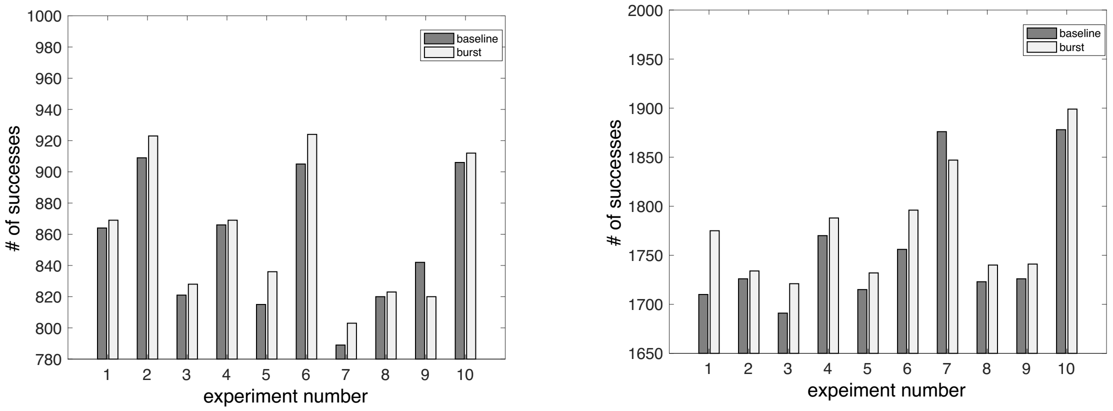
5.4 CONCLUSION

In this chapter we investigated fluctuations of link quality in wireless sensor networks and proposed a model to estimate the expected duration of stable transmission periods. We employed conditional probability distribution functions in our model where the link quality duration was conditioned by the signal-to-noise ratio thresholds of acknowledgment packets.



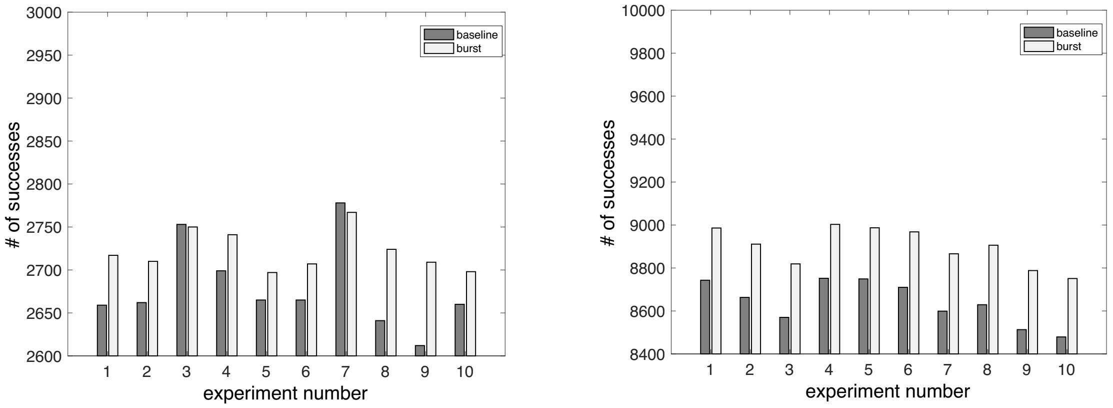
(a) 400 packets

(b) 500 packets



(c) 1000 packets

(d) 2000 packets



(e) 3000 packets

(f) 10000 packets

Figure 5.7: Comparison of the successfully transmitted packets when they were transmitted continuously without intermission and then they were transmitted by taking the expected durations of continuous success and continuous failure in to account.

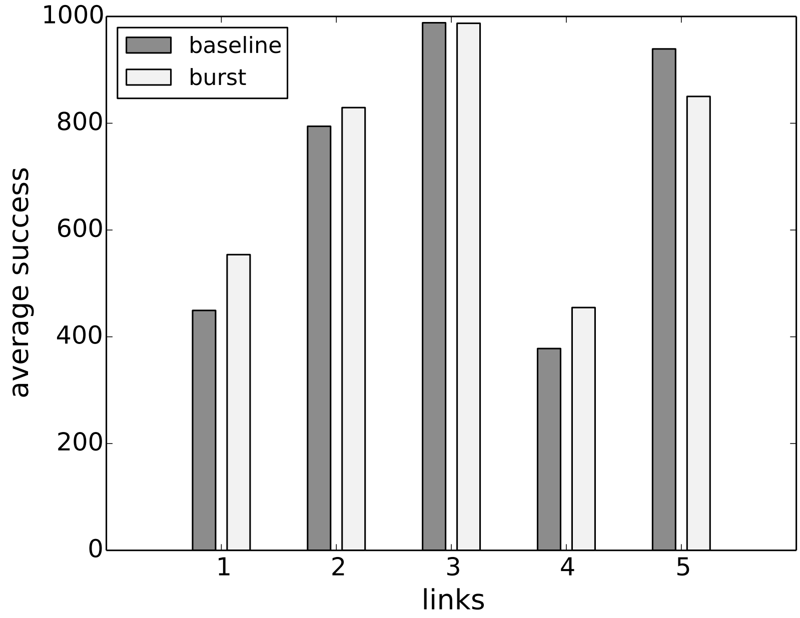


Figure 5.8: Comparison of the successfully transmitted packets in burst in different indoor and outdoor links.

We used the model to determine the number of packets that can be successfully transmitted in burst. In other words, nodes transmit packets in burst when the link quality is good but they refrain from transmitting packets when it is bad. Our model enables them to determine for how long on average the quality of a link remains good and for how long it remains bad. We deployed IMote2 nodes in various places and considered different separation distances and transmission power levels to obtain statistics pertaining to link quality.

The experiment results confirm the plausibility of our approach. We compared our approach with a transmission scheme that transmitted packets in succession without taking link quality fluctuations into account. Altogether we transmitted 70,000 packets to obtain statistics and 16,900 packets to evaluate the performance of our approach. 90% of the time, our approach outperformed in transmitting packets successfully.

In future, we shall continue improving the model, so that it can dynamically update the statistics pertaining to link quality fluctuations. One aspect we shall consider is using Bayesian Estimation Techniques [PP02] to update the conditional distribution function.

6 A HANDOVER TRIGGERING ALGORITHM FOR MANAGING MOBILITY IN WSNS

As we have already motivated in the introduction chapter, the combination of BAN and PAN enables a flexible and reliable sensing. Since the PAN consists of stationary relay nodes, they can be strategically placed to ensure that connectivity is preserved within the mobility range of the user. As the relay nodes can be placed such that the communication link between any of them and the base station is stable, it is sufficient to address the challenges surrounding a single-hop link connecting the BAN and the PAN gateway. Moreover, maintaining a stable link during mobility is formidable challenging. In order to highlight this challenge, we refer to Figure 6.1. We placed two stationary relay nodes in the foyer of our faculty, separated from one another by a distance of 30 m. There were no objects between these nodes to obstruct communication. A mobile robot carrying a transmitting node moves at a speed of approximately 0.13 m/s from one of the nodes to the other in a straight line whilst the transmitter continuously transmitted packets to both nodes simultaneously (4800 packets in all). We plotted the received signal strength indicator (RSSI) of ACK packets as a function of distance.

As can be seen in Figure 6.1, the signal strength of the received packets fluctuated considerably for both receiving nodes, regardless of the relative distance of the robot from the receivers. Another interesting aspect we observed in this experiment was that some packets were lost even though the received signal strength of neighboring packets indicated that they should have been successfully received. By the same token, some packets were successfully received even though their RSSI was too small. In order to deal with these challenges (to increase the reliability and throughput of mobile, wireless links), we propose a seamless handover. Unlike the handover strategies applicable for cellular networks, however, our proposed strategy does not rely on resource-rich base stations which determine when and how a mobile node should transfer communication. Instead, the mobile node itself, by examining the

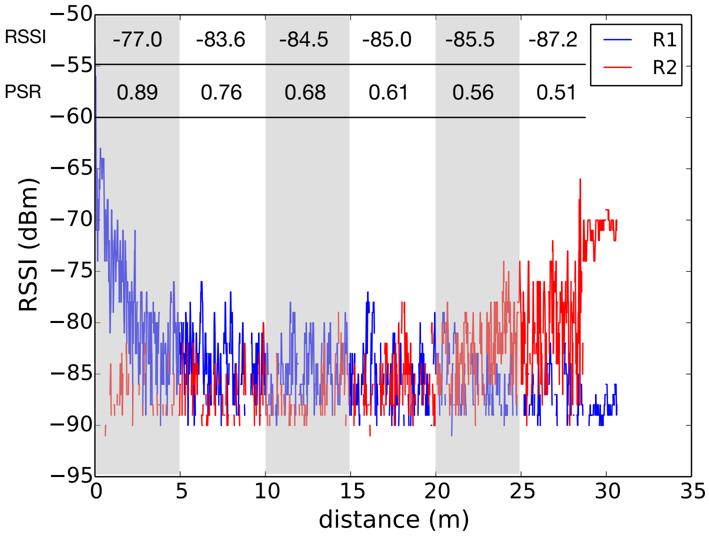


Figure 6.1: The fluctuation of link quality during mobility and the associated packet success rate in different regions.

fluctuation of the RSSI values of incoming acknowledgment packets and the packet success rate, seamlessly transfers a communication from one relay node to another without the need to first disconnect an existing communication. For this reason, it is vital for a mobile node to (1) determine whether a fluctuation in link quality eventually results in a disconnection, (2) foresee potential disconnection well ahead of time and establish an alternative link before the disconnection occurs, and (3) seamlessly transfer communication to the new link. In this chapter, we address (1) and (2).

6.1 BACKGROUND

In this section, we first present the fundamental background on bulk data transfer and discuss why it is necessary in mobile wireless sensor networks (MWSNs). Then, we introduces handover triggering algorithm in mobility management protocol in MWSNs.

6.1.1 BULK DATA TRANSFER

The existing IEEE 802.15.4 standard, which is widely used in wireless sensor networks, is not optimal for applications requiring a high throughput. Firstly, the bandwidth it supports is small (it can support a maximum of 250 kbit/s). Secondly, the packet-by-packet contention scheme (the need for clear channel assessment, random back-off, and the transmission of control packets, RTS and CTS, for every data packet) when there are a few or even no contending nodes (which is typically the case in a home environment), is wasteful. Thirdly, the statistics which can be established based on the evaluation of the RSSI values of incoming packets may not be reliable, as the relative interval between the received packets can be long and their correlation can be weak. All these limitations can be overcome or significantly reduced, if mobile nodes are permitted to transmit packets in burst once they seized the medium. The number of packets they can transmit in succession can be determined from the expected stable durations (in a statistical sense) of established links.

6.1.2 HANDOVER TRIGGERING ALGORITHM IN MOBILE WIRELESS SENSOR NETWORKS

The first step of dealing with mobility in wireless sensor networks is to monitor the link quality of current link between mobile node and its current relay node. If the link quality meets the predefined criterion, a handover trigger event is issued, we call this process handover trigger algorithm (HTA). According to the link quality metrics they used, the HTA can be classified into three categories: ACK-based, RSSI-based, and hybrid.

ACK BASED

For the ACK based algorithms, the idea is simple and intuitive. It requires no complex link quality process but only use MAC layer metric *ACK*. The most intuitive handover trigger method is using one packet failure as the trigger criterion [PKG+15][KMN13]. However, this method may lead to ping-pong handover[1] due to the high dynamic behavior of the links, especially when the nodes are mobile. To deal with this problem, the author [ZVC13] proposed a new handover trigger algorithm named Link Loss which combines two metrics together, burst loss (consecutive transmission failure) and packet failure rate (PFR), both metrics are calculated via *ACK*. Once either of the conditions is matched, PFR is larger than a predefined threshold and/or there are n consecutive transmission failure occurred, the handover procedure is initiated.

RECEIVE SIGNAL STRENGTH BASED

Several algorithms have been proposed by using RSSI value as the handover trigger algorithm, such as Smart-Hop [FZA+12], MobiSense [GLJ11], MX-MAC [ZD10] etc. They either use single threshold or double thresholds as the trigger criterion. If the RSSI value with current access point is detected below a predefined threshold, the handover procedure is initiated and until the mobile node find a new access point which has a better link quality, the handover procedure then finished. Using a single threshold, the mobile node may handover to an unreliable link and trigger a handover procedure again soon. To solve this problem, a hysteresis margin is added to enhance the selection of new access point.

---

[1] consecutive handover happens between two access points alternatively in a quite short period of time

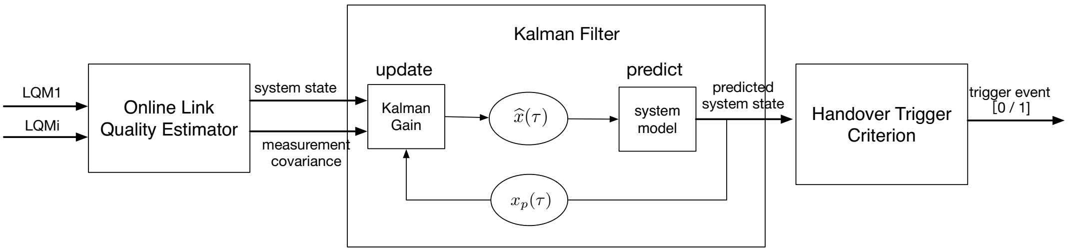


Figure 6.2: Simplified block diagram of KMF.

HYBRID SOLUTIONS.

Unlike the previous two groups of solutions, the hybrid ones combine multiple metrics together to trigger the handover. The most popular algorithm used is fuzzy logic based method. In [FAKB10], the author proposed a handover procedure by using F-LQE [BKY⁺10], a fuzzy logic based link quality estimator. It takes packet success rate, link asymmetry, stability and signal-to-noise-ratio (SNR) into consideration and combines the other three metrics (energy, traffic load and depth level) together to deal with the handover. However, this procedure is not implemented and tested. By carefully studying the characteristics of link quality in industrial environment, Zinonos et al. [ZCV14] proposed new handover trigger algorithm named fuzzy logic based mobility controller (FLMC). It only takes two metrics, RSSI value and link loss, as the inputs of the fuzzy inference system. If the output, trigger decision probability, is below a predefined threshold, the handover procedure is initiated.

6.2 MODELING

A quantifiable cost can be associated with every packet transmission a mobile node makes if its communication setting are known at least in a probabilistic sense. This cost can in turn be used to determine the most suitable transmission scheme. If, for example, the transmission should take place in a highly contentious setting, the MAC protocol can elect to turn on the collision avoidance mechanism. If, on the other hand, the medium is less contentious, the collision avoidance mechanism can be turned off because the packet retransmission cost (in case of collision) may be less than the transmission cost of RTS and CTS control packets (which introduce both latency and energy penalty). Similarly, if the cost of a handover is less than the retransmission cost, the mobile node can elect to search for an alternative link and transfer communication to it. The penalty it has to accept is the cost of predicting the link quality and neighbor discovery. Figure. 6.2 shows the architecture of our proposed handover triggering algorithm.

6.2.1 HANDOVER TRIGGERING CRITERION

Suppose a mobile node has n number of packets to transmit in succession and the expected packet success rate is psr, the retransmission cost, c_{re} (seen only from the mobile node's perspective), can be expressed as:

$$c_{re} = n\,(c_{tx} + c_{rx})\,(1 - psr) \tag{6.1}$$

where c_{tx} and c_{rx} are the transmission and ACK reception cost for a single packet. Similarly, if the node has to communicate with k number of neighbor nodes by sending them m number of packets in order to determine which of them can be the best rely node, the cost it incurs for neighbor discovery, c_s, can be expressed as:

$$c_s = mkc_{rx} \tag{6.2}$$

where c_{rx} is the cost of receiving a single ACK packet from a neighbor. From Equations 6.1 and 6.2, it is clear that a handover is a better option when the quality of a link deteriorates and the packet loss becomes considerably high. In other words, a handover is preferred when:

$$c_{re} > c_s \tag{6.3}$$

As can be seen, we have expressed the handover condition in a generic sense. The costs may refer to energy, latency, or some other criterion which is important for the application or the user. By substituting equation 6.1 and 6.2 into 6.3, we can get:

$$psr < 1 - \frac{mkc_{rx}}{n(c_{tx} + c_{rx})} \tag{6.4}$$

So the handover criterion is simplified to equation 6.4.

6.2.2 PACKET SUCCESS RATE

In order to evaluate the stability of a link, some aspects of the received packets can be evaluated. In the literature, parameters such as RSSI, SNR, and LQI are used. Nevertheless, these parameters exhibit considerable fluctuations even when packets are successfully received. Similarly, some packets are lost even though these parameters, taken from preceding and succeeding packets, indicate that the link was stable. In order to complement these parameter, the packet success rate (psr) can be considered in addition for the evaluation of link quality and link stability. The psr is computed as follows: If n packets are transmitted in succession and m of them are acknowledged, then psr is calculated by dividing m by n. The psr can be expressed as a function of time by using a moving average of n packets transmitted up to the time t. Figure 6.1 displays the psr of our robot for every 5 m distance it covered.

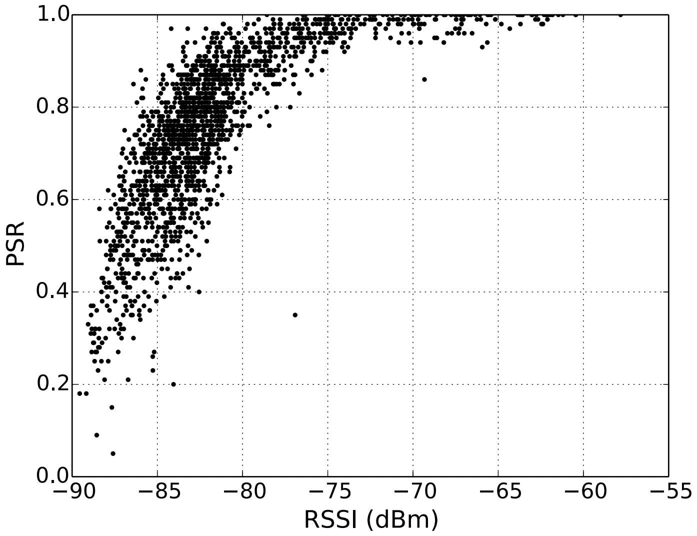


Figure 6.3: Packet success rate and RSSI value. *psr* is calculated for every 100 packets.

6.2.3 PREDICTION WITH KALMAN FILTER

The generic handover triggering condition we specified in Equation 6.3 implicitly requires the packet success rate. The packet success rate, in turn, is a function of the RSSI values of received packets, but it is impossible to establish a one-to-one relationship between RSSI and psr. Figure 6.3 displays the relationship we have established for the CC2420 radio chip after transmitting 450,000 packets in different locations, both indoor and outdoor.

In general, a handover triggering algorithm should deal with three sources of uncertainties: (1) the erratic fluctuation of RSSI values, (2) the uncertainty associated with the relationship between RSSI and psr, and (3) the error associated with predicting the RSSI and psr values of the future, so that a handover can be initiated in a timely fashion.

In order to deal with these uncertainties, we divided packet transmission time into epochs. The average RSSI value of the ACK packets received within an epoch serves as the RSSI value of that epoch. The RSSI and psr values of the past *n* epochs can be used to determine whether the deterioration of a link quality is a steady phenomenon and therefore a seamless handover should take place in the next epoch.

One of the advantages of dividing time into epochs is that the effect of the three types of uncertainties can be minimized in a systematic way. Specifically, the RSSI and psr values of successive epochs can be regarded as correlated with one another. However, it is impossible to express the RSSI or psr value of epoch τ in terms of past values in a deterministic sense, as they are subject to random fluctuations (we label this error as a process error). Secondly, even if averaging the RSSI values of a single epoch minimizes the error associated with the actual RSSI values of the received packets in the epoch, still this estimation contains error, which we regard as a measurement error. The Kalman filter can be employed to combine

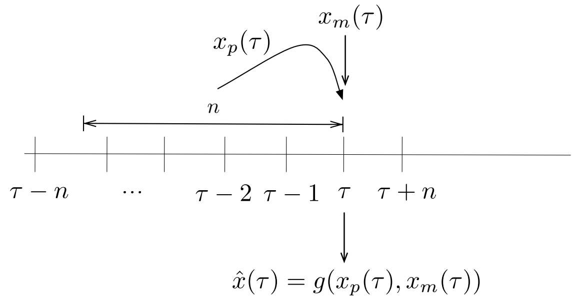


Figure 6.4: The steps of estimation to support a seamless handover.

prediction and measurement values in order to minimize the three sources of uncertainties.

In order to explain our approach, we refer to Figure 6.4. Suppose the parameters we wish to estimate at epoch τ can be represented by the generic random variable $x(\tau)$. The reason we describe it as a random variable is that we will never be able to obtain its real value at any given time, owing to the fact that it is subject to the three types of uncertainties. Suppose, at time epoch $\tau - 1$, based on the statistics we have up to that time, we predict the value of x for the epoch τ and label it as $x_p(\tau)$. The index p stands for prediction. At time epoch τ, however, we measure x and label this as $x_m(\tau)$. Both $x_p(\tau)$ and $x_m(\tau)$ contain the actual value of x for that epoch, but each contains a different kind of error. Using the Kalman formalism, we can estimate $x(\tau)$ by properly combining the evidence coming from the two sources:

$$\hat{x}(\tau) = x_p(\tau) + k(\tau)\left(x_p(\tau) - x_m(\tau)\right) \tag{6.5}$$

Note that:

$$x_m(\tau) = x(\tau) + v(\tau) \tag{6.6}$$

where $v(\tau)$ is the measurement error modeled as a random variable. Similarly,

$$x_p(\tau) = x(\tau) + w(\tau) \tag{6.7}$$

where $w(\tau)$ is the processor error modeled as a random variable. Hence, our goal should be finding the optimal k such that the difference between the actual $x(\tau)$ and its estimated value, $\hat{x}(\tau)$, is minimum. One way to achieve this goal is minimizing the mean square error:

$$e^2(\tau) = E\left\{\left[x(\tau) - \hat{x}(\tau)\right]^2\right\} \tag{6.8}$$

The value of k in Equation 6.5 which minimizes the mean square estimation error in Equation 6.8 is expressed as [Dar16]:

$$k(\tau) = P_p(\tau)\left[P_p(\tau) + R(\tau)\right]^{-1} \tag{6.9}$$

where $P_p(\tau)$ is the prediction error covariance, i.e.,

$$E\left\{\left[x(\tau) - x_p(\tau)\right]\left[x(\tau) - x_p(\tau)\right]\right\}$$

which can be expressed as:

$$P_p(\tau) = P(\tau - 1) + Q(\tau) \tag{6.10}$$

where $Q(\tau)$ is the process error covariance (to be defined shortly). Finally, $R(\tau)$ is the measurement error covariance, i.e.,

$$E\left\{\left[x(\tau) - x_m(\tau)\right]\left[x(\tau) - x_m(\tau)\right]\right\}$$

respectively, for epoch τ:

In order to take both the fluctuation in RSSI values and the psr of received ACK packets, we represent $x(\tau)$ as a vector quantity:

$$x(\tau) = \left[r(\tau), \quad psr(\tau)\right]^T \tag{6.11}$$

where $r(\tau)$ is the RSSI value and $psr(\tau)$ the packet success rate of the epoch τ. Compared to the packet transmission rate, the speed of the mobile node is very small (typically a human movement is below 5 km/h). Hence, for a very short time (500 ms to 1 s), the change in the RSSI values of received ACK packets can be approximated as a linear function of time:

$$r(\tau) = a\tau + b \tag{6.12}$$

from which we have: $r(\tau) = r(\tau - 1) + a$. Moreover, compared to the fluctuation in RSSI values, the change in psr between consecutive epochs is imperceptible. Hence, it is plausible to assume that $psr(\tau) = psr(\tau - 1)$. Putting together these two assumption yields:

$$\begin{bmatrix} r(\tau) \\ psr(\tau) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r(\tau - 1) \\ psr(\tau - 1) \end{bmatrix} + \begin{bmatrix} a \\ 0 \end{bmatrix} \tag{6.13}$$

where a and b are parameters which can be determined by a linear regression [NKNW96] and are associated with the covariance between the RSSI values and the time epochs. The two coefficients are determined by minimizing the error between the actual and estimated RSSI values using the mean square error estimation. The error associated with our assumption as regards $r(\tau)$ (the linear approximation) and $psr(\tau)$ (the assumption that $psr(\tau) = psr(\tau - 1)$) can be described as the process error, $Q(\tau)$. The process error in $r(\tau)$ can be described by the variance of the past n RSSI values:

$$\sigma^2_{r,p}(\tau) = \frac{1}{n-1} \sum_{k=(\tau-n-1)}^{\tau-1} \left(r(k) - \bar{r}\right)^2 \tag{6.14}$$

Likewise, the process error as regards the psr can be expressed as:

$$\sigma^2_{psr,p}(\tau) = \frac{1}{n-1} \sum_{k=(\tau-n-1)}^{\tau-1} \left(p_s(k) - \overline{p}_s\right)^2 \tag{6.15}$$

The process error of the vector $x(\tau)$ expressed as a matrix is:

$$Q(\tau) = \begin{bmatrix} \sigma^2_{r,p}(\tau) & 0 \\ 0 & \sigma^2_{psr,p}(\tau) \end{bmatrix} \tag{6.16}$$

where we assumed that $\sigma^2_{psr,p}(\tau)$ and $\sigma^2_{r,p}(\tau)$ are uncorrelated. The error associated with the measurement of the actual values of RSSI and the psr for a specific epoch can be determined by taking the variances and covariances of the two random variables for that epoch. Consequently:

$$\sigma^2_{r,m}(\tau) = \frac{1}{m-1} \sum_{i=1}^{m} \left(r_i^\tau - \overline{r}^\tau\right)^2 \tag{6.17}$$

where $\overline{r}^\tau$ is the mean RSSI value for the time epoch τ. In a single epoch, we have a single *psr* value, since *psr* is an average quantity. In order to compute the associated measurement error, we have to take into account the fact that $\sigma^2_{r,m}$ and $\sigma^2_{psr,m}$ are related with one another. This relation is described by the correlation coefficient, from which the measurement error as regards *psr* can be determined[2]:

$$\sigma_{psr,m}(\tau) = \rho_{r,psr,m}(\tau)\sigma_{r,m}(\tau)\sigma_{psr|r,m}(\tau) \tag{6.18}$$

where $\sigma_{psr,m}$ corresponds with the error associated with the measured psr for the time slot τ, $\rho_{r,psr,m}$ is the correlation coefficient between the measured RSSI and *psr* for the time slot τ, and $\sigma_{psr|r,m}$ is the conditional error associated with the *psr* given RSSI. The quantities in the right term save $\sigma_{r,m}(\tau)$ are determined experimentally, using Figure 6.3. Figure 6.5 displays the conditional *psr* error as a function of the correlation coefficient and the measurement error associated with the RSSI of epoch τ. Finally, the measurement covariance error is expressed as:

$$R(\tau) = \begin{bmatrix} \sigma^2_{r,m}(\tau) & \sigma_{r,psr,m}(\tau) \\ \sigma_{r,psr,m}(\tau) & \sigma^2_{psr|r,m}(\tau) \end{bmatrix} \tag{6.19}$$

With $Q(\tau)$ and $R(\tau)$, it is sufficient to compute the Kalman gain for each time epoch and with it, to predict the RSSI and the psr of the future $(\tau + 1)$ time epoch. Moreover, with the future

---

[2] The strength of correlation between two random variables, **X** and **Y**, can be expressed by the correlation coefficient:

$$\rho_{xy} = \frac{E\left[\mathbf{X} - \eta_x\right]\left[\mathbf{Y} - \eta_y\right]}{\sigma_x \sigma_y}$$

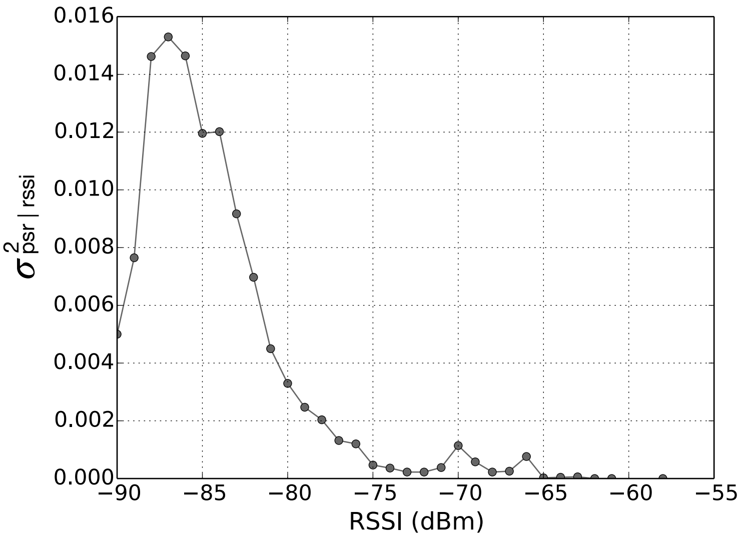


Figure 6.5: The conditional variance of *psr* given RSSI.

values predicted, it is possible to apply Equation 6.3 and determine whether a mobile node should trigger a handover at epoch τ so that in $\tau + 1$ it can switch to a new communication partner.

6.3 IMPLEMENTATION

We implemented our handover-triggering algorithm (KMF) and integrated it with the MX-MAC protocol [DW14]. It runs in a TinyOS runtime environment on the TelosB platform.

6.3.1 ONLINE LINK QUALITY ESTIMATOR

To estimate the current link quality within a window slot, intuitively, it has to store all the RSSI and ACK history data in the memory and calculate the summation and average it. Additionally, to calculate the variance, a large number of summation and multiplication operations are introduced. All these calculations are done at one time, it may take a long time to finish the computation, which may delay the next packet transmission. Even worse, when the window size increased, the memory consumption and execution time overhead is linearly proportional to the history size. In order to reduce the memory consumption and computation time, a cumulative moving average and variance algorithm is employed, as shown in Algorithm 1. The benefits are two folds, 1) the memory overhead is fixed no matter how large the window size, and 2) the execution time is amortized to each packet transmission.

Algorithm 1: Online link quality estimator

```
Input  : Link quality metrics: RSSI r, ACK
Output: averaged RSSI $r(\tau)$, variance $\sigma_r^2(\tau)$ and packet success rate $psr(\tau)$
if $counter >= window\_size$ then
    $counter \leftarrow 0$;
    $r(\tau) \leftarrow 0$;
    $\sigma\hat{}2_r(\tau) \leftarrow 0$;
    $psr \leftarrow 0$;
end
if $counter < window\_size$ then
    $counter \leftarrow counter + 1$;
    $\Delta_1 \leftarrow rssi - r(\tau)$;
    $r(\tau) \leftarrow r(\tau) + \Delta_1 / counter$;
    $\Delta_2 \leftarrow rssi - r(\tau)$;
    $m(\tau) \leftarrow m(\tau) + \Delta_1 * \Delta_2$;
    $\sigma_r^2(\tau) \leftarrow m(\tau) / counter$;
    $psr(\tau) \leftarrow psr(\tau) + (psr(\tau) + ack) / counter$;
end
return $r(\tau)$, $\sigma_r^2(\tau)$, $psr(\tau)$;
```

6.3.2 ADDITIONAL HANDOVER TRIGGERING ALGORITHMS

We also implemented four additional proposed handover-triggering algorithms to make an objective comparison. Some interface adaptations are made to integrate with the MX-MAC.

Single Packet Failure (SPF). The SFP trigger algorithm is used by ME-ContikiMAC [PKG+15] and MoX-MAC [KMN13]. It triggers a handover upon a single packet failure.

Link Loss (LL). This algorithm combines consecutive failure and packet failure rate to trigger a handover. Thus, if n packets continuously failed or the packet failure rate falls below a set threshold f within a specified duration, then it triggers a handover. The LL algorithm is first introduced by Zinonos et al. [ZVC13].

RSSI threshold based. The RSSI threshold based algorithm (or simply, RSSI) is the most popular handover trigger algorithm used in mobility management protocol, such as in Smart-Hop [FZA+12], MA-MAC [ZD10] and MobiSense [GLJ11]. It triggers a handover if the average RSSI value of successively received ACK packets drops below a pre-defined threshold. In our implementation, we follow the design in Smart-Hop.

MX-MAC default. It takes the RSSI values of present and future epochs into consideration in order to trigger a handover. It implements a normalized LMS filter for predicting the mean RSSI value of a future slot [DW14].

Table 6.1: The memory footprint and computational complexity of the different algorithms.

	LL	RSSI	MXMAC	KMF
ROM (bytes)	122	158	5498	4876
RAM (bytes)	8	12	118	98
execution time (ms)	-	-	16 ± 0.03	26 ± 0.05

6.3.3 MEMORY FOOTPRINT AND COMPUTATIONAL COMPLEXITY

Table 6.1 shows the memory footprint and computational complexity in terms of execution time for all algorithms we implemented. The additional memory overhead is mainly due to the implementation of the handover trigger algorithms and the link quality metrics cached in RAM. Since the simplicity of SPF algorithm, it almost costs no additional memory when integrated with MX-MAC. Thus, we use it as the baseline. For the current implementation of KMF, it utilizes 4876 more bytes in ROM and 98 more bytes in RAM. However, KMF consumes slightly less memory than the original design of MX-MAC. Moreover, compared to the overall available memory in TelesB platform (48 KB flash and 10 KB RAM), the additional memory overhead is not a big issue.

The computation overhead is mainly due to the execution of link quality estimation and prediction algorithm. Therefore, we programed TelesB node to execute each handover trigger algorithm for 10,000 times continuously at one run, and repeated 100 runs. Additionally, the execution time of link quality estimation depends on the window size, so we fixed this parameter to 10. Then we calculated the averaged value as the execution time. As discussed in section 6.2, the link quality estimator in KMF is executed for each packet transmission and the Kalman Filter runs every 10 packets. Totally these two computation take 26 ms for every 10 transmissions. Considering the minimum inter-packet-interval (IPI) 10 ms, the computation can be finished without introducing additional latency. From the energy consumption prospective, the computation consumes much more less energy than a single transmission. For example, the CC2420 radio consumes 17.4 mA at 0 dBm when in transmission mode [Ins13], while the MCU, MSP430, only consumes 2.64 mA in busy mode [Tex06]. To transmit one packet which is 128 bytes long (the maximum packet length), it consumes 0.24 mJ. And to do one calculation, it only takes 0.20 mJ.

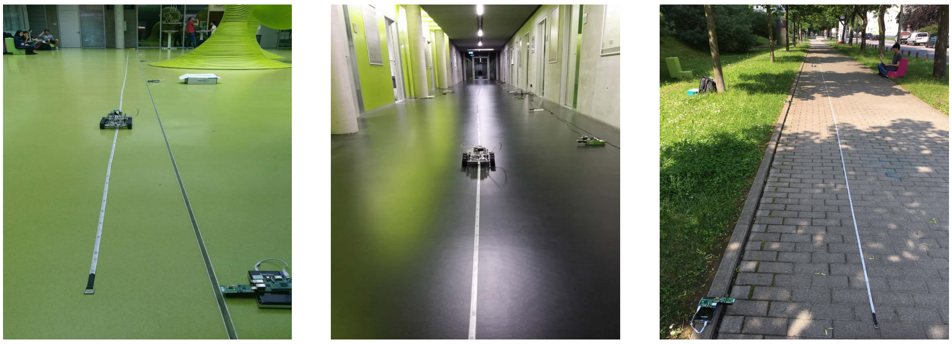
6.4 EVALUATION

In order to evaluate the performance of our algorithm, we conducted a series of experiments using the MobiLab testbed [WAD16a]. In our setup, the testbed consisted of 5 static TelosB nodes deployed in a straight line with a 5 m separating distance between them and a mobile node carried by a robot. We deployed the testbed in a lobby, a corridor, and outdoors (see Figure 6.6). The description of our experiment settings is summarized in Table 6.2. In order to draw a comparable conclusion for the other handover triggering algorithms, we first launched

Table 6.2: Parameters used for handover trigger experiments in different environment.

parameter	value
environment	lobby, corridor, outdoor
spacing between nodes	5 m
speed	0.13 m/s (constant)
motion pattern	straight line walk
duration	160 s
tx-power	-25 dBm, -10 dBm, 0 dBm
IPI	50 ms, 100 ms

a large number of preliminary experiments and carried out an in-depth analysis of the received packets. Our aim was to calibrate the parameters for each algorithm (the configuration parameters we obtained are listed in Table 6.4). Afterwards, we executed and repeated each experiment ten times. During each experiment, the robot was moving from one end of the deployment area to the other in a straight line, at a constant speed (approximately 0.13 m/s), whilst the transmitter carried by the robot transmitted packets in burst. As a result, in a single run of experiment, the characteristics of five distinct links could be evaluated (i.e., the communication link established by the robot with each relay node). The runtime characteristics of all sensor nodes and the robot were monitored by using the TFCP framework [WAD16b].



(a) (b) (c)

Figure 6.6: The deployment environments for our experiments: (a) Lobby. (b) Corridor. (c) Pathway.

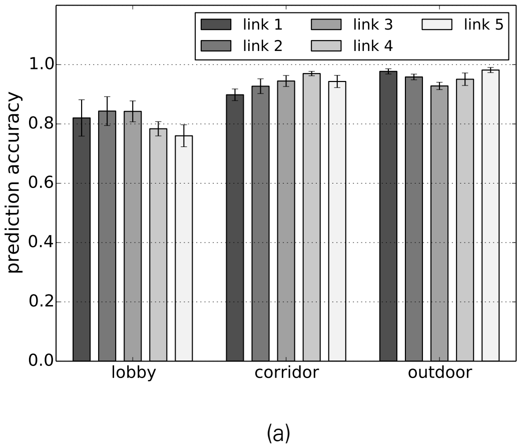
6.4.1 PREDICTION ACCURACY

One of the features upon which the performance of our approach depends is the prediction accuracy of the Kalman filter. This feature is important because the implementation of the Kalman Filter added complexity to our algorithm. Table 6.3 summarizes the *psr* for each link as the ratio of the total number of packets received to the total number of packets transmitted in a link. Figure 6.7 summarizes the prediction accuracy of the Kalman Filter for the different links, from which it can be seen that the prediction accuracy is above 0.8 (1 being the maximum)

Table 6.3: Average *psr* for different links.

environment	link1	link2	link3	link4	link5
lobby	0.67	0.75	0.72	0.70	0.60
corridor	0.87	0.91	0.92	0.95	0.90
outdoor	0.16	0.38	0.42	0.38	0.11

for most of the links. Only link 4 and link 5 in the lobby were less than 0.8. This is mainly due to the high false positive (as shown in Figure 6.7(c)).



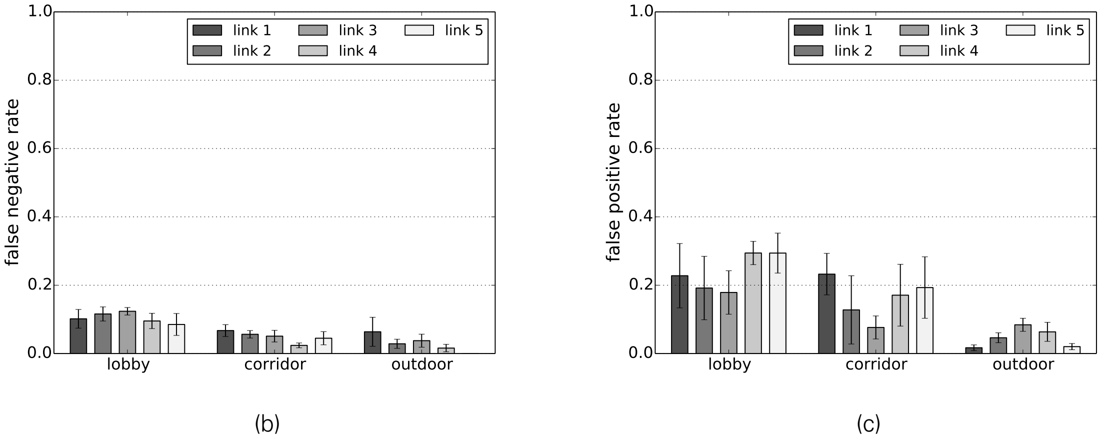


Figure 6.7: The prediction accuracy of Kalman filter based handover trigger algorithm applied to 5 different links in three environment: (b) and (c) show false negative and positive respectively

6.4.2 HANDOVER TRIGGER EVENT

A *handover trigger event* is generated when a handover triggering algorithm initiates a handover as a result of a "belief" by the former that a deterioration in the link quality leads to a

Table 6.4: Configuration parameters of the different algorithms.

Trigger Algorithm	Trigger Criterion	Slot Size
SPF	single packet failure	-
LL	CF = 2 or PFR > 0.2	10
RSSI	-80 dBm	10
MXMAC	L <= 1	10
KMF	$c_{re} > c_s$	10

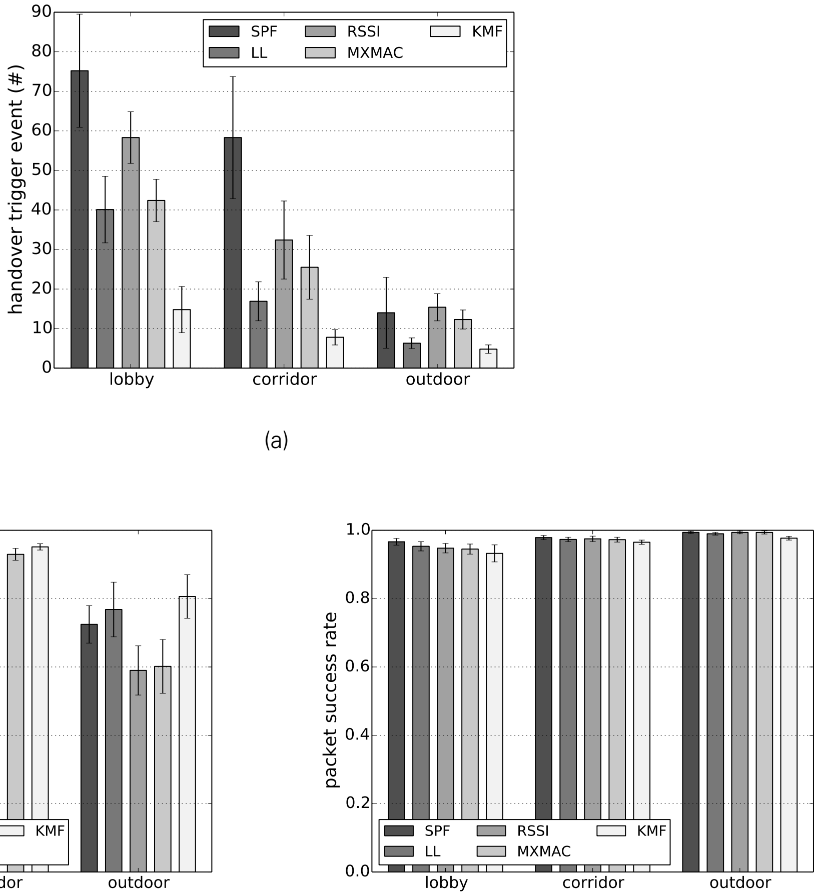
disconnection or that the packet loss rate is below a specified threshold. It is a measure of the sensitivity of the triggering algorithm. A highly sensitive algorithm leads to a frequent attempt to transfer a communication to an alternative relay node, and may cause a high handover cost. As most commercially available transceivers are low-powered and low-cost, the RSSI values of received ACK packets may fluctuate for a brief period of time despite the very low speed of the robot. In other words, a fluctuation in the RSSI values of received ACK packets may not necessarily indicate the disconnection of an established link. Thus, the handover triggering algorithm should be tolerant to such transient variations of link quality, otherwise it may lead to a ping-pong handover problem, unnecessarily increasing packet transmission latency and power consumption. Moreover, a mobile transmitter may not be successful in finding a new relay node whenever a handover is initiated, in which case it may waste resources in searching for relay nodes. Figure 6.8(a) suggests that our algorithms (KMF) generated a significantly less number of trigger events than all the other algorithms, because it was able to filter transient link fluctuations more efficiently than the other solutions, particularly, in the indoor environments (lobby and corridor). The SPF algorithm performed worst due to its reliance on a single packet failure to trigger a handover.

6.4.3 GOODPUT AND PACKET SUCCESS RATE

We define the *goodput* as the ratio of the number of successfully transmitted data packets to the maximum data packets which can be transmitted in an ideal link during the same transmission period:

$$Goodput = \frac{N_{success}}{N_{ideal}}$$

As shown in Figure 6.8(b), KMF gains the highest goodput overall in different environments. The reason is its high data packet transmission efficiency. Furthermore, KMF is the only algorithm the average goodput of which is above 80 %. It can be seen in Figure 6.8(c) that, compared to the other algorithms, the performance of KMF degraded a little bit in terms of packet success rate. It achieved 93.2 %, 96.5 %, and 97.7 % for lobby, corridor, and outdoor, respectively. The reason for the relatively low performance in this respect is its higher tolerance of transient packet failures.



(a)

(b)

(c)

Figure 6.8: Evaluation of the five algorithms in terms of: (a) Number of handover events. (b) Goodput. (c) psr.

6.4.4 SIMULATION

In order to compare the performance of all the algorithms under an identical condition, we let the robot move from one end of a deployment to another communicating with the relay nodes and recorded all essential parameters such as the RSSI values of the received ACK packets and the *psr*. Then we fed to the algorithms offline in a simulated environment and observed how they reacted to the link quality fluctuations and recorded the time points when they triggered a handover in order to visualize the communication transfer patterns. Figure 6.9 demonstrates the performances of the different schemes. The colored lines illustrates how a transmitter carried by a mobile robot was able to seamlessly maintain a link using the handover-triggering algorithms. The vertical bar lines display the frequency of attempts to transfer a communication as a result of a perceived change in the link quality. As can be seen, compared to all the other schemes, our approach was able to significantly minimize the handover attempt because of its superior prediction technique.

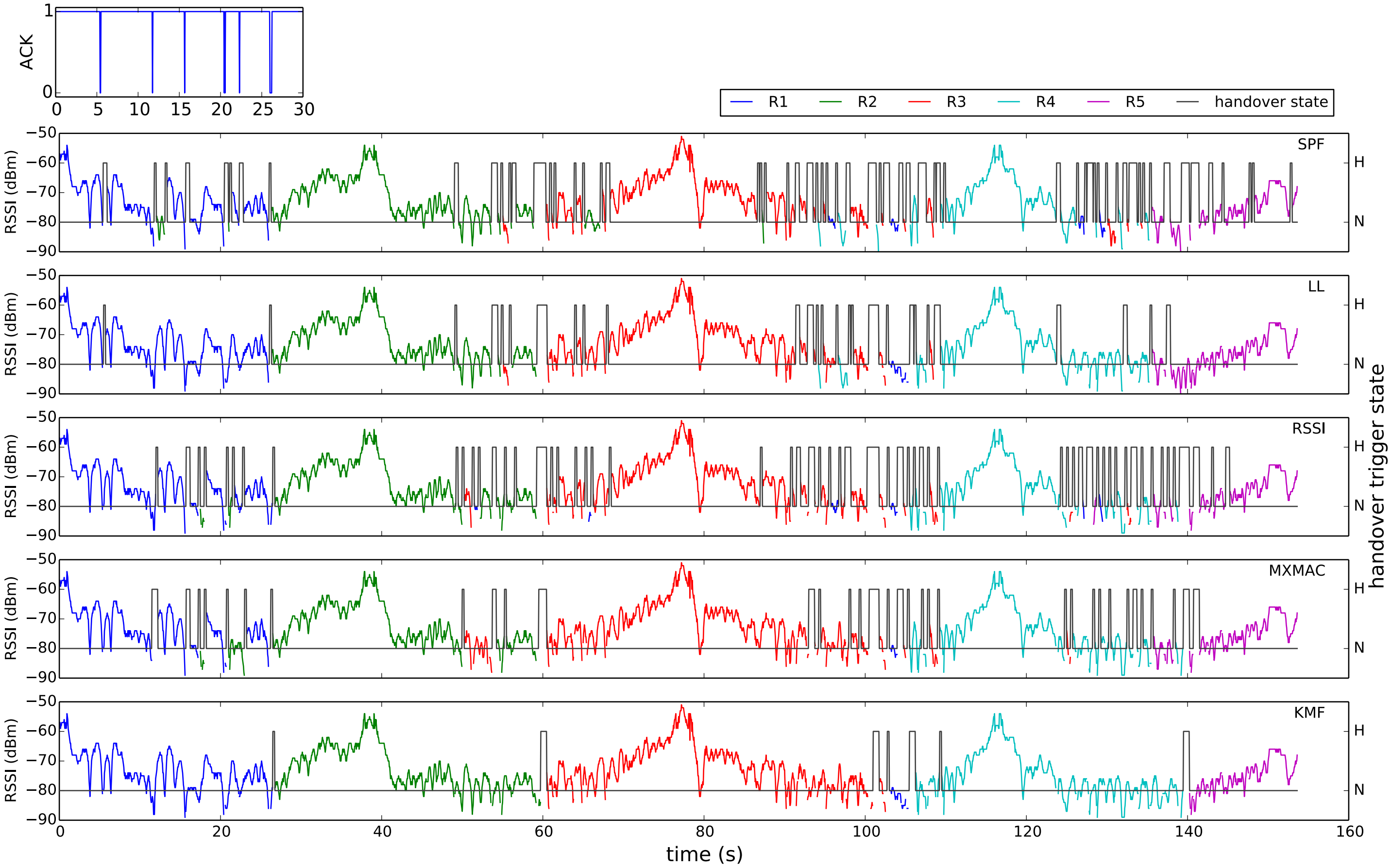


Figure 6.9: The runtime performance of the handover triggering algorithms for the same data trace. Colored lines: RSSI fluctuations. Black lines: Handover triggering instances. Top-left: The reception of ACK packets for the first 30 s at the first relay node.

6.5 CONCLUSION

In this chapter we proposed a handover-triggering algorithm which takes the RSSI fluctuation and the packet failure rate of a wireless link into consideration. To predict these two quantities, we modeled them as random variables and applied the Kalman filter by dividing time into epochs and analyzing the statistics representing their fluctuations in these epochs. Our aim was to tolerate transient fluctuations but accurately foresee middle term trends. We compared our approach with four proposed algorithms, all of which we implemented for the TinyOS and TelosB platforms.

7 MSI-MAC

A wide range of applications in wireless sensor networks require mobility support. Examples are healthcare applications [CLBR10], applications supporting independent living in residential areas [AE10], applications monitoring pollution in smart cities [SMG+14], and wildlife monitoring [GLJ11]. One of the main challenges in supporting mobility is the difficulty of establishing stable and reliable links when a continuous streaming of data is required. Independent studies have shown that wireless links established with low-power radios (i.e., those complying with the IEEE 802.15.4 specification) are often lossy and dynamic [DÖD11][SKAL08], the fluctuation in link quality becoming significant for mobile nodes (some have reported ±10 dB fluctuation for distances less than 30 m) [ZCV14][ZD10][DW14].

In cellular networks, the task of managing mobility (intra- and inter-cellular handover) is assigned to resource-rich base stations. Should the same feature be supported in wireless sensor networks, the management task should be undertaken by energy constraint, resource-limited, and low-power wireless sensor nodes. Furthermore, unlike cellular base stations, which are always powered on and active, the potential relay nodes with which a new link should be established before an existing one breaks, may be sleeping in order to save energy.

Ideally, a mobility management protocol in wireless sensor networks should be resilient to transient link dynamics but quick to react to persistent link quality degradations. This aspect entails:

- identifying the appropriate time to initiate a handover process,
- seamless discovery of candidate (neighbor) nodes, and
- selection of the most reliable relay node.

These steps have been addressed in the literature in different ways.

MRI-MAC [DD12] assumes that the relative location of a mobile node with respect to a stationary relay node can be estimated from the RSSI values of the packets it receives. In order to collect sufficient statistics from incoming ACK packets in short time, the protocol uses burst transmission. If the relative distance of a node with the current relay node is beyond a predefined threshold, the mobile node initiates a handover immediately. To discover candidate relay nodes, the mobile node eavesdrops on beacons transmitted by neighbor relay nodes.

MX-MAC [DW14] and MoX-MAC [BNG14] extend X-MAC [BYAH06] to support a seamless handover. The former employs Least Mean Square (LMS) filter to predict the link quality of a mobile node and defines a threshold to trigger a handover, while the latter triggers a handover upon experiencing a single packet failure. SmartHop [FZA+12] transmits beacons in burst to discover candidate neighbors and estimates the relative link quality of its neighbors by evaluating received ACK packets. The handover decision is made by setting a predefined RSSI threshold with a hysteresis margin. The protocol is designed on the basis of an extensive study on the impact of key PHY and MAC parameters on the handover performance. However, the protocol does not support duty-cycling and assumes that candidate relay nodes are active all the time.

In this chapter we propose a mobility management protocol to address the three features we identified above. Its typical features can be summarized as follows: It (1) enables mobile nodes to join a network quickly; (2) supports burst transmission in order to let a mobile node transfer as many packets as possible when the quality of a link is good and stable; (3) employs a Kalman filter in the background in order to predict the state of a mobile link with statistics obtained from received ACK packets; and (4) establishes the temporal evolution of all potential links during neighbor discovery in order to identify the best relay node to which a communication should be transferred.

The remainder of this chapter is organized as follows: In Section 7.1 and 7.2, we introduce our protocol and discuss its implementation detail respectively. In Section 7.3, we present experiment results and quantitative comparisons with three state-of-the-art mobility management protocols. Finally, in Section 7.4, we provide concluding remarks and outline future work.

7.1 SEAMLESS HANDOVER

In a wireless sensor network supporting mobile nodes, the predominant traffic flow is from the mobile nodes to a remote base station via intermediate, stationary relay nodes. Hence, the main task of the relay nodes is assisting the mobile nodes. This is the case for many residential applications (healthcare, independent living) where sensor nodes are deployed on the bodies of people who nevertheless move freely and carry out everyday tasks while vital biomedical measurements are collected from them.

In order to support uninterrupted monitoring and a steady streaming of packets, we support a seamless handover. In our protocol, a mobile node initiates a handover when it perceives that the link it has already established with a stationary relay node is becoming bad. This can be done by evaluating physical and link layer parameters of received ACK packets. Moreover, a handover can be initiated and completed without first breaking an established link. This can be achieved by embedding handover requests into the MAC header of data packets.

7.1.1 PROTOCOL DESIGN

Our protocol is a preamble-based MAC protocol [BYAH06] and supports burst transmission [Dun11]. Hence, when a mobile node first attempts to join the network, it transmits a preamble until a nearby relay node responds with an acknowledgment. The preamble is *anycast*, in that all neighbor relay nodes can access and respond to it, as illustrated in Figure 7.1. With the arrival of a beacon (acknowledgment) packet, the join phase will be completed (this process will be explained in more detail shortly).

After establishing connection with a specific relay node, the mobile node begins transmitting packets in burst with unicast/ACK scheme, as depicted in the right part of Figure 7.1. While the transmission is still going on, the mobile sender estimates and predicts the link quality by continuously evaluating physical and link layer parameters in the background. In case of a steady link quality deterioration (characterized by persisting packet loss rate and poor RSSI values of incoming ACK packets), the mobile node initiates a handover request to all nearby relay nodes without actually breaking the data transmission with the current relay node. When it discovers a better relay node, it then transfers communication to this node and resumes burst transmission with unicast/ACK scheme. The cycle of burst transmission, handover trigger and neighbor discovery/selection is repeated until the bulk data transfer is completed.

Our design approach takes many of the requirements of low-power wireless sensor networks into account:

- In contrast to existing or proposed preamble-based MAC protocols, our protocol establishes a link by *anycasting* the first data packet and with a relay node which wakes up and acknowledges the earliest. This minimizes the number of packets transmitted as preamble and leads to a fast network joining.
- Unlike many mobility-aware MAC protocols, except for the network join, our protocol employs a *unicast* communication during the whole transmission, even during neighbor discovery and selection phase. As a result no data packet duplication is introduced and, therefore, no duplication suppression mechanism is required.
- Our protocol does not require extra control packets to manage a handover process thereby reducing the signaling overhead, for example, when compared with ME-ContikiMAC [PKG+15] and SmartHop [FZA+12].
- Our protocol is compatible with duty-cycled operations.

7.1.2 FAST NETWORK JOIN

A mobile node may not have sufficient information about the relay node distribution in its surrounding. Therefore, it has to first search for an available relay node before it can trans-

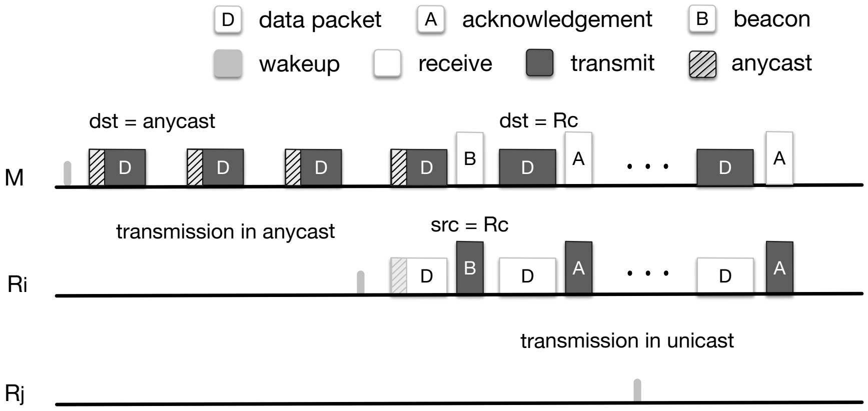


Figure 7.1: The initial phase of MSI-MAC: MSI-MAC introduces *anycast* to discover and join the network with the first awoken receiver in the vicinity. The label M and R in the figure represent mobile sender and receivers respectively

fer communication to a new link. In compliance with the IEEE 802.15.4 specification, it first performs clear channel assessment (CCA) to ensure that the medium is free. Then, it transmits the first data packet repeatedly in *anycast* mode, until it receives an acknowledgment from a nearby relay node. At the receiver's side, when a relay node receives an *anycast* data packet[1], it does not evaluate the entire packet in order to acknowledge it. Instead, it generates a beacon packet containing its own address and sends the beacon to the mobile node. The purpose is to simply indicate that it is available and ready to receive the remaining data packets. In case multiple relay nodes receive the *anycast* packet simultaneously, the probability of multiple beacons experiencing collision will be high.

To illustrate the impact of this: suppose that during the active period of a duty cycle, T, a mobile sender transmits at most N packets with inter-packet interval τ. We can express N as $N = \lfloor \frac{\tau}{T} \rfloor$. Suppose also the wakeup times of the relay nodes are statistically independent and uniformly distributed between $(0, T)$. If we divide the duty cycle interval T into N uniform slots, the probability that a relay node awakes at any one of the N slots to successfully receive a data packet and respond with a beacon at that slot is $1/N$. The beacon transmission collision occurs when at least two receivers awake in that slot. So the collision probability can be expressed as:

$$P_{collision} = 1 - \left[\frac{1}{N} \left(1 - \frac{1}{N} \right)^{m-1} + \left(1 - \frac{1}{N} \right)^{m} \right] \tag{7.1}$$

where m is the number of receivers in the neighborhood of the mobile sender. Figure 7.2 shows the beacon collision probability distribution for different wakeup intervals and number of receivers in the neighborhood. By properly desynchronizing the wakeup time at set up time, the probability of beacon collision can be reduced to an acceptable level for small-scale, small duty-cycled networks (which is the case for residential areas, for example). For instance,

---

[1] In our implementation, we reserve the address 0x8000 as anycast address.

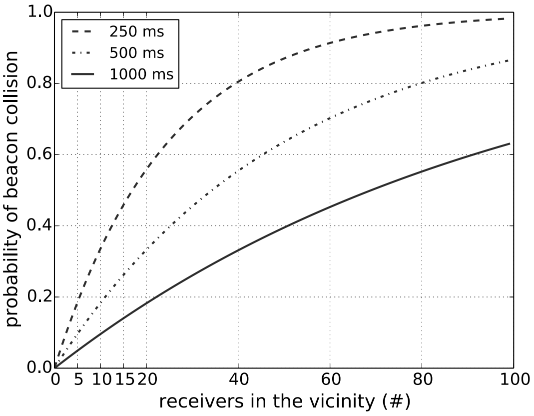


Figure 7.2: The collision probability distribution function during *anycast* packet transmission within a single wakeup period (the wake up intervals is set between 250 ms to 1000 ms).

the collision probability is between 8% to 30% when the wakeup interval is set from 1000 ms to 250 ms with 10 neighbors.

7.1.3 BURST TRANSMISSION

Once a mobile sender discovers a relay node, it switches the communication mode back to unicast/ACK mode and transmits packets in burst, with no CCA between successive packets [DÖD11]. The idea is to enable the mobile node to transfer as many packets as possible before the link deteriorates. Hence, our protocol trades fairness for high throughout.

Two of the key components of our handover management protocol are the link quality estimation and handover trigger algorithm. The first continuously evaluates the fluctuation of link quality and whether this is a steady-state phenomenon. It is a realization of the Kalman filter and takes as its input two link quality metrics from the physical and the link layer, namely, RSSI values and packet success rate (PSR). The filter predicts whether the deterioration of a link quality is a steady phenomenon (and, therefore, whether a handover request should be triggered) or not. The second component triggers a handover request, collects beacons from its environment, and selects the best candidate to transfer a communication.

7.1.4 NEIGHBOR DISCOVERY (ND-PHASE)

Instead of broadcasting a sequence of control packets for discovering potential relay nodes during a handover request, our protocol keeps data transmission with current receiver as *unicast* but embeds a handover request in the MAC header, as illustrated in Figure 7.3 (left part). Nearby relay nodes intercepting these packets need only evaluate the MAC header in order to determine whether the packets contain a handover request. Because surrounding relay nodes may be sleeping during this phase (as a consequence of duty-cycling), the mobile node should send multiple requests for a duration that equals the period of a single duty-cycle. Unlike the unicast/ACK scheme during a normal burst transmission, where packet transmission by the mobile node immediately follows the reception of an acknowledgment packet, the mobile node should now back-off after receiving an ACK packet from its current relay node. The reason is that those relay nodes which have intercepted the data packets and are ready to participate in a handover process have the possibility to transmit beacons to the mobile node. Figure 7.4 illustrates this period.

Relay nodes, participating in a handover process should also back-off before they transmit beacons in order to minimize the probability of collision. In case more than two relay nodes wake up and respond to a handover request at the same time, beacon collision will occur and the probability distribution of this collision can be determined by using Equation 7.1.

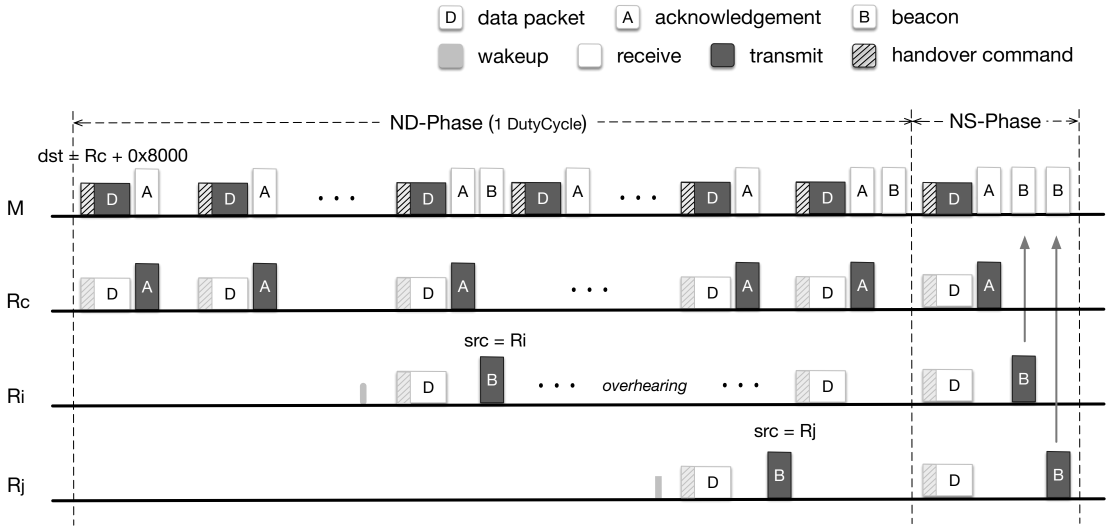


Figure 7.3: Neighbor discovery (ND) and selection (NS) phases: The mobile sender (M) embeds a handover request in a data packet to probe potential receivers while keeping communication with the current receiver (Rc). In the meantime, nearby active relay nodes (Ri and Rj) overhear the handover request and estimate the quality of the link they establish between the mobile node and themselves.

7.1.5 NEIGHBOR SELECTION (NS-PHASE)

As we have already mentioned, a neighbor discovery lasts an entire duty cycle. Following this, the mobile node decides to elect one of them as its future relay node. This decision is made based on the feedback it gathers from each potential neighbor at the end of the neighbor discovery period. The feedback is gathered thus: At the end of a neighbor discovery period, the mobile node sends to all its neighbors a request for feedback, whereupon each candidate relay node sends a beacon containing its *unicast* address and bidding information. The bidding information consists of:

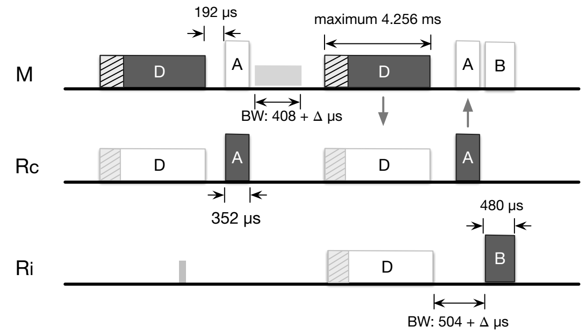


Figure 7.4: The expected duration of a neighbor discovery phase. Δ is the small guard time added to the back-off window (BW).

1. the averaged RSSI value for all the packets the relay node has intercepted since the beginning of a neighbor discovery phase;
2. the packet reception ratio; and,
3. the trend in the change of RSSI values in order to estimate whether the mobile node is moving towards the relay node or away from it. This phase is illustrated in Figure 7.3.

The computation of these parameter is as follows:

RSSI: The RSSI values are collected by overhearing data packets in which a handover request is embedded. We apply an online moving average algorithm to amortize the calculation cost to each reception. The averaged RSSI value is calculated as:

$$\bar{r}_n = \bar{r}_{n-1} + \frac{r_n - \bar{r}_{n-1}}{n}, \text{where n > 1} \tag{7.2}$$

Packet Reception Ratio (PRR): The packet reception pattern is aggregated by a counter from the reception of the first packet containing a handover request to the arrival of the feedback request. The total number of handover request transmitted can easily be determined by examining the digital sequence number in the header.

Mobility Trend: The main aim of neighbor selection is to choose the most reliable next relay node to which the remaining packets of a mobile node can be transferred. To this end, estimating the mobility trend of the mobile node with respect to a potential relay node is necessary. Since neither the mobile nor the relay node has an explicit location information or mobility model, whether or not a mobile node is approaching or moving away from a relay node can only be estimated locally by the fluctuating pattern of RSSI values. For a short duration and distance, it is reasonable to model the change in the RSSI values as a linear function of time: $rssi(t) = at + b$. Then the changing rate can be estimated by a simple linear regression model[2], which results in:

$$a = \frac{cov(r, t)}{var(t)} \tag{7.3}$$

---

[2] The constant a is estimated by minimizing the difference between $rsst(t)$ and its estimate $at + b$ in a mean square error sense.

Thus, the link quality bidding metric L_{bid} can be expressed as:

$$L_{bid} = \bar{r} + a * s * prr \tag{7.4}$$

where s is the remaining number of packets which should be transmitted in burst and prr is the packet reception ratio. Based on this input from each relay node, the mobile sender chooses the one with the highest bidding value as its next relay node.

7.2 IMPLEMENTATION

We implemented our protocol (henceforth called MSI-MAC) in TinyOS [LMP+05] for the TelosB [PSC05] and IMote2 platforms [NHS+08], both of which integrate an IEEE 802.15.4 compatible radio (CC2420). A good portion of the code is hardware independent and can easy be ported to other platforms and operating systems.

7.2.1 ANYCAST COMMUNICATION

The CC2420 radio chip does not support *anycast*, so we disabled the hardware address recognition and *auto-ack* features and delegated to the link layer the decision whether packet reception should be followed by the transmission of ACK packet or a beacon. This is implemented as follows: We reserved $0x8000$ as the *anycast* address. Any node receiving a data packet destined to this address and has a valid frame check sequence (FCS), responds with a beacon containing its address information without delay (without a CCA).

7.2.2 HANDOVER REQUEST

MSI-MAC does not introduce a new field in the MAC header but uses the most significant bit of the destination address to issue a handover request. For example, when a handover request is triggered, the destination address is set to $R_c + 0x8000$, where R_c is the current receiver address. This scheme has two benefits:

1. The unicast/ACK scheme remains intact during neighbor discovery. A designated relay node receiving the data packets in which a handover request is embedded can respond with an acknowledgment in the usual way by masking the handover request bit during the validation of the destination address.
2. All the other relay nodes, however, do not need to evaluate any part of the data packet except the header in order to determine whether this packet contains a handover request.

7.2.3 DATA AND BEACON FORMAT

We extended the 802.15.4 MAC header with two additional fields, namely, "remains" and "opt" to encode the number of remaining packets in burst and the beacon's feedback dur-

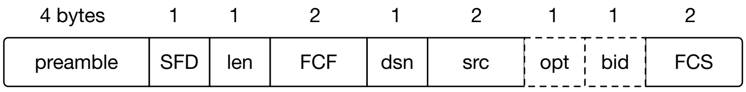


Figure 7.5: The format of a beacon for responding to a neighbor discovery request.

ing neighbor discovery, respectively. When a potential relay node responds to a handover request with a beacon, it randomly set one bit in the "opt" field. The mobile sender receives this beacon and sets the same bit in the "opt" field for next data transmission. If the relay node receives a handover request with the same "opt" bit, it will keep its radio on, continue overhearing handover request packets, but refrain from sending further beacons until the neighbor discovery period is over and the feedback request arrives. The "opt" field is used by relay nodes to determine the feedback transmission order during the neighbor selection phase. The beacon frame is varied from 13 bytes to 15 bytes and requires a maximum 480 μs to transmit, the format is illustrated in Figure 7.5.

7.3 EVALUATION

We evaluated our protocol experimentally and compared it with three state-of-the-art protocols, namely, ME-ContikiMAC, MX-MAC, and SmartHop (the authors provided us with the source code). We selected four metrics for our evaluation: packet success rate, the number of handover triggers, signaling overhead, and latency.

7.3.1 METHODOLOGY

We performed the experiment with the MobiLab testbed [WAD16a] consisting of 3 to 10 TelosB and Imote2 nodes, depending on the specific experiments. One of these nodes was a mobile robot. The static nodes acted as relay nodes and were deployed in the corridor of our faculty along a straight line, with a separating distance of 5 m. The mobile sender carried by the robot moved at a constant speed of 0.13 m/s from one end of the corridor to the other end while transmitting packets in burst. The inter-packet-interval (IPI) is set to 10 ms which is the minimum interval between two outgoing packets that is currently supported by the TinyOS implementation. The transmission power is limited to -25 dBm. Each experiment is repeated 10 times. The following figures show the averaged results with error bars (standard deviation). For the performance comparison with the-state-of-the art, unless explicitly stated, we used 5 static relay nodes to minimize the probability of collision on beacons during neighbor discovery. The evaluation of more than four neighbors is shown in Table 7.1.

Table 7.1: Performance comparison: The wakeup interval of relay nodes was set to 1000 ms. The number of neighbors is the potential nearby relay nodes.

Deployment	Protocol	PSR(%)	Handover Triggers (#)	Signaling Overhead (#)	Latency (ms)
neighbors: 2 spacing: 10 m	ME-Contiki	98.79%	63 (max: 83)	1151.8 (max: 1529)	11517.5
	MX-MAC	95.49%	37.5 (max: 51)	37.5 (max: 51)	372.9
	MSI-MAC	95.97%	11.5 (max: 15)	46 (max: 60)	293.3
neighbors: 4 spacing: 5 m	ME-Contiki	98.77%	68.5 (max: 99)	756.9 (max: 1166)	7568.8
	MX-MAC	96.81%	36.1 (max: 45)	36.1 (max: 45)	219.4
	MSI-MAC	97.57%	8.6 (max: 13)	69 (max: 104)	228.6
neighbors: 8	ME-Contiki	99.2%	55.5 (max: 72)	362.8 (max: 440)	3627.5
spacing: max 5 m	MX-MAC	97.0%	56 (max: 69)	56.0 (max: 69)	179.7
min 2.5 m	MSI-MAC	98.0%	8.5 (max: 12)	136 (max: 192)	242.2

7.3.2 PACKET SUCCESS RATE

As illustrated in Figure 7.6a, the packet success rate of the three protocols are all above 97%, under different wakeup intervals. The reliability of MSI-MAC is a little bit lower than ME-ContikiMAC, which is 98.77% against 97.57% for 1000 ms wakeup interval, while it is slightly higher than MX-MAC. The reason for the relatively low performance in this respect is that:

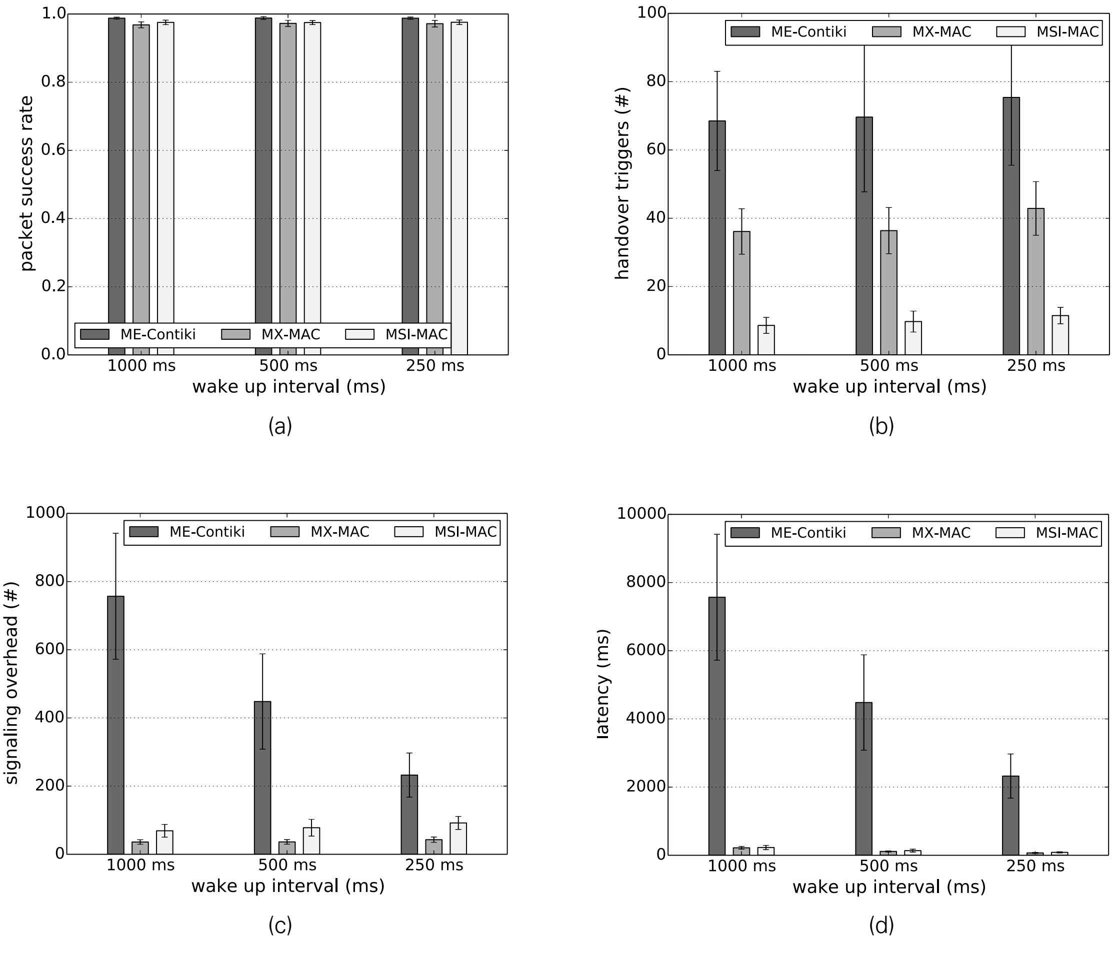
1. ME-ContikiMAC is sensitive to a single packet failure and triggers handover upon a single packet loss, regardless of the link dynamics (i.e., irrespective of whether a mobile node faces a transient or a persistent link quality deterioration).
2. In contrast, MSI-MAC relies on two metrics coming from the physical and link layers to estimate the link quality fluctuation and exhibits a greater tolerance to transient link fluctuations. As a result, ME-ContikiMAC experiences more handover oscillations (triggers), as depicted in Figure 7.6b.

7.3.3 HANDOVER TRIGGERS

Triggering a handover request at the appropriate time is essential to avoid unnecessary oscillations. If the handover trigger algorithm is too sensitive to link quality variations, more handover events are experienced, and consequently, the handover cost (signaling overhead, latency etc.) is high. On the contrary, if the algorithm is too tolerant to the link dynamics and fails to trigger a handover on time, the node may suffer from a considerable packet loss. Our results show that MSI-MAC reduces the number of handover triggers by about 12% and 23% compared to ME-ContikiMAC and MX-MAC, respectively.

7.3.4 SIGNALING OVERHEAD

The signaling messages are exchanged during neighbor discovery. Figure 7.6c shows the number of signaling messages transmitted on average. ME-ContikiMAC has the worst performance, because of the relatively poor neighbor discovery strategy it employs. To suppress the data packet duplication, instead of transmitting data frames, ME-ContikiMAC *anycasts* a burst of control packets to search for a new receiver. This led to the highest signaling overhead and, as a result, a large number of handover triggers. In contrast, both MX-MAC and MSI-MAC embed handover requests within data packets. Hence, the only signaling overhead is due to the response beacons generated by potential relay nodes. The difference is that MX-MAC receives only one beacon from a relay node which responds the earliest whereas in MSI-MAC, each potential relay node transmits two beacons to express their readiness and to bid their suitability. As a result, the signaling overhead of MSI-MAC is almost twofold when compared with MX-MAC, and amounts to 9% to 40% of the overhead produced by ME-ContikiMAC for different wakeup intervals. By contrast, MX-MAC introduces data packet duplication due to its data packet broadcasting scheme during neighbor discovery.



(a)

(b)

(c)

(d)

Figure 7.6: Performance comparison: (a) packet success rate (b) the number of handover triggers (c) signaling overhead (d) latency averaged per handover trigger.

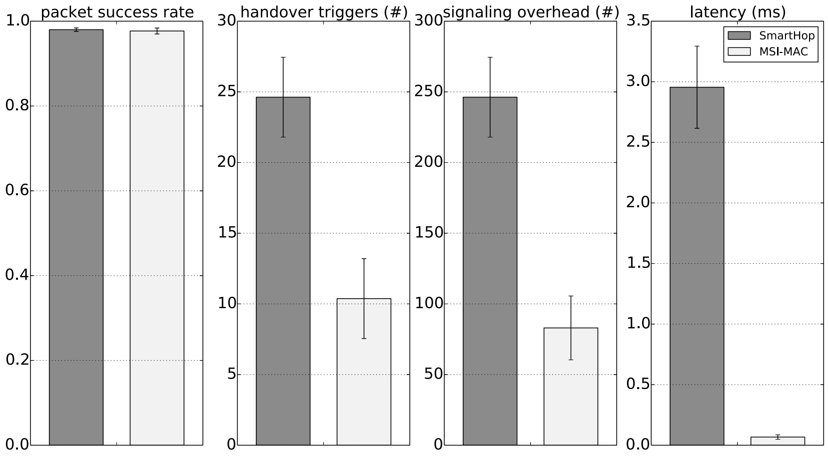


Figure 7.7: Performance comparison with SmartHop. The wakeup interval for MSI-MAC is 125 ms. The RSSI threshold for triggering handover is set to -85 dBm for SmartHop, and the decision margin is 1 dB.

7.3.5 LATENCY

In the context of seamless handover, latency is the time needed to establish a new link and resume burst communication via this link. Similar to the signaling overhead, latency is introduced during neighbor discovery. For ME-ContikiMAC, latency arises due to the time spent during the transmission of the control packet and the waiting for acknowledgment. For MX-MAC and MSI-MAC, it is caused by the back-off time during the transmission of beacons by potential relay nodes. The neighbor selection phase in MSI-MAC contributes additional latency. Figure 7.6d shows the averaged latency introduced by handover. As can be seen, the latency associated with MSI-MAC is significantly the smallest.

7.3.6 COMPARISON WITH SMARTHOP

Figure 7.7 shows the performance comparison between our protocol and SmartHop [FZA+12]. Since the duty cycling mechanism is not enabled in SmartHop, to make a fair comparison, we set the wakeup interval to 125 ms in MSI-MAC and the window size of neighbor discovery to 10 for SmartHop. In order to fix other parameters, such as the handover threshold and hysteresis margin for SmartHop, we performed a set of preliminary experiments and tuned them accordingly. SmartHop is a hard handover solution; in other words, the protocol first interrupts data transmission during neighbor discovery and associates a mobile node with an alternative relay node. Consequently, the signaling overhead and latency are significantly higher (4 to 44 times greater) in SmartHop than those introduced by MSI-MAC. Furthermore, when the wireless link is highly dynamic, SmartHop performs even worse because it relies only on a single, unreliable metric (RSSI values) to estimate link quality fluctuation.

7.4 CONCLUSION

In this chapter we proposed a protocol for enable a seamless handover in wireless sensor networks supporting mobile nodes. The protocol carries out seamless handover by (1) enabling mobile nodes to quickly join the network, (2) continuously evaluating link quality and stability using PHY and MAC layer parameters and by implementing a Kalman filter, and (3) defining a bidding metric to select the best relay node amongst competing nodes. Moreover, our protocol supports burst transmission in order to enable a mobile node to transfer as many packets as possible when the link is stable. This approach also has the added benefit of collecting sufficient statistics for the Kalman filter, so that it can make reliable prediction pertaining to link quality fluctuation.

We implemented our protocol for TinyOS runtime environment and for TelosB and Imote2 platforms. Furthermore, we compared our protocol with three state-of-the-art protocols. Thus, through repeated experiments we demonstrated that our protocol was able to make reliable handover; reduce handover latency, overhead and oscillation; and deal with transient link quality fluctuations. In future, we are aiming to focus on 1) optimizing the protocol to further reduce the signaling overhead and latency by introducing mechanisms to quickly identify bad links, so that aimless handover attempts can be quickly aborted, and 2) accommodating and scheduling multiple mobile senders simultaneously.

8 CONCLUSION

The widespread applications of wireless sensor networks require mobility support, such as health care and animal monitoring. Most traditional communication protocols in wireless sensor networks are not oriented from mobile scenarios, but assume stationary deployment. As a consequence, they are either lack of mechanisms to handle mobility or are not efficient in terms of latency and energy consumption. Some efforts have been made by extending existing protocols to deal with mobility, but they are not applicable to bulk data transfer or they are only evaluated by simulation. Therefore, it is necessary to seek for a new communication scheme to support mobility and bulk data transfer in energy constraint, resource-limited and low-power wireless sensor networks. To this end, this dissertation dives into designing protocols to enhance mobility and bulk data transfer from the beginning. The algorithms and protocols proposed in this dissertation are all evaluated on a mobility enabled testbed. In this chapter, we first conclude this dissertation with a summary of our contributions and then discuss some possibilities for future work.

8.1 FORMALIZING EXPERIMENT ON MOBILITY ENABLED TESTBED

Most of the exiting mobility aware protocols are evaluated in simulation and are not tested in a real environment. This is because on one hand, the proposed protocols with strict assumptions are not applicable and they are difficult to implement and test in resource constraint devices. On the other hand, the difficulties of handling mobile objects remotely result in few publicly accessible testbeds supporting mobility. However, there are several mobility enabled testbeds introduced in the WSN research community. They deeply rely on fixed infrastructure which makes it impossible to perform experiments in different environments. To address these problems, we developed MobiLab, a mobility enabled testbed for wireless sensor networks, on which repeated and reproducible experiments can be conducted. In Chapter 3, we elaborated the design and implementation of MobiLab. In the design of MobiLab, we separated the concern of the application development from the evaluation of the application in different mobile scenarios. To do this, we integrated commercial-off-the-shelf (COTS) hardwares into a unified system, which contains a testing field and the management components. These two parts are connected via wireless communication (WiFi), which makes the testbed flexible to deploy in different environments and on any scale.

In traditional testbeds, they only provide inter experiment management services, such as switching on or off sensor nodes and collecting data etc., while leaving intra-experiment control to experimenters. Usually, intra-experiment controls are done by either embedding the control logic in the testing application, or changing the matter of concern of source code and reloading to the hardware. In MobiLab, we release the intra experiment control to the control script. The control script specifies not only the number of nodes involved in the experiment, but also the runtime parameters and execution procedures. For the individual experiment, the control script is interpreted into a sequence of commands at the beginning of each running. These commands are sent to sensor nodes to adjust parameters or to change the node status timely. To enable sensor nodes to receive these commands, a traffic flow control protocol (TFCP) is developed. The TFCP is implemented as a middleware on top of TinyOS and has a footprint of 1058 bytes of ROM and 84 bytes of RAM. Additionally, to support mobility, several robots are employed. The motion pattern of each robot can be defined in the control script. It supports multiple random walk models in online or offline mode. In both modes, the motion actions are recorded in a script, which can be replayed to repeat the experiment. Finally, as an evaluation methodology for all experiments in this dissertation, in addition to the dedicated control scripts, MobiLab integrates specific data analysis and visualization tools.

To conclude, MobiLab has three main contributions:

- MobiLab provides a unified testing system to verify, test and evaluate the design of mobility aware protocols. MobiLab is a scalable mobility enabled testing framework. It is flexible and easy to be deployed on any scale and in any environment.
- MobiLab integrates with TFCP to enable both inter- and intra-experiment management that significantly simplifies the effort of conducting experiment.
- The procedure of experiment is directed by control script, which makes repeating experiment under the same condition possible.

8.2 ADAPTIVE BURST TRANSMISSION TO ADDRESS LINK DYNAMICS

In the early stages of wireless sensor networks, low data rate traffic patterns are assumed since applications have single purpose with simple sensing task and data packets are generated at a rate of minutes or hours. Thus, most of the proposed protocols focus on energy efficiency rather than high throughput. Emerging high data rate applications motivate bulk data transfer protocols to achieve high throughput. The basic idea is to enable nodes to transmit a sequence of packets in burst once they obtain a medium. However, due to the low-power, low-cost nature, the transceiver used in wireless sensor networks is prone to packet loss. Especially when the transmitters are mobile, packet loss becomes worse. The traditional bulk data transfer protocols are not energy efficient since they keep their radios on even when a large number of consecutive packet losses occur. To achieve a high throughput as well as energy efficiency, an adaptive burst transmission scheme is required.

In Chapter 5, we proposed an adaptive burst transmission scheme (ABTS), which is based on conditional probability distribution function (CPDF). The link quality information is collected during runtime and processed by a binarization algorithm. After binarization, the sequence of $[0, 1]$ values can be considered as the duration of unstable and stable periods. By measuring the width of each pulse of 1, the CPDF of the stable period in terms of number of packets can be obtained. In the same way, the CPDF of the unstable period can be calculated by measuring the width of 0 pulse. Once the CPDF is obtained, it is possible to determine the expected number of packets to transmit in burst and the duration of sleeping period. To evaluate the performance of ABTS, we conducted multiple experiments in indoor and outdoor environments. The evaluation results showed that ABTS achieves higher packet success rate in 90% of the cases, compared with a baseline algorithm which disregards link quality.

In conclusion, the proposed adaptive bust transmission scheme has three main contributions:

- We investigate link dynamics in wireless sensor networks and model link quality fluctuation in statistical manner.
- We propose a lightweight approach based on conditional probability distribution function (CPDF) to estimate the expected duration of stable and unstable links.
- We adapt the burst transmission size and sleep period according to the expected stable and unstable duration respectively.

8.3 ENHANCING MOBILITY BY SEAMLESS HANDOVER

Although an adaptive bust transmission scheme can efficiently handle link dynamics in mobile wireless sensor networks, other mobility problems can not be addressed, such as maintaining reliable connectivity. Bulk data transfer may be interrupted due to link break, which can result in the transmission being halted and lead to high energy consumption (e.g., keeping the radio switched on to retransmit, contend medium repeatedly). To prevent the unintended interruption before link disconnection actually occurs, the mobile node should find an appropriate alternative relay node to transfer its communication seamlessly. In order to achieve this aim, it has to address three questions as we suggested in the introduction chapter: when to trigger a handover process, how to search for a new relay node and where to transfer the communication. In Chapter 6 and 7, we proposed a novel handover trigger algorithm KMF and MSI-MAC to address these three questions.

KMF is a Kalman filter based handover trigger algorithm, which determines whether a fluctuation in link quality eventually results in a disconnection, rather than a transient failure. In the design of KMF, we utilized combined link quality metrics in physical and link layers, such as RSSI and packet success rate (PSR), to estimate link quality fluctuation online. The Kalman filter is adopted to predict link dynamics ahead of time. If a predicted link quality fulfills handover trigger criterion, a handover process will be initiated to discover alternative relay nodes and to establish a new link before disconnection occurs. We implemented KMF on top of TinyOS and integrated with a handover management protocol to preliminarily evaluate its performance. Our evaluation showed that KMF is able to significantly minimize handover trigger events, while maintaining the equivalent packet success rate. To further demonstrate the reason why KMF outperforms its competitors, we developed a simulator in Python. This simulator uses data traces from real experiments. The simulation result showed that KMF triggers less handover events due to its ability of distinguishing transient failure from disconnection.

In Chapter 7, we presented a full-fledged mobility management protocol, namely MSI-MAC. MSI-MAC is a mobile sender initiated medium access control protocol, which is originated to support seamless handover, instead of adapting existing protocols. It integrates with KMF to monitor and predict link dynamics. The proceeding of bulk data transfer starts with repeatedly anycasting the first data packet to join the network. Upon receiving a beacon from any stationary neighbor node, the mobile sender continues to transmit its remaining data packets in burst by using the unicast-ACK scheme. While transferring bulk data with the current relay node, link quality information is collected and fed to KMF to predict link dynamics. If a steady link quality deterioration is detected which may lead to disconnection, a handover process will be initiated by embedding a handover request into the MAC header of the data packet. This data packet with the handover request can be received and processed by all stationary nodes nearby as well as the one with which the mobile sender is currently communicating. In other words, the bulk data transfer with current relay node is not interrupted while the mobile node discovers alternative relay nodes. When the mobile node discovers a relay node with better link quality, it transfers the communication seamlessly to that node. We fully implemented MSI-MAC on top of TinyOS and evaluated in different environments. Our evaluation showed that MSI-MAC achieves reliability above 97% under different network setups, while it significantly reduces the signaling overhead by up to 95%.

In summary, the mobility management protocol proposed in this dissertation has four main contributions:

- We introduce anycast/beacon mechanism for fast network join. It shortens preamble transmission time for the mobile sender, thus the network join latency and overall power consumption significantly decrease.
- We utilize link metrics in physical and link layers to monitor link quality and apply Kalman filter to estimate, predict and trigger handover. Our handover trigger algorithm can effectively deal with temporal link dynamics and can efficiently trigger the handover in advance.
- We introduce unicast/ack, beacon scheme to discover neighbors in the vicinity of the

mobile sender. It can proceed with data transmission, while simultaneously searching for potential receivers. Thus, it eliminates the latency and signaling overhead.

- We present a new link quality bidding metric at the receiver side to assist the mobile sender in making a handover decision. This metric takes both current and predicted link quality into consideration, which can assist the mobile sender with making a local optimal handover decision.

8.4 FUTURE WORK

The algorithms and protocols proposed in this dissertation facilitate burst transmission in mobile wireless sensor networks. They gain better performance in dynamic environments compared to state-of-the-art. Nevertheless, there are still some possibilities to improve them in the future.

Duty cycling adaptation. Energy efficiency is one of the most important concerns when designing communication protocol in low power wireless sensor networks. To achieve a long life time, the duty cycling mechanism is widely adopted in practice. It can significantly lower down the overall energy expenditure. However, this parameter is usually determined ambiguously and is not adaptive during runtime. The non-optimal duty cycling setting can degrade network performance in terms of throughput and latency. Despite the efforts that have been made in the stationary network, the existing adaptive duty cycling algorithms are not suitable for mobile wireless sensor networks. This is because: 1) most of algorithms compute the optimal duty cycling value by estimating the data rate and traffic density around a receiver which takes a long time to achieve a convergent value. 2) Due to the mobile nature, it is more challenging to estimate the traffic density. So, to address these challenges, there is a need to exploit an efficient adaptive duty cycling mechanism for mobile wireless sensor networks.

Transmission power adaptation. By default, nodes in wireless sensor networks usually use the maximum power to transmit packets (e.g., 0 dBm), which aims to extend network coverage and improve reliability. But when the networks accommodate mobile nodes, using fixed transmission power will result in unnecessary energy wastage. This is because the relative distance between the mobile transmitter and the static receiver is changing over time. Intuitively, when the mobile node is approaching the relay node, it is preferable to lower down the transmission power to reduce power consumption while maintaining the required quality of service. In the same way, when the mobile node is moving away from the relay node, it has to increase the transmission power to maintain good link quality and prolong the connection time with the current relay node. To further reduce energy consumption in mobile wireless sensor networks, in addtion to duty cycling, an adaptive transmission power control strategy should be investigated.

Handover issues in sparse networks. In this work, we assume the network that accommodates mobile nodes is dense or moderate in deployment, where better alternative relay nodes can be always found during the handover process and at least one exists. However, in practice, the situation is more complex. For instance, large obstacles may block links for long time which results in temporal weak link quality in that area. To reduce the cost, less nodes should be deployed in some places. When handover is triggered, it is possible that no suitable node can be found. In this case, the mobile node will continue discovering neighbor nodes for a long period of time, resulting in high latency and energy consumption. To address this problem, an opportunistic transmission functionality should be added in the mobility management protocol.

BIBLIOGRAPHY

[80203] IEEE 802.15.4. Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans). 2003.

[AE10] Hande Alemdar and Cem Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

[ALL$^+$09] Muhammad Hamad Alizai, Olaf Landsiedel, Jó Ágila Bitsch Link, Stefan Götz, and Klaus Wehrle. Bursty traffic over bursty links. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 71–84. ACM, 2009.

[BKJ$^+$11] Nouha Baccour, Anis Koubaa, Maissa Ben Jamâa, Denis Do Rosario, Habib Youssef, Mário Alves, and Leandro B Becker. Radiale: A framework for designing and assessing link quality estimators in wireless sensor networks. *Ad Hoc Networks*, 9(7):1165–1185, 2011.

[BKM$^+$12] Nouha Baccour, Anis Koubaa, Luca Mottola, Marco Antonio Zuniga, Habib Youssef, Carlo Alberto Boano, and Mário Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.

[BKY$^+$10] Nouha Baccour, Anis Koubâa, Habib Youssef, Maissa Ben Jamâa, Denis Do Rosario, Mário Alves, and Leandro B Becker. F-lqe: A fuzzy link quality estimator for wireless sensor networks. In *EWSN*, pages 240–255. Springer, 2010.

[BMR$^+$11] James Brown, Ben McCarthy, Utz Roedig, Thiemo Voigt, and Cormac J Sreenan. Burstprobe: Debugging time-critical data delivery in wireless sensor networks. In *European Conference on Wireless Sensor Networks*, pages 195–210. Springer, 2011.

[BNG14] Papa Dame Ba, Ibrahima Niang, and Bamba Gueye. An optimized and power savings protocol for mobility energy-aware in wireless sensor networks. *Telecommunication Systems*, 55(2):271–280, 2014.

[BYAH06] Michael Buettner, Gary V Yee, Eric Anderson, and Richard Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *Proceedings* *of the 4th international conference on Embedded networked sensor systems*, pages 307–320. ACM, 2006.

[BZB$^+$14] Carlo Alberto Boano, Marco Zúñiga, James Brown, Utz Roedig, Chamath Keppitiyagama, and Kay Römer. Templab: A testbed infrastructure to study the impact of temperature on wireless sensor networks. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 95–106. IEEE Press, 2014.

[C$^+$06] LAN/MAN Standards Committee et al. Ieee standard for information technology telecommunications and information exchange between systems local and metropolitan area networks specific requirements. part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). *IEEE Std*, 802, 2006.

[CBA$^+$05] Brent N Chun, Philip Buonadonna, Alvin AuYoung, Chaki Ng, David C Parkes, Jeffery Shneidman, Alex C Snoeren, and Amin Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. Institute of Electrical and Electronics Engineers, 2005.

[CBD02] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[CFK$^+$10] Ioannis Chatzigiannakis, Stefan Fischer, Christos Koninis, Georgios Mylonas, and Dennis Pfisterer. Wisebed: an open large-scale wireless sensor network testbed. In *Sensor Applications, Experimentation, and Logistics*, pages 68–87. Springer, 2010.

[CLBR10] Octav Chipara, Chenyang Lu, Thomas C Bailey, and Gruia-Catalin Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 155–168. ACM, 2010.

[CT05] Jose Antonio Carrillo and Giuseppe Toscani. Wasserstein metric and large-time asymptotics of nonlinear diffusion equations. *New Trends in Mathematical Physics (In Honour of the Salvatore Rionero 70th Birthday)*, pages 240–254, 2005.

[Dar16] Waltenegus Dargie. Principles and applications of ubiquitous sensing, 2016.

[DC14] Manjunath Doddavenkatappa and Mun Choon. P 3: a practical packet pipeline using synchronous transmissions for wireless sensor networks. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, pages 203–214. IEEE, 2014.

[DD12] Qian Dong and Waltenegus Dargie. Analysis of the cost of handover in a mobile wireless sensor network. In *WWIC*, pages 315–322. Springer, 2012.

[DDHC$^+$10] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and

Andreas Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14. ACM, 2010.

[did] Diddyborg robot. https://www.piborg.org/diddyborg.

[DÖD11] Simon Duquennoy, Fredrik Österlind, and Adam Dunkels. Lossy links, low power, high throughput. In *SenSys*, pages 12–25. ACM, 2011.

[Don13] Qian Dong. *Analysis of the Cost of Handover in a Mobile Wireless Sensor Network*. PhD thesis, Saechsische Landesbibliothek-Staats-und Universitaetsbibliothek Dresden, 2013.

[DP10] W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wireless Communications and Mobile Computing. Wiley, 2010.

[DRCD+11] Clément Burin Des Roziers, Guillaume Chelius, Tony Ducrocq, Eric Fleury, Antoine Fraboulet, Antoine Gallais, Nathalie Mitton, Thomas Noël, and Julien Vandaele. Using senslab as a first class scientific tool for large scale wireless sensor network experiments. In *International Conference on Research in Networking*, pages 147–159. Springer, 2011.

[DRCF+11] Clément Burin Des Rosiers, Guillaume Chelius, Eric Fleury, Antoine Fraboulet, Antoine Gallais, Nathalie Mitton, and Thomas Noël. Senslab very large scale open wireless sensor network testbed. In *Proc. 7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCOM)*, 2011.

[DRR+15] Behnam Dezfouli, Marjan Radi, Shukor Abd Razak, Tan Hwee-Pink, and Kamalrulnizam Abu Bakar. Modeling low-power wireless communications. *Journal of Network and Computer Applications*, 51:102–126, 2015.

[Dun11] Adam Dunkels. The contikimac radio duty cycling protocol. 2011.

[DW14] Waltenegus Dargie and Jianjun Wen. A seamless handover for wsn using lms filter. In *LCN*, pages 442–445. IEEE, 2014.

[EAR+06] Emre Ertin, Anish Arora, Rajiv Ramnath, Vinayak Naik, Sandip Bapat, Vinod Kulathumani, Mukundan Sridharan, Hongwei Zhang, Hui Cao, and Mikhail Nesterenko. Kansei: a testbed for sensing at scale. In *Proceedings of the 5th international conference on Information processing in sensor networks*, pages 399–406. ACM, 2006.

[FAKB10] Hossein Fotouhi, Mário Alves, Anis Koubaa, and Nouha Baccour. On a reliable handoff procedure for supporting mobility in wireless sensor networks. In *RTN*, 2010.

[FFL06] R Fish, M Flickinger, and J Lepreau. Mobile emulab: A robotic wireless and sensor network testbed. In *IEEE INFOCOM*, 2006.

[FZA+12] Hossein Fotouhi, Marco Zúñiga, Mário Alves, Anis Koubâa, and Pedro Marrón. Smart-hop: A reliable handoff mechanism for mobile wireless sensor networks.

In *EWSN*, pages 131–146. Springer, 2012.

[GLJ11] Antonio Gonga, Olaf Landsiedel, and Mikael Johansson. Mobisense: Power-efficient micro-mobility in wireless sensor networks. In *DCOSS*, pages 1–8. IEEE, 2011.

[GOV+03] Craig A Grimes, Keat G Ong, Oomman K Varghese, Xiping Yang, G Mor, Maggie Paulose, Elizabeth C Dickey, Chuanmin Ruan, Michael V Pishko, James W Kendig, et al. A sentinel sensor network for hydrogen sensing. *Sensors*, 3(3):69–82, 2003.

[HKWW06] Vlado Handziski, Andreas Köpke, Andreas Willig, and Adam Wolisz. Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 63–70. ACM, 2006.

[HSFG09] SA Hameed, EM Shaaban, HM Faheem, and MS Ghoniemy. Mobility-aware mac protocol for delay-sensitive wireless sensor networks. In *2009 International Conference on Ultra Modern Telecommunications & Workshops*, pages 1–8. IEEE, 2009.

[HWXC10] Pei Huang, Chen Wang, Li Xiao, and Hongyang Chen. Rc-mac: A receiver-centric medium access control protocol for wireless sensor networks. In *Quality of Service (IWQoS), 2010 18th International Workshop on*, pages 1–9. IEEE, 2010.

[HXS+13] Pei Huang, Li Xiao, Soroor Soltani, Matt W Mutka, and Ning Xi. The evolution of mac protocols in wireless sensor networks: A survey. *IEEE communications surveys & tutorials*, 15(1):101–120, 2013.

[Ins13] Texas Instruments. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. *Reference SWRS041c*, 2013.

[JK07] Arshad Jhumka and Sandeep Kulkarni. On the design of mobility-tolerant tdma-based media access control (mac) protocol for mobile sensor networks. In *International Conference on Distributed Computing and Internet Technology*, pages 42–53. Springer, 2007.

[KA13] Bilal Muhammad Khan and Falah H Ali. Collision free mobility adaptive (cfma) mac for wireless sensor networks. *Telecommunication Systems*, 52(4):2459–2474, 2013.

[KFD+07] Sukun Kim, Rodrigo Fonseca, Prabal Dutta, Arsalan Tavakoli, David Culler, Philip Levis, Scott Shenker, and Ion Stoica. Flush: a reliable bulk transport protocol for multihop wireless networks. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 351–365. ACM, 2007.

[KMN13] Romain Kuntz, Julien Montavont, and Thomas Noël. Improving the medium access in highly mobile wireless sensor networks. *Telecommunication Systems*, 52(4):2437–2458, 2013.

[KN09] Romain Kuntz and Thomas Noël. Machiavel: Accessing the medium in mobile and dense wsn. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 1088–1092. IEEE, 2009.

[LC12] Tao Liu and Alberto E Cerpa. Talent: temporal adaptive link estimator with no training. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 253–266. ACM, 2012.

[Lev06] KSAP Levis. Rssi is under appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors, Cambridge, MA, USA*, volume 3031, page 239242, 2006.

[LEVG+13] Aime Lay-Ekuakille, P Vergallo, G Griffo, Olfa Kanoun, F Angelillo, and A Trabacca. Wireless network acquisition of joint eeg-ecg-ergospirometric signals for epilepsy detection. In *Measurements and Networking Proceedings (M&N), 2013 IEEE International Workshop on*, pages 41–45. IEEE, 2013.

[LFZ+13] Robert Lim, Federico Ferrari, Marco Zimmerling, Christoph Walser, Philipp Sommer, and Jan Beutel. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*, pages 153–165. IEEE, 2013.

[LMP+05] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos: An operating system for sensor networks. In *Ambient intelligence*, pages 115–148. Springer, 2005.

[LMZ+16] Shan Lin, Fei Miao, Jingbin Zhang, Gang Zhou, Lin Gu, Tian He, John A Stankovic, Sang Son, and George J Pappas. Atpc: Adaptive transmission power control for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 12(1):6, 2016.

[LPLT10] Chieh-Jan Mike Liang, Nissanka Bodhi Priyantha, Jie Liu, and Andreas Terzis. Surviving wi-fi interference in low power zigbee networks. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 309–322. ACM, 2010.

[MGC16] Mobashir Mohammad, XiangFa Guo, and Mun Choon Chan. Oppcast: Exploiting spatial and channel diversity for robust data collection in urban environments. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, page 19. IEEE Press, 2016.

[MLH+10] Sirajum Munir, Shan Lin, Enamul Hoque, SM Nirjon, John A Stankovic, and Kamin Whitehouse. Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 303–314. ACM, 2010.

[NH93] Homayoun Nikookar and Homayoun Hashemi. Statistical modeling of signal amplitude fading of indoor radio propagation channels. In *Universal Personal Communications, 1993. Personal Communications: Gateway to the 21st Century. Conference Record., 2nd International Conference on*, volume 1, pages 84–88. IEEE, 1993.

[NHS⁺08] Lama Nachman, Jonathan Huang, Junaith Shahabdeen, Robert Adler, and Ralph Kling. Imote2: Serious computation at the edge. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC'08. International*, pages 1118–1123. IEEE, 2008.

[NKNW96] John Neter, Michael H Kutner, Christopher J Nachtsheim, and William Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.

[ODE⁺06] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with cooja. In *Local computer networks, proceedings 2006 31st IEEE conference on*, pages 641–648. IEEE, 2006.

[PGN⁺14] Georgios Z Papadopoulos, Antoine Gallais, Thomas Noel, Vasileios Kotsiou, and Periklis Chatzimisios. Enhancing contikimac for bursty traffic in mobile sensor networks. In *SENSORS, 2014 IEEE*, pages 257–260. IEEE, 2014.

[PHC04] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.

[PJ04] Huan Pham and Sanjay Jha. An adaptive mobility-aware mac protocol for sensor networks (ms-mac). In *MASS*, pages 558–560. IEEE, 2004.

[PKG⁺15] Georgios Z Papadopoulos, Vasileios Kotsiou, Antoine Gallais, Periklis Chatzimisios, and Thomas Noël. Wireless medium access control under mobility and bursty traffic assumptions in wsns. *Mobile Networks and Applications*, 20(5):649–660, 2015.

[PP02] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[PSC05] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48. IEEE Press, 2005.

[ras] Raspberry pi 2 model b. https://www.raspberrypi.org/products/raspberry-pi-2-model-b/.

[RCBG10] Bhaskaran Raman, Kameswari Chebrolu, Sagar Bijwe, and Vijay Gabale. Pip: A connection-oriented, multi-hop, multi-channel tdma-based mac for high throughput bulk transfer. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 15–28. ACM, 2010.

[RHLG10] Olof Rensfelt, Frederik Hermans, Lars-Åke Larzon, and Per Gunningberg. Sensei-uu: A relocatable sensor network testbed. In *Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 63–70. ACM, 2010.

[RKM+09] Satish SC Rao, Braden Kuo, Richard W McCallum, William D Chey, John K DiBaise, William L Hasler, Kenneth L Koch, Jeffrey M Lackner, Carrie Miller, Richard Saad, et al. Investigation of colonic and whole-gut transit with wireless motility capsule and radiopaque markers in constipation. *Clinical gastroenterology and hepatology*, 7(5):537–544, 2009.

[RL09] Tal Rusak and Philip Levis. Burstiness and scaling in the structure of low-power wireless links. *ACM SIGMOBILE Mobile Computing and Communications Review*, 13(1):60–64, 2009.

[RMN11] Damien Roth, Julien Montavont, and Thomas Noel. Mobinet: Mobility management across different wireless sensor networks. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 351–356. IEEE, 2011.

[ROJS10] Thierry Rakotoarivelo, Maximilian Ott, Guillaume Jourjon, and Ivan Seskar. Omf: a control and management framework for networking testbeds. *ACM SIGOPS Operating Systems Review*, 43(4):54–59, 2010.

[SBR10] Petcharat Suriyachai, James Brown, and Utz Roedig. A mac protocol for industrial process automation and control. In *Proceedings of the 7th IEEE European Workshop on Wireless Sensor Networks (EWSN2010), Coimbra, Portubgal*, 2010.

[SDTL10] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16, 2010.

[SGJ08] Yanjun Sun, Omer Gurewitz, and David B Johnson. Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 1–14. ACM, 2008.

[SGS+09] Edmund YW Seto, Annarita Giani, Victor Shia, Curtis Wang, Posu Yan, Allen Y Yang, Michael Jerrett, and Ruzena Bajcsy. A wireless body sensor network for the prevention and management of asthma. In *Industrial Embedded Systems, 2009. SIES'09. IEEE International Symposium on*, pages 120–123. IEEE, 2009.

[SK13] Philipp Sommer and Branislav Kusy. Minerva: Distributed tracing and debugging in wireless sensor networks. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, page 12. ACM, 2013.

[SKAL08] Kannan Srinivasan, Maria A Kazandjieva, Saatvik Agarwal, and Philip Levis. The β-factor: measuring wireless link burstiness. In *SenSys*, pages 29–42. ACM, 2008.

[SMG+14] Luis Sanchez, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, et al. Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238, 2014.

[SSZ+13] Hugues Smeets, Chia-Yen Shih, Marco Zuniga, Tobias Hagemeier, and Pedro Josè Marrón. Trainsense: a novel infrastructure to support mobility in wireless sensor networks. In *Wireless Sensor Networks*, pages 18–33. Springer, 2013.

[SXFL15] De Lei Sha, Wei Cheng Xie, Xiao Long Fan, and Yu Li. Based on wireless sensor network (nwk) of non-contact tremor monitoring equipment improvement for parkinson's disease. In *Applied Mechanics and Materials*, volume 713, pages 491–494. Trans Tech Publ, 2015.

[SZSV11] Ricardo Silva, Zinon Zinonos, Jorge Sa Silva, and Vasos Vassiliou. Mobility in wsns for critical applications. In *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pages 451–456. IEEE, 2011.

[Tex06] Texas Instruments. *MSP430 Flash Memory Characteristics*, September 2006. Revised April 2008.

[VDL03] Tijs Van Dam and Koen Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180. ACM, 2003.

[WAD15] Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. A link quality estimation model for energy-efficient wireless sensor networks. In *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015.

[WAD16a] Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. Mobilab: A testbed for evaluating mobility management protocols in wsn. In *Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 49–58. Springer, 2016.

[WAD16b] Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. A system architecture for managing complex experiments in wireless sensor networks. In *ICCCN 2016*, 2016.

[WASW05] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. Motelab: A wireless sensor network testbed. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 68. IEEE Press, 2005.

[YBS$^+$08] Xiaojing Yuan, Sumegha Bagga, Jian Shen, M Balakrishnan, and D Benhaddou. Ds-mac: differential service medium access control design for wireless medical information systems. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 1801–1804. IEEE, 2008.

[YHE02] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576. IEEE, 2002.

[ZAI$^+$14] Mahdi Zareei, Muhammed I Alghamdi, AKM Muzahidul Islam, Sabariah Baharun, and Rahmat Budiarto. Energy efficient handover management in cluster based wireless sensor network. *Proceeding of the Electrical Engineering Computer Science and Informatics*, 1(1):433–436, 2014.

[ZCV14] Zinon Zinonos, Chrysostomos Chrysostomou, and Vasos Vassiliou. Wireless sensor networks mobility management using fuzzy logic. *Ad Hoc Networks*, 16:70–87, 2014.

[ZD10] Tang Zhiyong and Waltenegus Dargie. A mobility-aware medium access control protocol for wireless sensor networks. In *IEEE Globecom Workshops*, pages 109–114. IEEE, 2010.

[ZHKS04] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138. ACM, 2004.

[ZTBW11] Mahdi Zareei, Alireza Taghizadeh, Rahmat Budiarto, and Tat-Chee Wan. Emsmac: energy efficient contention-based medium access control protocol for mobile sensor networks. *The Computer Journal*, 54(12):1963–1972, 2011.

[ZVC13] Zinon Zinonos, Vasos Vassiliou, and Chrysostomos Chrysostomou. Handoff triggering for wireless sensor networks with performance needs. In *ISCC*, pages 000982–000988. IEEE, 2013.