SYMBOL-LEVEL INFORMATION EXTRACTION : FROM BITS TO SYMBOLS IN WIRELESS NETWORK CROSS-LAYER DESIGN

TAO XIONG

Ph.D

The Hong Kong Polytechnic University

2015

The Hong Kong Polytechnic University

Department of Computing

Symbol-Level Information Extraction: From Bits to Symbols in Wireless Network Cross-Layer Design

Tao Xiong

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

September 2014

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signature)

Tao Xiong (Name of Student)

Abstract

At ever-increasing rates, wireless networks are becoming an indispensable part of people's daily life due to the low cost in the deployment of networking infrastructures and the high availability to access the Internet. Current wireless technologies use bits to deliver both the digital contents and packets' information, e.g. MAC address or packet transmission duration, which require the packets to be fully received and correctly decoded. However, for most receivers in the networks, they only need packets' information to adjust their behaviors even if they suffer the bad channel conditions. In this thesis, I mainly focus on the cross-layer design of symbol-level information extraction which enables the vital information to be delivered between the transmitters and receivers at symbol-level. I further take advantage of that symbol-level information to design the cross-layer structure to deal with three popular research topics in wireless communications: the hidden terminal problem, the energy inefficiency of the packet overhearing problem and communications security (COMSEC) problem.

Hidden terminals are typical interference sources that can significantly reduce the throughput of a wireless network if it adopts the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) MAC protocol. Normally, the standard RTS/CTS (Request to Send / Clear to Send) mechanism is deployed to solve this hidden terminal problem. However, the standard RTS/CTS would fail to silence all the hidden terminal, as in the real world, the CTS packets might not be correctly received all the time due

to either the CTS packets are unable to be decoded at remote hidden terminals or the CTS packets are collided with other packets at the hidden terminals. To solve the two drawbacks of the CTS packets, the RTS/S-CTS mechanism is proposed, which uses the symbol-level information extraction mechanism to deliver the NAV time information. The S-CTS frame can be correctly detected from collisions and by remote hidden terminals. Thus, the NAV time information which is contained in the S-CTS frame can be used to silence those hidden terminals. A testbed of RTS/S-CTS with GNURadio/USRP2 software radio is built to demonstrate its feasibility and extensive ns-2 simulations are conducted to evaluate its performance. The simulations results show that the RTS/S-CTS can significantly improve the throughput in the random topology network scenario compared with the standard RTS/CTS.

In the energy inefficiency of the packet overhearing problem, I first reveal that the energy waste on the packet overhearing accounts for the majority of the energy inefficiency of wireless devices in high traffic wireless LANs by analyzing the real world traffic traces. Though there are many existing approaches trying to reduce this energy inefficiency, none of them can avoid the energy waste on the packet overhearing in high traffic wireless LANs effectively. Thus, I propose a novel SASD (Sample-Address Sample-Duration) scheme that uses the symbol-level information extraction mechanism to deliver the MAC address and packet transmission duration information at the PHY layer. The SASD enables the wireless devices to discern the information under the energy-saving downclocking mode through the SASD Detection and Identification decoder. Consequently, the devices which are not the intended receiver of the packet can switch to the sleeping mode to save the energy cost on the packet overhearing. The extensive hardware experiments and simulation results show that the SASD can greatly outperform the existing approaches in the high traffic wireless LAN scenario.

In the wireless communications security problem, I explore the feasibility of symbol

obfuscation to defend against the passive eavesdropping attack and fake packet injection attack during the wireless communications. I propose a novel Multiple Inter-symbol Obfuscation (MIO) scheme, which utilizes a set of artificial noisy symbols, which called "symbols key", to obfuscate the original data symbols in the PHY layer. As the symbols key information can only be extracted and verified by the legitimate receiver, the eavesdropper can hardly decrypt the obfuscated symbols from the eavesdropped packets, and the fake packet can be easily checked out. Thus, the MIO can effectively enhance the wireless communications security. Compared with other communications security methods, the mathematical analysis proves that the MIO can provide the better performance on computational secrecy against the fake packet injection attack. Moreover, MIO can provide an easier way to achieve information-theoretic secrecy against the passive eavesdropping attack.

To sum up, by investigating the symbol-level information extraction mechanism to solve the problems, we show the feasibility of delivering vital information at the symbol level. Moreover, this symbol-level information extraction mechanism opens up a new dimension to convey the information from the bit level to the symbol level.

Publications

Conference Papers

1. **Tao Xiong**, Jin Zhang, Junmei Yao and Wei Lou, "Symbol-Level Detection: A New Approach to Silencing Hidden Terminals", in the *20th IEEE International Conference on Network Protocols (ICNP)*, Austin, Texas, USA, October 2012.

2. **Tao Xiong**, Jin Zhang and Wei Lou, "On Eliminating Energy Inefficiency of the Packet Overhearing Problem in High Traffic Wireless LANs", in the *11th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON) poster*, Singapore, June, 2014.

3. Jin Zhang, **Tao Xiong** and Wei Lou, "Community Clinic: Economizing Mobile Cloud Service Cost via Cloudlet Group", in the *11th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS)*, Philadelphia, Pennsylvania, USA, October 2014.

4. Junmei Yao, **Tao Xiong** and Wei Lou, "Mitigation of Exposed Terminal Problem Using Signature Detection", in the *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Seoul, Korea, June, 2012.

5. Jin Zhang, **Tao Xiong** and Wei Lou, "What Steward Could BYOD Users Employ? A Differentiated DRM Service between the Cloud and Mobile Devices", in

preparation.

Journal Papers

1. **Tao Xiong**, Jin Zhang and Wei Lou, "It Can Drain out Your Energy: An Energy Saving Mechanism against Packet Overhearing in High Traffic Wireless LANs", submitted to the IEEE Transactions on Transactions on Mobile Computing (Under review).

2. **Tao Xiong**, Wei Lou, Jin Zhang and Hailun Tan, "MIO: Enhancing Wireless Communications Security through Physical Layer Multiple Inter-symbol Obfuscation", submitted to the IEEE Transactions on Information Forensics & Security (Accepted).

3. Junmei Yao, **Tao Xiong**, Jin Zhang, and Wei Lou, "On Eliminating the Exposed Terminal Problem Using Signature Detection", submitted to IEEE Transactions on Mobile Computing. (Major revision)

4. Junmei Yao, **Tao Xiong**, and Wei Lou, "Beyond the Limit: A Fast Tag Identification Protocol for RFID Systems", submitted to Elsevier Journal on Pervasive and Mobile Computing. (Accepted)

Acknowledgements

I would link to express my appreciation to all those who helped me during my Ph.D. study. First and foremost, I would like to take this opportunity to thank Dr. Wei Lou, my chief supervisor, for his continuous encouragement and rigorous supervision of my research. He always trains me to be a good researcher, such as how to find research issues, how to solve thorny problems, how to express the ideas clearly, and how to write high-quality academic papers. His motivation, enthusiasm, patience, and immense knowledge make him a terrific supervisor, and inspire me a lot. What I have learned and experienced will always help and encourage me in the future.

Second, I would like to thank and express my gratitude to the coexaminers of my guided study and confirmation defense, Dr. Qixin Wang and Dr. Zhijun Wang. Also, I am very grateful to all my co-authors, Mr. Jin Zhang, Ms. Junmei Yao, and Dr. Hailun Tan. They provided me with a lot of insightful comments and constructive suggestions to my research.

Third, I would like to convey my thanks to Dr. Junchao Ma, Dr. Honglong Chen, Libin Yang, Nianbo Liu, Bing Bai, and Lei Wang, of our research group. I thank them for their kind help in my research, such as interesting discussions and suggestions on my work. I also would like to thank Yingjie Li, Ran Wang and Yan Shu for sharing with me the pain and pleasure of study at the university.

Last but not least, I would like to dedicate this thesis to my family, including my

parents, Mr. Mianchao Xiong and Mrs. Peirong Li, my beloved wife, Mrs. Wanling Peng and my younger sister, Ms. Wanjun Xiong. I hope I can make it up to them for all the time they sacrificed to help me to be devoted to my study. Especially, I would like to express my gratitude to my parents, their support and encouragement have convinced me that I can achieve anything I put mind to. Without them, this work would not have been possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past decades, the development of wireless network technology has fundamentally changed the way we obtain information and communicate with others [18, 39]. The wireless network is becoming an indispensable part of people's daily life. People can easily access the wireless network via the cellphones, laptops, tablets, or other wireless handheld devices. Most wireless network systems deploy the 5-layer networking model (TCP/IP model) and use the bits to deliver the digital contents and packets' information [64], which inevitably require that the packets must be fully and correctly received. However, there is a factor that most receivers do not have to fully receive the packets. They only need some packet information to adjust their behavior.

The main objective of this research is to investigate the symbol-level information extraction mechanism which makes the information to be carried, delivered and discerned by the transmitters and receivers at symbol-level. In this thesis, the cross-layer structures between physical layer (PHY layer) and MAC layer are designed to utilize the symbol-level information to express different level of vital information, e.g. the packet's information or the confidential information, so that both transmitters and receivers are aware of the

vital information without fully receive the packet even if the channel conditions are not good enough for receivers to correctly decode the bits. We deploy this cross-layer design of symbol-level information extraction mechanism in three popular research topics in wireless communications: the hidden terminal problem, the energy inefficiency of the packet overhearing problem and wireless communications security (COMSEC) problem.

1.1 The Introduction

In this section, I first brief the symbol-level information extraction. Then, I give the introduction of the three popular research problems which I mentioned above, briefly discuss the motivation why I deploy the symbol-level information extraction mechanism in the problems and our approaches that how the cross-layer design of symbol-level information extraction mechanism defends against the problems.

1.1.1 The Symbol-Level Information Extraction

At ever-increasing rates, wireless networks are becoming an indispensable part of people's daily life. Current wireless technologies, e.g. 802.11, 802.15.4, or bluetooth, use digital bits to deliver both the digital contents and packets' information, which inevitably cause that the information acquirement in the wireless network is limited by two main requirements, 1) the signal to interference and noise ratio (SINR) requirement and 2) the packet integrity requirement, which means that, the SINR at the receiver side must be enough to correctly decode the digital bits, and the packet must be fully received to reveal the information carried in the packet.

To alleviate these two main requirements, recent research works [29,39,45,46,68,83,84]

trend to use the PHY layer baseband symbols to convey some information at the PHY layer. The symbol-level information is added at the transmitter side without any third party, and it can be achieved by the receiver. As the symbol-level information does not need to be decoded into digital bits and can be directly extracted at the PHY layer, the symbol-level information extraction can resist the low SNR and SINR environments. Moreover, as the symbol-level information extraction mechanism is independent of the whole packet, it can be added to any place of the packet frame at the PHY layer to deliver some vital information.

In the thesis, I use this symbol-level information extraction mechanism to convey some packets' control information, such as MAC address or packet transmission duration (Chapter 3 and 4), to solve the hidden terminal problem and the energy inefficiency of the packet overhearing problem. Moreover, I use this symbol-level information extraction mechanism to covey the secure information, such as noisy symbols key (Chapter 5), to enhance the wireless communications security.

1.1.2 The Hidden Terminal Problem

1.1.2.1 The Problems in the Existing Hidden Terminal Solutions

Hidden terminals are typically considered harmful in wireless networks since the interference from these hidden terminals can significantly reduce the throughput of wireless networks [21,23,41,60,77,80]. Current IEEE 802.11 MAC protocol mainly uses two mechanisms, carrier sense multiple access with collision avoidance (CSMA/CA) and RTS/CTS (virtual carrier sensing), to handle this hidden terminal problem [5, 23, 25, 50, 60, 80]. In the standard RTS/CTS mechanism, the NAV time field of the RTS/CTS packets plays

an important role that the terminals, which are not involved in the RTS/CTS handshake, can decode the NAV time and defer their transmissions for that time duration. Ideally, underlying the assumptions that (1) all hidden terminals are within the data transmission range of a receiver and (2) the CTS packet from the receiver suffers no collisions, the RTS/CTS mechanism is successful in contending for the wireless channel [5, 43].

However, in the real world, these two assumptions do not hold all the time. Consequently, it may cause two problems: (1) Remote hidden terminals that are out of the data transmission range of the receiver may not be able to decode the CTS packet correctly due to its low signal-to-noise-ratio (SNR). (2) The CTS packet may be collided with other concurrently transmitted packets so that hidden terminals cannot successfully decode the collided CTS packet due to its low signal-to-interference-plus-noise-ratio (SINR). Therefore, the hidden terminal problem cannot be fully solved by the standard RTS/CTS mechanism [39, 60, 80].

In this thesis, we address the above two problems as the *remote hidden terminal problem* due to the low SNR of the received CTS packet, and the *CTS collision problem* due to the low SINR of the received CTS packet. Both problems make the CTS packet undecodable at hidden terminals under low SNR/SINR environments.

1.1.2.2 The Approach to Solve the Low SNR/SINR Problems

To effectively solve the two problems, I propose a novel cross-layer structure of RTS/S-CTS mechanism that uses global-known bits sequences to carry the NAV time information. I catalogue the NAV time durations and use different bits sequences, which are called *S-NAV indicators*, to represent different catalogued NAV time durations. These

indicators can be detected and identified at symbol-level by a node under low SNR/SINR environments. The RTS/S-CTS frames require no change to the standard RTS/CTS packets at the MAC layer, but just append a new "S-NAV" field to the tail of the CTS frame at the PHY layer. The RTS/S-CTS handshake is same as the standard RTS/CTS handshake except that I devise a *symbol-level detection decoder* (SLDD) at the PHY layer to detect the S-NAV indicator, together with an *NAV decision algorithm* to pass the NAV time information up to the MAC layer. As the S-NAV does not have to be decoded into bits, the RTS/S-CTS can be compatible with current 802.11 MAC protocol.

Compared with the standard RTS/CTS mechanism, the RTS/S-CTS mechanism has following key features:

(1) The S-CTS works at the *symbol level*, i.e., the S-CTS frame's detectable range is enlarged from the data transmission range to the interference range, which is controlled by tuning the detectable threshold β_{S-NAV}. By contrast, the standard CTS works at the *bit level*, i.e., the CTS packet can only be correctly decoded within the data transmission range.

(2) The RTS/S-CTS mechanism can achieve good performance even under low SNR/SINR environments. It uses the symbol level correlation method to extract the S-NAV indicator's information from the S-CTS frame. Thus, the RTS/S-CTS mechanism can alleviate the problems in the existing hidden terminal solutions.

1.1.3 The Energy Inefficiency of the Packet Overhearing Problem

1.1.3.1 Energy Efficiency and the Packet Overhearing Problem

Energy efficiency is a critical issue of wireless devices. Generally speaking, due to the broadcast character of the wireless transmission medium, the WiFi protocols would deploy CSMA mechanism to sense the channel, which inevitably causes the energy-inefficiency on *channel sensing* and *packet overhearing* [16, 17]. The continuously channel sensing on sending packets or waiting for incoming packets would cause energy-inefficiency on the mobile devices, because the energy cost of the channel sensing is comparable with packet reception [10, 40]. Moreover, when a packet is broadcast in the wireless channel, all WiFi devices which sense the packet would receive and decode the packet even if the packet is not addressed to it. Despite the packet would be eventually dropped at the MAC layer after the receiver checks the packet's receiver MAC address, the energy spent on receiving this packet has already been wasted, and this energy-inefficiency of the unnecessary packets receiving, which is called *packet overhearing*, can be even worse if the wireless devices are placed in a high traffic wireless LAN [2, 7, 47, 83].

Some well-known solutions to alleviate the energy-inefficiency are to build MAC-layer Power Saving Model (PSM) protocols [2, 7, 37, 48, 67] which schedule the WiFi stations to periodically wake up to exchange the control packets with AP. Then, the stations decide whether to stay active to receive the data packets or to switch to the sleeping mode to save the energy. The PSM protocols can classify the network traffic and reduce the energy-inefficiency of channel sensing on waiting for incoming packets. However, it cannot reduce the energy-inefficiency caused by the channel sensing on sending packets

and unnecessary packet overhearing [2, 83].

Recently, another energy-efficient approaches have conducted at the PHY layer. As the power consumption of wireless devices is known to be proportional to its voltage-square and sampling clock-rate, putting the stations in the low voltage or downclocking rate during the channel sensing and packet overhearing can significantly save the energy [10,20,47,56,70,83]. A state-of-art work–E-MiLi [83] was proposed to let the WiFi chipsets work at the downclocking mode during the channel sensing or packet overhearing, and restore to the fullclocking mode when detecting the intended incoming packets. Although E-MiLi alleviates the energy consumption on the packet overhearing, it still needs to spend extra energy on sampling the unnecessary packets continuously.

1.1.3.2 The Approach to Reduce the Energy Waste on Packet Overhearing

To effectively reduce the energy consumption on channel sensing and packet overhearing, in this thesis I propose the Sample-Address Sample-Duration (SASD) scheme, which leverages the advantages of both the downclocking mode and sleeping mode for saving the energy of wireless devices. SASD can put a device in the downclocking mode during the channel sensing and turns the device into the sleeping mode when it detects that the transmitted packet is not addressed to it. To achieve this, two critical packet information, the packet's receiver MAC address and transmission duration, must be carried in the packet frame at the PHY layer and can be detected without the whole packet being received, specifically in the downclocking mode. Thus, SASD designs a MAC address mapping method that uses the local unique fixed-length bits sequence, called as *Sample-Address* (SA), to represent the MAC address. The station involved in the packet receiving

can detect the corresponding SA information at PHY layer, switches to the fullclocking mode and waits for the incoming packet. On the other hand, to reduce the energy cost on the packet overhearing, SASD adopts a "catalogued duration time" method that divides all possible transmission durations of a packet into catalogues and uses different global unique fixed-length bits sequences, called as *Sample-Durations* (SDs), to represent these transmission duration catalogues. Stations which cannot detect the SA information would turn to detect and identify the SD information in the downclocking mode at PHY layer, switches to the sleeping mode for the catalogued duration time to avoid the energy inefficiency of the packet overhearing problem.

1.1.4 The Wireless Communications Security Problem

1.1.4.1 The Problem in the Existing Physical Layer Security Techniques

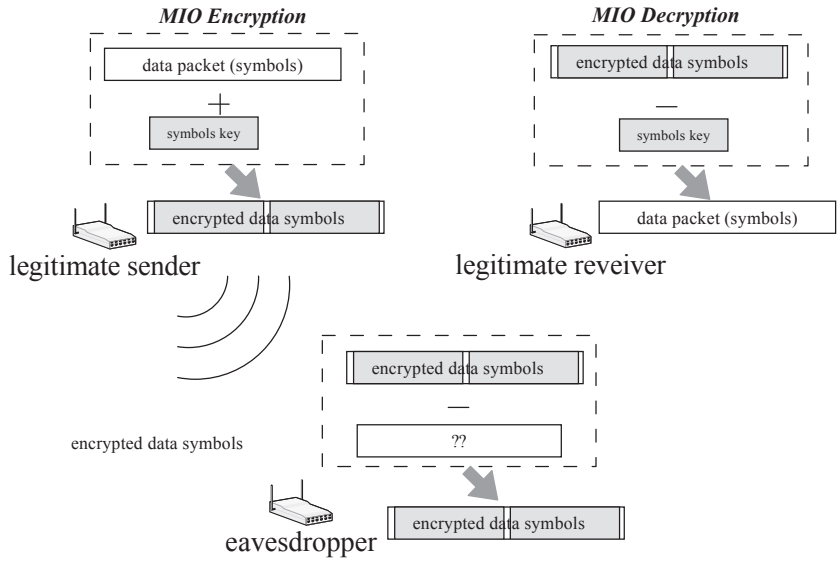
The wireless communications security is a critical and increasingly challenging issue in wireless networks since the users might transmit their sensitive personal information (e.g., credit card details) over the wireless networks. In addition, wireless channels are susceptible to eavesdropping [75] and malicious message injecting [28] due to the openness and sharing of the wireless medium.

Recent research has shown that physical layer security techniques become a more essential part in the wireless communications [27, 28, 30, 36, 49, 62, 71, 73, 79]. Compared with the traditional asymmetric/symmetric cryptographic techniques, such as RSA, DES [65, 66], that provide the computational secrecy, it has been proved that, physical layer security techniques, such as using a proper channel coding, can achieve the information-theoretic secrecy which makes the eavesdropper hardly break the encryption

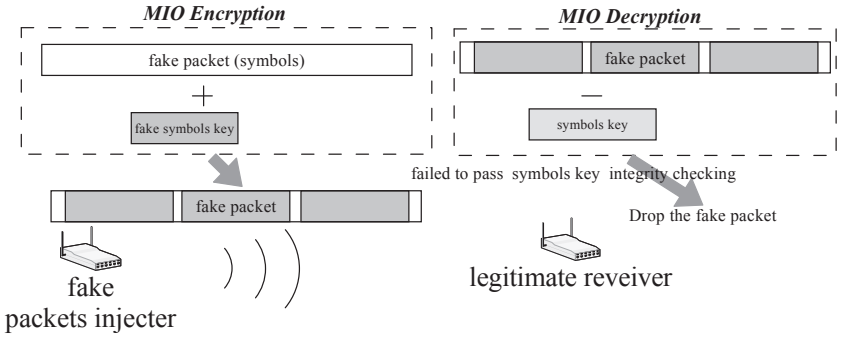
even it has unlimited computing power. However, the information-theoretic secrecy requires a strict positive secrecy capacity that the legitimate transmitter and receiver have to be in a better quality channel than the attacker [8,14,78]. Later works have shown that by artificially interfering the transmitting signal, the positive secrecy capacity requirement can be achieved [27,28,30,49,73] in practical wireless communications. But, most of these techniques need to deploy trusted third parties [27, 28, 30, 49] or multiple antennas (MIMO) [61] to generate the artificial noise. Moreover, the positive secrecy capacity of these works may be compromised if the eavesdropper locates at certain locations.

1.1.4.2 The Approach to Enhance the Wireless Communications Security

In this thesis, I adopt a Multiple Inter-symbol Obfuscation (MIO) scheme to enhance wireless communications security at the physical layer. In MIO, upon sending each data packet, a random subset of the corresponding data symbols are obfuscated with a set of artificial noisy symbols, which is called *symbols key*, so that (1) the eavesdropper's channel quality is worse than the legitimate receiver's and (2) the eavesdropper cannot decrypt the data symbols correctly since it does not know the symbols key, which is updated dynamically during the data packets' transmissions. For the legitimate receiver, it can offset the obfuscation of the symbols key by employing the reversed symbols key to derive the intended data symbols from the legitimate transmitter. In addition, the legitimate receiver can discern the fake packets sent from the adversary as it will fail the integrity check of the symbols key on the fake packets through symbol cross-correlation. Fig. 1.1 provides an overview about how MIO defends against both the passive eavesdropping attack and fake packet injection attack. The MIO can provide the better performance on computational secrecy against the fake packet injection attack, and an easier way to



(a) MIO against the passive eavesdropping attack.



(b) MIO against the fake packet injection attack.

Figure 1.1: The overview of Multiple Inter-symbol Obfuscation (MIO).

achieve information-theoretic secrecy against the passive eavesdropping attack. Thus. MIO can efficiently enhance the wireless communications security.

1.2 Contributions of the Thesis

The main contributions of this dissertation can be summarized in two aspects: the design contributions and the hardware experimental contributions.

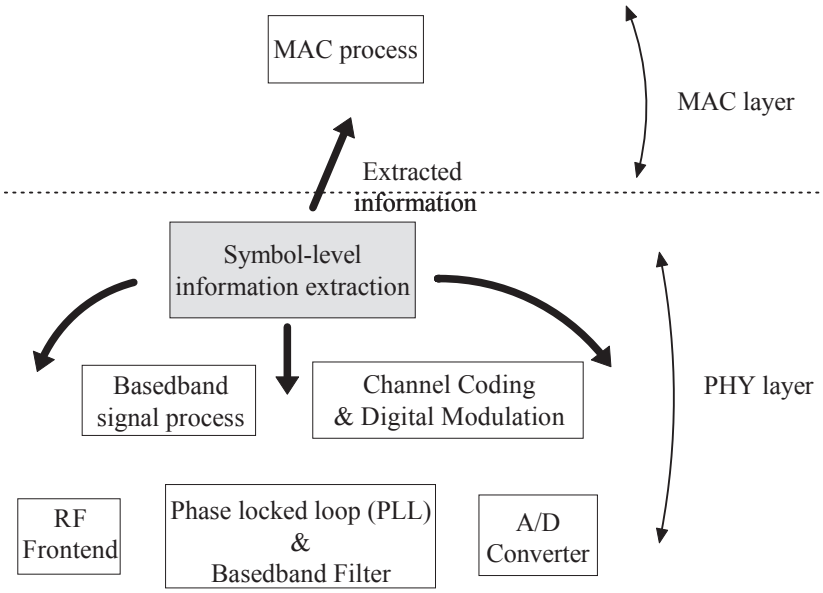


Figure 1.2: Design contributions of this dissertation. At the physical layer, I have implemented the symbol-level information extraction mechanisms (gray block) to solve the problems in the three popular research topics. The extracted information can be passed to the MAC layer, so that the MAC layer can give the corresponding responses and adjust the receivers behaviors.

- The design contributions: The design contributions are shown as in Fig. 1.2. Although the symbol-level information extraction mechanisms are implemented at the PHY layer, the actual design structures are slightly different from each other. Moreover, I focus on the techniques of this cross-layer design of symbol-level information extraction mechanism that can provide a new dimension to convey the vital information, e.g. MAC address information, packet transmission duration, or the confidential information from the bit level to the symbol level.

- The hardware experimental contributions: I have implemented and evaluated the symbol-level information extraction mechanisms on the hardware testbed with USRP2/GNURadio. The results show that the feasibility of the symbol-level information extraction mechanisms. Although there are still lots of work to do, the symbol-level

information extraction mechanisms show its potential in the indusial area.

The details of the contributions for each problem are listed in the rest of this section:

1.2.1 Contributions in Solving the Hidden Terminal Problem

The main contributions of solving the hidden terminal problem are summarized as follows:

- I propose the RTS/S-CTS scheme to deliver the NAV time information using symbol-level information encoding and decoding, and show the feasibility of delivering such information at the symbol level.
- I design, implement, evaluate and analyze the RTS/S-CTS on a 4-node testbed with GNURadio/USRP2 soft defined radio. The results demonstrate the feasibility of the RTS/S-CTS to combat both remote hidden terminal and CTS collision problems.
- To reveal the performance improvement, I conduct simulations for various network topology scenarios in ns-2. The results show that the RTS/S-CTS can achieve more than 63% performance improvement compared with the standard RTS/CTS in the random network topology scenario.

1.2.2 Contributions in Eliminating the Energy Inefficiency of the Packet Overhearing Problem

The main contributions of solving the energy inefficiency of the packet overhearing problem are summarized as follows:

- I introduce the SASD (Sample-Address Sample-Duration) scheme, which can deliver the packet's receiver MAC address and transmission duration information in one

shot at the PHY layer. Moreover, the SASD scheme combines the advantage of both the downclocking mode and sleeping mode for energy efficiency.

- I design, implement, evaluate and analyze the SASD scheme on a 3-node testbed with USRP2/GNURadio. The results demonstrate the feasibility of the SASD to reduce the packet overhearing problem under the downclocking mode.

- To reveal the performance improvement, we conduct simulations for various network topology scenarios in ns-2. The results show that our SASD can outperform the CAM (76.3%), PSM (71.8%), E-MiLi (64.5%) and PSM+ E-MiLi (65.3%) in term of energy usage with slight negative influence on the network throughput in the WLAN scenario.

1.2.3 Contributions in Enhancing the Wireless Communications Security

The main contributions of enhancing the wireless communications security are summarized as follows:

- I propose the MIO scheme that combines the data symbols encrypting and channel interfering at one step. In MIO, the symbols key not only encrypts the baseband data symbols but also interferes these symbols, which guarantee that the secrecy capacity of the wireless communication would always stay positive regardless of the location of the eavesdropper. Thus, the information-theoretic secrecy can be achieved. On the contrary, the traditional bit-level symmetric/assymmetric cryptographic schemes cannot guarantee this feature. Also, compared with other physical layer security schemes, MIO does not have to concern or assume the channel state

information (CSI), as the noisy symbols key has interfered the eavesdropping channel regardless of the location, which makes that the legitimate channel is better than the eavesdropping channel. Moreover, MIO does not need any trusted third party to deploy the noisy symbols key, and the secrecy capacity will not be decreased by the location of the eavesdropper.

- I design a dynamic key extraction mechanism to change the artificial noisy symbols key to defend against the eavesdroppers from retrieving the correct information of symbols key in MIO. In this mechanism, as the legitimate transmitter can randomly encrypt the data symbols without notifying the legitimate receiver, the receiver has to employ the key checking process to locate the symbols key's position and the corresponding dynamic encrypted symbols. Consequently, MIO can allow the legitimate receiver to decrypt those encrypted symbols without any further information.

- I prove that, without considering the initial key, MIO scheme can provide information-theoretic secrecy against the passive eavesdropping attack and computational secrecy against the fake packet injection attack. In addition, this information-theoretic secrecy would not be compromised by the location of the eavesdroppers. Moreover, we show that MIO can defend against the symbol detection attempts as well as the acknowledgement-based key disruption attack. Thus, MIO can greatly enhance the wireless communications security.

- I evaluate MIO's performance with the USRP2 [19] testbed and Simulink tools to further validate the effectiveness of MIO in real wireless environments.

1.3 Organization of the Thesis

The structure of this thesis is illustrated in Figure. 1.3. Chapter 1 is the introduction to this thesis. Chapter 2 briefly presents the literature review on the hidden terminal problem, the energy efficiency problem and the wireless communications security at the PHY layer respectively. Also some background knowledge related to the research issues in this thesis are briefly introduced in this chapter. The main body of this thesis is from Chapter 3 to Chapter 5. In Chapter 3, I detail a new approach of the symbol-level detection approach that solves the hidden terminals problem. Chapter 4, the energy inefficiency of the packet overhearing problem is discussed, along with the SASD (Sample-Address Sample-Duration) solution. In Chapter 5, the wireless communications security problem is discussed and I propose the MIO (Multiple Inter-symbol Obfuscation) solution to enhance the wireless communications security in this chapter. Chapter 6 summarizes this thesis and directions for future research.

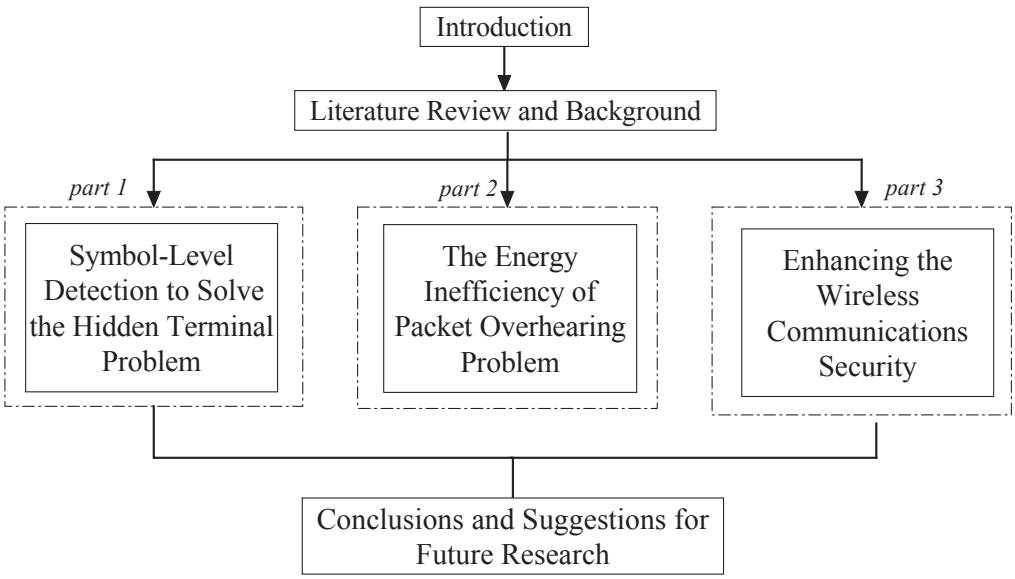


Figure 1.3: The structure of this thesis.

Chapter 2

Literature Review and Background

In this chapter, we provide the literature review of the related topics and some necessary background knowledge for the research in this thesis. The organization of this chapter is as follows. We first review the related work about the hidden terminal problem in Section 2.1. Then we review the related work about the energy efficiency in wireless communication in Section 2.2. In Section 2.3, we review some former works related to the wireless communication security. Finally, we introduce some background knowledge about the experimental hardware, the digital signal processing and the cross-correlation operation which are related to the research issues in Section 2.4 and 2.5, respectively.

2.1 Existing Work about Hidden Terminal Problem

In this section we briefly review some prior work related to the hidden terminal problem. The hidden terminal problem has been well-studied in the past two decades. Most solutions to the hidden terminal problem work at the MAC layer. MACA [43] proposes a mechanism using the RTS/CTS exchange without carrier sense to reserve the wireless channel. MACAW [5] revises the MACA and uses the ACK packet to acknowledge the

successful reception of data transmission. However, both of them assume the RTS/CTS exchange can perfectly received by hidden nodes, which is not likely the case in the wireless network for most of the time. Moreover, it introduces extra control overhead when the size of data packets is relatively small. Fullmer and Garcia-Luna-Aceves further proposed FAMA family MAC protocols (FAMA [22], FAMA-PJ [21] and FAMA-NCS/NPS [23]), which require the length of the RTS/CTS packets to be larger than a fixed size due to the awareness of RTS/CTS packet collisions. This partially solves the RTS/CTS collision. These MAC protocols relies on the virtual carrier sensing. However, they all suffer the Low-SNR/SINR-CTS drawbacks (details in Section 3.1), which motivates us to resort to new solutions to the hidden terminal problem.

Xu et al. [80] revealed the remote hidden terminal problem and proposed two solutions, selective response to RTS request and directional antenna. The former solution requires that a CTS can be granted only if the RTS's energy level is higher than a threshold. As a result, this solution reduces more than half of the effective data transmission range, which sacrifices the network connectivity. Moreover, it cannot defeat the CTS collision problem at the same time. The directional antenna could be a solution to both remote hidden terminal and CTS collision problems. However, because the beam-width of directional antenna is narrow, it requires 5$\sim$9 times more CTS retransmissions to cover the whole region, and may cause the jamming problem if they work with omni-antennas [12]. Also, it is costly to equip a directional antenna into wireless devices.

Recent studies exploited a new form of interference cancelation strategy [29, 33, 38, 44, 45, 82]. Instead of avoiding collision, the new strategy tries to reconstruct the collided packets at the PHY layer by using some known symbol level information. Jamieson and

Balakrishnan [38] proposed a partial packet recovery mechanism to recover the whole packet via SoftPHY. The SoftPHY interface collects bits information and requires the transmitter only to retransmit corrupted bits for saving wireless network bandwidth. ANC [44] provides an algorithm for canonical 2-way relay transmission; it doubles the capacity of typical 2-way network by designing an analog sample coding algorithm. But, it is based on the assumption that the receiver has already known one of the collision packets, and not suitable for the random network. ZigZag [29] works under 802.11 protocol scenarios and deals with general collisions. However, it can only perform well in the AP-Station mode and the collided packets require retransmitting multiple times.

Busy tone has been proposed in [32, 72] to silence the hidden nodes. In [32], the busy tone message is transmitted to hidden nodes through a separated control channel, which wastes the wireless spectrum. In [72], to improve spectrum efficiency, the wireless device sets up a full-duplex channel using two antennas. However, the transmitter and receiver have to stay within a short distance so as to decode the data signal correctly.

More recently, a new form of collision avoidance called CSMA/CN has been proposed in [68, 69]. Instead of collision avoidance, it uses a collision notification packet to send out the packet collision information, so that the transmitter can stop transmitting the data immediately. While it implements a kind of CSMA/CD mechanisms in wireless environments, it does not alleviate the hidden node problem, and would still be interfered by hidden nodes.

2.2 Existing Work about Energy Inefficiency of the Packet Overhearing Problem

The energy efficiency of WiFi devices has been a popular research topic in the research community of wireless networks. Most solutions to the energy inefficiency problem work at the MAC layer. A well-known power saving mode (PSM) protocol [37] in 802.11 standard requests the AP to buffer all the packets for the stations which work at the power saving mode and the stations to periodically wake up to receive a Traffic Indication Map (TIM) notice in the AP's beacon messages. If the station's corresponding TIM field is set to 1, it sends back a PS-POLL packet to the AP and prepares to receive the buffered packet from the AP. The PSM would cause a long response delay that particularly affects the QoS of the time sensitive applications [1]. Consequently, most WiFi devices would implement the adaptive PSM [48] that devices can switch between the constantly awake mode (CAM) and PSM based on certain mechanisms, so that the QoS would not be degraded in the adaptive PSM. However, it cannot prevent the stations from receiving unnecessary packets when it switches to the CAM.

NAPman [67] improves the energy-efficiency of the above PSM protocols by isolating the traffic of PSM clients, so that stations can reduce the time staying in the high power consumption CAM. Based on NAPman, Manweiler and Choudhury [57] proposed a TDMA manner mechanism that isolates the traffic from different wireless LANs, the stations would wake up in each scheduled time and receive the packets. μPM [54] proposed to save the energy by aggressively switching the stations to the sleeping mode in a very short interval. It uses a prediction mechanism to exploit short idle intervals and does not need any special support from the AP. However, it has to use the packet retransmission

mechanism to recover the missing packet.

Recent studies exploited a new form of energy-efficiency at the PHY layer. As the main energy consumption of a wireless card is caused by the modern wireless digital circuits and is proportional to $V_d^2 f$, where V_d is the supply voltage and f is the sampling rate [10, 15, 20], by reducing the voltage of radio circuits or decreasing the sampling clock-rate, the power efficiency of whole wireless system can be achieved [56].

The wake-on-wireless [70] employs a low power radio circuit to detect the packet, and switches to the full power circuit to receive the incoming packets. The system would keep in a low energy consumption in channel sensing. But the devices need a secondary low power circuit to implement the packet detection, and also, it cannot solve the packet overhearing problem.

Kim et al. [47] proposed a similar solution as [70], instead of using the secondary low power circuit to detect packets, it deploys the low power sensor, which is called accelerometer, to sense the channel. As a result, the overall energy cost at the PHY layer can be reduced. However, as the sensor is using the energy-based detection to sense the channel, the false negative/positive error cannot be controlled. Thus, it would cause the unfair transmission in busy networks and the QoS cannot be guaranteed.

R. Chandra et al. [10] conducted the hardware experiment on the energy cost of Atheros NIC cards. It revealed that the energy consumption of the wireless card can be reduced when the A/D clocking-rate (channel bandwidth) goes down. However, this paper is more focusing on the adaptive channel bandwidth usage, and does not discuss the packets' detection method under the downclocking rate.

Most recently, Zhang and Shin [83] proposed a state-of-art PHY layer energy saving

mechanism–E-MiLi. Similar as [10], it linearly decreases the A/D converter's sampling clocking-rate, and adds a M-preamble field to notify the station whether the incoming packet is addressed to it. The stations which are not involved in the packets receiving can still in the downclocking mode to save the energy on the idle listening. E-MiLi needs no extra circuits compared with wake-on-wireless [70] or sensors [47], and because it applies at the PHY layer, it can combine with other MAC layer PSM protocols. By deploying the A/D converter's downclocking rate 2, E-MiLi can save 36% energy cost of wireless NIC card. After that, Zhang and Shin [84] proposed the Gap Sensing which uses the same method of E-MiLi to detect packets and save energy between heterogeneous wireless devices.

However, all these MAC layer or PHY layer power saving protocols cannot avoid the energy consumption on packet overhearing in the high dense and traffic wireless LANs [2, 83]. To avoid this significant energy inefficiency on packet overhearing, Biswas and Datta [7] proposed a RTS/CTS based mechanism that forces the stations switch to the sleeping mode if the packet is not addressed to it. The approach relies on the receiving of the RTS/CTS packets at the MAC layer. It cannot take the advantage of the PHY layer energy saving. Also, as the high overhead of the RTS/CTS mechanism, it would also degrade the QoS at the clients side.

2.3 Existing Work about Wireless Communication Security at the PHY Layer

Although the communications security has been a popular research topic in the research community of wireless networks, the development of wireless communications security,

particularly in the physical layer, remains at its early stage. Prior physical layer security research mainly falls in the following three areas, *channel coding approaches*, *signal design approaches*, and *artificial noise approaches* [71]:

1) *Channel Coding Approaches*: Channel coding approaches can defeat packet interception and jamming problems. Code Division Multiple Access (CDMA) [6, 11, 51, 53] is a well-known channel coding scheme in the wireless communications security area. By using the bit level Pseudo Noise code (PN code), the encrypted transmission message can only be decrypted by the legitimate user. However, traditional CDMA has limited PN codes, and users have to share those PN codes. To solve this PN code size problem, Li et al. [53] enhanced the CDMA security based on the advanced encryption standard (AES) operation. It specifies 3 different AES-CDMA PN code sizes (128, 192, and 256 bits) to raise the security level against eavesdropping. Unfortunately, like other channel coding approaches, this long size security code lags the wireless transmission rate, reduces the network goodput. Moreover, CDMA is spread-spectrum multiple access techniques which data for transmission is combined via bitwise XOR with the PN code. Thus, CDMA is still a kind of bit level symmetric cryptographic techniques which cannot provide the information-theoretic secrecy in the wireless communication security.

Liu et al. [55] shown that the low-density parity-check (LDPC) code can achieve the secrecy capacity of the wiretap channel, and proved this code can be used to provide perfectly secret communications at low data rates. However, it is under the assumption that the main channel must be noiseless than the eavesdropping channel and the eavesdropping channel is a general binary-input symmetric-output memoryless channel, which can hardly be true in the real wireless communications environment.

2) *Signal Design Approaches*: The advantage of the signal design approaches is that, by designing a different signal constellation mapping method, the eavesdropper cannot correctly map the received digital symbols into bits, which leads to the incorrect decoding of the packets. Pöpper et al. [62] proposed the symbols flipping method by rotating a preset angle for the baseband data symbol vectors before transmissions. The legitimate receiver can retrieve the data symbol vectors by reversing the angle rotation. However, the rotating angle in their scheme is fixed and the eavesdropper can brute-force the rotating angle by intercepting sufficient data packets for demodulation.

Different from Pöpper's work, Husain et al. [36] proposed a constellation diversity mapping method to secure the wireless transmission. It increases the BER at the eavesdropper side by using different constellation maps (e.g., Circular constellation changes to Rectangular constellation) in wireless transmissions (under Gaussian noise), this constellation diversity mapping can hardly be detected by normal symbol detection attempts [59,63]. However, this scheme is demonstrated to be more suitable for the complex modulation, like M-QAM. Moreover, the information-theoretic secrecy can be compromised under certain specific symbol detection attempts.

3) *Artificial Noise (AN) Approaches*: Recent studies [27, 28, 30, 49, 73] exploit the advantage of deploying artificial noise that can easily make the intruders' channel more noisy than the legitimate users' channel to achieve the information-theoretic secrecy. Sperandio and Flikkema [73] proposed to obfuscate the original signal by imposing the multiple orthogonal artificial noise through the multi-path transmissions. The receiver can retrieve the correct signal by having multiple orthogonal noise to offset each other while the eavesdropper is not able to retrieve the correct signal without correct location.

However, their scheme is constrained to a static channel condition requirement for both the sender and receiver so that the receiver is able to receive the affected signals, together with that artificial orthogonal noise can offset each other through the multi-path effect. Such requirement on the static channel condition of sender and receiver might not be suitable for mobile networks.

Jorgensen et al. [42] proposed a wire-connected third party to send the synchronized artificial noise when intended receiver receives the packet. It uses the secrecy capacity to prove that the AN approach can achieve the information-theoretic secrecy. However, if the eavesdropper is more closed to the transmitter, the secrecy capacity of the scheme can decrease to 0, in which the information-theoretic secrecy is weakened by the locations of eavesdroppers.

Lai and Gamal [49] suggested deploying a trusted third party to send anti-artificial noise during the wireless transmission, thus, the useful information is hard to be intercepted. However, their scheme requires an additional device and static channel condition for the legitimate sender, receiver and trusted third party so that the anti-artificial noise can be synchronously offset with the artificial noise to retrieve the transmitted signals.

Gollakota and Katabi [30] adopted a redundancy mechanism to defend against the wireless signals' interception. Each signal will be randomly obfuscated with addictive noise and sent twice. The receiver is required to identify the obfuscated signal, and reconstruct the clean signal. Given the redundancy mechanism, the throughput is reduced; it also required the signal synchronization between sender and receiver. They further improved their work by employing the full-duplex hardware to impose the noise to the transmission between the Implantable Medical Devices (IMDs) and the sink. As a result,

the mission-critical commands to IMDs cannot be forged and overheard by the unauthorized third party [28]. Similar to Lai and Gamal's design [49], their scheme requires an additional hardware to jam the channel. Moreover, an adversary is able to overhear the transmitted signal if it is sufficiently close to the legitimate transmitter or to inject the unauthorized commands to the legitimate receiver if it is close enough to the legitimate receiver.

2.4 Background Knowledge about Software-Defined Radio and USRP2

As I said in Section 1.2, one of the main contributions of this dissertation is the hardware experimental contributions. I design and implement all the ideas on the GNURadio/USRP2 software-defined radio. Thus, in this section, I would give a brief background about the GNURadio/USRP2, which is a useful hardware tool to implement the software-defined radio.

Software-defined radio is the technique of getting your own code as close to the antenna as possible. It turns radio hardware problems into software problems. The fundamental characteristic of software-defined radio is that software defines the transmitted waveforms, and how to modulate/demodulate the received waveforms. This is in contrast to most radios in which the processing is done in either analog circuitry or analog circuitry combined with digital chips [19, 26]. Normally, it can be divided into two parts, the software radio part and the supported hardware part (Fig. 2.1).

The beauty of the software radio is that you can create your own radio code and implement it on the supported hardware. Compared with the old ways that people can only

simulate the idea in the Matlab or Simulink, the software radio is more close to the real world. Moreover, most software radio has already implemented many wireless standards, e.g. GPRS, 802.11 or MIMO. In this dissertation, we mainly use the GNURadio which is a free and famous software toolkit for building the soft radio[1]. As the GNURadio is the open source software, you can download and modify the standard codes for you own idea. The typical software radio block diagram is shown in the Fig. 2.1. The researchers can create, modify, store the digital signals for further study [26, 35].

Moreover, the supported hardware must be correctly chosen to run your own software radio code. Here, we choose the Universal Software Radio Peripheral 2 (USRP2) [19] with RFX2400 daughter-board as the supported hardware part because it is the newest hardware to support GNURadio in the year of 2010. All supported hardware should connect to the PC, so that the software radio code can be run on the PC. In the hardware experiment, the PCs have installed Ubuntu 10.04 as GNURadio can be only supported by the Linux system. The RFX2400 operates at the 2.4GHz frequency band. The DAC rate at the transmitter side is 400M samples/s; the ADC rate at the receiver side is 100M

[1]There are also other software radio, e.g. Sora system [58].

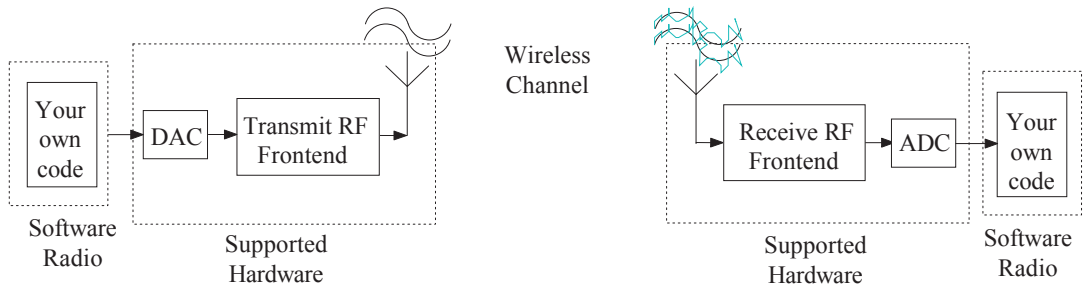


Figure 2.1: The architecture of the software-defined radio's transmitter and receiver side. For each side, the software-defined radio can be divided into the software radio part and the supported hardware part.

samples/s. Thus, the USRP2 can support 100Mhz bandwidth which is enough for any 802.11 standards. As the figuration and experimental methodology are different for each problem in this dissertation, I would give the details in the hardware experiment section in Chapters 3∼5.

2.5 Background Knowledge about Digital Signal Processing and Cross-Correlation Operation

In this section, we brief the digital signal processing and cross-correlation operation which are strongly related to this thesis. As the symbol-level information extraction is mainly accomplished by using cross-correlation between the incoming signal and the known sequence signal. Say that if the known sequence signal has l samples, the receiver aligns the first l incoming signal samples with the known l samples, computes the correlation value, then, shifts the incoming samples by one, and recomputes the correlation value. The correlation value reaches the peak when the incoming samples are perfectly matched to the known samples. Thus, a device can use this peak value to detect and identify this known sequence signal.

When a packet transmits over the wireless channel, the transmitter needs to modulate the digital bits to a series of digital baseband data symbols. These symbols will be further resampled as a sequence of baseband samples [34]. Mathematically, those digital baseband samples are represented as a stream of discrete complex samples.

$$x[n] \quad = \quad A[n]e^{j\phi[n]}, \tag{2.1}$$

where $A[n]$, $\phi[n]$ are the magnitude and the angle of the n^{th} resampled sample. Then, these samples are gone to the D/A converter and carried by the analog signal. It is noted

that the resampled samples are closely related to the input symbol. When the input symbol is changed, the corresponding resampled samples are also changed.

As the analog signal is transmitted over the wireless channel, at the receiver side, the RF down-converter samples the incoming analog signal, and the A/D converter derives a stream of discrete complex baseband samples. However, those received samples differ from transmitted samples. The received sample $y[n]$ can be represented as:

$$y[n] = Hx[n] + w[n]. \tag{2.2}$$

Here, H is also a complex number representing the channel coefficient between transmitter and receiver, $w[n]$ is the Gaussian noise at n^{th} sample.

In practice, samples are actually distorted due to hardware constraint and wireless channel effect: frequency offset, sampling offset, and inter-symbol interference [29, 46]. For example, considering the frequency offset between transmitter and receiver Δf over time Δt, should be:

$$y[n] = Hx[n] \cdot e^{j2\pi\Delta f\Delta t} + w[n]. \tag{2.3}$$

Assume $x_{kn}[i]$ is the i^{th} complex sample of the known sequence, i.e., $x_{kn}[i] = A_{kn}[i]e^{j\phi_{kn}[i]}$. $y_{tr}[i]$ is the incoming signal sample from the transmitter, $y_{int}[i]$ is the incoming signal sample from the interferer, l is the length of the known sequence and $0 \leq i \leq l-1$. The signal correlation value, $C_{kn}(l)$, can be calculated as:

$$C_{kn}(l) = |\sum_{i=0}^{l-1} \bar{x}_{kn}[i](y_{tr}[i] + y_{int}[i])| \tag{2.4}$$

Here, $\bar{x}_{kn}[i]$ represents the complex conjugate of $x_{kn}[i]$.

However, Eq. (2.4) cannot compute the correlation spike because the frequency offset

can destroy the correlation. The receiver needs to compensate the frequency difference Δf and shift each sample by $-2\pi\Delta f\Delta t$. Thus, from the above equations, we can have:

$$\begin{aligned} C_{kn}(l) &= |\sum_{i=0}^{l-1} \bar{x_{kn}}[i] \cdot (H_{tr} \cdot x_{kn}[i] \cdot e^{j2\pi\Delta f\Delta t} + y_{int}[i] + w[i]) \cdot e^{-j2\pi\Delta f\Delta t}| \\ &= |\sum_{i=0}^{l-1} \bar{x}_{kn}[i] \cdot H_{tr} x_{kn}[i] e^{j2\pi\Delta f\Delta t} \cdot e^{-j2\pi\Delta f\Delta t} + O(l)| \\ &= |\sum_{i=0}^{l-1} H \cdot |x_{kn}[i]|^2 + O(l)|. \end{aligned} \tag{2.5}$$

Here,

$$O(l) = \sum_{l=0}^{l-1} \overline{x_{kn}[i]} \cdot y_{int}[i] \cdot e^{-j2\pi\Delta f\Delta t} + \sum_{l=0}^{l-1} \overline{x_{kn}[i]} \cdot w[i] \cdot e^{-j2\pi\Delta f\Delta t}.$$

Since the known sequence is independent of the signal from y_{int} and the noise, as long as l is large enough, $O(l)$ would be close to 0. On contrast, when the known sequence is presented in the incoming signal samples, the correlation value C_{kn} will reach a spike:

$$C_{kn}(l) \approx |H| \cdot \sum_{i=0}^{l-1} |x_{kn}[i]|^2. \tag{2.6}$$

To detect the known sequence, a threshold β_{kn} is introduced to compare with $C_{kn}(l)$: If $C_{kn}(l) \geq \beta_{kn}$, the receiver detects the presence of the sequence; otherwise, the sequence is considered absent.

Normally, threshold β_{kn} can be defined as $\beta_{kn} = \psi \cdot l \cdot RSSI_{kn}$ [29, 68], where ψ is a constant (Normally, $0.55 \leq \psi \leq 0.95$) and $RSSI_{kn}$ is the received signal strength indicator of the known-sequence signal. Thus, the comparison inequality can be changed

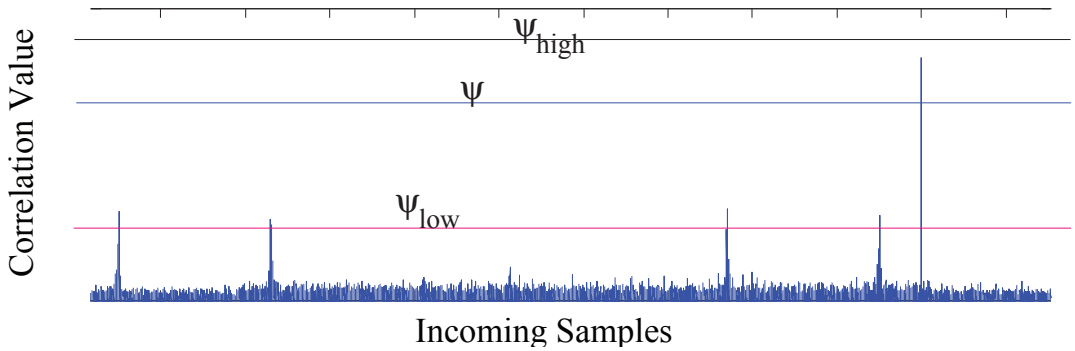


Figure 2.2: Two types of errors caused by the ψ. It can cause false negative (missing alarm) error if we use ψ_{high}, and false positive (false alarm) error if we use ψ_{low}

to:

$$\frac{C_{kn}(l)}{l \cdot RSSI_{kn}} \geq \psi. \tag{2.7}$$

We call $\frac{C_{kn}(l)}{l \cdot RSSI_{kn}}$ as "normalized correlation value".

Thus, we can use this cross-correlation operation to extract the vital information which we give to this known sequence signal. We call this "the known sequence signal's detection and identification". However, this method would have two kinds of errors, false negative (missing alarm) error and false positive (false alarm) error, which are caused by using improper ψ (Fig. 2.2). We give details that how to choose this parameter ψ wisely in the later chapters.

Chapter 3

A Symbol-Level Information Detection Approach against the Hidden Terminals Problem

Hidden terminals are typical interference sources that can significantly reduce the throughput of a wireless network if it adopts the CSMA/CA MAC protocol. The RTS/CTS mechanism is a well-known solution to this hidden terminal problem. However, some drawbacks of the CTS packet can make the standard RTS/CTS mechanism fail to silence all hidden terminals, and deteriorate the throughput of the wireless network. In this chapter, I present the RTS/S-CTS mechanism, a novel symbol-level detection mechanism that combats the drawbacks to solve the hidden terminal problem.

3.1 Two Drawbacks in the Standard RTS/CTS Mechanism

3.1.1 Remote Hidden Terminal Problem

The standard RTS/CTS works when a node's data transmission range is equal to its interference range. However, recent studies [23, 39, 60] have revealed that packets' trans-

mission power can raise the environmental noise of the wireless channel, which causes the node's interference range to be much larger than its data transmission range. To avoid interference, the distance between transmitter-receiver d and the distance between receiver-interferer d_i should satisfy $d_i \geq \sqrt[k]{\beta_{SNR}} \cdot d$, where β_{SNR} is the SNR threshold that the received packet can be correctly decoded into bits and k is the signal decay factor [80]. That means, when $d_i < \sqrt[k]{\beta_{SNR}} \cdot d$, the receiver cannot correctly decode the packet. Note that the impact of the interferer on the receiver's reception of the packet is related to the distance between transmitter and receiver.

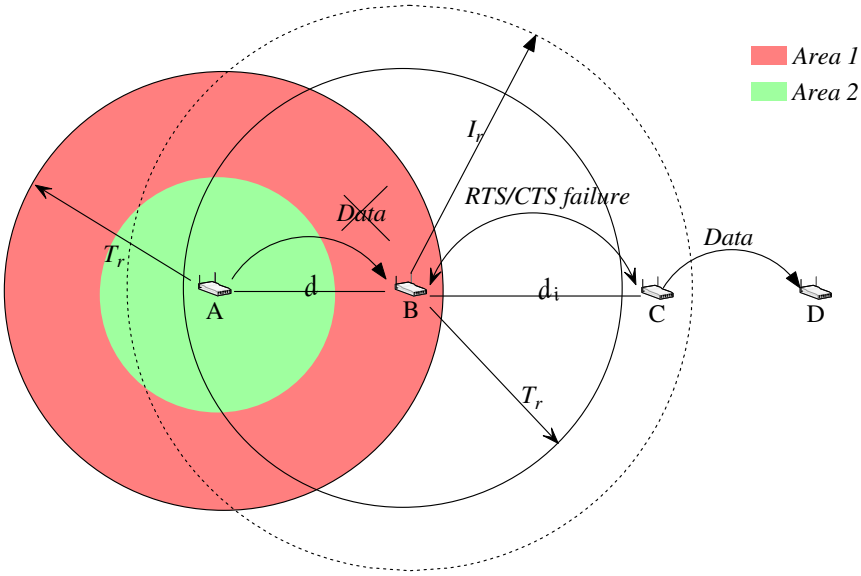


Figure 3.1: A remote hidden terminal scenario. Since the CTS packet from node B cannot be decoded by the remote hidden node C, the standard RTS/CTS handshake fails to silence node C. Node C's data transmission causes node A's data transmission corrupted at node B. Here, d and d_i satisfy the relationship $T_r < d_i < \sqrt[k]{\beta_{SNR}} \cdot d$.

Assume all nodes have the same data transmission range T_r. When $d \leq \frac{T_r}{\sqrt[k]{\beta_{SNR}}}$, the interference range I_r satisfies $d_i \leq I_r \leq T_r$, i.e., the interferer is within the data transmission range of the receiver. Since the CTS packet from the receiver can be correctly decoded by the interferer, the RTS/CTS handshake can successfully silence the hidden

interferer. In that case, there is no remote hidden terminal problem (i.e., *Area 2* in Fig. 3.1). However, when $\frac{T_r}{\sqrt[k]{\beta_{SNR}}} < d \leq T_r$, we get $T_r < d_i \leq I_r \leq \sqrt[k]{\beta_{SNR}} \cdot T_r$. As the interferer is a remote terminal that locates outside the data transmission range of the receiver, the SNR of the CTS packet at the receiver is below β_{SNR}. The RTS/CTS handshake fails to silence this remote hidden terminal. Thus, any transmission from this remote hidden node will cause the receiver's packet corruption (i.e., *Area 1* in Fig. 3.1). In this case, the standard RTS/CTS fails to silence node C's transmission to node D, which in turn causes node A's transmission to be collided at node B. We also call this remote hidden terminal drawback as *Low-SNR-CTS problem*.

3.1.2 CTS Collision Problem

The standard RTS/CTS mechanism makes the assumption that RTS/CTS packets would not collide with other packets. This assumption cannot hold when the network's workload becomes high. When multiple nodes concurrently transmit packets, the RTS/CTS packets may be collided with other packets. The corruption of RTS packet does not trouble the system much because, if the RTS is missed, the RTS/CTS handshake will re-initiate after a waiting time. However, the corruption of CTS packet at hidden node(s) can deteriorate the throughput of the system. If the CTS sent by the receiver fails to silence its neighbor(s) for enough NAV time, its data reception may be corrupted by the hidden node(s)' transmissions, which will force the transmitter to retransmit the entire data packet. To emphasize this problem, we also call this CTS collision drawback as *Low-SINR-CTS problem*.

A simple CTS collision scenario is illustrated in Fig. 3.2: Nodes A∼G are wireless

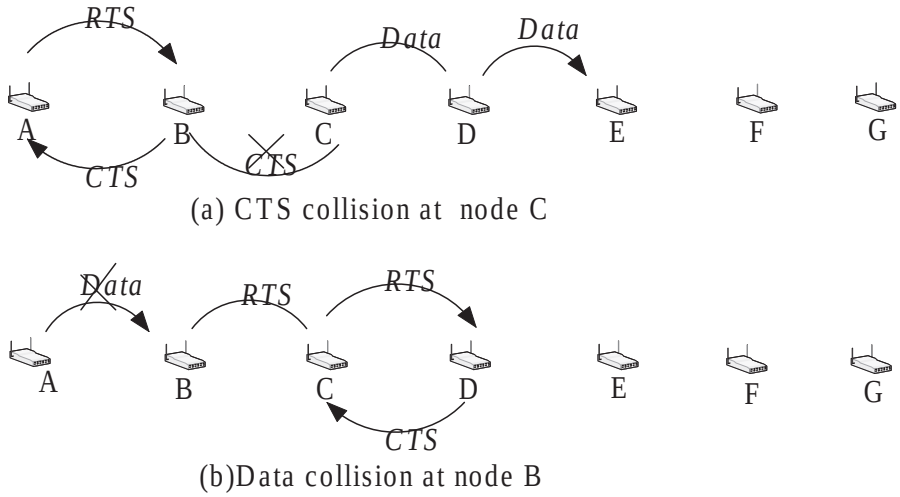


Figure 3.2: A CTS collision scenario: (a) Node B's CTS packet is collided with node D's data packet at node C. (b) Because the CTS packet is collided at node C in (a), node C cannot defer its transmission for the NAV time and node C's RTS packet will cause node A's data packet to be corrupted at node B.

stations that form a straight line topology. Fig. 3.2(a) shows that node B's CTS packet is collided with node D's data packet at node C. Due to the low SINR, the CTS packet cannot be correctly decoded by node C, and node C do not renew its NAV waiting timer. Hence, when node A transmits the data packet to node B, node C may initialize a new RTS/CTS handshake with node D, as shown in Fig. 3.2(b). Consequently, node A's data packet is collided at node B. Note that though the RTS/CTS packets may be hardly collided with each other due to their small packet sizes, the probability that the CTS packets are collided with data packets will be high when the network's workload becomes high.

The Low-SNR-CTS and Low-SINR-CTS problems are two main drawbacks that significantly deteriorate the performance of the standard RTS/CTS mechanism. To combat these two drawbacks, we propose a novel RTS/S-CTS mechanism to make the S-CTS frame detectable at the symbol level even under low SNR/SINR environments.

3.2 RTS/S-CTS Mechanism

Similar to the standard RTS/CTS, the RTS/S-CTS silences the neighboring nodes through the exchange of RTS/S-CTS frames: If a node that is not involved in the transmission can successfully decode or detect the RTS/S-CTS frames, it defers its transmission for the NAV time.

As we have addressed in previous section, because of the Low-SNR/SINR-CTS problems, the RTS/CTS cannot silence all interferers. On the contrary, the RTS/S-CTS can silence these interferers through the symbol-level correlation method. As the low-SNR/SINR-CTS problems only relate to the CTS frame, there is no change to the RTS frame format. To make the S-CTS detectable under low SNR/SINR scenarios, a new "S-NAV" field, which contains the symbol-level NAV time information, is appended to the CTS frame at the PHY layer, which makes the S-CTS packet same as the standard CTS packet at the MAC layer (Fig. 3.3). To specify the format difference of a packet between the MAC and PHY layers, we call it as "packet" at the MAC layer and as "frame" at the PHY layer in this dissertation.

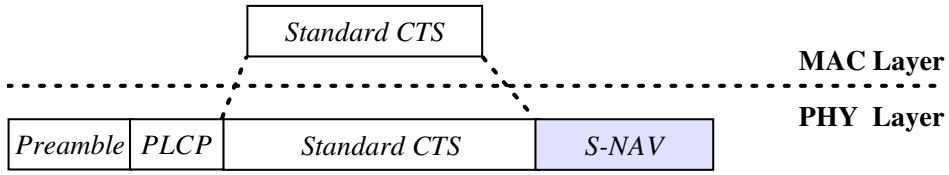
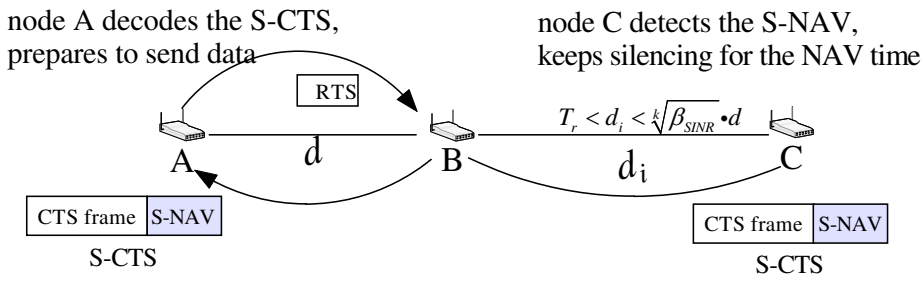


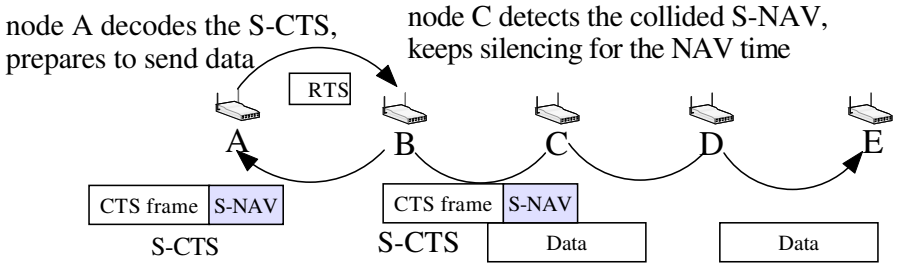
Figure 3.3: New S-CTS frame format at the physical layer. We do not change the standard CTS packet at the MAC layer, but only append a S-NAV filed at the PHY layer.

The new S-CTS frame can combat both Low-SNR/SINR-CTS problems (Fig. 3.4):

(a) For the Low-SNR-CTS problem case (Fig. 3.4(a)), because the distance between node B and node C, d_i, satisfies $T_r < d_i < \sqrt[k]{\beta_{SNR}} \cdot T_r$, node C cannot decode node B's



(a) Low-SNR-CTS scenario



(b) Low-SINR-CTS scenario

Figure 3.4: The RTS/S-CTS can combat both Low-SNR/SINR-CTS problems.

S-CTS frame into bits correctly. However, node C can detect the S-NAV field at the symbol level and obtain the NAV time information. Thus, node C can keep silencing for the NAV time.

(b) For the Low-SINR-CTS problem case (Fig. 3.4(b)), although node B's S-CTS is collided by node D's data transmission at node C, node C can still detect the S-NAV field at the symbol level, obtain the NAV time information, and keep silencing for the NAV time.

3.2.1 S-CTS Frame Generation

When a station receives a RTS packet, it checks if it is the designated receiver. If yes, it achieves the NAV time from the RTS's duration field, minus the time that is required

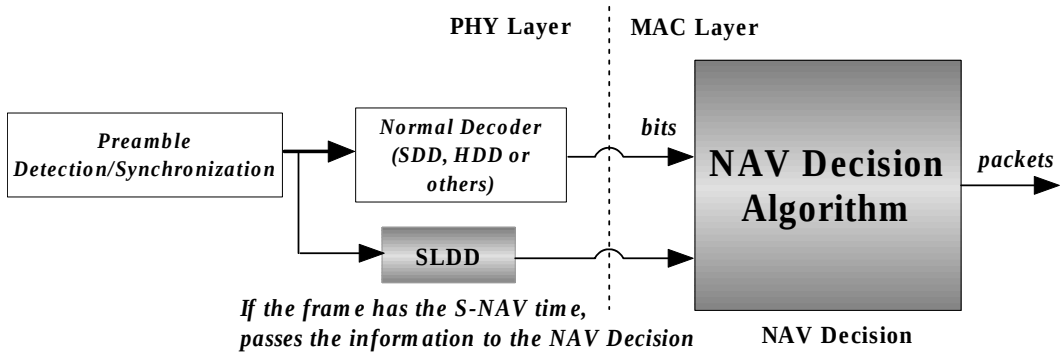


Figure 3.5: The receiving process of S-CTS frame. Gray blocks are the new components. The SLDD detects the S-CTS frame's S-NAV field and passes this S-NAV time information to the NAV decision block at the MAC layer.

to transmit the CTS frame and ACK frame plus three SIFS intervals. That means, only the time for transmitting the pending data is remained.

After the calculation, the receiver encodes the resultant NAV time, called "S-NAV time", into the S-NAV field (see details in Section 3.2.3), and appends this S-NAV field to a standard CTS frame at the physical layer to build a S-CTS frame. The receiver responses this S-CTS frame to the transmitter.

3.2.2 S-CTS Frame Reception

A station normally uses two mechanisms, hard decision decoder (HDD) and soft decision decoder (SDD), to decode the incoming signals at the PHY layer [38, 39]. Either one can deliver the bits up to the MAC layer correctly if received signals meet the SNR/SINR threshold requirement. However, as the S-CTS frame needs to be detected under low SNR/SINR environments, we design a new decoder, called *symbol-level detection decoder* (SLDD), to detect the S-CTS frame at the PHY layer, as shown in Fig. 3.5:

1) Under High SNR/SINR Environments: When the incoming signal meets the

threshold requirement, the correct bits are delivered to the MAC layer. For a received S-CTS frame, as the MAC layer can correctly obtain the standard CTS packet from the PHY layer, both the receiver address RA and the NAV time T_{NAV} are derived. In this scenario, the S-NAV time information in the S-NAV field becomes useless.

2) Under Low SNR/SINR Environments: Due to the low SNR/SINR, the hidden terminals cannot correctly decode the CTS packet at the MAC layer. However, for any node not involved in the data transmission, the only useful information carried by the CTS packet is the NAV time. This information is also presented as the S-NAV time information, which is stored in the S-NAV field and can be detected by the SLDD at the PHY layer (see details in Section 3.2.3).

When the NAV decision block only receives the S-NAV time information and no RA is obtained from the S-CTS frame, the station knows that it does not involve in the data transmission. Then the NAV decision adds the S-NAV time T_{S-NAV} achieved from the SLDD, together with the time required to transmit one ACK frame plus two SIFS intervals to calculate a new NAV time $T_{NAV-TIME}$. If the new NAV time is larger than

Algorithm 3.1 NAV Decision Algorithm

Input: The digital bits from the normal HDD/SDD decoders; the S-NAV time from the SLDD decoder.

1: **if** the packet is correctly decoded **then**
2: Extract RA and T_{NAV} from the digital bits; set $T_{NAV-TIME}$ to T_{NAV}.
3: **else**
4: Set RA to $NULL$; obtain T_{S-NAV} from the S-NAV field; set $T_{NAV-TIME}$ to $T_{S-NAV} + T_{ACK} + 2 \cdot T_{SIFS}$.
5: **end if**
6: **if** RA is the station's address **then**
7: Prepare for sending the data packet.
8: **else**
9: Renew the NAV-timer if $T_{NAV-TIME} > T_{NAV-timer}$; cease for $T_{NAV-timer}$.
10: **end if**

the time kept in the NAV-timer $T_{NAV-timer}$, the NAV decision renews the NAV-timer and ceases for the duration. The *NAV decision algorithm* is listed in Algorithm 3.1.

3.2.3 S-NAV Detection and Identification

S-CTS's two main challenges pertain to detecting the S-NAV field and discerning the different NAV time information, both of which depend on how a known sequence signal is detected. As describe in Section 2.5, by using the cross-correlation Ineq. (2.7), the known sequence can be detected.

$$\frac{C_{kn}(l)}{l \cdot RSSI_{kn}} \geq \psi.$$

However, as the S-NAV field plays a critical role in defeating the Low-SNR/SINR-CTS drawbacks. When receiving the S-CTS frame, a hidden terminal can achieve the S-NAV time information from the S-NAV field and keep silencing for the requested time. Thus, there are still two issues for the S-NAV detection and identification process: (1) How to provide enough global-known sequences, which are called *S-NAV indicators*, to carry different S-NAV time information? (2) How to identify those S-NAV indicators under low SNR/SINR environments? To solve these problems, we propose both *indicators mapping function* and *best candidate algorithm* to present and identify different NAV time information.

In the IEEE 802.11 standard, the size of MAC service data unit (MSDU) is limited to 2272 bytes. However, in the IPv4 and Ethernet standard (Version 2), the maximum transmission unit (MTU) cannot exceed 1500 bytes. Since the IEEE 802.11 MAC still uses IPv4 as its upper layer, the MTU is set as 1500 bytes. Consequently, the data transmission

time cannot exceed a maximum transmission time T_{max}. The indicator mapping function divides all possible data transmission time into N catalogues and maps each catalogued time T^i_{cata} to a global-unique S-NAV indicators. T^i_{cata} is calculated as follows:

$$T^i_{cata} = \frac{T_{max}}{N} \cdot i, \tag{3.1}$$

where i is the catalogued index of the S-NAV indicator and $1 \leq i \leq N$. Note that the indicators mapping function could keep the station(s) waiting longer than actual data transmission time, which is called "catalogue overhead" in this thesis.

When the station prepares for the S-CTS frame, it obtains the NAV time from the RTS frame, calculates the data transmission time, finds the catalogued time just longer than the data transmission time, maps the catalogued time to the corresponding S-NAV indicator, and stores that S-NAV indicator in the S-NAV field of the S-CTS frame at the PHY layer.

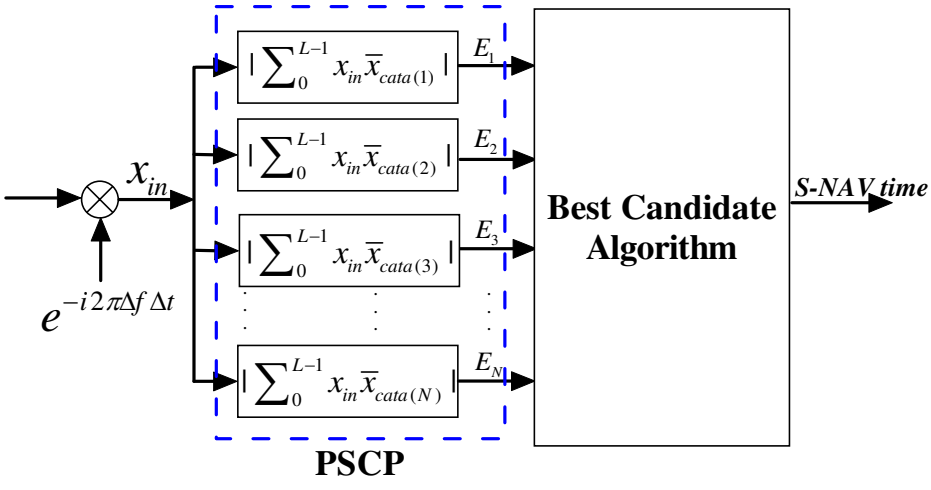


Figure 3.6: The structure of SLDD.

To decode the S-NAV time from the S-CTS frame, the SLDD resorts to a *parallel*

signal correlation process (PSCP) (Fig. 3.6): The SLDD correlates the incoming samples with N different S-NAV indicators and picks up the results that exceed β_{S-NAV}, which are called the *candidates*. The SLDD compares those candidates to find the one with maximum value E_i, which is the *best candidate*. The SLDD gets the index i and sets the S-NAV time T_{S-NAV} as the catalogued time T^i_{cata}. The best candidate algorithm is listed as Algorithm 3.2.

Algorithm 3.2 Best Candidate Algorithm

Input: Incoming symbol samples from the RF down-converter.
Output: The S-NAV time T_{S-NAV}.

1: Correlate the incoming samples with N different S-NAV indicators; pick up the correlation values that exceed β_{S-NAV} as the candidates.
2: **if** the number of candidates > 0 **then**
3: Compare those candidates and find out the one with maximum value E_i as the best candidate.
4: Get the catalogued index i of the best candidate; set T_{S-NAV} to the catalogued time T^i_{cata}; .
5: **else**
6: Do nothing.
7: **end if**

Fig. 3.7 illustrates the decoding procedure of SLDD with N S-NAV indicators. Different from the conventional correlation detections which only allow one correlation value to exceed the threshold, the SLDD allows several candidates to exceed the threshold β_{S-NAV}, and the best candidate algorithm can pick up the best candidate to output the S-NAV time.

3.3 Hardware Experiments

In this section, we reveal the hardware implementation and experimental methodology. Also, we discuss some practical issues about S-NAV detection and identification.

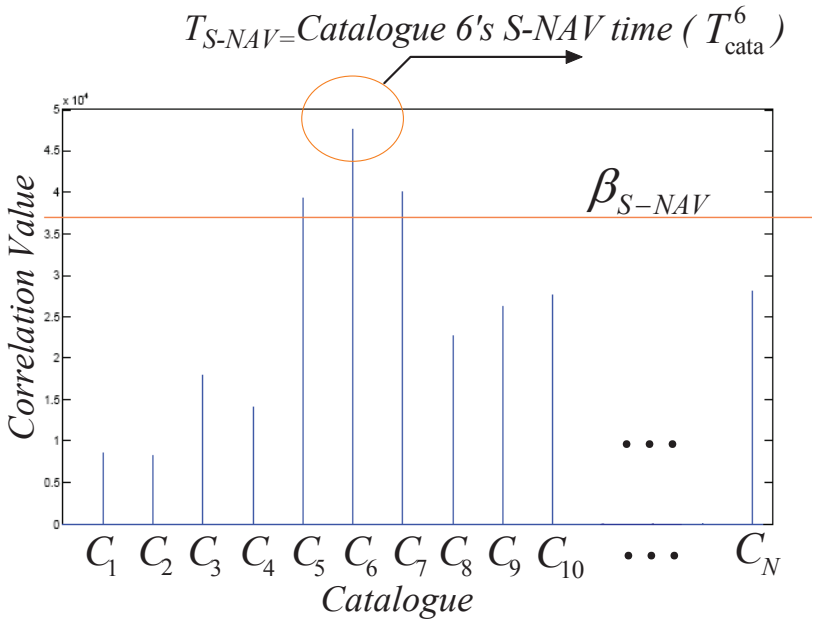


Figure 3.7: N-catalogued S-NAV indicators scheme. The example shows that catalogue 6 is the best candidate to be selected.

3.3.1 Hardware Implementation and Experimental Methodology

3.3.1.1 Hardware Implementation

We implemented the RTS/S-CTS mechanism on a 4-node GNURadio/USRP2 testbed. Each node is a commodity PC connected to a Universal Software Radio Peripheral 2 (USRP2) [19] with RFX2400 daughter-board. The RFX2400 operates at the 2.4GHz frequency range. All PCs are installed Ubuntu 10.04 and GNURadio [26].

The RTS/S-CTS uses the BPSK modulation/demodulation module, which is commonly used in the 802.11 standard. We used the default GNURadio configuration for the communications, i.e., on the transmitter side, the DAC rate is 400e6 samples/s, the interpolation rate is 200 (4 interpolation rate in the DAC chip itself and 50 interpolation rate controlled by GNURadio), and the number of samples per symbol is 2; on the re-

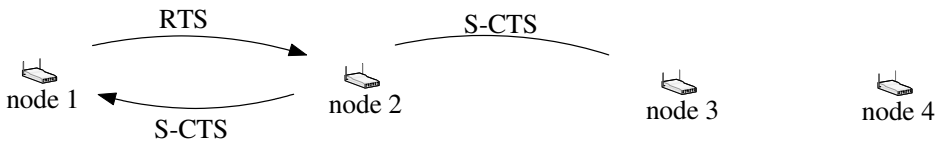


Figure 3.8: The network topology of the 4-node testbed.

ceiver side, the ADC rate is 100e6 samples/s and the decimation rate is 50. Given the above parameters and a BPSK modulation, the resulting bit rate is 1Mbps.

3.3.1.2 Experimental Methodology

USRP2 has hardware delays in transmitting samples from the RF front-end to its connected commodity PC, also GNURadio incurs artificial software delay to process these samples. Thus, it is difficult to conduct a real time evaluation of the RTS/S-CTS in high bit rates. Hence, we resorted to the trace-based evaluation that is also used in [29, 68]. Each node saves all the outgoing and incoming samples for off-line processing.

We set up the 4-node GNURadio/USRP2 testbed shown as Fig. 3.8: (1) To evaluate the remote hidden terminal problem, we increased the distance between nodes 2 and 3 to make them not communicate with each other. We made nodes 1 and 2 exchange RTS/S-CTS frames to set up link 1. Similarly, nodes 3 and 4 set up link 2. Here, node 3 is a remote hidden node of node 2. (2) To evaluate the CTS collision, we set the distance between nodes 2 and 3 less than the transmission range. Nodes 1 and 2 exchange RTS/S-CTS frames to set up a link, node 3 is a hidden node of node 1, and node 4 broadcasts some random data as an interferer, causing the CTS collision at node 3.

The design of S-NAV plays a crucial role in the proposed RTS/S-CTS mechanism. There are three factors that affect the design of S-NAV indicators: the length of S-NAV

indicators, the detectable threshold β_{S-NAV} of S-NAV indicators, and the minimum Hamming distance among S-NAV indicators. We conducted hardware experiments to study how these factors affect the S-NAV detection and identification under low SNR/SINR environments.

3.3.2 Length of S-NAV Indicators

A pertinent concern is how long the S-NAV indicator should be. Evident from Fig. 3.9(a), we can see that longer indicators can make higher normalized correlation values, which also make the indicators easily detected under lower SINR environments (Fig. 3.9(b)).

However, a longer indicator also means more channel occupation time. Table 3.1 gives the channel occupation time overhead with various indicator lengths.

Standards \ Indicator Length	80	160	240	320
802.11a/g	6.67	13.3	20	26.6
802.11b	7.27	14.55	21.82	29.09
	Time (Microsecond)			

Table 3.1: Channel occupation time overhead.

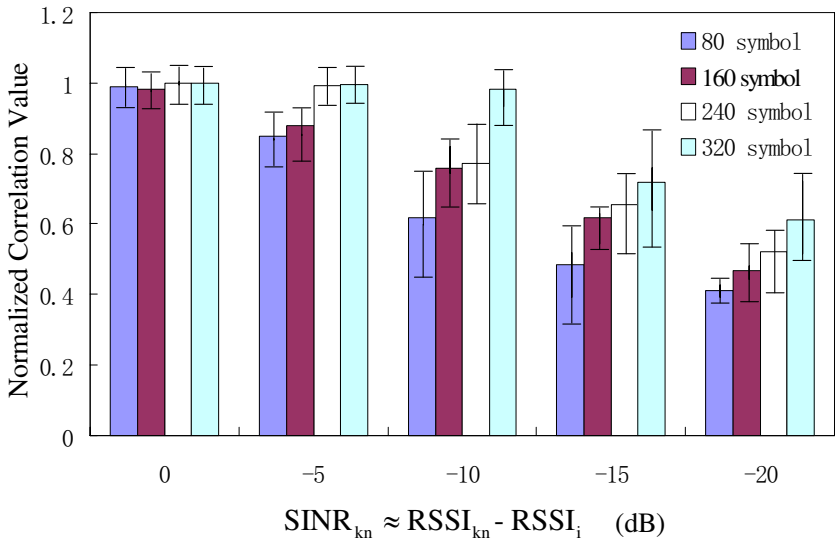
Indicator Length	80	160	240	320
Utilization	0.075	0.069	0.046	0.041

Table 3.2: Indicator length utilization.

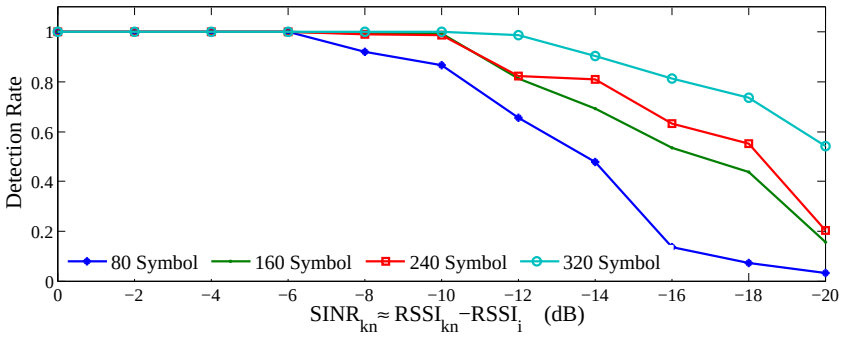
To make a trade-off between SINR and channel occupation overhead, a metric *indicator length utilization* U_l is defined as:

$$U_l = |\frac{SINR_{min}}{l}|, \tag{3.2}$$

where $SINR_{min}$ is the lowest SINR that, given length l, the indicator can be detected (with detection rate $\geq 95\%$, Fig. 3.9(b)). Table 3.2 gives the various indicator length



(a)



(b)

Figure 3.9: (a) Normalized correlation value with various indicator lengths under different SINRs. (b) Detection rate with various indicator lengths under different SINRs.

utilization. By further considering the available number of S-NAV indicators, in our testbed, we set the length of S-NAV indicator as 160 symbols.

3.3.3 Threshold β_{S-NAV}

Another key factor that affects the performance of S-NAV detection (false negative/positive error) is the threshold β_{S-NAV}. According to Ineq. (2.7), we can have

$$\frac{C_{S-NAV}(l)}{l \cdot RSSI_{S-NAV}} \geq \psi_{S-NAV}. \tag{3.3}$$

ψ_{S-NAV} is closely related to both false negative error and false positive error (Section 2.5). Fig. 3.10 shows the two error rates with 160 symbols. Obviously, the false negative error dominates the detection's performance within $-14dB$. From Fig. 3.9(b), we can see that the false negative error is more related to the SINR and indicator length. The false positive error rate is mainly due to that the correlation value of S-NAV indicator and data is larger than β_{S-NAV}, which can be defeated by a large Hamming distance between the indicator and data.

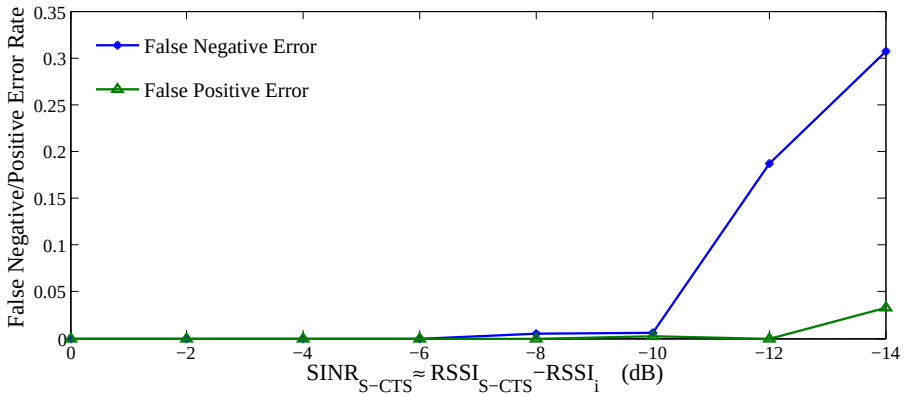


Figure 3.10: False negative/positive error rate with 160 symbols. ($\psi_{S-NAV} = 0.55$)

Table 3.3 shows the decreasing trend as the Hamming distance increases. When the Hamming distance becomes 52, the false positive error rate is 0.2%.

Clearly, Eq. (3.3) mainly deals with the Low-SINR-CTS problem. When a station

Hamming Distance	34	40	46	52
FPE Rate	0.170	0.047	0.008	0.002

Table 3.3: The relationship between Hamming distance and false positive error (FPE) rate. (SINR $= -10dB$, $\psi_{S-NAV} = 0.55$)

is out of the data transmission range, $RSSI_{S-NAV} \leq \beta_{SNR} + RSSI_{no}$ where β_{SNR} is the SNR threshold for correctly decoding packets and $RSSI_{no}$ is the environment noise (typically, -98∼-95dBm). To balance the Low-SNR-CTS problem and exposed terminal problem [41], in this case, $\beta_{SNR} + RSSI_{no}$ is used to instead of $RSSI_{S-NAV}$. Thus, Eq. (3.3) changes to

$$\frac{C_{S-NAV}(l)}{l \cdot (\beta_{SNR} + RSSI_{no})} \geq \psi_{S-NAV}. \tag{3.4}$$

Fig. 3.11 shows the hardware result of the detection rate under different SNRs.

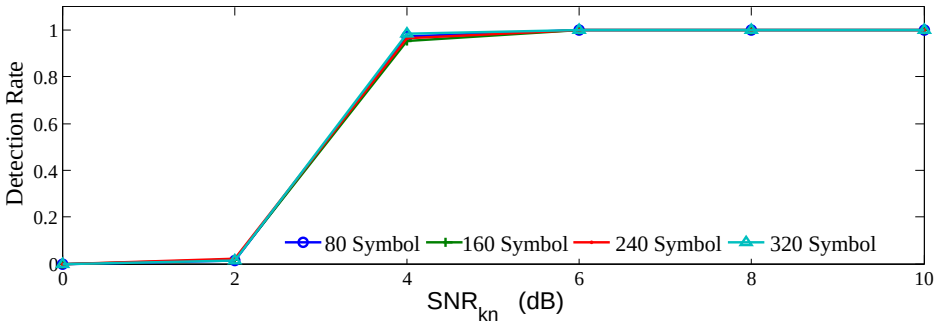


Figure 3.11: Detection rate with various indicator lengths under different SNRs.

3.3.4 Minimum Hamming Distance among S-NAV Indicators

The minimum Hamming distance between any pair of S-NAV indicators will affect the performance of S-NAV identification. Similar to the false positive error rate, the best candidate algorithm may also choose a wrong candidate as the S-NAV indicator, causing the S-NAV time information to be decoded incorrectly. We call this error as "indicator decoding error". The indicator decoding error rate has close relationship with the minimum

Hamming distance between any pair of S-NAV indicators. From Table 3.4, we can see that, by increasing the Humming distance between two S-NAV indicators, the indicator decoding error rate can be less than 0.1% when the Humming distance is 22. Therefore, the indicator decoding error rate can be minimized by enlarging the Humming distance between any pair of S-NAV indicators. However, enlarging the Hamming distance would reduce the available number of S-NAV indicators that the system can use.

Hamming Distance	6	10	14	18	22
PDE Rate	0.0570	0.0200	0.0079	0.0038	0.0010

Table 3.4: The relationship between Hamming distance and indicator decoding error (PDE) rate. (SINR = -10dB)

Compared with the conventional correlation detection [29,38,68] that only allows one candidate, which requires the minimum Humming distance to be 52, the best candidate algorithm reduces the required minimum Humming distance to be 22, which means that we can design more S-NAV indicators to alleviate the catalogue overhead. In our USRP2 experiment, the Humming distance of the 160 symbol-length indicators is set as 22. We can design more than 150 different S-NAV indicators, consequently, the catalogue overhead is below 13.3μs in 802.11a.

3.4 Performance Evaluation

In this section, we give ns-2 simulation results that show the effectiveness of our RTS/S-CTS mechanism to solve the Low-SNR/SINR-CTS problems. We simulated the RTS/S-CTS under different network scenarios: a 4-node line topology for the Low-SNR-CTS problem scenario (Fig. 3.1), a 7-node line topology for the Low-SINR-CTS problem scenario (Fig. 3.2), and a 16-node random network topology for general scenario (Fig. 3.14).

We compared the performance of the RTS/S-CTS with standard CSMA/CA and RTS/CTS protocols. We modified ns-2's source code at the physical layer to support the S-CTS's symbol-level detection by using the hardware experiment results. We also considered the catalogue overhead of the RTS/S-CTS. To evaluate the performance, we employed two common metrics, throughput and packet delivery rate. Table 4.5 lists the parameter configurations used in our simulation.

Parameter	**Value**	**Parameter**	**Value**
Transmission range	$500m$	Preamble	$16\mu s$
Carrier sensing range	$870m$	SIFS	$16\mu s$
S-NAV	$13.3\mu s$	DIFS	$34\mu s$
Catalogue overhead	$13.3\mu s$	CWmax	$1023\mu s$
Link capacity	6Mbps	CWmin	$15\mu s$
Packet size	700∼1500 bytes	Time slot	$9\mu s$

Table 3.5: Parameter configurations for ns-2 simulation.

3.4.1 Low-SNR-CTS Problem Scenario

To evaluate the RTS/S-CTS's performance in the Low-SNR-CTS environment, we built up two directional links (A→B and C→D) as Fig. 3.1. The distance between A→B (and C→D) is $480m$. We set data flow on each link as 2.5Mbps. We varied the distance between B and C (denoted as d_i) from $400m$ to $1000m$ to study the changes of throughput on both links. The simulation result (Fig. 3.12) shows that RTS/S-CTS can completely solve the remote hidden terminal problem (when d_i ranges from $500m$ to $800m$) while the other two schemes cannot compete under this circumstance. More interestingly, we observed that the RTS/CTS suffers more severely than the CSMA/CA when the remote hidden terminal problem occurs. The reason is that node A's RTS/data packet can be

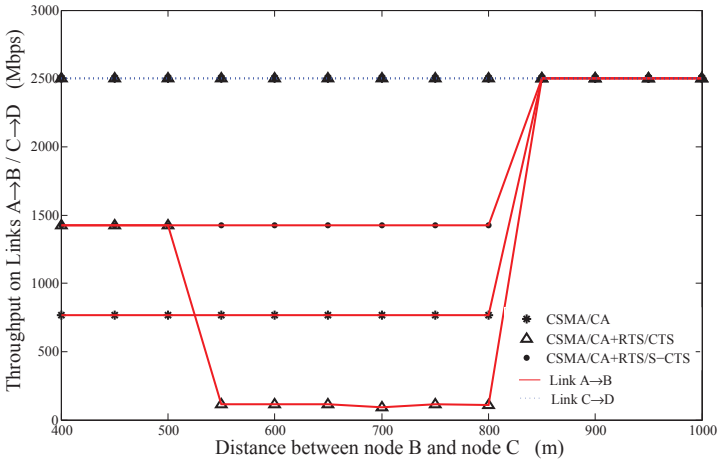


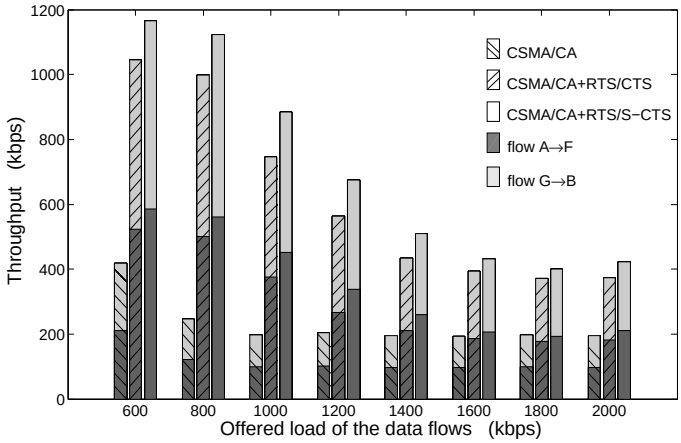
Figure 3.12: The throughput in the Low-SNR-CTS scenario.

corrupted by node C's RTS/data packet with a high probability.

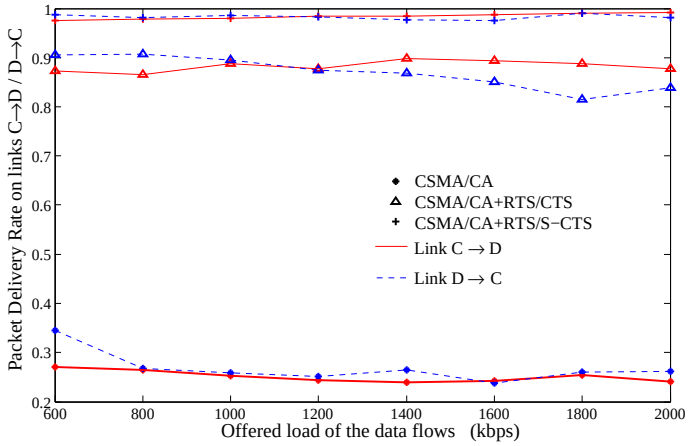
3.4.2 Low-SINR-CTS Problem Scenario

To simulate the low SINR scenario, we built up two unidirectional data flows (A→F and G→B) as Fig. 3.2. The distance between each pair of adjacent nodes is $480m$. We injected two data flows concurrently at nodes A and G. We varied the data flow from 600Kbps to 2Mbps to study the throughput and packet delivery rate on each link.

The simulation reveals that: (1) For unidirectional flows, the RTS/S-CTS achieves 8∼15% improvement compared with standard RTS/CTS, and 200∼250% improvement compared with traditional CSMA/CA in each data flow throughput and total end-to-end network throughput (Fig. 3.13(a)). 2) By solving the CTS collision problem, the RTS/S-CTS can safeguard each link's packet delivery rate to be above 97%. More importantly, this guarantee would not be affected by the offered load of the data flows (Fig. 3.13(b)), and can significantly save the retransmission energy cost.



(a)



(b)

Figure 3.13: Results in the Low-SINR-CTS scenario: (a) The throughput. (b) Packet delivery rate on link C→D/D→C.

3.4.3 Random Network Topology Scenario

To evaluate the RTS/S-CTS's scalability and generality, we generated a random network topology with 16 nodes and randomly set up 6 links (Fig. 3.14). We varied the data flow on each link from 1Mbps to 4.5Mbps. For each data flow, we varied the packet size randomly. We run CSMA/CA, standard RTS/CTS and our RTS/S-CTS respectively. We measured the throughput of each link and summarized the average throughput of the

six links in Fig. 3.15. The result shows that, when the data flow on each link exceeds 4Mbps, the RTS/S-CTS can improve the throughput of those affected links for more than 63% compared with CSMA/CA and standard RTS/CTS.

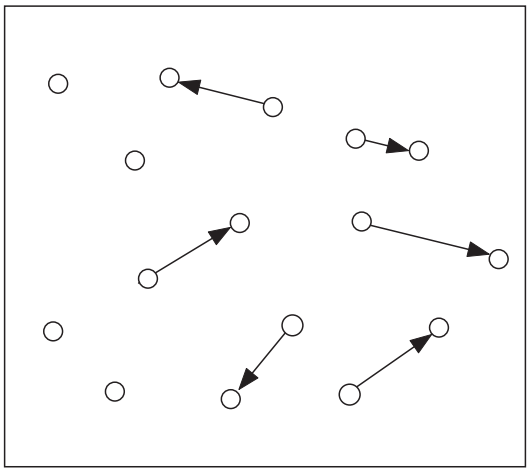


Figure 3.14: The random network topology.

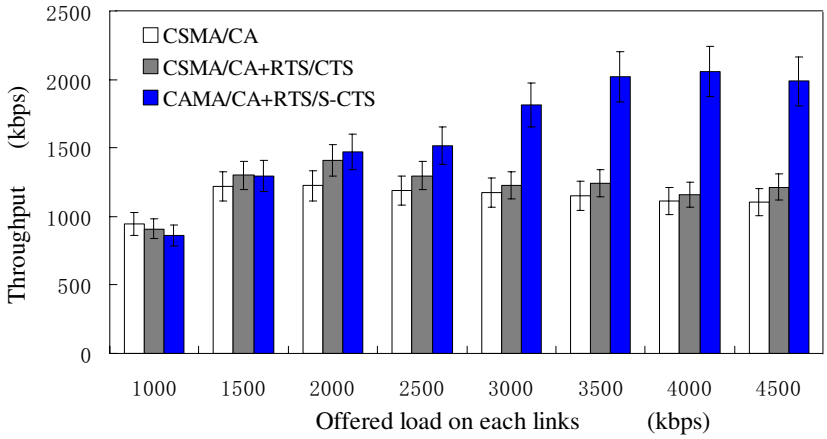


Figure 3.15: The throughput in the random network topology scenario.

However, when the link's data flow is low (<1.5Mbps), the RTS/S-CTS becomes useless (Fig. 3.15). Furthermore, because of the overhead of the S-NAV field and the catalogue overhead, the throughput of RTS/S-CTS is even slightly below the standard RTS/CTS.

As a summary, we can see from the improvement of throughput that the RTS/S-CTS outperforms CSMA/CA and RTS/CTS. Moreover, Fig. 3.15 explains why standard RTS/CTS is just a "backup and supplement" mechanism to CSMA/CA [25, 37]. With high link workload, the standard RTS/CTS cannot protect the receiver from hidden terminals due to the CTS collision (Low-SINR-CTS problem). The RTS/S-CTS defeats those drawbacks by designing a new method to solve the hidden terminal problem.

3.5 Discussions

We further discuss some issues arisen from the RTS/S-CTS mechanism that remain unaddressed in this chapter:

3.5.1 Complexity

Although using a larger number of different S-NAV indicators can reduce the catalogue overhead, it may also introduce computation overhead to conduct signal correlation of the incoming signal sample by sample. Fortunately, we use the preamble detection and synchronization (Fig. 3.5) to activate the SLDD. As the S-NAV field in the S-CTS frame has a constant size, instead of correlating all the incoming samples, we can just *cut* the appropriate S-NAV samples and do the correlation operation *only* for that set of S-NAV samples. Thus, the computation complexity of the PSCP is $\theta(cN)$, where N is the total number of S-NAV indicators adopted and c is a constant cost for conducting one signal correlation. Note that this computation complexity is also a constant cost when N is fixed. Comparing with the preamble detection that needs to correlate all the incoming samples with the preamble, this constant cost is significantly less than the preamble

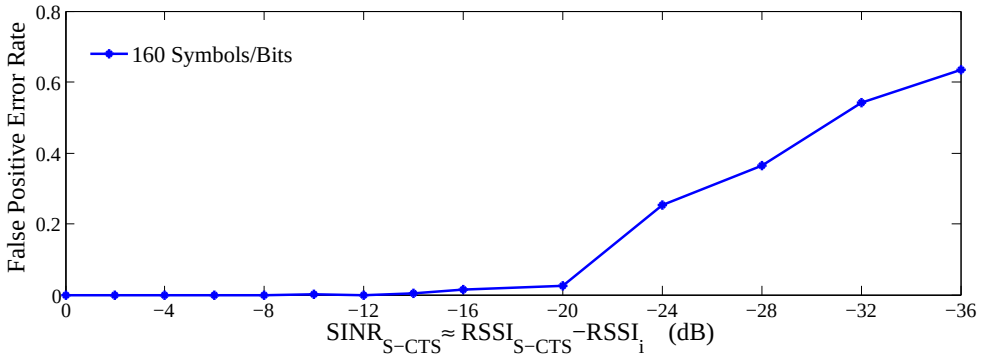


Figure 3.16: False positive error rate with 160 symbol length.

detection cost.

3.5.2 Self-Test and Cancelation

The RTS/S-CTS is a cross-layer mechanism. This feature makes it easily compatible with old protocols, but this may bring new issues. Generally, each packet needs to pass the CRC self-test and will be abandoned when it fails the CRC checking. When the RTS/S-CTS is under a very low SINR scenario ($SINR < -20dB$), the S-CTS's false positive error rate becomes remarkably high (Fig. 3.16). Recall that the false positive error incurs because the station erroneously detects a S-NAV indicator from data. This will cause unnecessary time waiting. We propose a *self-test and cancelation mechanism* for the RTS/S-CTS to eliminate this problem: When the parameters of S-NAV indicators, such as length and minimum Hamming distance, have been settled, the maximal number of candidates that exceed the threshold β_{S-NAV} in the best candidate algorithm has also been fixed. We fix this number N_{S-NAV} for the SLDD (e.g., in Fig. 3.7, N_{S-NAV} is set to 3). If the PSCP generates more than N_{S-NAV} candidates, the SLDD would cease delivering the S-NAV time information to the MAC layer.

3.5.3 Impact of Packets Size

The RTS/S-CTS mechanism is similar to the standard RTS/CTS in the MAC layer. This brings us another concern: the RTS/CTS packets may waste the wireless channel when the size of data transmission packets is small. In fact, the RTS/CTS mechanism is not activated unless the size of data packets exceeds a threshold. The standard RTS/CTS mechanism does not specify the value for this threshold, since it relates to many network parameters, such as network topology and network traffic patterns, which have been well studied [25,37]. In this thesis, we do not give detailed discussion on this threshold neither.

3.6 Summary

Comparing to the RTS/CTS mechanism and previous works, the proposed RTS/S-CTS scheme requires no changes to the standard 802.11 MAC, has no constraint on the transmitter-receiver distance, and does not need expensive hardware such as directional antenna or full-duplex transceiver. Therefore, it is more effective in silencing hidden terminals. The hardware implementation and software simulations show its feasibility and performance improvement for the large scale network. The complexity of the RTS/S-CTS is further analyzed and the self-test/cancelation mechanism is proposed to defeat detection errors. All these efforts make the RTS/S-CTS more practical to real world network scenarios.

Chapter 4

A Downclocked Symbol-Level Information Extraction Approach to Eliminate the Energy Waste on Packet Overhearing Problem

Energy efficiency is a critical issue of wireless devices. However, as the packets are broadcast to the stations in the wireless transmission medium, all active neighboring stations have to spend their energy to decode the packets even though the packets are not addressed to them, which is called as the energy inefficiency of packet overhearing problem in this thesis. In this chapter, I present the novel SASD (Sample-Address Sample-Duration) scheme to solve the energy inefficiency of the packet overhearing problem. Different from the Chapter 3, the symbol-level infotmation is detected under fullclocking mode, the SASD scheme allows the symbol-level information to be extracted under downclocking mode. Thus, the SASD can significantly improve the energy effieiency of the wireless devices.

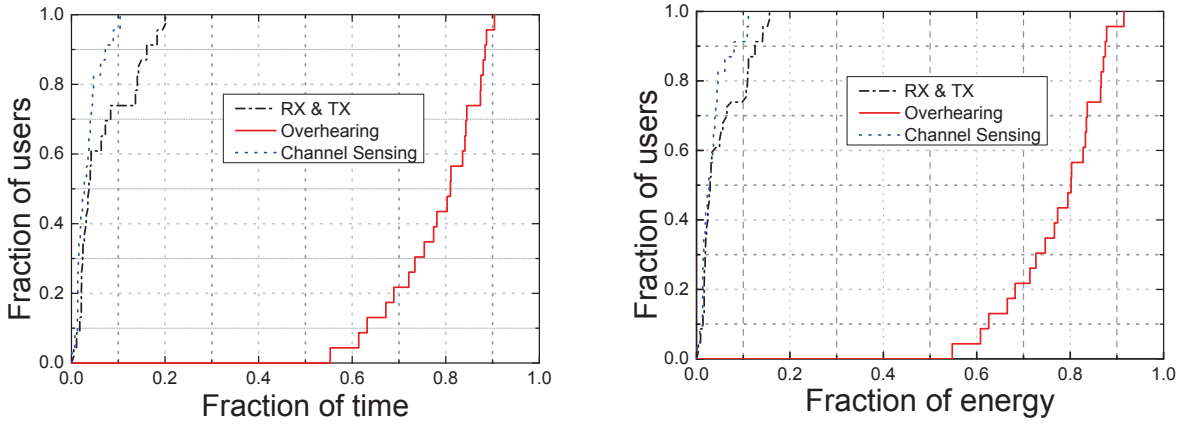
4.1 Motivation

We first explore the energy inefficiency of the packet overhearing problem, especially, in the high traffic network, by detailing the time and energy cost on the channel sensing and packet overhearing through real world WiFi traces, respectively.

Among the wireless devices' energy cost list, the idle listening accounts for the major reason of the energy-inefficiency [83]. However, the energy cost on the idle listening can also be detailed into two parts [17]: (1) the energy cost on packet overhearing, and (2) the energy cost on channel sensing on sending packets or waiting for incoming packets. We describe the energy cost on the three aspects as below, and analyze the energy cost on these aspects by using the real world WiFi traces [81], respectively:

- **Energy cost on TX & RX**: the energy cost on transmitting or receiving packets.
- **Energy cost on packet overhearing**: the energy cost on receiving and decoding packets which are not addressed to the station. The energy spent on this aspect is similar to that on TX & RX. However, the packets will be eventually dropped at the MAC layer, and this kind of energy cost is considered as energy-inefficiency.
- **Energy cost on channel sensing on sending packets or waiting for incoming packets**: the energy cost on sensing the wireless channel before sending packets or monitoring the channel for unpredictable incoming packets. Although the channel sensing is necessary in wireless networks to defend against the packet collision, we still consider this energy cost on the channel sensing as energy-inefficiency.

Noted that we do not consider the energy cost when the station is in the sleeping mode,



(a) Time (b) Energy

Figure 4.1: Time and energy characteristics of mobile devices in a high traffic wireless scenario (The data comes from the trace given in [81]).

as it is quite small compared with the three aspects in high traffic wireless LANs.

Fig. 4.1(a) shows the normalized fraction of time in the three aspects discussed above. More than 90% of the devices spent over 63% of the total time on the packet overhearing while more than 80% of the devices spent less than 5% of the total time on the channel sensing. We believe that it is because the trace was taken from a busy network. In contrast, around 60% stations spent less 10% time on sending (TX) and receiving (RX) packets.

With respect to the energy cost on the three aspects, we adopt a typical Atheros card's energy profile (TX: 127mW, RX: 223.2mW, channel sensing: 219.6mW [13]) which was used in [83]. Fig. 4.1(b) shows that in a high traffic wireless LAN, 80% stations spent more than 70% energy on the packet overhearing. This significant energy-inefficiency on the packet overhearing would rapidly drain out the mobile devices' battery power.

The above analysis reveals that the packet overhearing accounts for the majority of wireless devices' energy cost. Although the traces were taken from a busy network,

we believe that this phenomenon is quite common in high traffic wireless LANs [2, 7]. Obviously, this energy inefficiency on the packet overhearing should be considerably dealt with. If this energy inefficiency can be effectively eliminated, it will clearly improve the energy efficiency of wireless devices in high traffic wireless LANs.

4.2 Architecture and Design

4.2.1 Overview of the SASD Scheme

Different from the Power Saving Mode (PSM) protocols [37, 48, 57, 67] or other energy saving mechanisms [47,70,83,84], SASD tries to enable the stations to detect and identify the packet's receiver MAC address and transmission duration information in one shot without receiving the whole packet at the PHY layer. It leverages the advantages of both the sleeping mode and downclocking mode (as shown in Fig. 4.2): The stations which are not involved in the packet receiving can turn to the sleeping mode to avoid energy waste on the packet overhearing. Also, as SASD is implemented at the PHY layer, it can take the advantage of the PHY-layer energy-efficient mechanism, i.e., low energy cost under the downclocking mode, to save the energy. To achieve this energy-efficiency goal, the main challenges are how to carry/discern the two vital MAC layer information, the packet's receiver MAC address and transmission duration, at the PHY layer without requiring the receiver to fully decode the whole packet, specifically, at the downclocking mode.

The SASD does not change the data packet's structure at the MAC layer, just adding a SASD header in front of the data frame at the PHY layer. As shown in Fig. 4.3, for each SASD header, it contains three fixed-length fields: the notification field, the

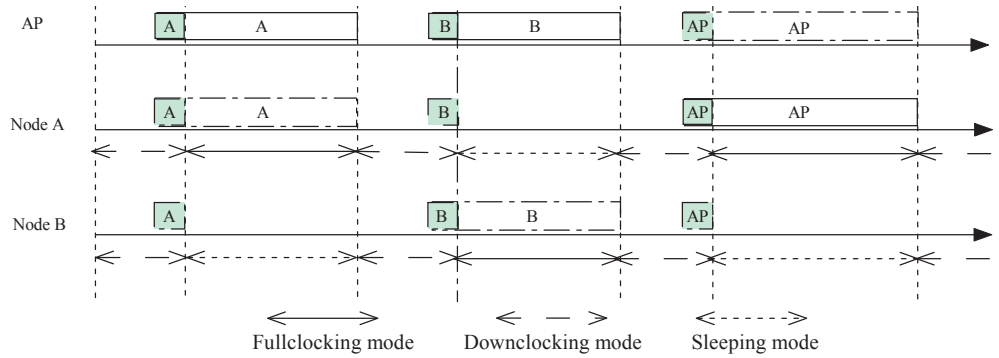


Figure 4.2: An overview of the SASD mechanism. (The full-line box denotes the packet sending and dot-dash-line box denotes the packet receiving.)

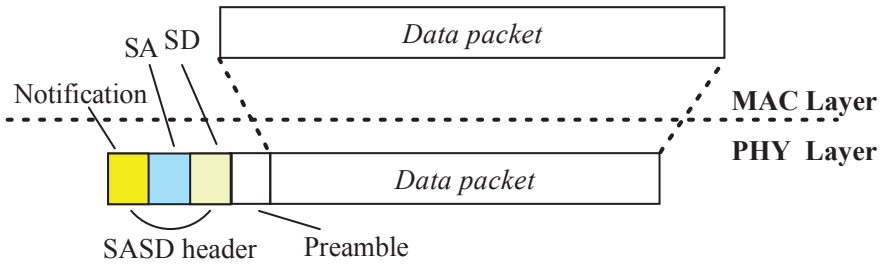


Figure 4.3: The SASD data frame at the PHY layer.

Sample-Address (SA) field and the Sample-Duration (SD) field. The notification field is used to notify all the wireless devices that there is an incoming packet even when the devices work at the downclocking mode. The SA and SD fields are used to carry the two important MAC layer information which are mentioned above.

At the transmitter side, the transmitter will generate the SASD header in front of each packet and send them together. At the receiver side, the receiver adopts the cross-correlation to detect and identify the three fields under the downclocking mode. To decode the information carried in the SASD header, the receiver deploys a new decoder, called "SASD Detection and Identification Decoder" (SSDI) (the gray block in Fig. 4.4), to obtain the SA and SD information from the received packet at the PHY layer and then

to inform the A/D converter to switch to the fullclocking or sleeping mode accordingly: When the device detects an incoming packet under the downclocking mode, it uses the SSDI to detect and identify the SA field to determine whether the incoming packet is addressed to it or not. If yes, it restores to the fullclocking mode to receive the packet. Otherwise, it turns to detect and identify the SD field and changes to the sleeping mode to avoid the energy cost on the packet overhearing. The SSDI procedure is listed in Algorithm 4.1:

Algorithm 4.1 SSDI Procedure

Input: Incoming samples from the A/D converter under the downclocking mode.

1: **while** detect and identify the notification field in the incoming samples **do**
2: Cut the samples in the fixed-length SA filed.
3: **if** the MAC address information can be extracted from the samples in the SA filed **then**
4: Inform the A/D converter to switch to the fullclocking mode for receiving the packet.
5: **else**
6: Cut the samples in the fixed-length SD field.
7: **if** the packet transmission duration information can be extracted from the samples in the SD filed **then**
8: Inform the station to switch to the sleeping mode to avoid packet overhearing.
9: **end if**
10: **end if**
11: **end while**
12: Go to Step 1.

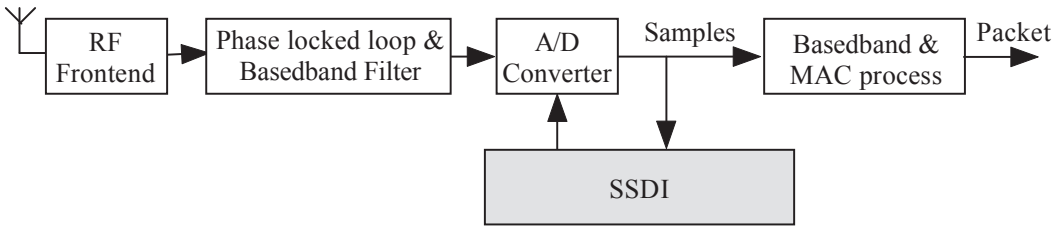


Figure 4.4: The architecture of the SASD scheme at the receiver side. Only the gray SSDI block is a new component.

In the following parts of this section, we first detail the detection and identification process under the fullclocking mode, then we describe the process under the downclocking mode.

4.2.2 Detection and Identification under the Fullclocking Mode

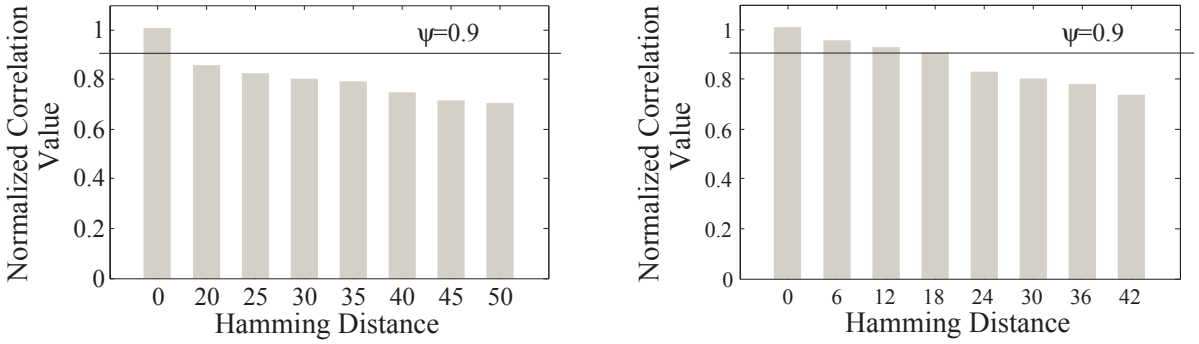
As describe in Section 2.5, we use the cross-correlation Ineq. (2.7) to detect and identify the known samples sequence.

$$\frac{C_{kn}(l)}{l \cdot RSSI_{kn}} \geq \psi.$$

As to the false negative (missing alarm) error and false positive (false alarm) error, we use the similar solution described in Section 3.3.3. We use the parameters ψ and Hamming distance between two known bit sequences to decrease the false negative/positive error rate. And we detail the two errors in Section 4.3. Fig. 4.5(a) shows the normalized correlation value between the transmitted known bit sequence and any other known bit sequences with various Hamming distances. From this illustration figure, we can see that, with the threshold $\psi = 0.90$, once the Hamming distance between two known bit sequences is larger than 20, the correlation process can correctly detect and identify the transmitted known bit sequence, making the false positive error rate close to 0 ($SINR \approx 0dB$).

4.2.2.1 How to Process the Receiver's MAC Address?

In SASD, the AP would maintain a *local* unique known bit sequence pool and a MAC address mapping table. When a new station joins the AP's wireless network, after authorization, the AP associates an unused known bit sequence from the pool with the new



(a) The conventional correlation.

(b) The enhanced correlation approach.

Figure 4.5: The Hamming distance requirements for two detection and identification methods. (bit sequence size = 160 bit, $SINR \approx 0dB$)

station's MAC address and puts them in the MAC address mapping table. Then, the AP informs this station both its known bit sequence and AP's known bit sequence (Fig. 4.6). Once the AP wants to send a data packet to a station, the AP puts the associated known bit sequence of the station in the SA field of the SASD header. Also, if a station wants to send packets to the AP, it puts AP's known bit sequence in the SA field.

The station would use the detection and identification approach to check whether the incoming packet is addressed for it or not. If yes, the station switches to the fullclocking mode for receiving the packet. Otherwise, it tries to detect and identify the SD field under the downclocking mode.

4.2.2.2 How to Process the Packet Transmission Duration?

One of the SASD's main challenge pertains to the delivery of the packet transmission duration at the PHY layer, which can enable those non-receivers to switch to the sleeping mode to avoid the packet overhearing and to switch back to the downclocking mode after the packet transmission is completed. As we mention in Section 3.2.3, the packet

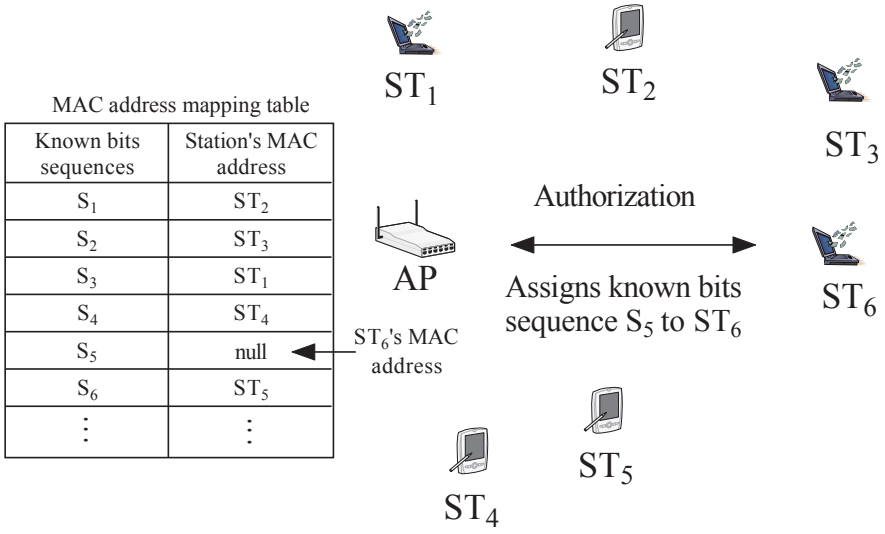


Figure 4.6: The MAC address mapping table in AP.

transmission duration would be a finite value, SASD takes the same way as the S-NAV, divides the packet transmission duration into N catalogues, and maps each catalogued duration time T^i_{cata} to the *global* unique known bit sequence. By using the Eq. (3.1), T^i_{cata} can be calculated as follows:

$$T^i_{cata} = \frac{T_{max}}{N} \cdot i,$$

where T_{max} is the maximum packet transmission duration, i is the catalogued time index of the global unique known bit sequence and $1 \leq i \leq N$. As the actual packet transmission duration could fall in any catalogues, the AP and station must preload all the global unique known bit sequences for generating or discerning the correct catalogued time index.

When the AP (or station) prepares to send a packet, it obtains the packet transmission duration T_D, calculates the catalogued time index $i = \lceil \frac{N \cdot T_D}{T_{max}} \rceil$. After that, the AP (or station) puts the corresponding known bit sequence in the SD field. Note that the

catalogued duration time function could keep the stations sleeping longer than the actual data transmission duration, which introduces extra catalogue cost. Clearly, enlarging the catalogue value N can alleviate this cost.

When the station receives the packet, SASD adopts the *enhanced correlation approach* (Algorithm 4.2), which not only considers that the correlation value must excess the threshold but also compares the correlation values between different catalogued time indexes to detect and identify the correct catalogued time index. Similar in Section 3.2.3, first, the station parallel correlates the incoming samples with N different known sample sequences corresponding to the N catalogued time indexes and picks up the normalized correlation results which exceed the threshold ψ_{SD} as the candidates. After that, it picks up the largest correlation result among the candidates and maps it to the catalogued time index i. The station gets the index i, calculates the catalogued duration time T^i_{cata}, and derives the sleeping time T_{sleep} by adding T^i_{cata} with a SIFS time (T_{SIFS}) and an ACK duration (T_{ACK}). Then, the station switches to the sleeping mode during the sleeping time T_{sleep}.

Algorithm 4.2 Enhanced Correlation Approach

Input: Samples in the SA filed.
Output: The sleeping time T_{sleep}.

1: Correlate the samples with N different catalogued time index bit sequences; pick up the normalized correlation values which exceed $\psi_{S-Duration}$ as the candidates.
2: Compare these candidates and pick up the maximum value $NC_i(l)$.
3: Get the catalogued index i of the selected one; calculate the catalogued duration time T^i_{cata}; set the sleeping time $T_{sleep} = T^i_{cata} + T_{SIFS} + T_{ACK}$.
4: Switch to the sleeping mode for T_{sleep}.

Fig. 4.5(b) shows that, when the Hamming distance between two known bit sequences ≥ 6 and $\psi = 0.90$, by using the enhanced correlation approach, the transmitted known

bit sequence can be correctly identified. Noted that the enhanced correlation approach allows multiple normalized correlation values excess the threshold.

4.2.3 Detection and Identification under the Downclocking Mode

In Section 4.2.2, we show that when the A/D converter works at the fullclocking mode, the receiver's MAC address and packet transmission duration can be detected and identified at the PHY layer. In this section, we explore that, even if the A/D converter works at the downclocking mode, these approaches are still functional to generate and discern the receiver's MAC address (SA) and packet transmission duration (SD) information at the PHY layer.

4.2.3.1 Is SASD Still Functional under the Downclocking Mode?

At the receiver side, according to the Nyquist-Shannon sampling theorem, to fully reconstruct the signal which the transmitter was sent out, the receiver has to use twice of the transmitting bandwidth to sample the incoming signal. Thus, the A/D sampling frequency is $f_r = 2B$, where B is bandwidth.

When the A/D converter reduces the sampling clock-rate in order to save the energy, it also decreases the number of the known samples that SASD can use to calculate the correlation value. Based on Eq. (2.6) in Section 2.5, the correlation value after downclocking would be:

$$C_{kn}(l/\tau) \approx 1/\tau \cdot |H| \cdot \sum_{i=0}^{l-1} |x_{kn}[i]|^2, \tag{4.1}$$

where τ is the downclocking rate of the A/D converter. Remember that Eq. (2.6) can only

be true when l is large enough to make $O(l) \approx 0$. Thus, Eq. (4.1) also needs to meet the requirement that l/τ is large enough to make $O(l/\tau) \approx 0$. Fig. 4.7 shows the relationship between the correlation value $C(l/\tau)$ and downclocking rate τ from the USRP2 hardware experiment. The known bit sequence is transmitted at 10 times. It is shown that when $\tau \leq 4$, $C(l/\tau) \approx 1/\tau \cdot C(l)$. When $\tau = 64$, due to the reason that l/τ cannot make $O(l/\tau) \approx 0$, Eq. (4.1) cannot be held anymore. However, 10 correlation spikes can be detected from Fig. 4.7(d). It evidently shows that by using the cross-correlation, even when the A/D converter works at the downclocking mode, the correlation value $C(l/\tau)$ can still be used to detect the known bit sequence. Consequently, SASD can use the known bit sequences to deliver the SA and SD information at the downclocking mode.

As we know, the difference between two bit sequences can be measured by their Hamming distance. At the fullclocking mode, every different bit in these two different bit sequences is fully sampled into different sample sequences, which guarantee the difference of their correlation values under the fullclocking mode. However, this difference cannot be guaranteed when the A/D converter works at the downclocking mode, i.e., the different samples which the A/D converter samples at the fullclocking mode would be lost. Thus, different bit sequences may lead to the same sample sequence at the downclocing mode. Consequently, the identification approach, especially, the enhanced correlation approach would fail to distinguish these various known bit sequences.

As $f_r = 2B_s$[1] and the samples are closely related to the incoming symbol, when the A/D works at the downclocking rate $\tau = 2$, the receiver can still get different sample

[1]The relationship between the bandwidth and symbol rate can be written as $B = B_s(1 + \alpha)$, where B_s is the symbol rate and α is the roll-off factor of a low-pass filter. Normally, $\alpha \geq 0$. In this chapter, we assume the bandwidth is fully used, which means we take $\alpha = 0$.

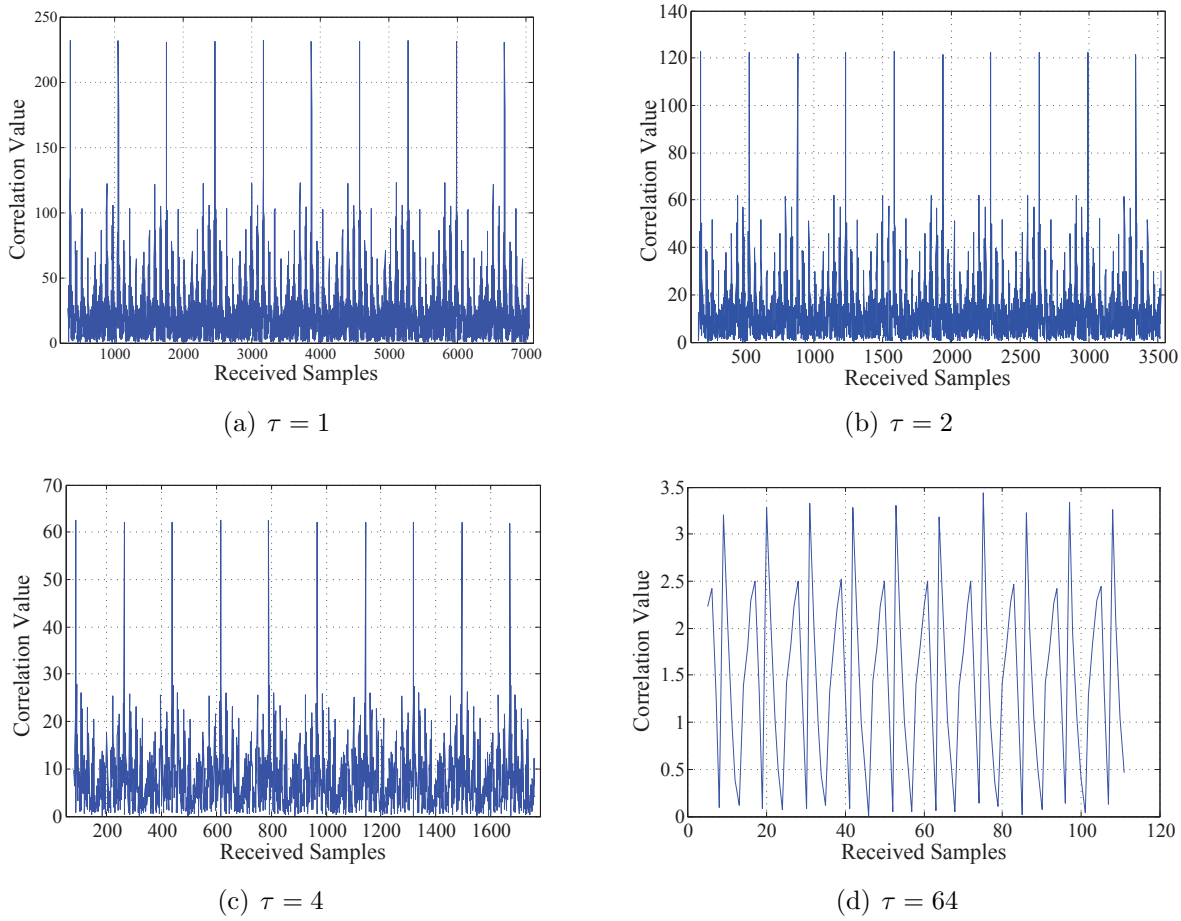
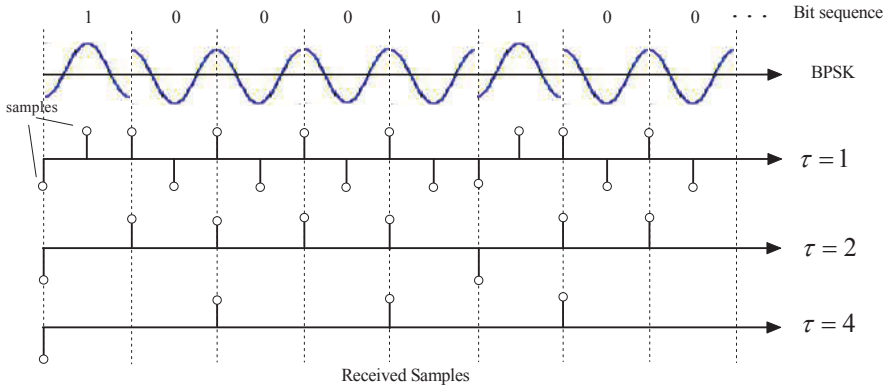
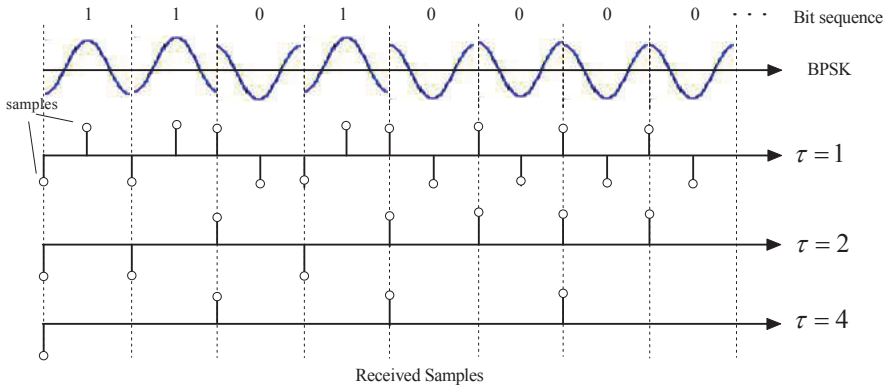


Figure 4.7: The relationship between correlation value $C(l/\tau)$ and downclocking rate τ. ($SINR \approx 0dB$, $l = 320$)

sequences to help the SSDI identify the known bit sequences. As shown in Fig. 4.8, two bit sequences ("10000100" and "11010000") are sampled into different sample sequences when the A/D works at the fullclocking rate. When the A/D uses the downclocking rate $\tau = 2$, the resultant samples of two bit sequences are still different samples. However, if the A/D goes to the downclocking rate $\tau = 4$, the resultant samples of two bit sequences would be the same. At that time, the SASD scheme fails to identify these two bit sequences.



(a) The bit sequence "10000100".



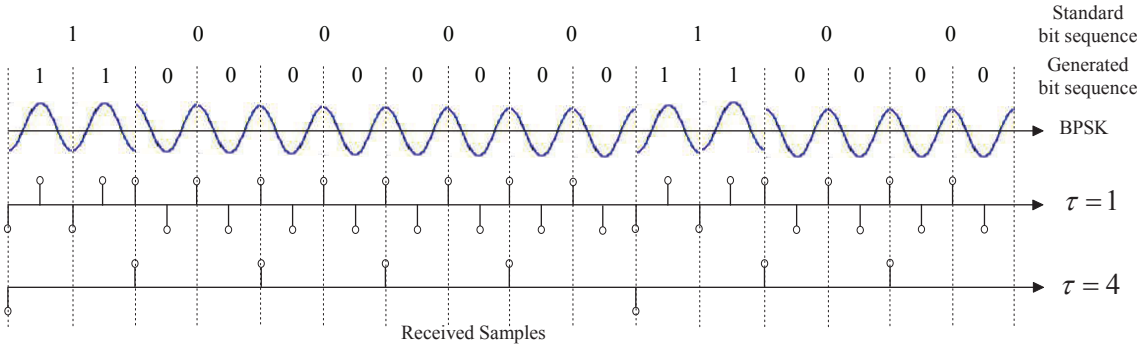
(b) The bit sequence "11010000".

Figure 4.8: The received samples of two bit sequences under different A/D's downclocking rate τ. Here, we assume the A/D start sampling at the first place.

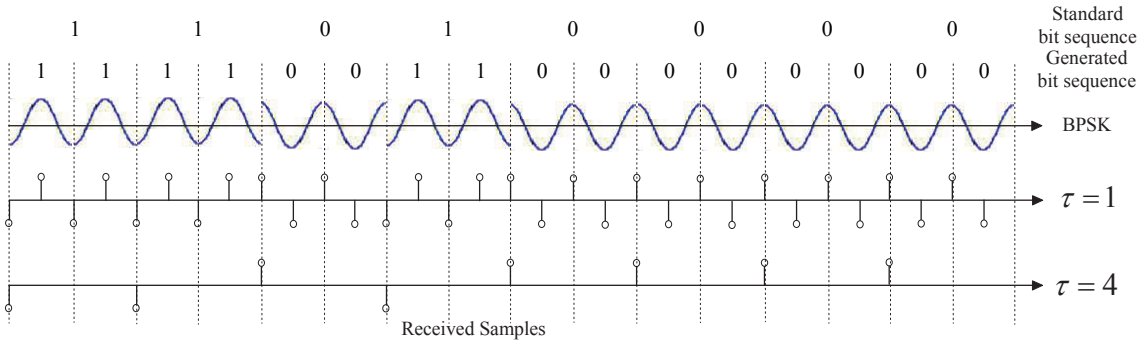
4.2.3.2 How to Generate Known Bit Sequences under A Higher Downclcoking Rate?

To generate the known bit sequences under a higher downclocking rate, e.g. $\tau = 4$, we deploy the interpolation method to generate a new known bit sequence from each fullclocking standard bit sequence. As the higher downclocking rate will cause some bits to miss sampled, the interpolation will duplicate the last sampled bit to fill the missed bits to guarantee the difference of the received samples under the higher downclocking rate. E.g., in Fig. 4.9, when $\tau = 4$, every two bits will only generate one sample, which means

that one bit is missed sampled. For each bit in the two standard bit sequences ("10000100" and "11010000"), the interpolation method will duplicate it to two bits, resulting in two new bit sequences "1100000000110000" and "1111001100000000". Apparently, the new bit sequences can keep the difference of the received samples when $\tau = 4$, but with the cost that the bit sequence's length has $\tau/2$ times increasing.



(a) The generated bit sequence "1100000000110000".



(b) The generated bit sequence "1111001100000000".

Figure 4.9: The received samples of two generated bit sequences under A/D's downclocking rate.

4.2.3.3 How to Identify the First Sampling Place?

When the A/D converter works at the downclocking mode, the A/D cannot guarantee to sample the bit sequence at the first sampling place, which inevitably forces the SSDI to check all possible downclocking sample sequences with the incoming sample sequence

to locate the first sampling place of the sequence. By choosing the downclocking sample sequence with the largest correlation value (details in Section 4.3.2), the SSDI is aware of the first sampling place of the sequence under the downclocking mode.

From what we have discussed above, the SASD scheme can work under the downclocking mode when the difference requirement of the sample sequences can be guaranteed. Particularly, when the A/D takes the downclocking rate $\tau = 2$, the known bit sequences which are used at the fullclocking mode can be directly deployed at the downclocking mode with downclocking rate $\tau = 2$. Moreover, if the A/D wants to work under a higher downclocking rate, the new known bit sequences can be generated from the fullclocking standard bit sequence by using the interpolation method which is described in Section 4.2.3.2.

4.3 Hardware Experiments

In this section, we reveal the hardware implementation and experimental methodology. As the three fields (Notification field, SA field and SD field) in the SASD header are separately detected and identified, each field would have different requirements for the detection and identification.

4.3.1 Hardware Implementation and Experimental Methodology

4.3.1.1 Hardware Implementation

The SASD scheme is implemented on a 3-node GNURadio/USRP2 testbed. Each node is a commodity PC connected to a Universal Software Radio Peripheral 2 (USRP2) [19]

with RFX2400 daughter-board. The RFX2400 operates at the 2.4GHz frequency band. All PCs are installed Ubuntu 10.04 and GNURadio [26]. As the USRP2 could not use the external clock to downclock the incoming signal, we emulated the A/D downclocking procedure by adjusting the decimation rate. Please noted that changing the decimation rate would not save much energy, but it would give the same outcome of baseband samples as that from the real A/D downclocking procedure. Also, we consider the interference model, which is more common in real-world wireless LANs, and denote $RSSI_{SASD}$ and $RSSI_{INT}$ as the received signal strength indicators of the SASD header and interference, respectively. Thus, the SINR is calculated as $SINR \approx RSSI_{SASD} - RSSI_{INT}$.

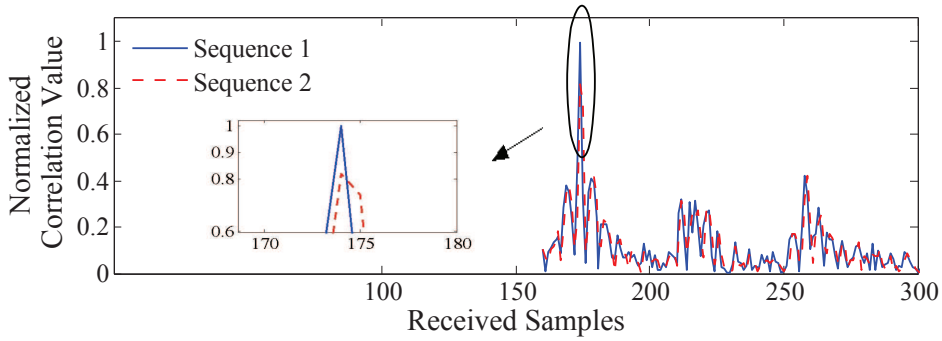
The SASD uses the BPSK modulation/demodulation module, which is commonly used in the 802.11 standard. we use the default GNURadio configuration for the SASD evaluation, i.e., at the transmitter side, the DAC rate is 400M samples/s, the interpolation rate is 100 (interpolation rate in the DAC chip is 4 and interpolation rate controlled by GNURadio is 25), and the number of samples per symbol is 2; at the receiver side, the ADC rate is 100M samples/s, and the decimation rate is set to be 50 to emulate the scenario when A/D works at the downclocking rate 2.

4.3.1.2 Experimental Methodology

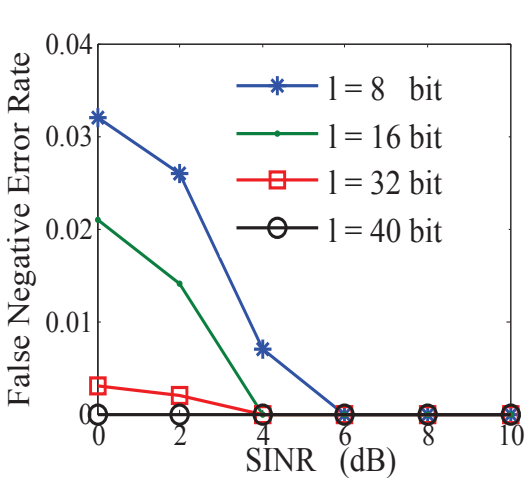
USRP2 has hardware delays in transmitting samples from the RF front-end to its connected commodity PC, and GNURadio also incurs some artificial software delay to process these samples. Thus, it is difficult to conduct a real time evaluation of the SASD in high bit rates. Hence, we resorted to the trace-based evaluation [68, 83], where each node would save all the outgoing and incoming samples for off-line processing.

4.3.2 Notification's Detection and Identification

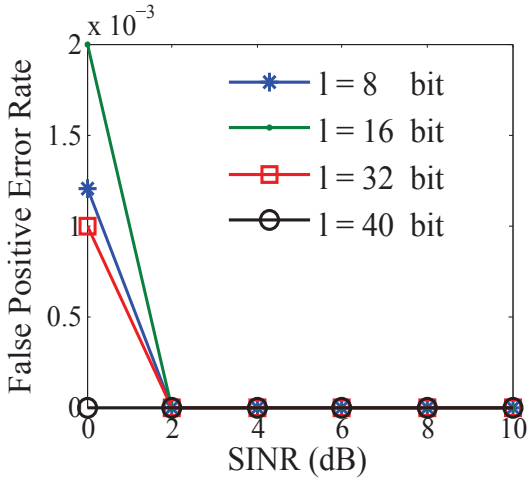
Before the SA and SD fields' detection and identification, the receiver has to sense if there is a incoming packet by correlating the notification field under the downclocking mode. Unfortunately, under the downclocking mode, the SSDI has to test τ possible known sample sequences to locate the first sampling place of the known sample sequences (Section 4.2.3). In the experiment, as we take the downclocking rate $\tau = 2$, the total number of possible downclocking sample sequences would be 2. After that, the SSDI is aware



(a) Find the A/D's first sampling place.



(b) The false negative error rate.



(c) The false positive error rate.

Figure 4.10: Notification detection performance ($\tau = 2$, and l = 8, 16, 32, 40 bit. Here, $SINR \approx RSSI_{SASD} - RSSI_{INT}$).

of the A/D's first sampling place, and it can directly calculate the right SA/SD's downclocking sample sequences during the SA/SD's detection and identification procedures. Fig. 4.10(a) shows the normalized correlation values between the received samples and the two known downclocking sample sequences (sequence 1 and sequence 2)[2]. By choosing the sequence with the largest normalized correlation value, the SSDI is aware of A/D's first sampling place when working under the downclocking mode.

In the SSDI Procedure (Algorithm 4.1), the detection of the notification field takes the key place because it is the foundation of the rest SA/SD's detection and identification. The false negative error is more related to the SINR, known bit sequence's length and threshold ψ. To minimize the false negative error, we took $\psi_N = 0.85$ during the detection of the notification field. Fig. 4.10(b) gives the false negative error rates for different l-length known bit sequences under the interference model. As shown in Table 4.1, for the false positive error, we can decrease this error rate by enlarging the Hamming distance between the data and known sample sequence. In the experiment, we took 40-bit length as the length of the notification field.

Hamming Distance	2	3	6	7
FPE Rate	0.0012	0.0020	0.0010	0
Sequence Length l (bit)	8	16	32	40

Table 4.1: The relationship between Hamming distance and false positive error (FPE) rate with various lengths of known bit sequence. (SINR $= 0dB$, $\psi_N = 0.85$)

[2]The two downclocking sample sequences are generated from the same known fullclocking sample sequence.

4.3.3 SA's Detection and Identification

In the SA field's detection and identification, as the SSDI has detected the notification field and identified the A/D's starting downclocking position, the SSDI can directly calculate the station's own SA known downclocking sample sequence for the SA field's detection and identification.

One of the key design issue of the SA field is to have a large set of sample sequences to support the stations. Fortunately, the SSDI can store all the sample sequences for carrying the SA information only. We arise the threshold ψ_{SA} for the SA's detection and identification. In our experiment, we set $\psi_{SA} = 0.9$ and SA's length to be 40 bits, the false negative error rate is nearly 0 even when $SINR = 0dB$. At the same time, when the Hamming distance reaches 5, the false positive error rate is also minimized to 0 ($SINR = 0dB$). However, this would cause the SSDI to extract the SA information, switch to the fullclocking mode for receiving packets, even if the SINR is not high enough to correctly decode the packet[3]. However, because the packet is addressed to the station, we do not consider this booting-up and unnecessary packet receiving as the packet overhearing problem.

Considering all the factors above, we can design more than 40 different bit sequences for the SA field. We believe these known sequences are enough for the normal usage, even if the AP reserves some sequences as the broadcast and multicast addresses. Moreover, if the AP wants to support more stations, we can increase the length of the SA field to design more sequences.

[3]802.11 needs at least $9.7dB$ SNR to decode a packet [83].

4.3.4 SD's Detection and Identification

The station that needs to switch to the fullclocking mode after the SA field's detection and identification would use the transmission time of SD field to switch its mode. This switching time is various among different devices, normally from 9.5 μs to 128 μs [83], which is enough to transmit the SD field in the SASD header. In our experiment, we choose 320-bit length as the SD field size to test the SASD scheme. Note that SASD is not restricted to this length, the SD field can be different lengths to fit for the A/D's switching time. Thus, stations have to restore all possible lengths in the SASD registers for the SD field's detection and identification. Table 4.2 gives the transmission times for various lengths of SD field in standard 802.11 wireless LANs.

SD Field Length (bit) / Standards	80	160	240	320
802.11a/g	6.67	13.3	20	26.6
802.11b	7.27	14.55	21.82	29.09
	Time (μs)			

Table 4.2: Channel occupation time overhead.

Same as the SA field's detection and identification, the SASD can directly "cut" the samples in the SD field as the input for the enhanced correlation approach, and those samples are thought to carry the SD information only. Moreover, compared with the conventional correlation detection and identification, which only allows one correlation value to exceed the threshold, the enhanced correlation approach would enable several correlation values to exceed the threshold $\psi_{SD} = 0.9$ which significantly relaxes the Hamming distance requirements (as shown in Table 4.3). Consequently, SASD can design more known bit sequences (N) to minimize the catalogue cost in the enhanced correlation

approach. In our USRP2 experiment, we designed more than 320 different bit sequences for 320-bit length SD field. Thus, the catalogue cost is below 6.3 μs in 802.11a. Also, considering all the factors above, the false negative/positive error rates of 320-bit length sequences are already close to 0 when $SINR = 0dB$.

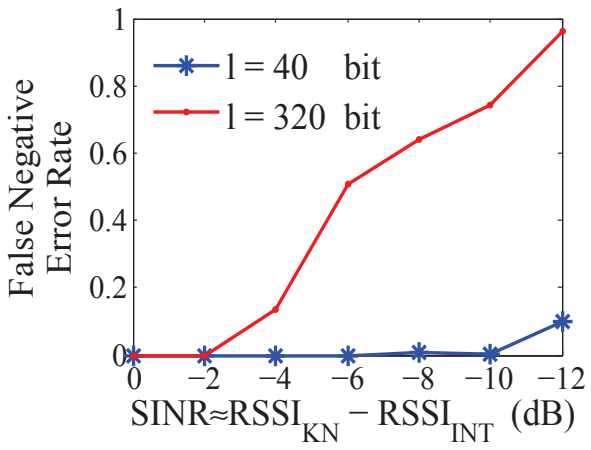
SD Field Length (bit) / Method	80	160	240	320
Conventional Correlation	10	20	30	40
Enhanced Correlation Approach	4	6	12	16
	Required Humming Distance			

Table 4.3: The Humming distance requirements between the conventional correlation and enhanced correlation approaches. (SINR = $0dB$, $\psi_{SD} = 0.9$)

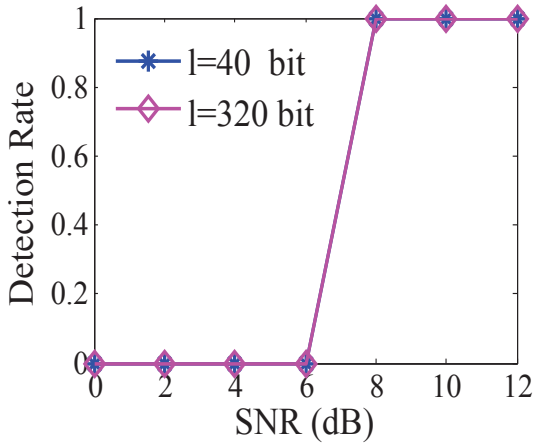
4.3.5 Aggressive Model - Low SINR and Low SNR Scenarios

The above sections (4.3.2$\sim$ 4.3.4) only reveal the scenario that the $RSSI_{SASD}$ of the SASD header is larger than the $RSSI_{INT}$ of the interference. However, the high interference is an inevitable situation in the high traffic wireless LANs [2]. Also, due to the signal fading effect, the SASD can detect a SASD header even if the SNR is not high enough to decode a packet. In this section, we discuss the aggressive model, i.e., how the SASD works under the low SINR and low SNR scenarios.

One critical issue caused by the low SINR is the rising of the false negative error rate. As the false negative error is highly related to the SINR, known bit sequence's length l and threshold ψ. To minimize the false negative error rate, we use $\psi = 0.55$ in the low SINR scenario to evaluate the performance. Fig. 4.11(a) shows the false negative error rate under the low SINR for known bit sequences with two sizes (40 bit, 320 bit and $\tau = 2$) we used in the hardware experiment. Even though we decrease the threshold ψ



(a) The false negative error rate of the known bit sequences under the low SINR. ($\psi = 0.55, \tau = 2$)



(b) The detection rate of known bit sequences under the low SNR. ($\psi = 0.85, \tau = 2$)

Figure 4.11: The detection and identification performance of known bit sequences with two sizes (40 bit, 320 bit) under the low SINR/SNR scenarios.

to 0.55, the missing rate of 40-bit known sequence would rise to 74% ($SINR = -8dB$), which means that the SASD would miss most of the chances to detect the notification field. Although the SASD can increase the length of the notification filed to decrease the false negative error rate, the overall overhead of the SASD header would be increasing. Considering the SASD works under the downclcoking mode, this missing detection would not cause much energy waste on the packet overhearing.

Another critical issue is the false positive error rate of the notification field. It would cause the unnecessary booting-up due to the SA and SD's detection and identification. To deal with this problem, we modify the correlation value normalization (Ineq. (2.7)) as:

$$\frac{C_{kn}(l)}{l \cdot RSSI} \geq \psi. \tag{4.2}$$

Here, we use the $RSSI$ of the incoming signal instead of the estimated $RSSI_{SASD}$. In

the low SINR scenario, as $RSSI \approx RSSI_{INT} > RSSI_{SASD}$, the normalized correlation value would be much smaller under the low SINR. When the data meets the Hamming distance requirement in Table. 4.1, the false positive error rate of the notification field can be minimized to 0.

Different from the low SINR scenario, the low SNR scenario would cause the SASD detection and identification decoder (SSDI) to detect and identify the SA field, and make the station switch to the fullclocking mode to decode the packet even if the SNR is not high enough to correctly decode the packet. To deal with this issue, we set a minimum detectable SNR_{min} [83] and modify Ineq. (4.2) as:

$$\frac{C_{kn}(l)}{l \cdot Max(RSSI, SNR_{min} + RSSI_{en})} \geq \psi. \tag{4.3}$$

Here, $RSSI_{en}$ is the environment noise (typically, $-98 \sim -95dBm$). Fig. 4.11(b) gives the detection rate of known bit sequences with two sizes (40 bit, 320 bit and $\tau = 2$) when $SNR_{min} = 8dB$. Note that the modified Ineq. (4.3) does not change the detection and identification's performance in Sections (4.3.2$\sim$ 4.3.4). When $RSSI_{SASD} \geq RSSI_{INT} \geq (SNR_{min} + RSSI_{en})$, $RSSI \approx RSSI_{SASD} \geq RSSI_{INT}$. Also, in the low SNR scenario, as $RSSI < (SNR_{min} + RSSI_{en})$, Ineq. (4.3) would use $SNR_{min} + RSSI_{en}$ to normalize the correlation value.

4.3.6 Complexity

The SSDI will introduce extra computation overhead to conduct the cross-correlation of the incoming signal sample by sample to identify the SASD's notification, SA or SD information. We take the computational complexity of the preamble detection and synchronization operation as the base to evaluate the SSDI's complexity as the preamble

detection and synchronization also deploys the cross-correlation to detect and synchronize the preamble [39]. The complexity of the notification field's detection and identification operation is comparable to the complexity of the preamble detection and synchronization part. After the notification field is detected, as the SA and SD fields have fixed sizes, SSDI can directly calculate their correlation values without sample by sample. Thus, the complexity of SA or SD's detection and identification will be significantly less than that of the preamble detection and synchronization. Thus, the total complexity of the SSDI is comparable to that of the preamble detection and synchronization part.

4.4 Performance Evaluation

4.5 Simulation Settings

In this section, we give ns-2 simulation results that show the effectiveness of our SASD scheme to save the energy cost on the packet overhearing problem. we have implemented and tested the SASD in ns-2.34 under two different network scenarios: single AP with multiple stations and multiple APs with multiple stations. To fairly compare with the E-MiLi, we have used the same energy profile given in [83], which is listed in Table 4.4, to calculate the energy cost. We have compared our SASD with the standard CAM (constant active mode), PSM, E-MiLi and PSM+E-MiLi (a combination scheme of the PSM and E-MiLi). As the energy cost that the mobile device spends on the wireless transmission part is much greater than that on the computation part, we have considered the energy cost on receiving the SASD header but ignored the energy cost introduced by the SSDI block in the SASD scheme. We have modified ns-2's source code at the PHY layer to support the SSDI procedure by using the hardware experiment results. We have

also considered the time overhead caused by the SASD header as well as the catalogue cost of the SD field in the simulations.

Energy Mode	Value (mW)
TX	127
RX	223.2
Sleeping	10.8
Channel Sensing	219.6
Half Sampling	140.54

Table 4.4: Energy profile [83] for ns-2 simulation.

As the above energy profile $(W_{TX}, W_{RX}, W_{SP}, W_{CS}, W_{HS})$ is used, the total energy usage per station can be calculated as:

$$E = W_{TX} \cdot T_{TX} + W_{RX} \cdot T_{RX} + W_{SP} \cdot T_{SP} + W_{CS} \cdot T_{CS} + W_{HS} \cdot T_{HS}, \qquad (4.4)$$

where $(T_{TX}, T_{RX}, T_{SP}, T_{CS}, T_{HS})$ are the time durations the station spends in different energy modes. In all the simulations, we have set a constant 5-minute time duration to evaluate the energy usage, that is, $T_{TX} + T_{RX} + T_{SP} + T_{CS} + T_{HS} = 5mins$.

Table 4.5 lists the parameter configurations used in our simulations.

Parameter	Value	Parameter	Value
Transmission range	$250m$	Preamble	$16\mu s$
Interference range	$500m$	SIFS	$16\mu s$
SASD header[4]	$33.35\mu s$	DIFS	$34\mu s$
Catalogue cost	$6.3\mu s$	CWmax	$1023\mu s$
Link capacity	6Mbps	CWmin	$15\mu s$
Packet size	300∼1500bytes	Time slot	$9\mu s$

Table 4.5: Parameter configurations for ns-2 simulation.

[4]The SD field can be different among different devices, which causes different SASD headers. Here, we choose the maximum SD field length which is used in our hardware experiment.

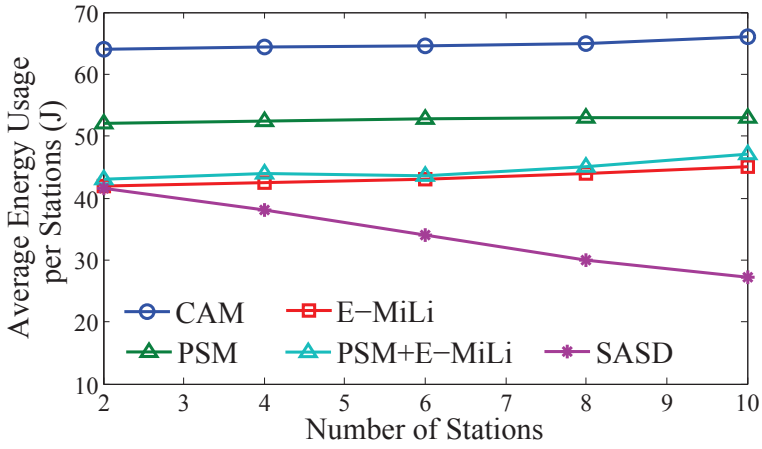
4.5.1 Single AP with Multiple Stations Scenario

In the single AP with multiple stations scenario, we have set up a WLAN that consisted of one AP with multiple stations. The stations are uniformly distributed within the AP's transmission range. First, we study the energy saving performance by increasing the number of stations. We use the fixed data rate ($200Kbps$, 1100-byte packet size) between the AP and each station. Fig. 4.12(a) shows that, although the SASD has similar performance when there are only two stations, it outperforms the CAM, PSM, E-MiLi and PSM+E-MiLi when the network density increases. One interesting phenomenon is that with the increasing of the stations, the energy usage per station in the SASD is decreasing. The reason why the SASD has different energy performance is that, when the network density increases and all stations want to communicate with the AP, for each station, it has more chances to switch to the sleeping mode for the SASD scheme. By using Eq. (4.4), the energy usage of the SASD is

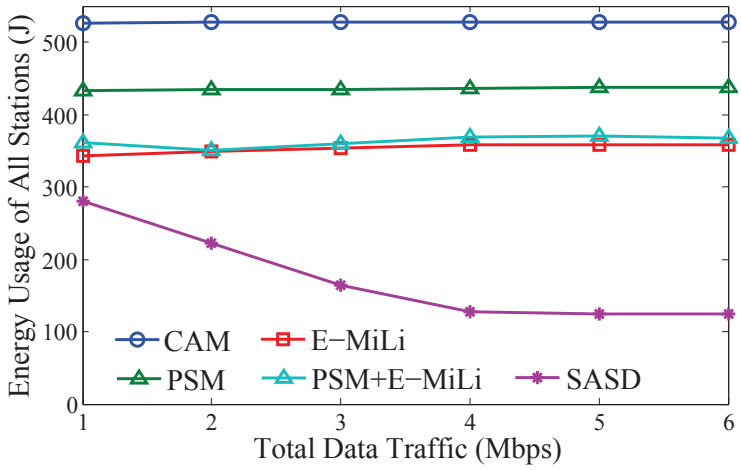
$$E_{SASD} = W_{TX} \cdot T_{TX} + W_{RX} \cdot T_{RX} + W_{SP} \cdot T_{SP} + W_{HS} \cdot T_{HS}. \tag{4.5}$$

Please note in this simulation, T_{TX} and T_{RX} for each station do not change as the network traffic is not fully loaded even with 10 stations. Consequently, with the increasing of T_{SP} and decreasing of T_{HS}, E_{SASD} is decreasing. In contrast, the stations in the PSM or PSM+E-MiLi are periodically active and sleeping, T_{SP} in the PSM or PSM+E-MiLi is almost fixed, but the stations have to spend extra energy in T_{HS} to decode the control packets in the PSM+E-MiLi. Thus, the energy performance in the PSM+E-MiLi is even a little worse than the E-MiLi.

To verify the energy performance under the full workload, we have fixed the number



(a) Energy vs number of stations.



(b) Energy vs data traffic.

Figure 4.12: Energy saving performance comparison in single AP with multiple stations scenario.

of stations to 8 and tested the energy cost performance under different data traffic loads. We have fixed the total data traffic loads to different levels and randomly generated the data traffic load on each station. Fig. 4.12(b) shows that, with various data traffic loads, the SASD can achieve a better energy saving performance than the CAM (76.3%), PSM (71.8%), E-MiLi (64.5%) and PSM+ E-MiLi (65.3%) when the data traffic load reaches $6Mbps$. The energy usage of all stations in the SASD is decreasing when the total data

traffic loads are increasing (from $1Mbps$ to $4Mbps$). We believe that it is because the time duration that the station spends in the sleeping mode, T_{SP} in Eq. (4.4), is rapidly increasing. After that, even if we increases the data traffic load (from $4Mbps$ to $6Mbps$), the energy usage of all stations in the SASD stops decreasing and maintains a constant level due to the network throughput reaches to the maximum and the time durations in different modes are almost fixed.

Fig. 4.13 shows that, for the network throughput, the SASD has almost the same network throughput as the CAM, which means that the SASD has very little impact on the network throughput.

Moreover, we have varied the packet size to investigate its impact on the SASD's performance. Fig. 4.14 shows that, although the SASD is more sensitive than the E-MiLi, the impact of the packet size on the SASD's performance is very little under various network traffic loads.

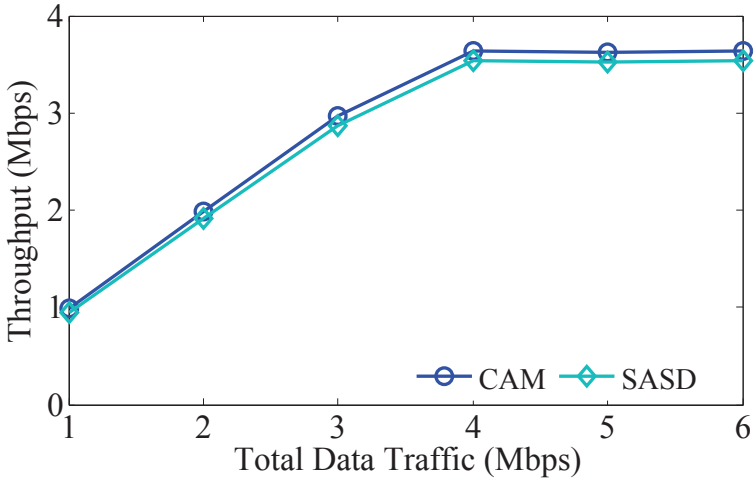


Figure 4.13: Throughput comparison under different data traffic loads.

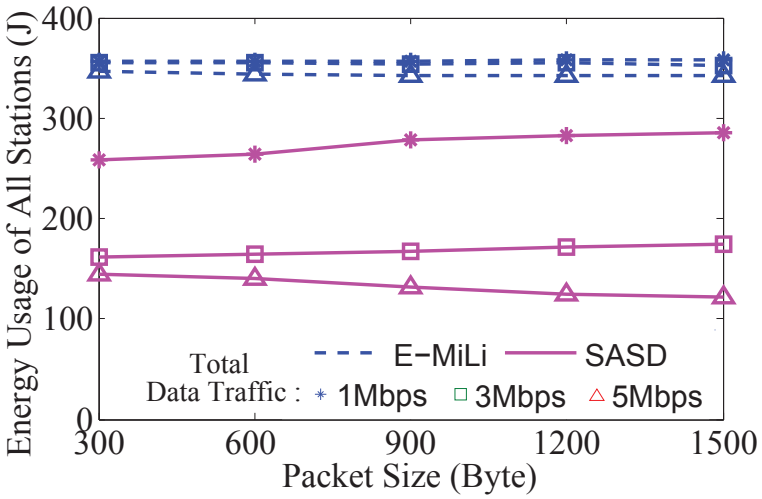


Figure 4.14: Energy saving performance vs packet size.

4.5.2 Multiple APs with Multiple Stations Scenario

To evaluate the SASD's scalability and generality, we have generated a multiple APs with multiple stations scenario that consists of 3 APs and 18 stations. Specifically, we have setup 3 WLANs, each of which has 1 AP and 6 stations and the stations are randomly distributed around the AP (Fig. 4.15). For each WLAN, we have varied the data traffic loads from $1Mbps$ to $6Mbps$ with 1100-byte packet size. Thus, we have evaluated the SASD in a more general environment. Fig. 4.16 shows the result that, even under high traffic loads ($6Mbps$), the SASD can still outperform the CAM (76.2%), PSM (70.7%), E-MiLi (64.9%) and PSM+E-MiLi (65.7%) on the energy efficiency.

The above simulation experiments reveal that the SASD scheme can detect and identify the SASD header in the downclocking mode and switch the device to the sleeping mode to save the energy cost on the packet overhearing. Therefore, the SASD can solve the energy inefficiency on the packet overhearing and achieve better energy efficiency performances in different WLAN environments.

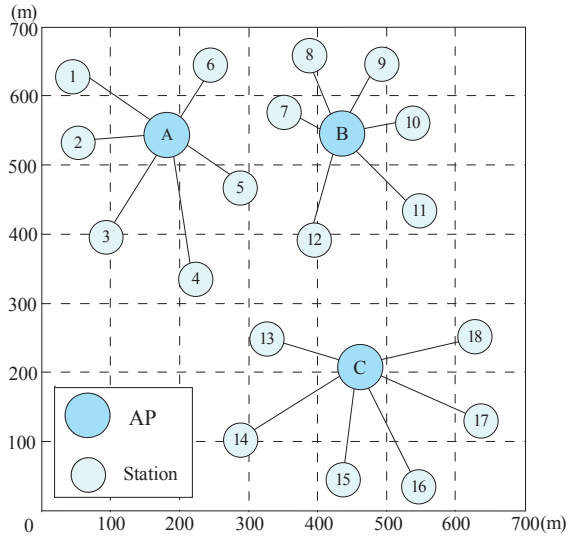


Figure 4.15: The multiple APs with multiple stations network topology.

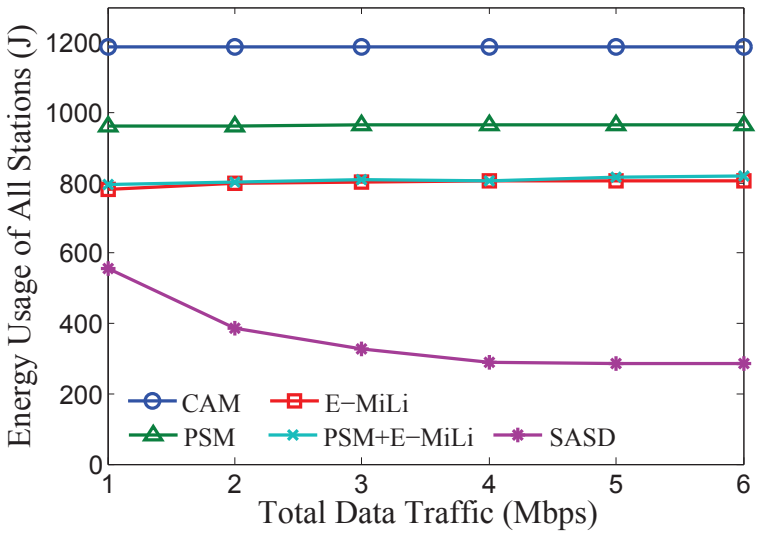


Figure 4.16: Energy saving performance comparison in multiple APs with multiple stations scenario.

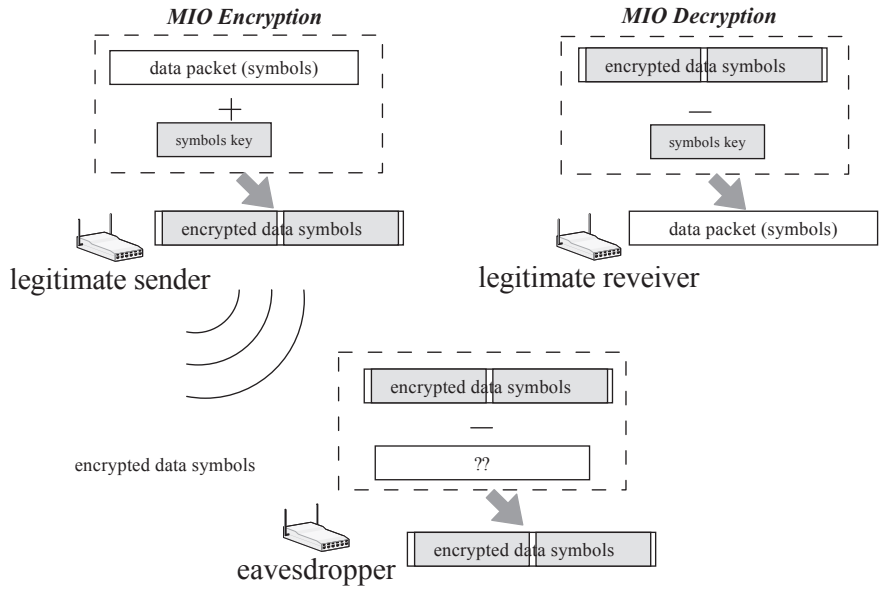
4.6 Summary

In this chapter, a novel SASD scheme is proposed to enable the wireless devices to discern the required information under the energy-saving downclocking mode. Comparing to the previous works, the proposed SASD scheme does not need expensive hardware such as extra circuits or sensors to leverage the advantages of both the sleeping mode and downclocking mode for energy savings. As the SASD scheme effectively eliminates the energy waste on the packet overhearing, it can significantly improve the energy-efficiency of wireless devices. Moreover, the hardware implementation and software simulations demonstrate that the SASD is a practical solution that to the packet overhearing problems which can improve the energy efficiency of the wireless devices.

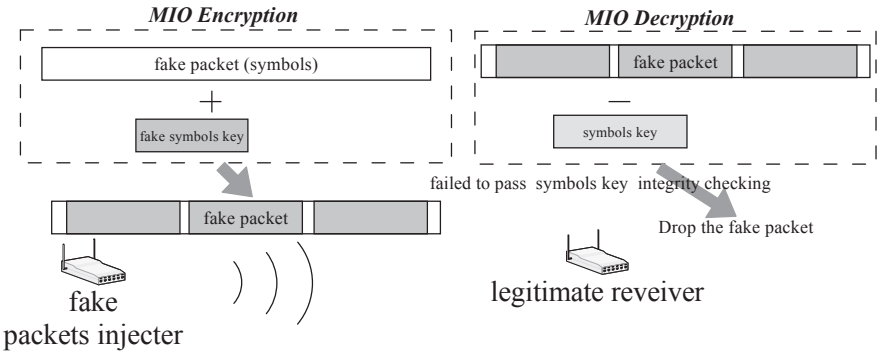
Chapter 5

A Symbol-Level Information Obfuscation Approach to Enhance the Wireless Communications Security

Wireless communications security is a critical and increasingly challenging issue in wireless networks. Due to the broadcast character of the wireless transmission medium, intruders can intercept the transmitting signal, and decode the information. The traditional asymmetric/ symmetric cryptographic techniques which can only provide the computational secrecy are becoming weak in these days, as the power of the computer is rapidly developed. In this chapter, I present the physical layer Multiple Inter-symbol Obfuscation (MIO) which can provide the information-theoretic secrecy to enhance the wireless communications security. The MIO utilizes a set of artificial noisy symbols (symbols key) to obfuscate the original data symbols in the physical layer that can effectively defend against the passive eavesdropping attack and the fake packet injection attack (Fig. 5.1). First, we detail the two attack models that the MIO defends against with.



(a) MIO against the passive eavesdropping attack.



(b) MIO against the fake packet injection attack.

Figure 5.1: The overview of Multiple Inter-symbol Obfuscation (MIO).

5.1 Attack Model

The wireless communications security is to prevent attackers from intercepting the wireless communications, while still delivering contents to the intended recipients. In this paper, we address two types adversaries, *passive eavesdropping attack* and *fake packet injection attack*, during the wireless communications, just like some former works [28, 30, 36, 49, 55, 62]:

1) *Passive eavesdropping attack*: An adversary eavesdrops on the wireless medium and intercepts the wireless transmission between the legitimate transmitter and receiver. It can attempt to decode the signal from the intercepted signal with the presence of the MIO scheme. The MIO scheme will provide the information-theoretic secrecy to enhance the wireless communications security.

2) *Fake packet injection attack*: An adversary injects fake packets to the legitimate users, triggering the events to further disrupt the users's manner (e.g., mislead the users' operations). Unlike the passive eavesdropping attack, it can deploy the brute-force to test all possible symbols keys to inject a fake packet. The MIO scheme will enhance the computational secrecy to defend against this attack.

However, we do not consider the cases where the legitimate transmitter or receiver is physically compromised because the data confidentiality is no longer ensured no matter what security measure is adopted to secure the wireless communications between two hosts if any one of them is not secured. Additionally, we do not consider the jamming-based Denial of Service (DoS) attack in this paper, where the adversary simply jams the channel with extraordinary transmission power, since the legitimate sender and receiver fail to communicate with each other under this DoS attack.

5.2 System Design

This section provides the design of the Multiple Inter-symbol Obfuscation (MIO) which includes two stages: MIO encryption (adding the artificial noisy symbols key), and MIO decryption (offsetting the artificial noisy symbols key). Although the MIO scheme is designed based on the multiple inter-symbol obfuscation at the physical layer, it still

needs an initial key to start the secure wireless communications. For ease of presentation, Table 5.1 lists the notations used in the following sections.

Table 5.1: Notations

notation	**meaning**
γ	the size of the symbols key
Key_k	the symbols key to be superimposed for k^{th} data packet
$Key_{k,j}$	the j^{th} key symbol of Key_k to be encrypted with the data symbol ($0 \leq j \leq \gamma - 1$)
$E_{Key_{k,j}}(S)$	the encrypted data symbol using key symbol $Key_{k,j}$ on data symbol S
$V_{k,j}$	the angle between the key symbol $Key_{k,j}$ and the Real-axis
α	the magnitude ratio of the key symbol and unit-power symbol ($\alpha = \|key_{k,j}\|/1$)
$\hat{\theta}$	the expected normalization factor of the encrypted symbols ($0 < \hat{\theta} < 1$)
β_c	the cross-correlation threshold that the encrypted symbols can be detected
β_{SNR}	the SNR threshold that the received packet can be correctly decoded
R_{re}	the maximum retransmission times for each packet

5.2.1 Initialization

To initiate the first symbols key in a non-secure wireless channel, we first take the conventional key agreement protocols, e.g., EKE or augmented EKE [3, 4], to achieve a bit-level authenticated key. Then, the bit-level authenticated key can be used to generate param-

eters by a one-way harsh function. After that, the parameters, which include the size of the symbols key γ, the angle between the key symbol and the Real-axis $V_{k,j}$ and the magnitude ratio of the key symbol and unit-power symbol α, are used to generate the first symbols key without any trusted third party. Obviously, the legitimate transmitter and receiver have to exchange some redundant packets and deploy the same set of hash functions for generating these parameters.

As the bit-level key agreement schemes can only provide computational secrecy but not the information-theoretic secrecy, the key can be compromised if the eavesdropper has enough computational power (detailed in Section 5.3.1). Moreover, the initial key still requires the legitimate transmitter and receiver to exchange redundant packets to generate different keys for different data packets. Thus, it introduces a high overhead. During the later data packet transmissions, the legitimate parties would employ the MIO scheme to generate the subsequent dynamic noisy symbols keys and deploy the multiple inter-symbol obfuscation scheme to interfere the eavesdropping channel, which can provide the information-theoretic secrecy to enhance the wireless communications security.

5.2.2 MIO Encryption

We first consider that legitimate transmitter A is about to send N data packets to legitimate receiver B. As shown in Fig. 5.2, for each data packet, it goes through the MIO encryption process by two steps: (1) symbols obfuscation and normalization and (2) symbols key update at the transmitter.

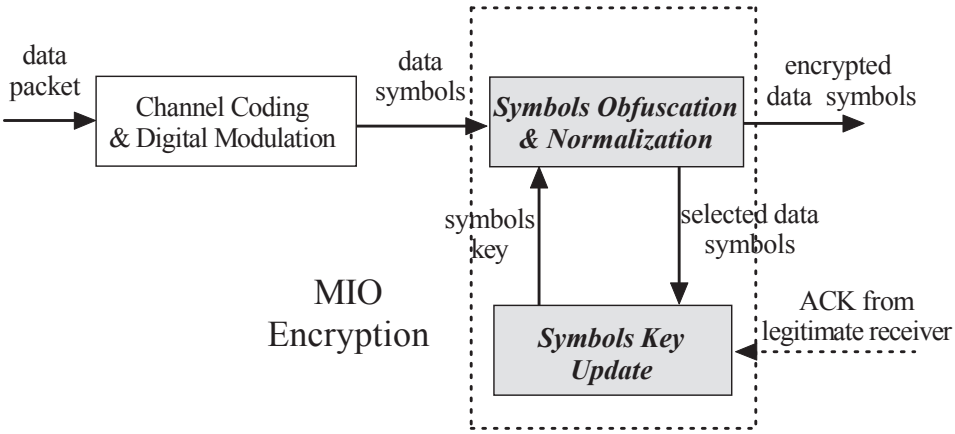


Figure 5.2: The MIO encryption process at a legitimate transmitter.

5.2.2.1 Symbols Obfuscation and Normalization

When a data packet P_k $(1 \leq k \leq N)$ is transmitted, transmitter A will map P_k to a series of L baseband data symbols $M_k = \{m_{k,0}, ...m_{k,l}, ...m_{k,L-1}\}$ using the modulation constellation diagram. Each data symbol $m_{k,l}$ $(0 \leq l \leq L-1)$ is represented as:

$$m_{k,l} = |m_{k,l}|e^{j\phi_{k,l}}, \tag{5.1}$$

where $|m_{k,l}|$, $\phi_{k,l}$ are the magnitude and angle of the l^{th} symbol vector, respectively. These data symbols are generated by the Channel Coding & Digital Modulation block in Fig. 5.2.

After mapping, the transmitter randomly picks up ξ blocks of data symbols, where $\xi = \lfloor \frac{L}{\gamma} \rfloor$, from M_k for encryption[1]. For each chosen data symbols block that begins with the i^{th} data symbol, the corresponding $(i+j)^{th}$ data symbol vector $m_{k,i+j}$ is added with the j^{th} key symbol vector $Key_{k,j}$ to generate an encrypted data symbol $E_{Key_{k,j}}(m_{k,i+j}) =$

[1]In case $L < \gamma$, the MIO randomly appends some dummy data symbols to make $L = \gamma$.

$Key_{k,j} + m_{k,i+j}$, which is illustrated in Fig. 5.3.

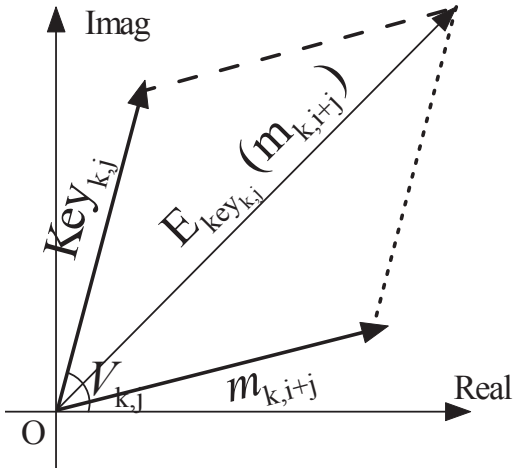


Figure 5.3: The obfuscation of a baseband data symbol $m_{k,i+j}$ with a key symbol $Key_{k,j}$ on the constellation diagram.

As shown in Fig. 5.4, the average power of the encrypted symbols (dot-line curve) would not be the same as that of the original data symbols (solid-line curve) at the transmitter. This energy difference can make the eavesdropper distinguish the encrypted symbols from the non-encrypted ones according to the surge of transmission power. To avoid this problem, the encrypted symbols should be normalized before they go to the digital to analog converter (DAC). After normalization, the energy of the encrypted symbols (dot-dash-line curve) is almost the same as that of the data symbols. Consequently, the eavesdropper is hard to determine whether the received symbols are non-encrypted data symbols or the encrypted symbols. The normalized factor θ can be calculated as:

$$\theta = \frac{\frac{1}{\gamma\xi}\sum_{j=0}^{\gamma\xi-1}|m_{k,i+j}|}{\frac{1}{\gamma\xi}\sum_{j=0}^{\gamma\xi-1}|E_{Key_{k,j}}(m_{k,i+j})|}. \tag{5.2}$$

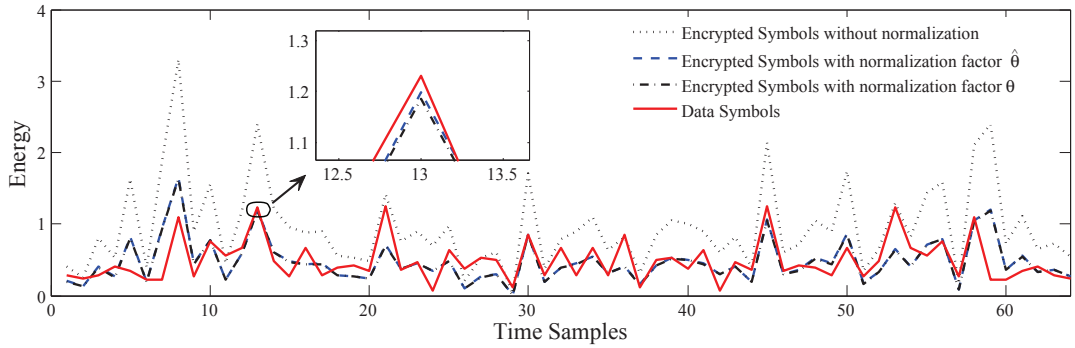


Figure 5.4: Energy of signal samples in the time domain.

Eq. (5.2) implies that the calculation of θ relies on the information of all original and encrypted data symbols, which is hard for the receiver to obtain before the MIO decryption. Fortunately, as the data symbols and key symbols are generally uniformly distributed, when γ is large enough, MIO can use the expected normalized factor $\hat{\theta}$ to replace θ.

The calculation of $\hat{\theta}$ considers all combination possibilities of data symbols and key symbols. Assume that the symbol set used for mapping data on the constellation diagram is $\{S_u\}$ and the probability that S_u is chosen is a_u; the set of key symbols is $\{Key_v\}$ and the probability that Key_v is used for encrypting data symbol is b_v. Then, the encrypted symbol $E_{Key_v}(S_u) = S_u + Key_v$ and the probability that $E_{Key_v}(S_u)$ is generated is $a_u \cdot b_v$. The $\hat{\theta}$ can be computed as:

$$\hat{\theta} = \frac{\sum_u a_u |S_u|}{\sum_{v,u} a_u \cdot b_v \cdot |E_{Key_v}(S_u)|}, \tag{5.3}$$

where $|S_u|$, $|E_{Key_v}(S_u)|$ are the magnitudes of S_u and $E_{Key_v}(S_u)$ on the constellation diagram. Obviously, both the legitimate transmitter and receiver can calculate the value

of $\hat{\theta}$ without knowing information of original and encrypted data symbols in advance.

In Fig. 5.4, the QPSK modulation is used for data mapping. Both data symbol and key symbol are unit-power vectors. The angles of the data symbols are $\{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$, and the probability of each data symbol is $\frac{1}{4}$. For the key symbols, the angles are set $\{\frac{\pi}{2}, -\frac{\pi}{2}\}$ and the probability of each key symbol is $\frac{1}{2}$. As the figure shows, with $\gamma = 60$, the energy of the two normalized encrypted data symbols (dot-dash-line curve and dash-line curve) are almost the same.

Please note that this normalization may decrease the transmission power of the data symbols and we detail this power loss, called as dB loss, in Section 5.4.2.

5.2.2.2 Symbols Key Update at Transmitter

After symbols encryption and normalization, the symbols key to encrypt next data symbols is dynamically updated by using the privacy amplification with one-way hash function [9]. The symbols key Key_{k+1} for the next data packet is generated from the data symbols which are encrypted in the current data packet. Because $\gamma\xi$ data symbols are randomly and independently selected, and encrypted with the noisy symbols key Key_k, when they are transmitted, the noise symbols interfere the eavesdropping channel, which makes the eavesdropping channel's quality much worse than the legitimate channel, the adversary has a low chance to decrypt the $\gamma\xi$ data symbols without knowing the noisy symbols key Key_k. Thus, the $\gamma\xi$ data symbols are completely confidential to the adversary.

After the MIO encryption, as the selected $\gamma\xi$ data symbols are stored in the array t, this array is completely confidential to the adversary. By using the array t as input to

the privacy amplification with one-way hash function, the distilled symbols key Key_{k+1} is also confidential to the adversary[2]. Also, the one-way hash function can guarantee that the symbols keys are not correlated even if the data symbols in consecutive packets are correlated [31].

A problem with this key update scheme is that, the first noisy symbols key is not protected by the noise symbols, just like other physical layer security schemes [8]. Fortunately, under certain situations, even if the first symbols key is cracked, it cannot help the adversary decrypting other encrypted data packets from the first symbols key (detailed in Section 5.6.1) because the succedent noisy symbols keys are dynamically updated. However, this dynamic symbols key update mechanism requires all the symbols to be decrypted successfully for the next data packet at the legitimate receiver side to synchronize the noisy symbols key, consequently, the transmitter has to wait for the *correct* acknowledgment (ACK) from the receiver before it can process the next packet. In a hostile scenario, an adversary might inject forged ACKs to disrupt the symbols key update process between the legitimate transmitter and receiver, which will be further discussed in Section 5.3.5.

The MIO encryption process algorithm is shown in Algorithm 5.1:

[2]The $\gamma\xi$ data symbols in the array t can be mapped into bits and these bits are also confidential to the adversary. After we get the new distilled bits key, MIO would use the one-way hash function to map the bits key into the symbols key.

Algorithm 5.1 MIO Encryption Process

Input: Key_1 is generated at initialization stage. N data packets are to be transmitted.
Output: Encrypted symbols of P_k.

for $k = 1$ to N **do**
  Map the k^{th} packet P_k to L data symbols $m_{k,0}, ...m_{k,i}, ...m_{k,L-1}$;
  Randomly select ξ blocks of data symbols out of L data symbols;
  Store all $\gamma\xi$ selected data symbols in the array t; /*for next symbols key generation*/
  for each selected data symbols block begins with the i^{th} data symbol **do**
    for $j = 0$ to $\gamma - 1$ **do**
      $E_{Key_{k,j}}(m_{k,i+j}) = Key_{k,j} + m_{k,i+j}$;
      $m_{k,i+j} \leftarrow \hat{\theta} \cdot E_{Key_{k,j}}(m_{k,i+j})$; /*encrypted symbol normalization*/
    end for
  end for
  Set retransmission counter $c_k = 0$;
  Send the encrypted data symbols M_k to the receiver;
  while receive no ACK packet P_{ack_k} from the receiver before timeout $\wedge$ $c_k \leq R_{re}$ **do**
    Retransmit M_k to the receiver;
    c_k++;
  end while
  Generate Key_{k+1} for P_{k+1} by using array t as input to the privacy amplification with one-way hash function;
end for

5.2.3 MIO Decryption

As shown in Fig. 5.5, when those encrypted symbols arrive at the legitimate receiver through the wireless channel, the receiver would conduct the MIO decryption process in two steps: (1) key checking and symbols decryption and (2) symbols key update at the receiver.

5.2.3.1 Key Checking and Symbols Decryption

Upon receiving signals by the legitimate receiver (or adversary), the RF down-converter samples the incoming signal, and observes a stream of discrete complex baseband sym-

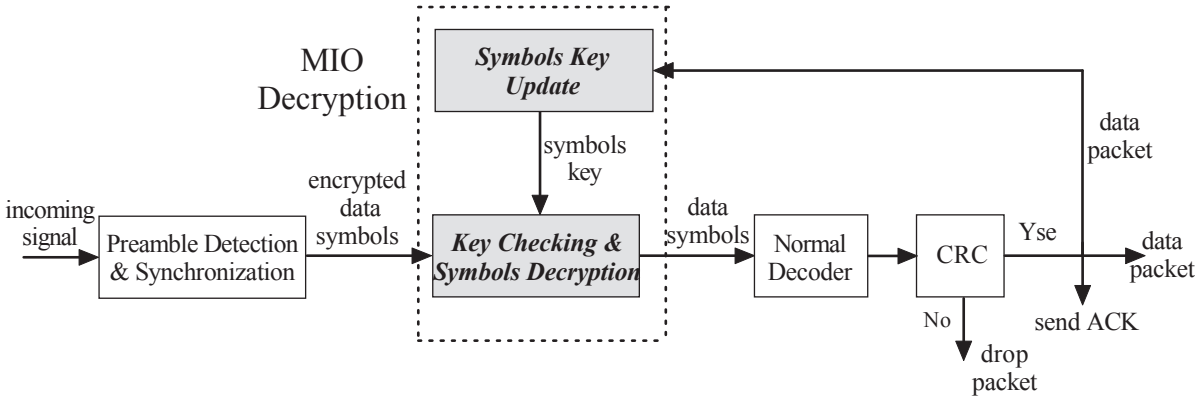


Figure 5.5: The MIO Decryption process at legitimate receiver.

bol vectors. In MIO, for any given transmitted encrypted symbol ($E_{Key_{k,j}}(m_{k,i+j})$), the received encrypted symbol $y_{k,i+j}$ can be represented as:

$$\begin{aligned} y_{k,i+j} &= H \cdot \hat{\theta} \cdot E_{Key_{k,j}}(m_{k,i+j}) + w_{k,i+j} \\ &= H \cdot \hat{\theta} \cdot (Key_{k,j} + m_{k,i+j}) + w_{k,i+j} \\ &= H \cdot \hat{\theta} \cdot Key_{k,j} + H \cdot \hat{\theta} \cdot m_{k,i+j} + w_{k,i+j}, \end{aligned} \tag{5.4}$$

where H and $w_{k,i+j}$ denote the wireless channel coefficient and Gaussian noise (in complex vectors), respectively. H is learned from the pilot sequence [44,46] while $Key_{k,j}$ is updated from the previous data packet. The decrypted data symbol $\hat{y}_{k,i+j}$ can be computed as:

$$\begin{aligned} \hat{y}_{k,i+j} &= y_{k,i+j} - H \cdot \hat{\theta} \cdot Key_{k,j} \\ &= H \cdot \hat{\theta} \cdot Key_{k,j} + H \cdot \hat{\theta} \cdot m_{k,i+j} + w_{k,i+j} - H \cdot \hat{\theta} \cdot Key_{k,j} \\ &= H \cdot \hat{\theta} \cdot m_{k,i+j} + w_{k,i+j}. \end{aligned} \tag{5.5}$$

As described in Section 5.2.2.1, the encrypted symbols blocks are randomly selected

when a new packet (data symbols) goes to the Symbols Obfuscation & Normalization block at the legitimate transmitter. This randomly pick-up mechanism can enhance the security level. However, at the receiver side, it would cause the legitimate receiver hard to locate those encrypted symbols blocks due to: (1) the positions of those encrypted symbols blocks cannot be carried in the last packet because the sizes of adjacent data packets are independent from one other; (2) the receiver cannot precisely determine whether the received symbols are the packet's data symbols at the physical layer during the wireless communications[3].

To precisely discern those encrypted symbols blocks, the legitimate receiver adopts a *cross-correlation* (Section 2.5) operation with the assistance of the symbols key, called *Key Checking*. The cross-correlation value at position i with γ symbols key for k^{th} encrypted packet, $C(i, \gamma, k)$, can be computed as:

$$C(i, \gamma, k) = |\sum_{j=0}^{\gamma-1} \overline{Key}_{k,j} \cdot y_{k,i+j}|, \tag{5.6}$$

where $\overline{Key}_{k,j}$ is the complex conjugate of $Key_{k,j}$. Assume $m_{k,i}$ is the first encrypted data symbol of one selected encrypted symbols blocks at transmitter A, by replacing $y_{k,i+j}$ in Eq. (5.6) with Eq. (5.4), the correlation value is:

$$C(i, \gamma, k) = |H| \cdot \hat{\theta} \cdot \sum_{j=0}^{\gamma-1} |Key_{k,j}|^2 + |O(i, \gamma, k)|, \tag{5.7}$$

[3]In [38], it deploys a redundant "postamble" field to explore a packet's end at the physical layer. We do not consider this kind of field because it is not a standard field in current wireless communications.

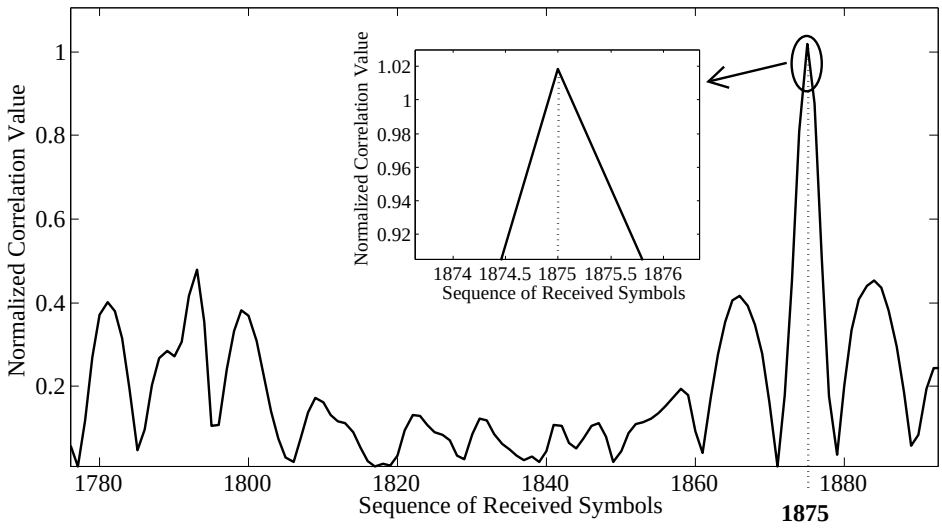


Figure 5.6: Identification of one encrypted data symbols block with normalized cross-correlation: The normalized peak correlation value being larger than the threshold indicates the corresponding encrypted data symbols (e.g., from 1816^{th} to 1875^{th}, $\gamma = 60$) are obfuscated with the given symbols key.

where

$$|O(i,\gamma,k)| = |H \cdot \hat{\theta} \cdot \sum_{j=0}^{\gamma-1} \overline{Key}_{k,j} \cdot m_{k,i+j} + \hat{\theta} \cdot \sum_{j=0}^{\gamma-1} \overline{Key}_{k,j} \cdot w_{k,i+j}|.$$

Since $Key_{k,j}$ is independent of either the data symbol $m_{k,i+j}$ or the noise symbol $w_{k,i+j}$, $|O(i,\gamma,k)| \approx 0$ [68]. In MIO, when the correlation value $C(i,\gamma,k)$ is larger than a threshold value β_c, the corresponding symbols are identified as the encrypted symbols, which is shown in Fig. 5.6. Normally, the threshold β_c can be defined as $\beta_c = \psi_c \cdot \gamma \cdot RSSI_{signal}$ [29, 68], where ψ_c is a constant (Here, we use $\psi_c = 0.9$) and $RSSI_{signal}$ is the received signal's strength indicator. Thus, the encrypted symbols' localization is conducted by checking the inequation:

$$\frac{C(i,\gamma,k)}{\gamma \cdot \hat{\theta} \cdot RSSI_{signal}} \geq \psi_c. \tag{5.8}$$

It is clearly that by using this cross-correlation operation, the legitimate receiver can

eliminate the channel noise influence to locate the correct position i for each encrypted symbols block without any packet information (e.g., the first symbol of the packet and the relative positions of the encrypted symbols blocks in the packet). This makes MIO more practical during wireless communications.

After identifying the position of an encrypted symbols block, the legitimate receiver can offset the symbols key by using Eq. (5.5) to calculate the clean data symbols in each block. We call this *Symbols Decryption*. To demodulate the clean data symbols same as the non-encrypted data symbols in the normal decoder (Fig. 5.5), the receiver has to increase the power of the clean data symbols by the factor $\frac{1}{\hat{\theta}}$. Thus, we can have:

$$\begin{aligned} \hat{y}_{k,i+j} &= \frac{1}{\hat{\theta}} \cdot (H \cdot \hat{\theta} \cdot m_{k,i+j} + w_{k,i+j}) \\ &= H \cdot m_{k,i+j} + \frac{w_{k,i+j}}{\hat{\theta}}. \end{aligned} \tag{5.9}$$

Note that the MIO requires frequent cross-correlation operations in the key checking and symbols decryption block to identify the encrypted symbols blocks, which introduces extra time and computation overhead on correlation calculations. However, the complexity of the key checking and symbols decryption block is at the same order as that of the preamble detection and synchronization block, because the preamble detection and synchronization block also deploys the cross-correlation to detect and synchronize the preamble.

5.2.3.2 Symbols Key Update at Receiver

Once the data symbols are decrypted, the receiver maps all these plain data symbols to digital bits in the Normal Decoder (Fig. 5.5) so that the channel coefficient and the

noise (i.e., H and $\frac{w_{k,i+j}}{\hat{\theta}}$ in Eq. (5.9)) can be filtered out. After decoding the digital bits, receiver B will check if the packet P_k is correct through Cyclic Redundancy Check (With some small probability, it may contain undetected errors even if the packet passes the CRC checking. We detail this in Section 5.6.2.). If the received data packet is correct, the packet acknowledgment will be sent back to transmitter A and this acknowledgment[4] will trigger A to update the symbols key for the next packet (Fig. 5.2). Synchronously, the symbols key for the next packet at receiver B will be updated exactly the same as at transmitter side (Section 5.2.2.2). Otherwise, the receiver drops the corrupted data packet and waits for the packet retransmission.

It is noted that in the MIO decryption process, after filtering noises and channel coefficients, the digital bits which are mapped into the data symbols for the key updating are exactly the same as those for the transmitter. Associating with the selected data symbols's position information in the array r, the receiver can store the corresponding selected data symbols in the array t. Thus, it would guarantee that the array t at the receiver for the key update is the same as the array at the transmitter. The MIO decryption process algorithm is shown in Algorithm 5.2.

5.3 Security Analysis

In this section, we first brief the computational secrecy of the initial key. Then, we demonstrate that, without considering the initial key, MIO scheme can provide both information-theoretic secrecy to the passive eavesdropping attack in Section 5.3.2 and computational secrecy to the fake packet injection attack in Section 5.3.3, respectively.

[4]The ACK may get lost during the transmission, which is dealt with in a similar way as the CRC checking.

Algorithm 5.2 MIO Decryption Process

Input: Key_1 generated at the initialization stage; encrypted data symbols of the k^{th} packet P_k;
Output: the k^{th} packet P_k.

while receiving encrypted data packet P_k **do**
  if the first encrypted data symbol $y_{k,i}$ is identified through the cross-correlation with symbols key Key_k (Eqs. (5.6) $\sim$ (5.8)) **then**
    for $j = 0$ to $\gamma - 1$ **do**
      Calculate clean decrypted data symbol $\hat{y}_{k,i+j}$ by Eqs. (5.5) and (5.9);
      $y_{k,i+j} \leftarrow \hat{y}_{k,i+j}$;
      Append the position information $i + j$ of $\hat{y}_{k,i+j}$ in array r;
    end for
  end if
  Map the received decrypted data symbols y_k to digital bits;
end while
if P_k passes the CRC check **then**
  Send P_{ACK_k} to the transmitter;
  Map P_k to L data symbols $m_{k,0}, ..., m_{k,i}, ..., m_{k,L-1}$;
  Find the selected data symbols according to the position information in the array r, and store the data symbols into the corresponding positions in the array t.
  Generate Key_{k+1} for P_{k+1} by using array t as input to the privacy amplification with one-way hash function;
else
  discard P_k and wait for retransmission;
end if

Furthermore, we analyze the MIO's defense against several symbol detection attempts in Section 5.3.4 and the acknowledgment-based key disruption attack in Section 5.3.5.

5.3.1 Computational Secrecy of the Initial Key

As we have described in Section 5.2.1, the MIO scheme has to use the conventional key agreement protocols to start the secure wireless communications. As the MIO scheme does not deploy any trusted third party to issue a certificate authority to the legitimate pairs, it would inevitably cause the secrecy of the first symbols key, which is generated by the password-authenticated key agreement scheme, computationally bounded. Thus,

the adversary can crack the first symbols key if it has enough computational power and the same symbols key has used over a long time. Furthermore, if the eavesdropper can correctly receive all the subsequent encrypted data packets, it can determine all the symbols keys and crack the encrypted data packets. In this situation, the MIO's secrecy is bounded by the first key agreement scheme. Moreover, to avoid the long-term use of the pairwise authentication password, the password would also have to be updated when each communication session is finished. Please note the MIO scheme is not limited to the key agreement protocol which we describe in this paper. It can apply to any bit-level key agreement schemes as the symbols key's parameters and first symbols key can be generated from any bits key by the one-way hash function.

5.3.2 Information-theoretic Secrecy against the Passive Eavesdropping Attack

In this section, we adopt the *secrecy capacity* model to prove that the MIO scheme can achieve the information-theoretic secrecy to the passive eavesdropping attack without considering the initial key. Normally, the secrecy capacity C_s in [52] is defined as:

$$C_s = C_M - C_T, \tag{5.10}$$

where C_M and C_T denote the Shannon capacities of the legitimate and eavesdropping channels, respectively. As MIO is focused on the real time wireless communications, by considering the realization of the quasi-static fading channels [8] and white Gaussian noise, the secrecy capacity in MIO can be written as:

$$C_s = \begin{cases} \log_2(1+\vartheta_M) - \log_2(1+\vartheta_T), & \text{if } \vartheta_M \geq \vartheta_T; \\ 0, & \text{if } \vartheta_M < \vartheta_T. \end{cases} \tag{5.11}$$

Here, ϑ_M and ϑ_T are the SNRs of the legitimate and eavesdropping channels, respectively. Obviously, the secrecy capacity C_s can be enlarged by either increasing ϑ_M or decreasing ϑ_T. It is proved that, when $C_s > 0$, the information-theoretic secrecy can be achieved [8, 52].

According to Eqs. 5.4 and 5.5, ϑ_M and ϑ_T are

$$\vartheta_M = \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2}, \tag{5.12}$$

$$\begin{aligned} \vartheta_T &= \frac{\hat{\theta}^2 \cdot |H_E|^2 \cdot |m_{k,i+j}|^2}{\hat{\theta}^2 \cdot |H_E|^2 \cdot |Key_{k,j}|^2 + |w^T_{k,i+j}|^2} \\ &= \frac{\hat{\theta}^2 \cdot |H_E|^2 \cdot |m_{k,i+j}|^2}{\alpha^2 \cdot \hat{\theta}^2 \cdot |H_E|^2 \cdot |m_{k,i+j}|^2 + |w^T_{k,i+j}|^2}, \end{aligned} \tag{5.13}$$

where H_L, H_E, $w^M_{k,i+j}$, $w^T_{k,i+j}$ are the channel coefficients and Gaussian noise of the legitimate and eavesdropping channels, respectively.

Generally, with no Channel State Information (CSI) of the eavesdropping channel, Eq. (5.11) in MIO can be equally as:

$$\begin{aligned} C_s &= \log_2(1+\vartheta_M) - \log_2(1+\vartheta_T) \\ &= \log_2(1+\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2}) \\ &\quad -\log_2(1+\frac{\hat{\theta}^2 \cdot |H_E|^2 \cdot |m_{k,i+j}|^2}{\alpha^2 \cdot \hat{\theta}^2 \cdot |H_E|^2 \cdot |m_{k,i+j}|^2 + |w^T_{k,i+j}|^2}) \\ &> \log_2(1+\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2}) - \log_2(1+\frac{1}{\alpha^2}). \end{aligned} \tag{5.14}$$

From Eq. (5.14), when

$$\alpha \geq \frac{|w^M_{K,i+j}|}{\hat{\theta} \cdot |H_L| \cdot |m_{k,i+j}|}, \tag{5.15}$$

we can have

$$\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2} \geq \frac{1}{\alpha^2}$$

and

$$C_s > \log_2(1+\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2}) - \log_2(1+\frac{1}{\alpha^2}) \geq 0$$

, then, the requirement for the information-theoretic secrecy, i.e., $C_s > 0$, can be guaranteed [14, 78].

Note that $\vartheta_M = \frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2}$ is the SNR of the legitimate channel. In 802.11, this SNR's value should be larger than a threshold β_{SNR} (Normally, $\beta_{SNR} > 0dB$), so that legitimate transmitter and receiver can have normal communications, which means

$$\frac{\hat{\theta}^2 \cdot |H_L|^2 \cdot |m_{k,i+j}|^2}{|w^M_{k,i+j}|^2} > 1. \tag{5.16}$$

Combining Eqs. (5.14) and (5.16), when $\alpha \geq 1$, C_s in MIO can be a positive secrecy capacity. Associating with the privacy amplification with one-way hash function which is used in MIO's dynamic symbol key update mechanism, MIO can achieve information-theoretic secrecy without considering the first symbols key. Furthermore, different from other artificial noise approaches [27, 28, 30, 42, 49, 73], MIO does not need any CSI of the eavesdropping channel. Most importantly, this secrecy capacity C_s would always stay positive regardless of the eavesdropper's location. Thus, MIO's information-theoretic secrecy would not be compromised by the location of the eavesdropper. However, this information-theoretic secrecy of the MIO scheme still needs the assumption that the first symbols key is computationally unbounded.

5.3.3 Computational Secrecy against the Fake Packet Injection Attack

As the symbols key cannot be correctly derived from the received encrypted symbols, the attacker may attempt the brute-force strategy[5] to test all possible symbols keys to inject the fake packet, i.e., it has to try $\tau^{\frac{\gamma}{2}}$ combinations on average to test a symbols key, where γ is the length of the symbols key and τ is the total number of possible key symbols. Compared with standard AES and DES, in which the computational complexity of brute-forcing the key is $2^{\frac{\gamma}{2}}$ (with same key size γ), the MIO's computational complexity against the fake packet injection attack would be much greater than the traditional ones when τ is larger than 2. Fig. 5.7 gives the computational complexity for brute-forcing the symbols key with different key symbols number τ and key size γ. It is clearly that more

[5]In fact, we do consider some other attacks, such as the dictionary attack. As the privacy amplification with one-way hash function can effectively defend against these attacks, due to the pages limitation, we only provide the brute force attack's computational complexity in our paper.

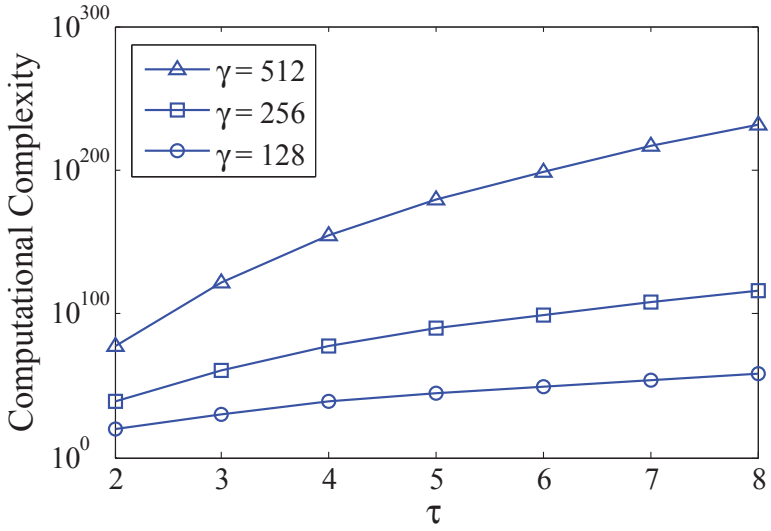


Figure 5.7: The Computational Complexity for Brute-forcing the key with various τ ($2 \leq \tau \leq 8$) and key size γ.

computations are required to test the symbols key as τ increases. Moreover, the attacker has to inject $\tau^{\frac{\gamma}{2}}$ fake packets to achieve a successful fake packet injection attack, which is hardly true in wireless communications. According to Fig. 5.3, the total number of possible key symbols, τ, is determined by the V_j and α, which can be much larger than 2. Thus, using the symbols key can achieve a better computational secrecy to the fake packet injection attack compared with the traditional bit level key.

5.3.4 Defense against Symbol Detection Attempts Attack

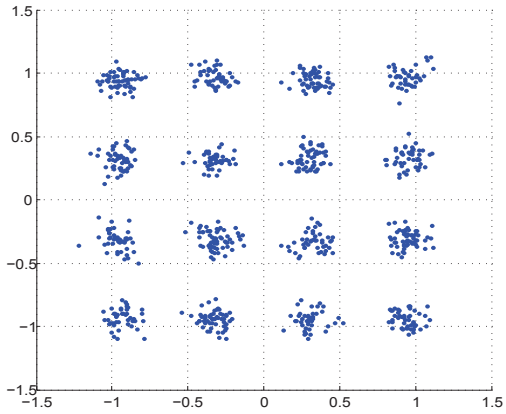
As the MIO encryption would change the location of the original data symbol in the constellation map, the eavesdropper may attempt to deploy symbol detection techniques, such as AMC [63], DMC [59] or constellation map, to distinguish the encrypted/non-encrypted symbols. In this subsection, we explore whether MIO can defeat these symbol detection approaches.

AMC [63] is based on a cyclic feature that different digital modulations have different periodical information associated with time, which could be deployed by the attacker to distinguish different modulations, such as BPSK, QPSK and QAM. However, MIO does not belong to any digital modulations. Even with large amount of training data and supervised learning, AMC still cannot find the encrypted symbols from the received symbols.

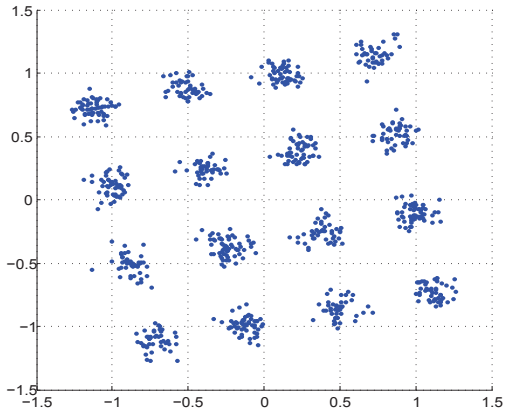
DMC uses the standard constellation shape as basis for finding received symbol's modulation [59]. In its algorithm, the eavesdropper constructs a scatter constellation map of the received symbols and uses the fuzzy c-means clustering to recover the robust constellation map. The reconstruction of constellation map is based on the maximum likelihood with predefined digital modulation templates. Similar as AMC [63], DMC cannot identify the unknown constellation map template (i.e., the MIO's constellation map) from the received symbols. Also, it requires symbol training and supervised learning which cannot be done through a single packet [36].

As we discussed above, the traditional symbol detection techniques can hardly identify the encrypted symbols due to that they more focus on exploring the received symbols' modulation. Next, we explore another symbol detection attempt that is based on a large amount of received symbols: The attacker plots all the received symbols on the constellation map to check if the key can be disclosed from this constellation map. We simulate this constellation map method based on 16-QAM and compare the MIO with symbol flipping based encryption [62] and CD-PHY encryption [36].

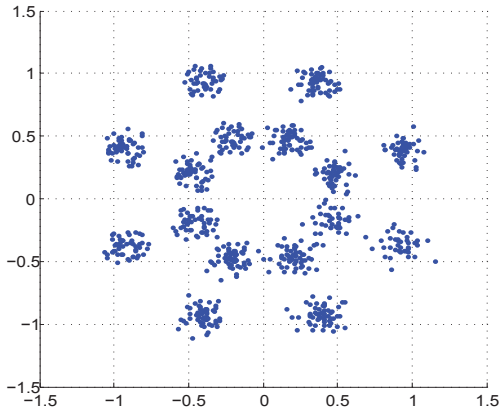
As shown in Fig. 5.8(a), without encryption, the rectangular 16-QAM can be easily identified from the constellation map. So does the symbol flipping based encryption (Fig.



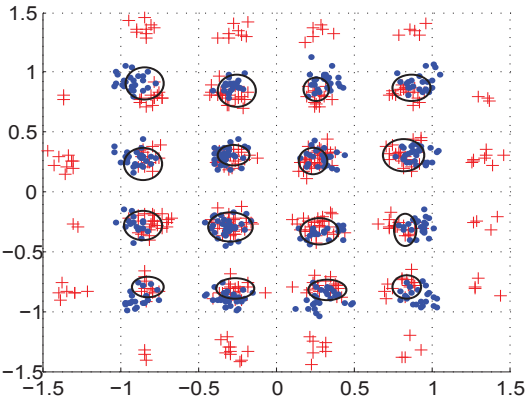
(a) Standard 16-QAM rectangular constellation map.



(b) Symbol flipping based encryption [62].



(c) CD-PHY based encryption [36].



(d) MIO based encryption. (“.” denotes non-encrypted data symbols, “+” denotes encrypted symbols.)

Figure 5.8: Comparison of constellation maps (SNR = 20 dB).

5.8(b)). Compared with Fig. 5.8(a), the key (rotation angle) can be easily conquered from the figure. The same problem exists in the CD-PHY modulation (Fig. 5.8(c)). Although the attacker may not know the symbol-to-bit mapping yet, the constellation diversity key can be easily identified from the large amount of symbols' plotting. However, due to the encryption feature that the MIO will mix up the normalized encrypted symbols with the non-encrypted symbols (Fig. 5.8(d)), the attacker can hardly identify the encrypted symbols from non-encrypted symbols. Even with clear evidences that the received symbols

are manipulated by the MIO scheme, the attacker cannot correctly locate the positions of the encrypted symbols blocks because they are mixed up with data symbols (the circled regions in Fig. 5.8(d)). Consequently, the eavesdropper fails to conquer the symbols key from this symbol detection attempt.

5.3.5 Acknowledgment-based Key Disruption Attack

As described in Sections 5.2.2.2 & 5.2.3.2, acknowledgment plays a crucial role in MIO's real time wireless communications. Each data packet is acknowledged before the next data packet is prepared to be transmitted since the previous data packet generates the symbols key to decrypt the data symbols in the next data packet. As a result, an adversary might try to disrupt MIO's key updating by sending the ACK packets to the legitimate transmitter even though the legitimate receiver fails to receive the correct data packet. Upon the receipt of the ACK, the transmitter might send the next data packet to the receiver, which has not received the previous correct data packet for the key update (Line 15 in Algorithm 5.2). Therefore, the receiver fails to retrieve any subsequent data packet from the transmitter. Such attack is called *acknowledgment-based key disruption attack*.

In MIO, a similar symbols key update mechanism (Sections 5.2.2.2 and 5.2.3.2) for the ACKs is adopted to defend against the acknowledgment-based key disruption attack. In this case, this attack will be the same as the packet injection attack except the roles of the transmitter and receiver are swapped. Although the size of ACK is not large as the size of the data, the privacy amplification with hash function can still generate significantly different symbols key even the input is quite similar [9, 24, 74]. The symbols key which encrypts the ACK's symbols is updated for each ACK. The transmitter will

deny the ACK which is not encrypted by the correct ACK's symbols key through the cross-correlation in Section 5.2.3. Moreover, even if an adversary might occasionally guess correctly an ACK's symbols key and inject a fake ACK, leading to an acknowledgment-based key disruption attack, the legitimate receiver will be aware of the attack because it cannot receive the subsequent data packets any more.

5.4 Implementation Issues

Some implementation issues in MIO are further analyzed as follows:

5.4.1 Symbols Key Checking

The symbols key checking is a crucial part in the MIO system (Section 5.2.3). Missing the first encrypted symbol (false negative error) or finding a wrong symbol (false positive error) is fatal to the MIO's decryption. From Eq. (5.8), the γ and ψ are key parameters in symbols key's identification. Enlarging both γ and ψ can minimize the false positive/negative errors [68]. In our testbed, when $\psi_c = 0.90$ and $\gamma \geq 40$, the false positive/negative errors can be reduced to 0.

Additionally, if the packets are transmitted in a low SNR environment, these packets cannot be correctly demodulated. Consequently, there is no need to find the first encrypted symbol. Therefor, to minimize the computational cost of symbol decryption under the low SNR environment, Eq. (5.8) can be changed to:

$$\frac{C(i,\gamma,k)}{\gamma \cdot max(RSSI_{signal},\ \beta_{SNR} + RSSI_{noise})} \geq \psi_c. \tag{5.17}$$

Here, $RSSI_{noise}$ is the environment noise (typically, $-98 \sim -95dBm$) and β_{SNR} is the SNR threshold that can correctly decode a packet. If the $RSSI_{signal}$ is not high enough to decode packets correctly, by using Eq. (5.17), the receiver would drop this packet because it would not pass the symbols key checking.

5.4.2 dB Loss in MIO

In the MIO system, the legitimate transmitter normalizes the encrypted symbol $E_{Key_{k,j}}(m_{k,i+j})$ by multiplying the normalization coefficient θ (Line 8 in Algorithm 5.1). As a result, this will attenuate the transmission power of the data symbol, we call it *dB loss* in MIO, and such power loss can be computed as:

$$\begin{aligned} dB_{loss} &= Original\ data\ energy - Normalized\ data\ energy \\ &= 10 \cdot \lg(|m(k,i+j)|^2) - 10 \cdot \lg(|\hat{\theta} \cdot m(k,i+j)|^2) \\ &= 10 \cdot \lg \frac{|m(k,i+j)|^2}{|\hat{\theta} \cdot m(k,i+j)|^2} \\ &= 10 \cdot \lg \frac{1}{\hat{\theta}^2} \\ &= -20 \cdot \lg \hat{\theta}. \end{aligned} \tag{5.18}$$

Obviously, associated with Eq. (5.3), the dB loss is closely related to α. When α increases, there is more dB loss in the encrypted symbol normalization.

5.5 Hardware Experiment and Performance Evaluation

We employ 5 USRP2 [19] with 2.4GHz-based RFX2400 daughter-board, and each USRP2 is connected to a notebook. All the notebooks are operated under Ubuntu 10.04 and installed the GNURadio software [19]. We evaluate the signal to noise ratio (SNR) by calculating the $SNR \approx RSSI_{signal} - RSSI_{en}$, where $RSSI_{signal}$ and $RSSI_{en}$ are the received signal strength indicators of the transmitted signal and the environment noise, both of which are measured from the USRP2 hardware. As we can adjust the transmitter's sending power and distance between the transmitter and receiver, we can get various $SNRs$ to test the MIO's performance. We choose QPSK as our data modulation scheme, which is widely adopted in the IEEE 802.11 standards for WLAN implementation [28]. For the key symbols, we take the key angles to be $\{\pm\frac{\pi}{2}\}$ and $\alpha = 1, 1.2$ and 1.4 to test the performance of the MIO scheme. The data packet consists of a 80-bit preamble, and a 200-byte payload, which means there are 800 data symbols for each packet. For each run of the experiment, 5000 data packets are sent and we repeat the experiment for 12 times. Also, we simulate the MIO scheme using the MATLAB with Simulink.

5.5.1 Experimental Results

In the hardware experiments, we first verify the BER performance at legitimate receiver with different α value. Compared with the BER in a non-MIO scenario, the decrypted data at legitimate receiver has a slight dB loss ($1\sim1.6dB$) when the $\alpha = 1$ (Fig. 5.9). However, this dB loss at legitimate receiver would go up to $2.5dB$ and $3.7dB$ when the $\alpha = 1.2$ and $\alpha = 1.4$, respectively.

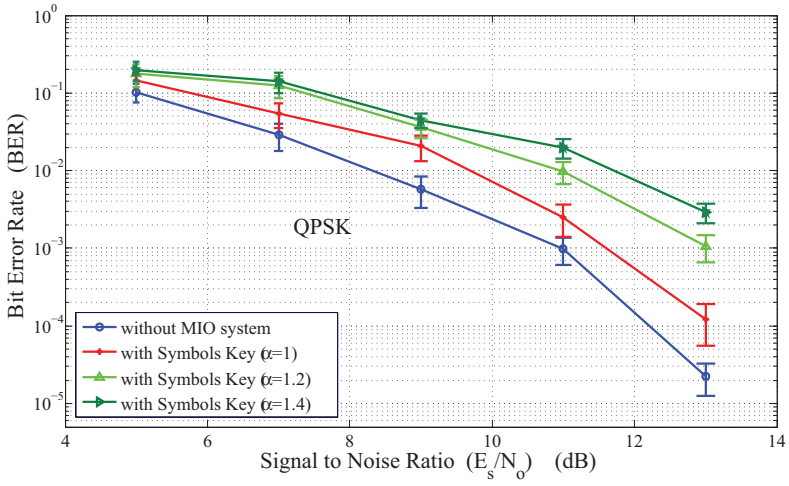


Figure 5.9: Bit error rate at the legitimate receiver (with symbols key, $\gamma = 800$).

At the eavesdropper side, without knowing the symbols key, the BER would remain constant, roughly at 0.27 ($\alpha = 1$) no matter what signal to noise ratio (SNR) is adopted (Fig. 5.10). We believe this BER can already ruin the packet reception at the eavesdropper. Moreover, this BER does not increase along with the rising α value and it stabilizes around 0.48 when the $\alpha > 1$.

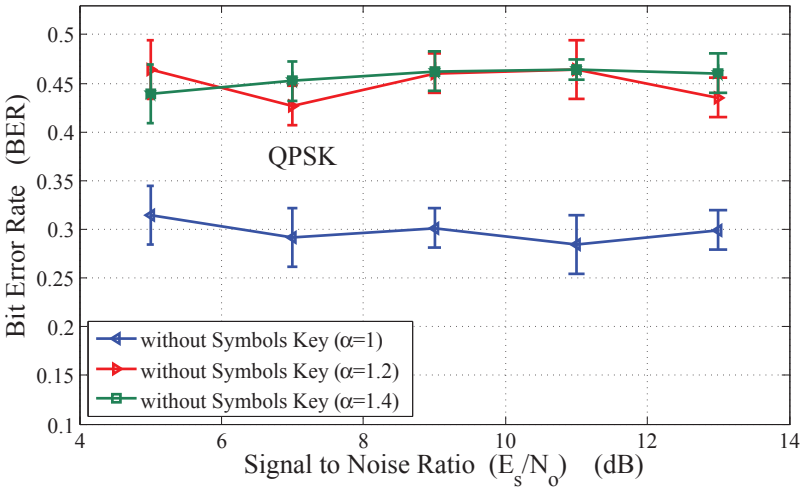


Figure 5.10: Bit error rate at the eavesdropper (without symbols key, $\gamma = 800$).

Regarding the symbols key checking, two size of symbols key ($\gamma = 40$ and $\gamma = 60$) are

testified. The result shows that the false negative(FN) error rate and false positive(FP) error rate can be 0% at the legitimate receiver when $\beta_{SNR} = 11dB$ (Fig. 5.11). It guarantees that the legitimate receiver would not make any mistakes at key checking process when the SNR is good enough to decode the packet[6].

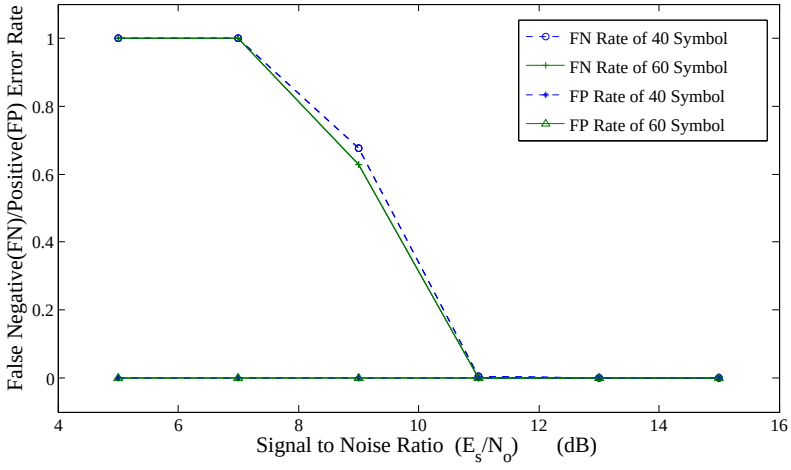


Figure 5.11: False Negative (FN)/Positive (FP) error rate at the legitimate receiver, with two different symbols key sizes under different SNRs. ($\alpha = 1$, $\beta_{SNR} = 11dB$, $\psi = 0.8$)

5.5.2 Simulation Results

To fully evaluate the MIO scheme's performance, we simulate the MIO scheme in various digital modulations (QPSK, 16/64-QAM). Fig. 5.12 depicts the BER between the non-MIO scenario and the MIO one. In regard to 16/64-QAM, it is clear that the MIO system does not need to suffer too much dB loss ($1.3dB$ with $\alpha = 0.6$) to meet the encryption requirement. Please note that, in 802.11, the QAM modulation requires high SNRs ($\beta_{SNR} > 11dB$) to demodulate the bits, as a result, $\alpha = 0.6$ can still guarantee a positive secrecy capacity. Also, in the QPSK, it has the same BER trend for the testbed as the

[6]This SNR value is much related to the modulation and channel coding [76]. For example, for the BPSK modulation with gray coding, the SNR requires above 9.7 dB for correctly receiving a packet. This value becomes 11 dB if QAM-16 is adopted.

one for the simulation. Furthermore, without knowing the symbols key, the BERs are 0.27 (16-QAM) and 0.18 (64-QAM), those BERs are enough to ruin the packet reception even with a channel coding [37].

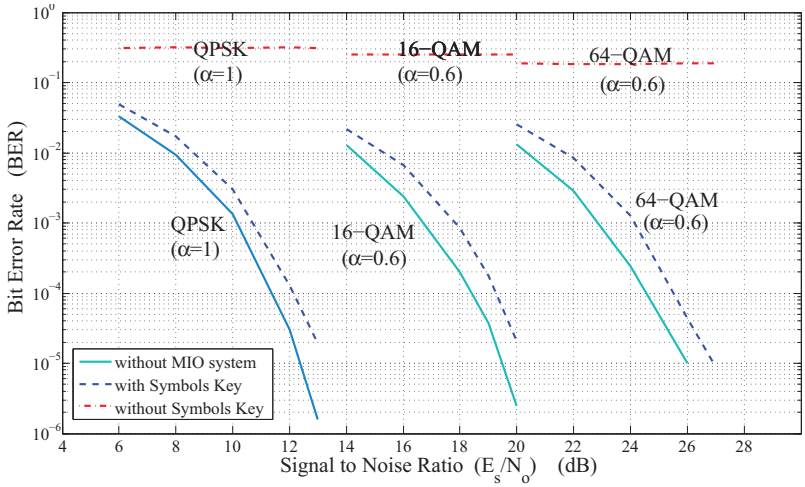


Figure 5.12: Bit error rate in various digital modulations (QPSK: $\alpha = 1$, 16/64-QAM: $\alpha = 0.6$).

As described in Section 5.4.2, the dB loss in the MIO scheme should be carefully considered in system implementation. Although, in Section 5.3.2, it has been proved that when $\alpha = 1$, the positive secrecy capacity can be guaranteed, we still simulate the dB loss trend as long as α increases. Fig. 5.13 reveals the dB loss is increasing with the rising α values.

Also, we simulate the BER performance at the eavesdropper side by using the QPSK modulation. Compared with Fig. 5.10, which is the hardware experimental result, the simulation result in Fig. 5.14 shows that the BER at the eavesdropper side is stabilized around 0.25 ($\alpha = 1$) and 0.5 ($\alpha > 1$). This simulation result is almost the same as the hardware experimental result.

Moreover, from Figs. 5.13 and 5.14, when $\alpha > 1$ ($\alpha = 1.2$ and $\alpha = 1.4$), the BER is

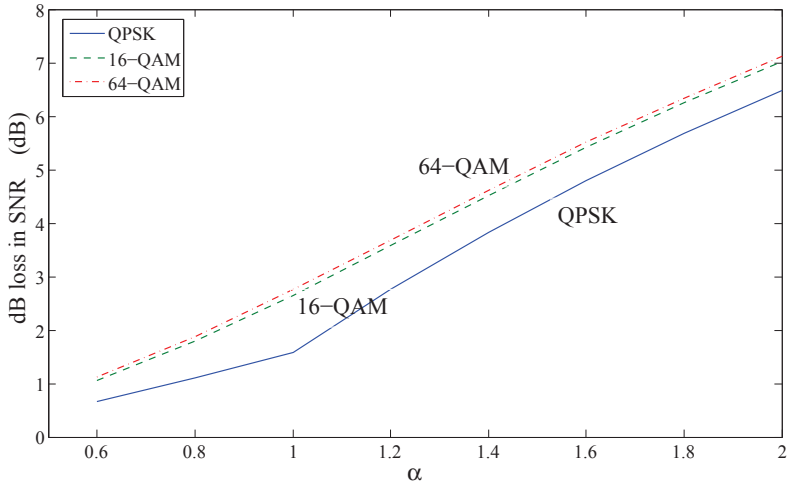


Figure 5.13: The dB loss in the MIO system.

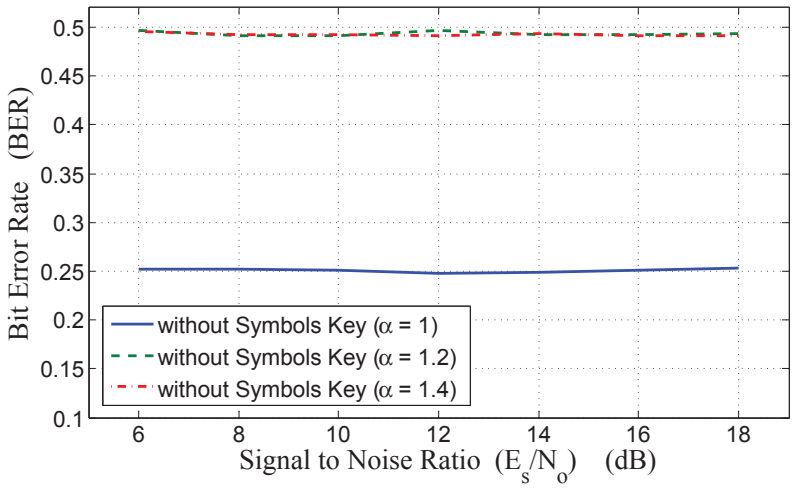


Figure 5.14: The relationship among BER, SNR and α (QPSK).

stabilized around 0.5, but the dB loss is around $3dB$ and $3.8dB$ (QPSK), respectively. The similar results are also shown for the USRP2 testbed (Fig. 5.9 and Fig. 5.10). Thus, increasing α would not increase the BER, but only enlarge the gap of dB loss in the MIO system when the BER reaches the maximum (BER = 0.5).

5.6 Discussion

We further discuss some issues arisen from the MIO scheme that remain unaddressed in this chapter.

5.6.1 Dynamic Key Updating vs. Static Key Updating

Most existing security schemes would adopt a static key updating algorithm to generate the key. Once the legitimate users know the initial key, it would use the key generation algorithm to calculate the corresponding keys for synchronous key updating both at the legitimate transmitter and receiver. However, MIO adopts a dynamic key updating mechanism to generate the symbols key. As described in Sections 5.2, MIO's dynamic key updating contains two aspects of dynamic updating: (1) The new key is generated from the current encrypted packet. (2) The encrypted data symbols are randomly and independently picked up from data packets.

Note that the new symbols key is generated from the current decrypted data symbols (Sections 5.2.2.2 and 5.2.3.2). Even if the eavesdropper is aware of the perfect CSI of the legitimate channel, it is hard for the eavesdropper to calculate the symbols key from one to another, because it is nearly infeasible for the eavesdropper to correctly receive all subsequent encrypted packets to track the symbols key in real time wireless environments

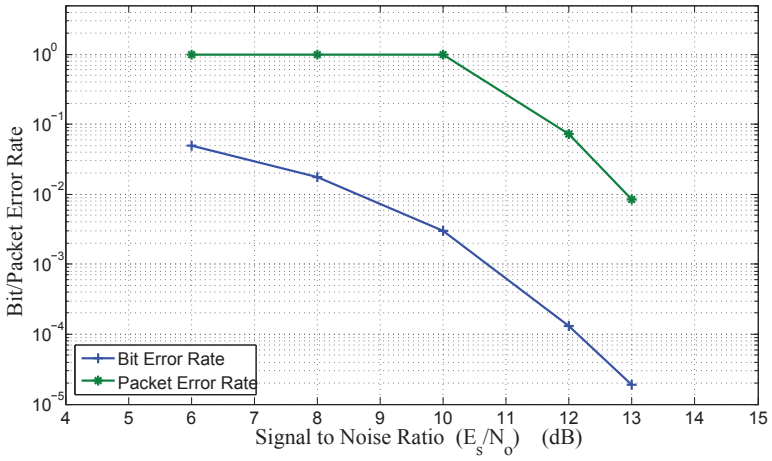


Figure 5.15: Bit/Packet Error Rate with symbols key in MIO. (QPSK modulation with $\alpha = 1$)

with background noises. When the decrypted packet cannot pass the CRC checking at the legitimate receiver, it is dropped and will be retransmitted to the legitimate receiver (within maximum retransmissions). However, if this happens at the eavesdropper side, because the dynamic key cannot be calculated from the previous keys, the eavesdropper fails to update the new symbols key, and loses the chance to decipher the subsequent encrypted packets. Thus, under the condition that the eavesdropper hardly receives all encrypted packets correctly, even if the eavesdropper knows the first symbols key, MIO can still achieve the information-theoretic secrecy of the symbols keys when a transmission error occurs at the eavesdropper. Fig. 5.15 gives the bit/packet error rates under different SNRs, as it shows that even under high SNR environments (SNR $= 13dB$), a packet error occurs within 1000 packet transmissions, which means that, if this error occurs at the legitimate receiver, the receiver would wait for the retransmission of the packet; however, if this error occurs at the adversary side, the adversary loses the chance to track down the new symbols key, the subsequent encrypted packets can achieve the information-theoretic secrecy. Note that in this case the former correctly received encrypted packets can be

cracked by the eavesdropper.

5.6.2 CRC Checking Failure

As we mentioned in Section 5.2.3, an incorrect packet may (with small probability) pass the CRC checking. That may disrupt the synchronization process of key update between the transmitter and receiver (It only happens if the incorrect bits will be used for the key updating). To solve this CRC checking failure, Error Correct Coding can be built in the channel coding procedure. However, this would increase the redundant part in the packet and decrease the network's goodput.

For MIO, the symbols key initialization procedure can solve this CRC checking failure. After the maximum retransmissions, the legitimate transmitter would go back to initialize the first symbols key Key_1 (Section 5.2.1) for the legitimate receiver. As the unique private wireless channel parameter has changed in the wireless environment, the symbols key initialization procedure would not compromise the MIO's security.

5.6.3 The Influence to the Network Throughput

The MIO scheme requires the symbols key to be synchronously updated at both the legitimate sides. It inevitably causes the legitimate transmitter to wait for a feedback by which the legitimate receiver acknowledges the correct reception of an encrypted packet. Then, the ACK can trigger the key update process at both the legitimate sides. Thus, the whole communication system which implements the MIO scheme actually adopts the DATA-ACK model.

In current IEEE 802.11 MAC protocols, two standard mechanisms, CSMA/CA and

RTS/CTS, are used to handle the packet collision problem in the wireless environments. The DATA-ACK model is also implemented for the unicast transmission scenario in the 802.11 MAC protocols[7]. The MIO scheme adopts the same DATA-ACK model for the packet unicast transmissions, consequently, the network throughput of the MIO scheme should be the same as the standard 802.11 MAC protocols in the packet unicast scenario.

However, for the broadcast and multicast scenarios, as the MIO scheme requires the synchronized symbols key to secure the wireless communications, it does not work in the broadcast and multicast scenarios since the transmitter does not require any ACKs for the packet transmissions. A possible solution to this problem is that the MIO scheme forces the packet transmission in the broadcast or multicast scenarios to adopt the packet unicast transmission mechanism. However, it would significantly affect the packet reception, as in the broadcast and multicast scenarios, a packet is transmitted once to all the receivers, but in the unicast scenario, the packet would be transmitted n times to n receivers. We leave this MIO broadcast and mutilcast problem as our future work.

5.7 Summary

In this chapter, the Multiple Inter-symbol Obfuscation (MIO) scheme is proposed to secure the wireless transmissions between two legitimate entities. MIO does not need any trusted third party to interfere the packet interception by the eavesdropper or static channel condition to cancel artificial noise. Rather, it employs the data symbols from the previous data packets to generate the symbols key which obfuscates the current data packets. By dynamically updating the symbols key as the packets are disseminated, it

[7]With the RTS/CTS, the unicast transmission is implemented with the RTS-CTS-DATA-ACK model.

is hard for an adversary to brute-force the symbols key by intercepting a number of encrypted symbols and analyzing them off-line. The mathematical models prove that MIO can provide both information-theoretic secrecy and computational secrecy. Additionally, the experimental results illustrate that without knowing the symbols key, the BER in the MIO scheme can effectively ruin the packet reception at the eavesdropper side, and the key checking process would defend against the packet injection attack in wireless networks.

Chapter 6

Conclusions and Suggestions for Future Research

In this chapter, I first conclude this thesis by summarizing the original contributions in Section 6.1, and outline the directions for future research in Section 6.2.

6.1 Conclusions

In this dissertation, I investigate the symbol-level information extraction mechanism which enables the vital information to be delivered between the transmitters and receivers at the symbol level. I deploy the cross-layer design of symbol-level information extraction mechanism to solve three popular research problems in wireless communications: the hidden terminal problem, the energy inefficiency of the packet overhearing problem and wireless communications security problem. Moreover, the hardware experiments are conducted to testify the effectiveness, flexibility and the robustness of our cross-layer design of the symbol-level information extraction mechanism.

I first deploy the cross-layer design of symbol-level information extraction mechanism, RTS/S-CTS, against the hidden terminal problem. As I explore that most of the existing

solutions of the hidden terminal problem would suffer two drawbacks: the Low-SINR-CTS drawback and the Low-SNR-CTS drawback, the RTS/S-CTS mechanism uses the symbol-level information extraction to deliver the NAV time information at the PHY layer which significantly alleviates the influence of the Low-SNR/SINR-CTS problems. The simulation results show that the feasibility of the performance improvement. More importantly, the hardware experiments analyze the RTS/S-CTS's design parameter. All these efforts make the RTS/S-CTS more practical in the real world network scenarios.

With the design experience of the RTS/S-CTS, I employ the symbol-level information extraction mechanism to eliminate energy inefficiency of the packet overhearing problem. I propose the SASD (Sample-Address Sample-Duration) scheme to deliver the packet's receiver MAC address and transmission duration information in one shot without receiving the whole packet at the PHY layer. Moreover, SASD scheme leverages the advantages of both the sleeping mode and downclocking mode to save wireless devices' energy. However, the challenge of the SASD is whether the symbol-level information extraction mechanism can work under downclocking mode. The hardware experiments are conducted to emulate the A/D downclocking procedure. The results show that the SASD is still functional under downclocking mode. I also simulate the SASD in the different network scenarios, and the simulations results demonstrate that SASD can significantly improve the energy efficiency of wireless devices by eliminating energy inefficiency of the packet overhearing problem.

Finally, I deploy the symbol-level information extraction mechanism in the wireless communications security problem. The MIO (Multiple Inter-symbol Obfuscation) scheme is proposed to enhance the wireless communications security. The symbol-level informa-

tion, which is call "symbols key", combines the data symbols encrypting function and channel interfering function at one step. The MIO scheme can provide the information-theoretic secrecy against the passive eavesdropping attack and better computational secrecy against fake packets injection attack. Moreover, the information-theoretic secrecy would not be compromised by the location of the eavesdroppers compared with the other solutions. I conduct the hardware experiment to verify the dynamic key localization mechanism, which can defend against the eavesdroppers from retrieving the correct information of symbols key. Also, the extensive hardware experiments and simulations in various digital modulations verify that the MIO is feasible in the real wireless networks to guard the communications security.

6.2 Future Research

In this section, I first summarize some limitations and possible solutions in the symbol-level information extraction in Section 6.2.1, then, some suggestions for future research are provided in Section 6.2.2.

6.2.1 Limitations and Possible Solutions in the Symbol-Level Information Extraction

Although the symbol-level information extraction can convey the information under the low SNR/SINR environment, the delivered information is very limited. It is hard for the symbol-level information extraction to carry any information field at the PHY layer, e.g. data information, digital content.

Of course, the information carried in the symbol-level information extraction can be

increased if the known samples sequence's length is increased, but the channel occupation time would also be proportionally increased (Section 3.3.2). The designer should carefully consider this trade-off between the known samples sequence's length and the channel occupation time. In my research work, I only use the symbol-level information extraction to carry the finite control information, e.g. local MAC address, packet transmission duration, which is effective through simulations.

6.2.2 Suggestions for Future Research

The research work that has been completed so far can be extended in the following directions:

- When we propose the RTS/S-CTS and SASD schemes, the NAV time or the packet transmission time is divided into N catalogues using Eq. (3.1)

$$T_{cata}^{i} = \frac{T_{max}}{N} \cdot i,$$

and

$$i = \lceil \frac{N \cdot T_D}{T_{max}} \rceil$$

where T_D is the actual NAV time or the packet transmission time. From the above equations, we can have $T_{cata}^{i} \geq T_D$ which means the catalogued time T_{cata}^{i} can keep the station(s) waiting/sleeping longer than the actual data transmission time, which is called "catalogue overhead" in this thesis. Although we develop the best candidate algorithm to increase the possible catalogue number N to alleviate this

catalogue overhead, this overhead still exists. In the hidden terminal problem, if some stations can correctly decode the S-CTS packet into bits and achieve the precise NAV time T_D, the "catalogue overhead" can cause the unfairness of the stations which uses the T^i_{cata} time information to keep silence. Also, in the energy efficiency problem, the "catalogue overhead" keeps the station(s) sleeping longer than actual data transmission time which causes the waste of the channel bandwidth. How to modify the above catalogue equations to make the average T^i_{cata} approximate to T_D can be extended in the future work.

- In the energy inefficiency of the packet overhearing problem, we only consider the energy waste on the packet overhearing problem. However, the energy waste on the receiving collision packets [17] can also be an energy inefficiency in the wireless network. In Section 4.3.5, we discuss the aggressive model used in low SINR and low SNR scenarios for the packet overhearing problem. We may use some inequations to identify whether the packet is collided or not. Also, the Zigzag [29] and CSMA/CN [68] provide the collision point checking method. We can use this method to avoid the energy inefficiency in receiving collision packets. Moreover, we can consider the impact of various digital modulations on the packet receiving.

- In the wireless communications security problem, the MIO provides the information-theoretic secrecy and computational secrecy with the price of dB loss (Section 5.2.2.1). Can we design a new wireless communications security scheme that provides information-theoretic secrecy and computational secrecy without sacrificing any dB loss? As the artificial noise (AN) approaches would always lose some decibel, we can turn to the signal design approaches, e.g. symbols flipping method by rotating an angle

for the baseband data symbol vector. Different from Pöpper's solution [62] which can be easily conquered by the symbol detection attempts (Section 5.3.4), we can consider that for each selected symbol, the rotating angle would be different. Those rotating angles can form the "angle key" and this key can be dynamically changed like the MIO's key update mechanism. However, how to identify those selected symbols would be a problem in the further research work.

Bibliography

[1] Y. Agarwal, R. Chandra, A. Wolman, P. Bahl, K. Chin, and R. Gupta. Wireless Wakeups Revisited: Energy Management for Voip Over Wi-Fi Smartphones. In *ACM MobiSys*, 2007.

[2] F. Ashraf. *Survival Guide for Dense Networks*. PhD thesis, University of Illinois at Urbana-Champaign, 2013.

[3] S. M. Bellovin and M. Michael. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *IEEE Computer Society Symposium on Research in Security and Privacy*, 1992.

[4] S. M. Bellovin and M. Michael. Augmented Encrypted Key Exchange: A Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise. In *ACM Conference on Computer and Communications Security*, 1993.

[5] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A Media Access Protocol for Wireless LAN's. In *ACM SIGCOMM*, 1994.

[6] S. Bhashyam and B. Aazhang. Multiuser Channel Estimation and Tracking for Long-Code CDMA Systems. *IEEE Trans. on Communications*, 2002.

[7] S. Biswas and S. Datta. Reducing Overhearing Energy in 802.11 Networks by Low-Power Interface Idling. In *IEEE International Conference on Performance, Computing, and Communications*, 2004.

[8] M. Bloch, J. Barros, , M. Rodriguesand, and S. McLaughlin. Wireless Information-Theoretic Security. *IEEE Trans. on Information Theory*, 2008.

[9] M. Bloch and J. Barros. *Physical-Layer Security: From Information Theory to Security Engineering*. Cambridge University Press, 2011.

[10] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A Case for Adapting Channel Width in Wireless Networks. In *ACM SIGCOMM*, 2008.

[11] C. C. Chen, K. Yao, K. Umeno, and E. Biglieri. Design of Spread-Spectrum Sequences using Chaotic Dynamical Systems and Ergodic Theory. *IEEE Trans. on Circuits Systems*, 2001.

[12] Cisco. Omni Antenna vs. Directional Antenna. In *Cisco Document*, 2007.

[13] A. Communications. AR5213 Preliminary Datasheet. *Cisco white paper*, 2004.

[14] I. Csiszar and J. Korner. Broadcast Channels with Confidential Messages. *IEEE Trans. on Information Theory*, 1978.

[15] W. R. Dieter, S. Datta, and W. K. Kai. Power Reduction by Varying Sampling Rate. In *ACM/IEEE ISLPED*, 2005.

[16] A. El-Hoiydi and J.-D. Decotignie. WiseMAC: An Ultra Low Power MAC Protocol for the Downlink of Infrastructure Wireless Sensor Networks. In *IEEE ISCC*, 2004.

[17] C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, and V. Peiris. WiseNET: An Ultralow-Power Wireless Sensor Network Solution. *Computer*, 37(8):62–70, 2004.

[18] T. K. S. et al. *History of Wireless*. John Wiley & Sons, 2006.

[19] Ettus Inc. Universal Software Radio Peripheral. http://ettus.com.

[20] K. Flautner, S. Reinhardt, and T. Mudge. Automatic Performance Setting for Dynamic Voltage Scaling. In *ACM MobiCom*, 2001.

[21] C. L. Fullmer and J. J. Garcia-Luna-Aceves. FAMA-PJ: A Channel Access Protocol for Wireless LANs. In *ACM MobiCom*, 1995.

[22] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks. In *ACM SIGCOMM*, 1995.

[23] C. L. Fullmer and J. J. Garcia-Luna-Aceves. Solutions to Hidden Terminal Problems in Wireless Networks. In *ACM SIGCOMM*, 1997.

[24] G. Van Assche. *Quantum Cryptography and Secret-Key Distillation*. Cambridge University Press, 2006.

[25] M. Gast. *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, 2002.

[26] GNU Radio. GNU Radio - The Open Source Software Radio Project. http://gnuradio.squarespace.com/.

[27] S. Goel and R. Negi. Guaranteeing Secrecy using Artificial Noise. *IEEE Trans. on Wireless Communications*, 2008.

[28] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They Can Hear Your Heartbeats: Non-Invasive Security for Implantable Medical Devices. In *ACM SIGCOMM*, 2011.

[29] S. Gollakota and D. Katabi. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *ACM SIGCOMM*, 2008.

[30] S. Gollakota and D. Katabi. Physical Layer Wireless Security Made Fast and Channel Independent. In *IEEE INFOCOM*, 2011.

[31] V. Goyal, A. ONeill, and V. Rao. Correlated-input secure hash functions. In *Theory of Cryptography*, pages 182–200. Springer, 2011.

[32] Z. J. Haas and J. Deng. Dual Busy Tone Multiple Access (DBTMA): A Multiple Access Control Scheme for Ad Hoc Networks Communications. *IEEE Trans. on Communications*, 50(6):975–985, 2002.

[33] D. Halperin, T. Anderson, and D. Wetherall. Taking the Sting out of Carrier Sense: Interference Cancellation for Wireless LANs. In *ACM MobiCom*, 2008.

[34] F. J. Harris. *Multirate Signal Processing for Communication Systems*. Prentice Hall, 2004.

[35] G. N. T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste. Enabling MAC protocol implementations on software-defined radios. In *Preparation. Draft: http://www.cs.cmu.edu/ prs/cmu-only/sdr-mac.pdf*.

[36] M. I. Husain, S. Mahant, and R. Sridhar. CD-PHY: Physical Layer Security in Wireless Networks through Constellation Diversity. In *IEEE MILCOM*, 2012.

[37] IEEE Computer Society. 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. 2007.

[38] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *ACM SIGCOMM*, 2007.

[39] K. A. Jamieson. *The SoftPHY Abstraction: From Packets to Symbols in Wireless Network Design*. PhD thesis, MIT, 2008.

[40] K. Y. Jang, S. Hao, A. Sheth, and R. Govindan. Snooze: Energy Management in 802.11n WLANs. In *ACM SIGCOMM*, 2011.

[41] L. B. Jiang and S. C. Liew. Improving Throughput and Fairness by Reducing Exposed and Hidden Nodes in 802.11 Networks. *IEEE Trans. on Mobile Computing*, 7(1):34–49, 2008.

[42] M. L. Jorgensen, B. R. Yanakiev, G. E. Kirkelund, P. Popovski, H. Yomo, and T. Larsen. Shout to Secure: Physical-Layer Wireless Security with Known Interference . In *IEEE GLOBECOM*, 2007.

[43] P. Karn. MACA-A New Channel Access Method for Packet Radio. In *the 9th ARRL Computer Networking*, 1990.

[44] S. Katti, S. Gollakota, and D. Katabi. Embracing Wireless Interference: Analog Network Coding. In *ACM SIGCOMM*, 2007.

[45] S. Katti, D. Katabi, H. Balakrishnan, and M. Mdard. Symbol-Level Network Coding for Wireless Mesh Networks. In *ACM SIGCOMM*, 2008.

[46] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Mdard, and J. Crowcroft. XORs in The Air: Practical Wireless Network Coding. In *ACM SIGCOMM*, 2008.

[47] K.-H. Kim, A. W. Min, D. Gupta, P. Mohapatra, and J. P. Singh. Improving Energy Efficiency of Wi-Fi Sensing on Smartphones. In *IEEE INFOCOM*, 2011.

[48] R. Krashinsky and H. Balakrishnan. Minimizing Energy for Wireless Web Access with Bounded Slowdown. *Wireless Networks*, 11(1-2):135–148, 2005.

[49] L. Lai and H. E. Gamal. The Relay-Eavesdropper Channel: Cooperation for Secrecy. *IEEE Trans. on Information Theory*, 2008.

[50] E. A. Lee and D. G. Messerschmitt. *Digital Communication.* Kluwer Academic, 1993.

[51] D. Leon, S. Balkir, M. Hoffman, and L. C. Perez. Fully Programmable, Scalable Chaos-Based PN Sequence Generation. *Electronics Letters*, 2000.

[52] S. Leung-Yan-Cheong and M. E. Hellman. The Gaussian Wire-Tap Channel. *IEEE Trans. on Information Theory*, 1978.

[53] T. T. Li, J. Ren, Q. Ling, and W. G. Liang. Physical Layer Built-in Security Analysis and Enhancement of CDMA Systems. In *IEEE MILCOM*, 2005.

[54] J. Liu and L. Zhong. Micro Power Management of Active 802.11 Interfaces. In *ACM MobiSys*, 2008.

[55] R. Liu, Y. Liang, H. V. Poor, and P. Spasojevic. Secure nested codes for type ii wiretap channels. In *Information Theory Workshop, 2007. ITW'07. IEEE.*

[56] F. Lu, G. M. Voelker, and A. C. Snoeren. SloMo: Downclocking WiFi Communication. In *USENIX, NSDI*, 2013.

[57] J. Manweiler and R. Roy Choudhury. Avoiding the Rush Hours: WiFi Energy Management via Traffic Isolation. In *ACM MobiSys*, 2011.

[58] Microsoft Research. Microsoft Research Software Radio (Sora). http://research.microsoft.com/en-us/projects/sora/.

[59] B. G. Mobasseri. Digital Modulation Classification using Constellation Shape. *Signal Processing*, 2000.

[60] P. C. Ng, S. C. Liew, K. C. Sha, and W. T. To. Experimental Study of Hidden Node Problem in IEEE 802.11 Wireless Networks. In *ACM SIGCOMM* Poster, 2005.

[61] F. Oggier and B. Hassibi. The Secrecy Capacity of the MIMO Wiretap Channel. *IEEE Trans. on Information Theory*, 2011.

[62] C. Popper, N. O. Tippenhauer, B. Danev, and S. Capkun. Investigation of Signal and Message Manipulations on the Wireless Channel. In *ESORICS*, 2011.

[63] B. Ramkumar. Automatic Modulation Classification for Cognitive Radios using Cyclic Feature Detection. *IEEE Circuits and Systems Magazine*, 2009.

[64] T. S. Rappaort. Wireless Communications: Principles and Practice. 2002.

[65] G. K. Raymond. Data Encryption Standard (DES). *U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology.*, Jan. 1999.

[66] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 1978.

[67] E. Rozner, V. Navda, R. Ramjee, and S. Rayanchu. Napman: Network-Assisted Power Management for WiFi Devices. In *ACM MobiSys*, 2010.

[68] S. Sen, R. R. Choudhury, and S. Nelakuditi. CSMA/CN: Carrier Sense Multiple Access with Collision Notification. In *ACM MobiCom*, 2010.

[69] S. Sen, N. Santhapuri, R. Choudhury, and S. Nelakuditi. Moving Away from Collision Avoidance: Towards Collision Detection in Wireless Networks. In *ACM HotNets*, 2009.

[70] E. Shih, P. Bahl, and M. J. Sinclair. Wake On Wireless: An Event Driven Energy Saving Strategy for Battery Operated Devices. In *ACM MobiCom*, 2002.

[71] Y. Shiu, H. Wu, S. Huang, and H. Chen. Physical Layer Security in Wireless Networks: A Tutorial. *IEEE Wireless Communications*, 2011.

[72] N. Singh, D. Gunawardena, A. Proutiere, B. Radunovic, H. V. Balan, and P. Key. Efficient and Fair MAC for Wireless Networks with Self-interference Cancellation. In *IEEE WiOpt*, 2011.

[73] C. Sperandio and P. G. Flikkema. Wireless Physical-Layer Security via Transmit Precoding over Dispersive Channels: Optimum Linear Eavesdropping. In *IEEE MILCOM*, 2002.

[74] D. R. Stinson. Universal Hashing and Authentication Codes. *Designs, Codes and Cryptography*, 1994.

[75] J. Suman, S. N. Premnathand, M. Clark, S. K. Kasera, N. Patwari, and K. V. Krishnamurthy. On the Effectiveness of Secret Key Extraction from Wireless Signal Strength in Real Environments. In *ACM MobiCom*, 2009.

[76] J. Thomson, B. Baas, et al. An integrated 802.11a baseband and mac processor. In *Solid-State Circuits Conference*, 2002.

[77] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[78] A. D. Wyner. Wire-Tap Channel. *The Bell System Technical Journal*, 1975.

[79] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication. In *IEEE ICC*, 2007.

[80] K. Xu, M. Gerla, and S. Bae. Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks. In *IEEE Ad Hoc Networks*, 2003.

[81] S. Yang, J. Kurose, and B. N. Levine. Disambiguation of Residential Wired and Wireless Access in a Forensic Setting. In *IEEE INFOCOM Mini-Conference*, 2013.

[82] S. Zhang, S. C. Liew, and P. P. Lam. Hot Topic: Physical-Layer Network Coding. In *ACM MobiCom*, 2006.

[83] X. Zhang and K. G. Shin. E-MiLi: Energy-Minimizing Idle Listening in Wireless Networks. In *ACM MobiCom*, 2011.

[84] X. Zhang and K. G. Shin. Gap Sense: Lightweight Coordination of Heterogeneous Wireless Devices. In *IEEE INFOCOM*, 2013.