WI-FI ATTACK MITIGATION:

ANALYSIS, DESIGN, AND

IMPLEMENTATION

by

KE MENG

YANG XIAO, COMMITTEE CHAIR
SUSAN V. VRBSKY
JINGYUAN ZHANG
XIAOYAN HONG
SHUHUI LI

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2011

ABSTRACT

Wi-Fi network provides local area network with mobility, convenience and low cost. At the same time, it suffers several typical security problems, such as being unsafe and more vulnerable than traditional wired network. The shared nature of wireless medium makes wireless network attacks easily to be launched by adversaries. As a result, it is requisite to have the research and evaluation on Deny of Service (DoS) attacks and the strategy to mitigate DoS attacks.

In this dissertation, we review several wireless attack methods, attack detection schemes and attack mitigation strategies. Afterward, we analyze the 802.11 wireless network behaviors during attack and propose our improved attack detection and mitigation strategies. More importantly, we implement and evaluate our attack detection and mitigation schemes on our developing platform which is based on commercial Wi-Fi adapters by using the Atheros chipset network adapters. Moreover, we configure the Ubiquiti Routerstation (RS) with the Atheros wireless adapters to support the attack detection and access point control functions. By connecting the Routerstation to the access point, the ordinary 802.11 wireless networks can be endued with our attack detection and mitigation strategy.

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Dr. Yang Xiao, for his invaluable guidance and continuous support throughout my entire study at the University of Alabama. I would not have been able to complete my dissertation and Ph.D. program without his advice, patience and encouragement. I'm also grateful to W4-NET lab which is founded by Dr. Xiao. A significant gratitude must be made to my colleagues in W4-NET lab for providing a great research environment and emotional support.

I'd like to express deepest acknowledgement to my committee members, Dr. Susan Vrbsky, Dr. Jingyuan Zhang, Dr. Xiaoyan Hong and Dr. Shuhui Li for spending valuable time reviewing my dissertation draft and sharing their research experience. Their excellent suggestion and instructions help to improve my research skills and guide my future studies.

Finally, I would like to express my deepest appreciation to my parents who have provided endless support and caring. The special thanks should be given to my wife, Yan, for her continuous understanding and encouragement. My gratitude is extended to all those who gave me the possibility to complete my dissertation and program.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Wi-Fi (Wireless Fidelity), as a trademark of the Wi-Fi Alliance, has become so prevalent nowadays that most companies and many families have setup their wireless networks. The deployments of Wi-Fi wireless networks are ranging from Wireless Local Area Network (WLAN) to Wireless Distribution System (WDS) and Wireless Mesh Network (WMN) [2].

No matter for home use or office deployment, wireless infrastructure network has become popular as a centralized network (with the center of access point and several wireless stations connecting to the access point) due to ease of installation and location freedom with the gaining popularity of laptops and other mobile devices. On the other hand, wireless ad hoc network has always been set up as decentralized network for temporary communication. The WDS mode allows two or more access points to connect with each other without the wired backbone links. Furthermore, the access point may transmit wireless signal to several miles or even tens of miles once configured with large power supply and special directional antenna. Those features enable WDS to connect two or more separated local networks (especially two branch companies) together with wireless connection and cut down the expenditure of connecting local networks which are miles away from each other. In WMN network, wireless nodes are organized in a mesh topology which may contain multi-hops from one wireless node to another one and cover wider spaces to provide internet access.

The wireless network has its own characteristics which are different from traditional wired network and will be illustrated in the later chapter. Wi-Fi describes only a narrow range of technologies including wireless local area network based on the IEEE 802.11 standards which make Wi-Fi networks own the deficiency of 802.11 protocols. The scarce channel numbers and shared medium in 802.11 wireless networks especially in 802.11 b/g wireless networks, which have only 11 channels, make it easy to suffer Denial of Service (DoS) attacks by the adversary. Such attacks can be easily launched by using commercial Wi-Fi products. The 802.11 family uses the MAC layer known as CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance). Compared to the CSMA/CD (Carrier Sense Multiple Access /Collision Detection) which is used by the classic Ethernet to deal with transmissions after collisions occurred, CSMA/CA algorithm is so vulnerable that it can be influenced by other wireless devices, to say nothing of the jamming attackers [3].

In CSMA/CA, the wireless device needs to listen on the desired channel until the channel is idle (no other node is transmitting at the time) before transmitting the packets. If the channel is not clear (idle), the device waits for a randomly chosen period of time, and then checks the channel again to see whether it is clear. This period of time is called the back off factor, and is counted down by a back off counter. If the channel is clear when the back off counter is greater than zero, the node transmits the packet. If the channel is not clear and the back off counter reaches zero, the back off factor is set again, and the process is repeated.

Based on the implementation of CSMA/CA algorithm, the vulnerability of 802.11 wireless networks is exposed to the jamming attackers obviously. Let’s consider a simple situation in which the attackers do not follow the CSMA/CA algorithm by disabling the back off time when transmitting packets. In this case, the attackers will send out packets continuously and

the legal wireless devices will always get busy status when they listen on the attacked channel. The result is apparently that legal devices cannot send out any packets in the current channel due to the busy media.

Besides the above vulnerability of CSMA/CA algorithm, 802.11 wireless networks can also be influenced by the packet conflicts. As the wireless network indicates, the transmitting media of the Wi-Fi network is the air, which is open and can be utilized by any people or any devices with the same frequency. In other words, the transmitting media may be interfered by either 802.11 devices or other radio devices with the same frequency, either legal or illegal wireless devices. If the interfering devices keep emitting signals with considerable power strength, the packets sent by legal Wi-Fi devices (if they can send out) will be collided. As a result, they cannot be identified as the original packets even if the packets may reach to the receivers. Even badly when the interfering devices do not following any 802.11 protocols by using big transmitting power and wide bandwidth of frequency, the whole channels used by the legal wireless network will be influenced. When the sender transmits the packets, the legal signals will be completely destroyed and the receiver cannot acquire any valid signal or packets. As a result, the nature of 802.11 wireless networks determines that the network is still vulnerable besides the use of CSMA/CA algorithm.

It is well known that 802.11 b/g network provides 11 channels from channel 1 to channel 11 for the U.S. Federal Communications Commission (FCC) where each channel has the bandwidth of 22 MHz. However, the center frequency separation of adjacent channels is only 5 MHz. For example, the center frequency of channel 1 is 2412 MHz and the center frequency of channel 2 is 2417 MHz. Consequently, there are as more as 5 continuous adjacent channels partly overlap with each other and their signals will interfere with each other, causing that there

are only three non-overlapping (orthogonal) channels (channel 1, 6 and 11) available in the 802.11 b/g wireless networks as explained in figure 1.1 [14].

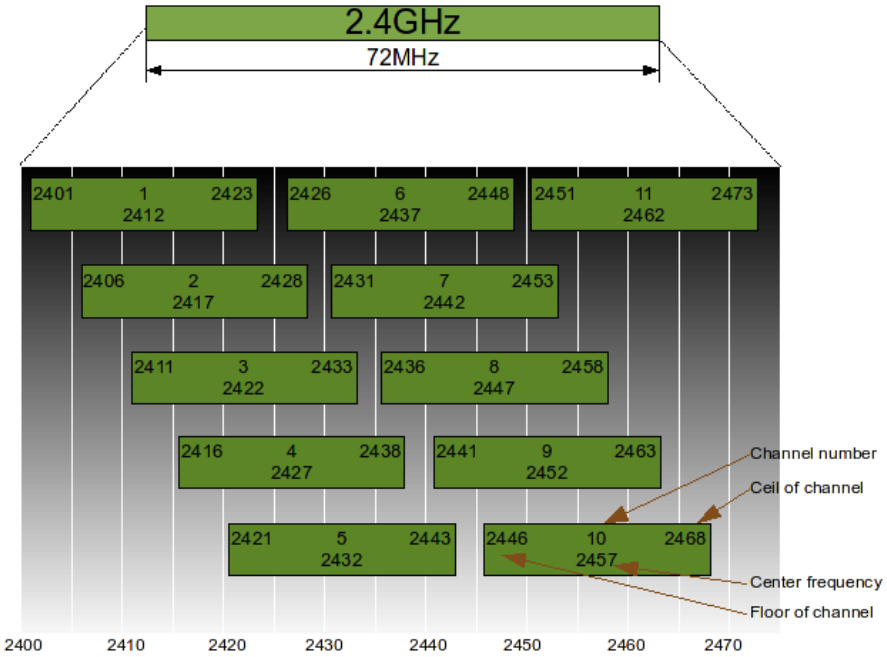

Figure 1.1 Channel overlap

Since most of the 802.11 products are commercial products, there is fewer research works have been done in 802.11 wireless networks than in wireless sensor network (WSN) which is an active research area with numerous workshops and conferences. Many researches have proved the feasibility of detecting and defeating the wireless jamming attacks in WSN [18-21]. Since communication of both 802.11 and WSN networks are based on the radio and suffered from the vulnerability of wireless transmission, similar strategies used to detect attack and mitigate attacking effect in WSN may be applied to 802.11 wireless networks. There are also several related researches on 802.11 wireless network channel interference and jamming attack. However, most of the researches are focused on the analysis and strategy. Whether they can be implemented in commercial 802.11 wireless products is unknown. In this dissertation we will

review the researches, analyze the Wi-Fi network behaviors and propose our new DoS detection and mitigation strategies on Wi-Fi wireless network.

The rest parts of this dissertation are organized as following. Section II introduces 802.11 wireless attack behaviors and attack techniques. Section III presents 802.11 wireless network attack detection schemes. Section IV discusses the attack mitigation strategies and implements our own mitigation strategies in Atheros platform. Section V talks about our advanced 802.11 DoS attack mitigation strategy, the Routerstation-controlled mitigation strategy, and attack mitigation strategy in multi-hop ad-hoc wireless network. Section VI gives the future work and conclusion.

CHAPTER 2

WI-FI ATTACK TECHNIQUES

The 802.11 wireless local area networks (WLAN) have transformed our network life in many advanced ways. Comparing with traditional Ethernet network, the wireless network has its special characteristics, including both advantages and challenges. One of the significant advantages is providing mobility to allow Laptops, PDAs and other mobile devices to escape the bondage of wired cables. In addition, the wireless network can be easily installed and deployed comparing with the wired network. However, one of the main challenges of employing wireless networks is that it is easier to attack the wireless networks than the conventional wired networks. In this chapter, we will review, evaluate and implement 802.11 wireless attack techniques which are the prerequisite of the remaining chapters.

2.1 Introduction

Jamming is defined as interfering with communications or surveillance. We differentiate the jamming attack influences to the wireless networks into two categories. In the first category, the vulnerability of CSMA/CA protocol is utilized to take up the available channel all the time and block legal wireless devices in the channel from transmitting any data as shown in figure 2.1. The second category is related to the hidden node problem where the sender cannot see the attacker. As a result, the sender is allowed to transmit packets which are disturbed in the receiver side as shown in figure 2.2. Since the attacker may not follow the legitimate rule, the CTS/RTS packets cannot overcome the hidden node problem from the attacked scenario.

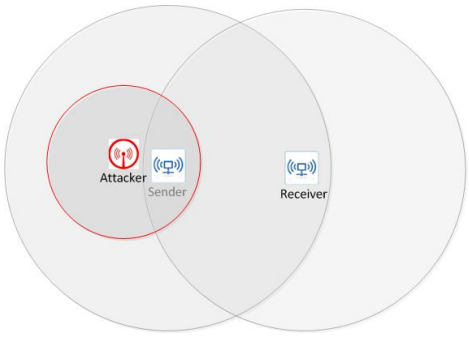

Figure 2.1 Attacking at the sender side

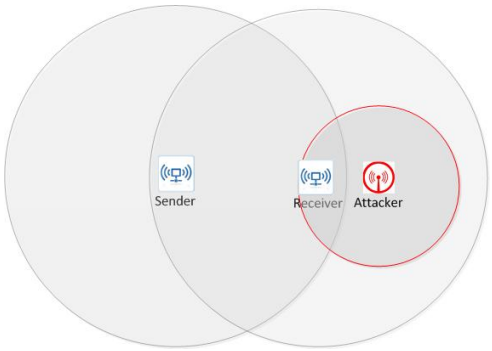

Figure 2.2 Attacking the receiver side

The working principle of the simple jammer is as simple as that the jammer scans all the 802.11 channels to search for the legitimate communication, then switches to the channel and starts jamming. Theoretically in the wireless domain, a jammer with unlimited power and resources can always successfully jam any nearby wireless transmission by flooding the frames or signals and taking up the entire spectrum that could be used by the legitimate client, making the resistance to the jamming impossible [22].

Most of the times however, the jammer may be restricted by hardware equipment, such as the limited battery capacity and transmit power, the number of jammers, the size of jamming devices and so on. Generally, the devices of jammers are similar to the legitimate wireless

devices, especially the size, in order to hide themselves among the legal users. For example, in many situations, the equipment used in the jamming would be visible to the legitimate participants and they should not be obviously special equipment. To remain inconspicuous, the attackers need to use conventional 802.11 wireless devices, such as a laptop, PDA, or other small equipment with one or two wireless interfaces for most of their attacking activities. Under the hardware constraints, the attackers have limited ability to jam the entire 802.11 spectrum at the same time, which increases the feasibility of defense to the jamming attacks.

Although significant research efforts have been made through designing appropriate network security architectures to fulfill the traditional security services, such as confidentiality, authentication, and integrity, one important class of security threats, Denial of Service (DoS)[7][8][9][10], cannot be simply addressed through these conventional security mechanisms. The DoS attack is also known as the Jamming attack, targeting at the shared and open wireless medium. It has two influences to the network. The first is taking up the available medium and blocking legal 802.11 users from sending out any packets. The second is allowing legal users to transmit packets, which are disturbed by the attacker and become invalid packets whey reaching to the receiver. In reality, DoS is an easily implemented attack which can be launched by an inexperienced attacker, howbeit may cause significant performance degradation of system, or even network partition and failure.

The Wi-fi wireless networks are easily to be attacked than the traditional wired network. First of all, the open air transmission media is easily to be disturbed by other radio signals. Any radio devices, which have the capability of sending signals at the frequency of 2.4GHz, may have the similar effect as an attack. We name it as physical device attack. For example, when a microwave oven, working at the frequency of 2.4GHz, is turned on, it will degrade the

performance of the 802.11 devices in its neighboring areas. The cordless phone or Bluetooth may also work at the frequency of 2.4GHz, and can be used by the attacker to conduct a DoS attack to impact the communication of normal 802.11 users. The attacker may also choose a power amplifier to completely block the wireless media in relevant frequency.

Second, the IEEE 802.11 wireless standard [11][12] has the PHY and MAC layer security weakness, and unavoidably become the target of attackers. Due to the broadcast nature of wireless communication, if more than one data frame meets in the air, they may superimpose together or disturb each other, leading to the bad frames with wrong CRC number and frame re-transmissions. In order to avoid frame collisions, IEEE 802.11 standard applies Clear Channel assessment (CCA) mechanism for Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The CCA deters collisions by requiring stations to monitor the channel and avoid transmitting while another one is doing so. The PHY layer is required to perform CCA using energy detection, carrier sense, or a combination of these two. It is possible for an attacker to perform a DoS attack by exploiting the CCA algorithm. The MAC layer request-to-send (RTS) and clear-to-send (CTS) mechanisms create additional weakness to the attackers, as they may not follow the protocol at all. We name this type of attacks targeting at the 802.11standard weakness as software attack, since it can be performed through commercial IEEE 802.11b/g devices with only software modifications.

The physical device attack is usually conducted by the greedy attackers, which are trying to jam any wireless transmission by flooding the entire spectrum that could be used by the client. Without much consideration of resource constraints, they maximize the attacking gain with the high risk of detection. Compared to the physical device attack which requires particular hardware device, the software attack is much easier to conduct and leaves a lot of spaces for the attacker to

play. In this attack, the attackers try to break down the communication with careful consideration of energy efficiency and the easiness of being detected. There are many such instances that the Jammer may be restricted to a configuration similar to that of a legitimate user. In many situations, such as in a secure group meeting or in an open field, the equipment used to jam would be visible to the legitimate participants. To remain inconspicuous, the jammer would need to jam with conventional (IEEE 802.11) hardware such as a single laptop with one or two wireless interfaces. The software can be modified, but under the hardware constraints, the jammer has limited capability to flood the spectrum, which makes mounting a defense feasible. In this research, we target at the software attacks and present the attack detection and mitigation mechanisms and some initial implementations.

It is noted that the DoS or jamming attack is a widely studied topic in the domain of wireless communication, and one directly related term is interference [13][14]. If the interference is caused by the malicious user, then it becomes a DoS attack. In literature, various research efforts have been conducted in investigating co-channel interference [13], adjacent channel interference [14], multi-radio coexistence [15], attacker techniques [7][8][9][10][38][39][40], and possible mitigation strategies in different types of wireless networks, such as cellular networks [16], wireless sensor networks (WSN)[1][18][19][20][21], and WLANs [22][23][24][25]. The authors in [16] analyzed the adjacent channel interference in cellular systems. The researches in [25] quantized the adjacent-channel and quadrature-channel interference by minimum shift keying method. Many different security attacks in IEEE 802.11 WLANs were identified in [7][8][9][10][38][39][40]. Four types of jamming attacks and four related defense strategies for 802.15.4-based sensor networks were present in [20]. In [26], the authors proposed some spatial-temporal techniques to reduce the impact of statistical and random

jammers. In [1][19][20] a set of jamming techniques, methods of detecting attacks, and strategies to mitigate attacks were proposed.

We also notice that there are significant researches in studying the co-channel and adjacent channel interference issue in IEEE 802.11 WLANs, but fewer detailed implementations of the DoS attacks and mitigation mechanisms compared with that in WSNs. One possible reason is that the attacks and mitigation methods can be easily implemented in WSNs as the source codes in WSNs are widely available (e.g., MAC layer codes). However, in IEEE 802.11 WLANs, the MAC codes are commonly embedded in the hardware, thus it becomes difficult to access MAC codes of commercial 802.11 adaptors. In this paper, we will present our initial implementations of attacks, attack detection, and mitigation methods in IEEE 802.11 WLANs. Compared to many previous research efforts in related areas, one major advantage of this work is to provide real implementations using IEEE 802.11 devices instead of simulations, emulations, or some other abstractions. As our implementation operates in the software (both application software and MAC layer device driver) above the standardized WiFi hardware, it is compatible with existing commercial IEEE 802.11 devices.

2.2 Related Work

2.2.1 PHY Layer Attack

The most differences between 802.11 wireless network and conventional wired Ethernet network are the PHY layer and the MAC sub-layer which is belong to the data link layer[27]. Therefore, the 802.11 wireless attacks which focus on the 802.11 MAC and PHY layers are different from other kinds of attacks targeting at influencing the network services above the data link layer. The characteristics of 802.11 attacks are the radio and channels which are the

foundation of the wireless data transmission. Figure 2.3 shows the format of 802.11 PHY layer structure.

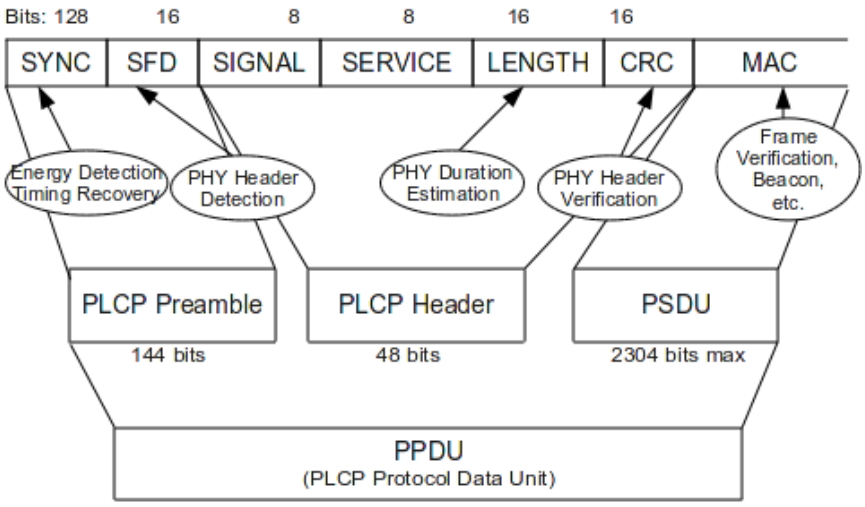

Figure 2.3 PHY encapsulation [13]

The total bandwidth of 802.11 b/g domain is 72 MHz (2401 – 2473 MHz). Despite of the complexity of 802.11 PHY layer, the efficient and easy way for the attacker to attack the channels can be achieved by using high power amplifier. The radio sent by the amplifier which works at the frequency of 2.4 GHz may take up the current 802.11b/g frequencies thoroughly and block the legal devices from sending and receiving any intact packets. Also, the attackers may utilize the format of 802.11 PHY headers to implement attacks according to the relevant weakness. There are many kinds of PHY wireless attacks shown in the following contents.

1) SYNC Deceive Interference

Each 802.11 frame has the SYNC bits in its PHY preamble as shown by the first 128 bits in the above figure. The major function of the SYNC field is to synchronize the sender and receiver devices together. The presence of the SYNC signal indicates that a frame is imminent. Wireless stations search for the SYNC pattern to prepare to receive subsequent data. The

receiver stations can measure the frequency of the incoming signal relative to its nominal values and perform any corrections needed to the received signal.

Under the use of PHY preamble with SYNC bits, the attacker may be implemented by emitting a continuous all 1s pattern, which makes the receivers mistakenly believe there are data following the preamble and directly interferes with the receiver's Timing Recovery module. Since the interferer's clock and the transmitter's clock are unsynchronized, the Timing Recovery module at the receiver cannot lock onto the transmitter's clock. As a result, the receiver only records energy detection events (SYNC), but does not detect any packet transmission. Authors in [13] also define this interference as Timing Recovery Interference.

2) Signal Range Selection Limitation [13]

The packets received by the 802.11 wireless devices may have a very large range of signal strengths from -10dBm to -70dBm (the difference is -60dBm or a factor of 10^6). In order to work over this range, the receiver normalizes these signals internally into a fixed range. However, there are two limitations of such a design in many commodity wireless network adapters, such as PRISM and Intel.

After receiving the SYNC bits, the automatic gain control unit (AGC) checks the signal to see whether the voltage level is greater than the threshold. If the signal voltage level is greater than the threshold, the signal is considered as strong and the AGC asks the RF amplifiers to subtract a 30dB gain from the incoming signals. However, the RF amplifiers will not correspondingly attenuate interference. Instead, the signal voltages are linearly subtracted by linear voltage comparators, which remove the high-order bits of the signal voltage and do not attenuate interference carried in the low-order bits of the signal voltage.

Another limitation is that the gain control and dynamic range selection are only done once per packet during the PLCP preamble processing. The interference may influence the packets either before or after the gain control is done. If interference is introduced after the gain control, the signal voltage levels will go above the signal range and become overflow. Similarly, if the interference is removed after gain control, the voltage levels will go under the range of signal selection and become under-flow.

3) Header Processing Interference [13]

Start Frame Delimiter (SFD) marks the end of the preamble in the 802.11 PHY encapsulated frame. Packet loss may also be caused by the interference which continuously transmits modulated data value used by the SFD in the PLCP preamble. The SFD field signals to the receiver that the following PLCP header is about to send and let the receiver to prepare for initialization of the receive process. The SYNC bits are also used to allow receivers to have sufficient time for the preparation. This also means that receivers are ready for the receiving of SFD before it arrives.

When the interferer continuously transmits the SFD field, the receivers will see the SFD pattern from the interferer before the SFD from the transmitter. As a result the receivers will start processing the header before the actual header from the transmitter arrives. As a result, bad packets and packets drop will happen because of the incorrect packet length and CRC bit.in the header fields.

2.2.2 MAC Layer Attack

802.11 MAC layer has flaws and are vulnerable to the attackers. There are many kinds of attackers utilize the flaw of “wait before sending” rule to jam the wireless network. Those attacks are focus on the MAC layer with the purpose of disturbing the normal behaviors of 802.11 MAC

layer. In [1] [18] [19], Xu proposes 4 generic jammer models: the constant jammer, the deceptive jammer, the random jammer and the reactive jammer.

Constant jammer continually sends out random bits to the channel without following any MAC-layer protocol. The constant jammer, which does not wait for the channel to become idle before transmitting, can make the channel busy and effectively prevent legitimate traffic sources from getting hold of channel and sending packets or corrupt the packets already sent out by legitimate wireless devices.

The deceptive jammer constantly injects regular packets to the channel without any gap between subsequent packet transmissions instead of sending out random bits. As a result, the attacked wireless nodes will be deceived into believing there are legitimate packets transmitting in the medium and will be duped to remain in the receive state. Therefore, even if a node has packets to send out, it cannot switch to the send state because a constant stream of incoming packets will always be detected.

A random jammer alternates between sleeping and jamming status, instead of continuously sending out a radio signal. Specifically, after jamming for some period of time (tj), it turns off its radio, and enters a "sleeping" mode. It will resume jamming after sleeping for some time (ts). The sleeping and sending time may be either random or fixed values.

For the reactive jammer, it is not necessary to jam the channel when nobody is under communication. Instead, the jammer stays quiet and keeps listening when the channel is idle; but starts transmitting the radio signal as soon as it senses the activity on the channel. The primary advantage besides saving energy for the reactive jammer is that it may be harder to detect.

In [28], the authors also propose four types of jamming methods. Spot jamming, which is the most popular jamming method emits all the transmitting power on a single channel to

override the legitimate signal. Spot jamming may be mitigated by jumping to another safe channel. The jammer shifts its attack frequency rapidly from one channel to another channel in the Sweep jamming. In such a jamming method, the jammer cannot affect the communication all the time and only cause part of packets loss. In barrage jamming, the jammer augments its signal emitting frequencies and may jam multiple channels at the same time. However, the output power at each channel will be reduced and the jamming effect will become smaller due to the jammer's limitation of total transmit power. The deceptive jamming is similar to the deceptive jamming as described in [1] in which the jammer resembles network traffic and sends out packets to disturb the communication.

2.3 OPNET Simulations of 802.11 Wireless Attack

OPNET is a business software that provides performance analysis for computer networks and applications. The 802.11 wireless attacks can be simulated in OPNET simulation platform. The following items will be included in our simulations: jamming attack models, attack effects on mobile wireless network and acknowledgement packets attack.

2.3.1 Wireless Jamming Attack Model in OPNET

We use OPNET to develop a simulation model for the 802.11 WLAN jamming attacks. Three jammer nodes (the pulsed jammer, single band jammer and frequency-swept jammer) are provided for OPNET to offer jamming function. However, those jammer nodes in OPNET are not designed for 802.11 wireless networks. Here we submit a mechanism of jammer that can be worked in OPNET 802.11 WLAN simulation.

2.3.1.1 Attack mechanism and configuration

The mechanism of Wi-Fi attack model we proposed in OPNET is to exploit the vulnerability of 802.11 wireless networks. According to the mechanism, the attacker will send

frames with very small interval, causing the busy of the wireless network. Then the other stations cannot send out frames until the wireless media is available. Consequently, the throughput of common communication is poorer and the network response delay is longer.

Figure 2.4 shows the topology of our attack model. Two wireless stations (Attack1 and Attack2) and one access point (AP1) were used as the attacking devices (the upper three devices). The Client, Server and AP2 are common devices, which are suffering attack during their communications. All the wireless devices are put in a small range and set to the same channel in order to enhance the simulation effect.

The whole simulation lasts 40 seconds. For the first 20 seconds, the client (AP2) and server communicate normally without any attack. Then the attackers start to send broadcast frames for the later 20 seconds. We will compare the statistics before and after the attack to prove the effects of attack. The statistic items include throughput, delay, media access delay, and so on

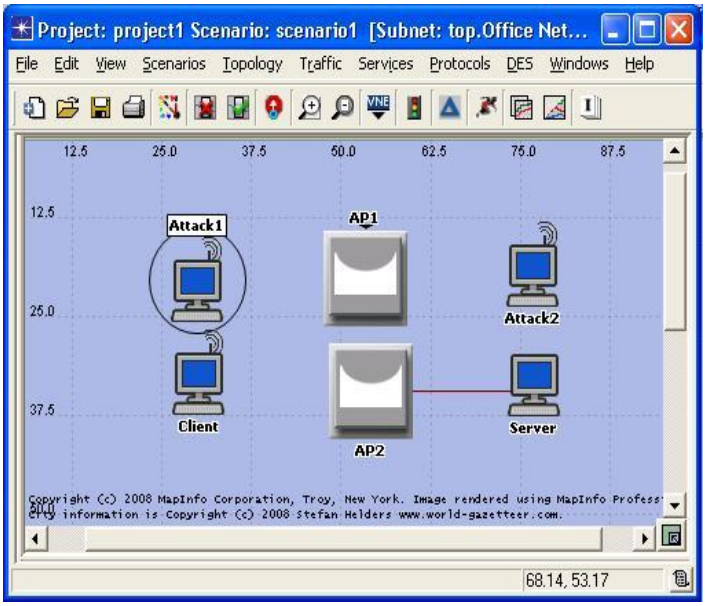

Figure 2.4 Attack model topology

Attack1 and Attack2 have the same parameter configuration. The start time is set to 20 seconds, and the interarrival time (the time interval between sending out packets) of the attackers is set as 0.00001 seconds. According to the configuration, the attacker will start after 20 seconds and send frames very frequently in the simulation. The detailed configurations of wireless nodes are shown in figure 2.5:

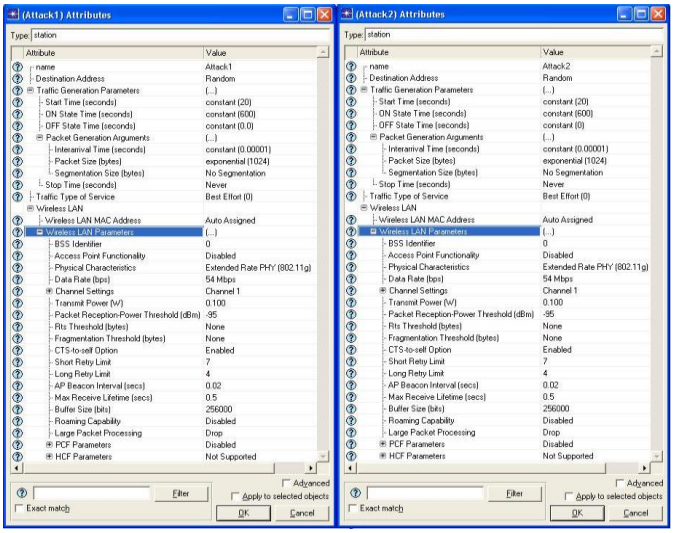

Figure 2.5 Attack model parameters

2.3.1.2 Wireless attack model simulation Results

After the configuration, we run the scenario for forty seconds and get the simulation results. Figure 2.6 shows the global statistics of the scenario. It is very clear that when we start the attack devices in the 20th second the global throughput drops dramatically. At the same time the Wireless LAN delay and retransmission attempts increase obviously.

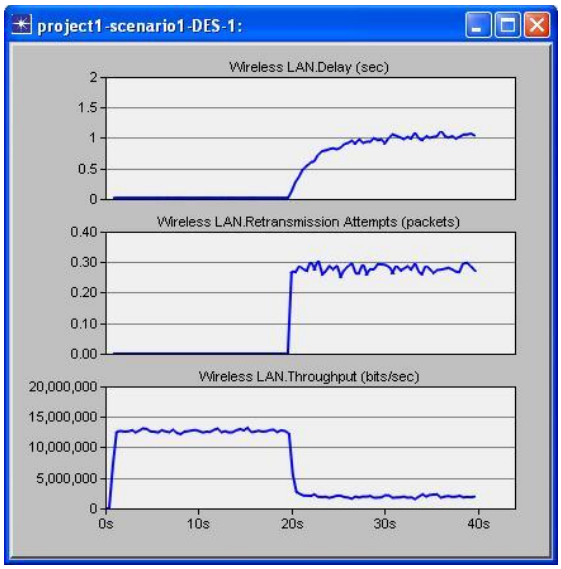

Figure 2.6 Global statistics

Figure 2.7 and 2.8 show the statistics of client, server and AP2. In the first twenty seconds, the delay and retransmission attempts are very low, and the throughput and traffic are high. In the latter twenty seconds after the attacker is started, the delay and retransmission attempts get high. At the same time period the throughput and the received traffic become low.

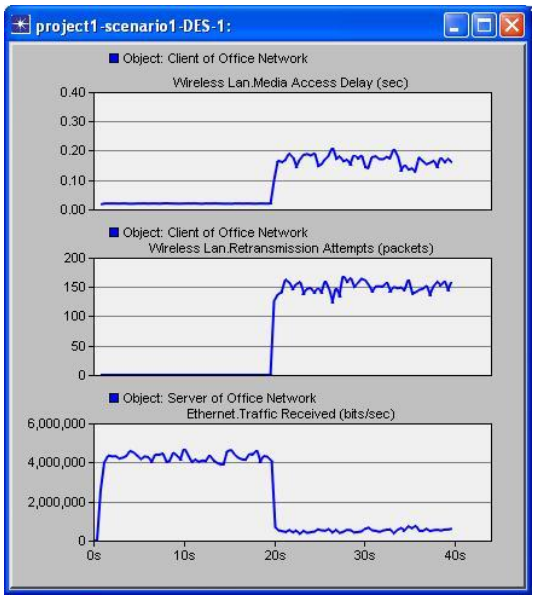

Figure 2.7 Client and server statistics

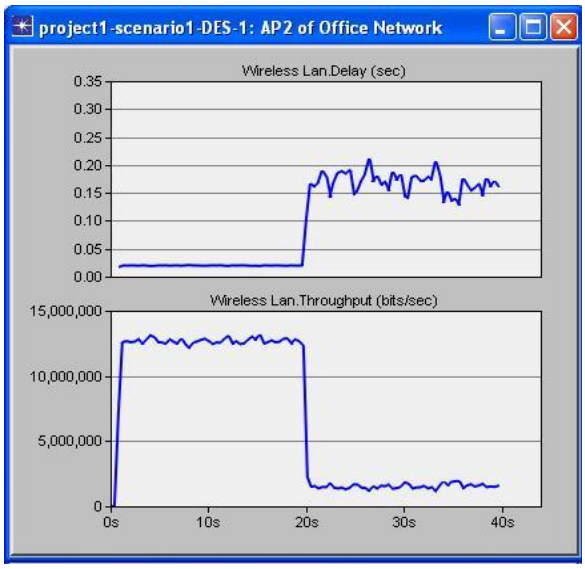

Figure 2.8 AP statistics

2.3.2 Simulation of Mobile wireless network during attack

The above OPNET simulation proves that the wireless attacker may influence the throughput, delay time and retransmission attempts in the wireless network. In this part, we will continue to simulate the relation between the attacking effect and the distances from the attackers to the wireless network.

2.3.2.1 Simulation configuration and topology

Figure 2.9 shows the topology of the simulation. We use 3 mobile nodes to simulate the common 802.11 network (the left 3 nodes in the figure). The mobile nodes include 2 wireless stations connect with one AP. Another 3 nodes is used to simulate the attackers (the 3 nodes in the right side). The attacker nodes include 2 stations connect with an access point. They send frames in very small intervals to simulate the attack. The white lines in the figure show the trajectory of the mobile nodes. At first, the common stations and the attackers are about 180 meters away from each other for about 10 seconds. Then the attackers is turned on and we let the

3 mobile nodes move to the place of the attackers together in 20 seconds, stay 5 seconds and then come back in 20 seconds. The total process spends 55 seconds.

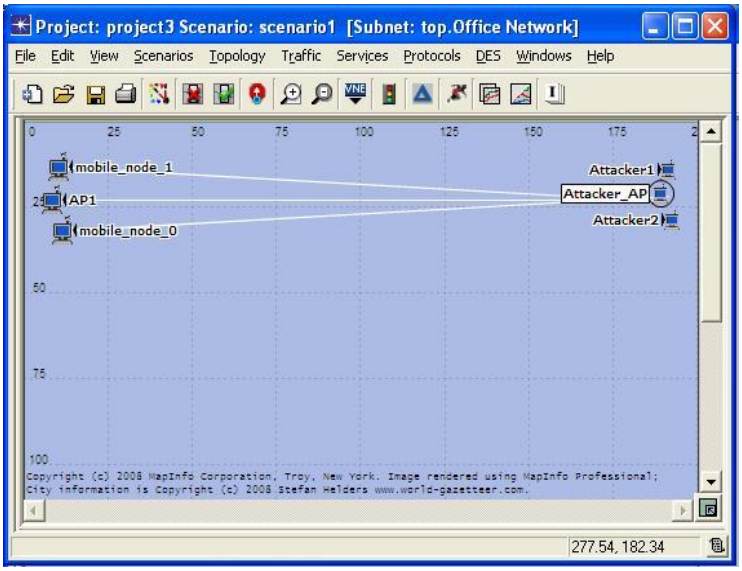

Figure 2.9 Mode 2 topology

The difference between the stations and the APs are the support of access point functionality and traffic generation. We set the BSS identifier as 1 and the interarrival time as 0.0001 for the common communication stations. While for the attacker nodes, the BSSID and interarrival time are set as 2 and 0.00001, respectively. The attackers will be turned on at the 10th seconds in the scenario.

2.3.2.2 Mobile wireless network attack simulation result

From the topology and the configuration file we know that the attackers are turned on at the 10th seconds and the place of mobile stations is the nearest to the attackers at the 30th seconds. Then the mobile stations move back after staying 5 seconds. Simulation result of delays and throughput on both station and access point will be shown in this part.

Figure 2.10 and 2.11 tell us that the delay increases and the throughput drops when the attackers are turned on although the attackers are about 180 meters far away from the stations. Still, the transmission delay increases and the throughput drops again when the mobile stations near the access point. At last, the throughput increase and delay drop a little when the mobile stations move away.

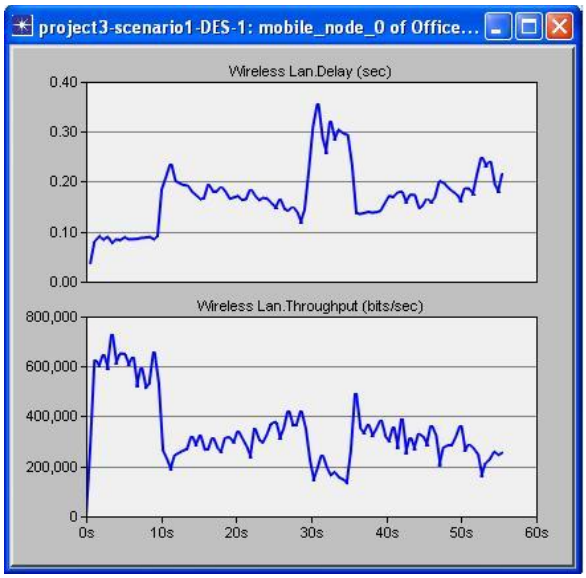

Figure 2.10 AP statistics

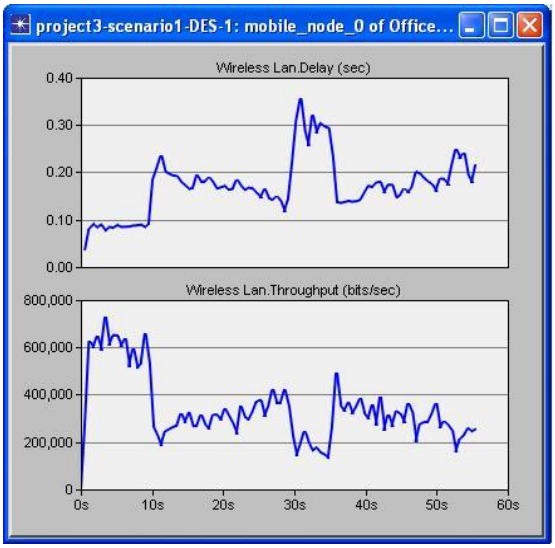

Figure 2.11 Node statistics

2.3.3 Simulation of Acknowledgment attack

The 802.11 protocols incorporate the use of acknowledgment (ACK) frames in order to recover from the situation of data frame collision. According to the protocol, a destination station always sends out an acknowledgment frame to the prior sending station if the destination station receives a directed data frame without errors. If the sending station doesn't receive an acknowledgment for a directed frame within a specific period, the data frame will be attempted retransmit. Retransmissions will take place several times before the sending station gives up. If the sending station is not able to successfully send the data frame and receive an acknowledgment, higher level protocols (such as TCP) can provide error recovery.

2.3.3.1 Acknowledgment attack topology

Our objective is to simulate the ACK attack in OPNET. As we know the common station only replies ACK frame to the frames whose destinations of the received frame are the station itself in an 802.11 wireless network. In the ACK attack simulation, the 802.11 MAC layer code of the attacker will be changed to reply ACK frames to all the unicast frames received. The ACK from the legitimate sender will be collided with the ACK from the attacker, making the ACK packet invalid at the receiver side. The process model “wlan_mac” of the attacker is modified to support the ACK attack.

The topology of the simulation is composed of four nodes (client1, client2, attacker and AP) which are shown in figure 2.12. The client1 and client2, which connect to the access point, will generate frames every 0.0001 seconds. Client2 is the destination of frames from client1, and vice versa. The attacker will not generate any data frames actively.

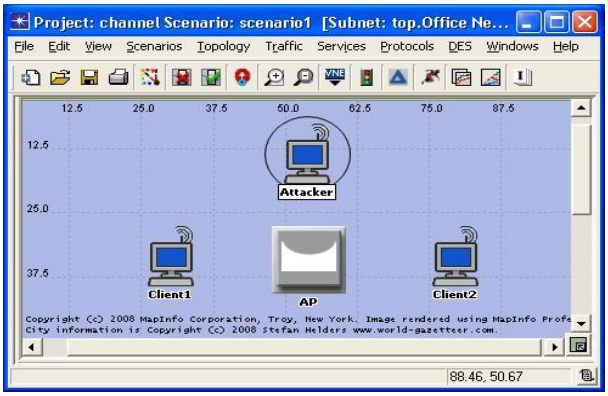

Figure 2.12 ACK attack topology

2.3.3.2 ACK attack simulation result

The total process is constituted by two steps. At first, we start the simulation for about 40 seconds with the attacker off. In this period, the attacker will not implement attacks to the clients and AP network. Figure 2.13 shows the simulation result where the throughput may reach to 600kb/second and the network delay may be as low as 0.3 seconds.

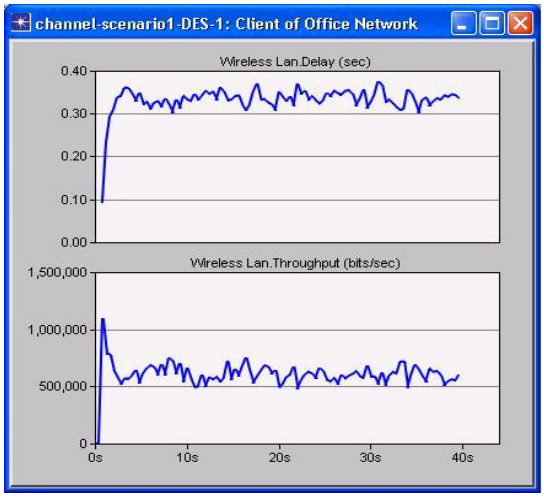

Figure 2.13 Statistics of Non-attack

Then we turn on the attacker and start the simulation for another 40 seconds. Figure 2.14 shows the test result. During this period, the attack will reply ACK frames to all the frames it

received. As a result, the ACK frames from the attacker may collide with the ACK frames from the clients and that AP. Test results show that the throughput is as low as 20kb/second and the average network delay may be more than 10 seconds. The test results demonstrate that the ACK attack may influence the throughput and delay effectively in the 802.11 wireless networks.

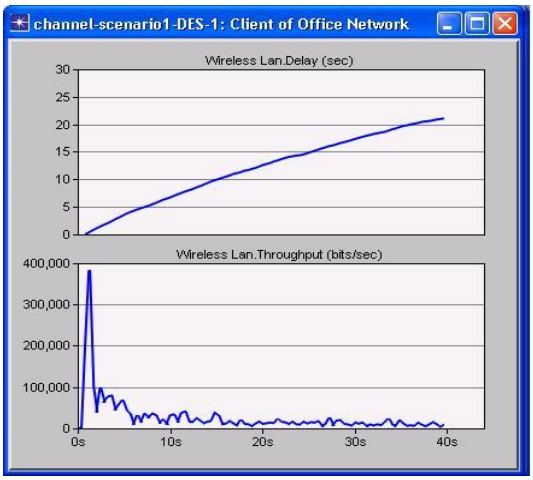

Figure 2.14 Statistics of ACK attack

2.4 Real Device Jamming Attack

2.4.1 Chariot Measurement

We have performed simple tests to see the impacts of DoS attack on network performance. The commercial 802.11 wireless devices are used to emulate the attack. Figure 2.15 shows the network setup where two access points, shown as AP1 and AP2, connect to a router which connects to a server computer. The client computer can connect to one of the two access points to access the server. We first configure the Client to connect to AP1 which is set to channel 1. AP2 is at the channel 11. There is a notebook emulating the attacker by running the Harris tool (an application used to broadcast packets for both wireless and wired network) to send broadcast packets at the selected channel (channel 1). As channel 1 is disturbed, the communication of client in this channel will be influenced, e.g. the carrier sensing time might

increase, the network throughput might reduce, the packet retransmission might increase, and the packet response time might increase.

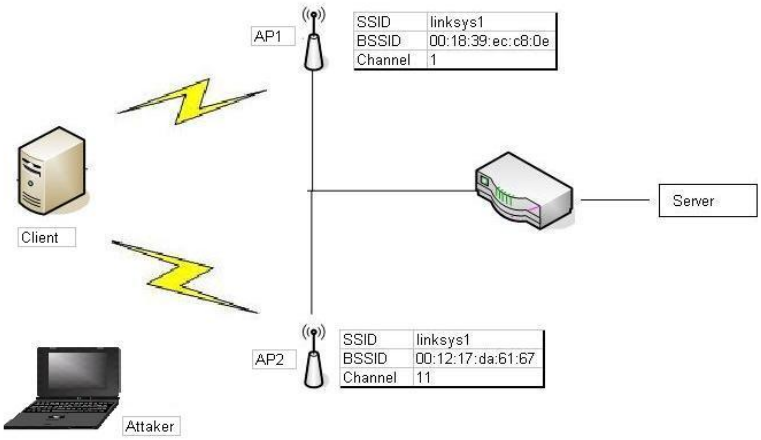

Figure 2.15 Device setup for chariot measurement

The communication throughput and the packet response time are measured during an emulated jamming attack. The software of IxChariot [24] is used to measure the throughput test in the wireless network between the Client and Server. At first, we turn off the Harris and run IxChariot between the Client and Server. After 12 seconds, the attacker is turned on and kept on for 66 seconds, and then turned off again. The test results are shown in the three figures of 2.16, 2.17 and 2.18. The attack time period is from the 12 seconds to the 78 seconds.

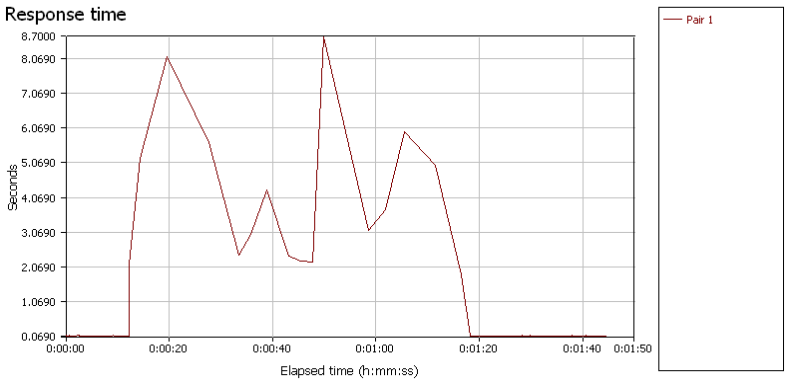

Figure 2.16 Chariot response time

Through the response time figure (2.16) we can get the result that when the attack station is turned off, the response times between station1 and station2 are very small (about 0.07 second). While during the attack period, response times are between 2 and 8.7 seconds.

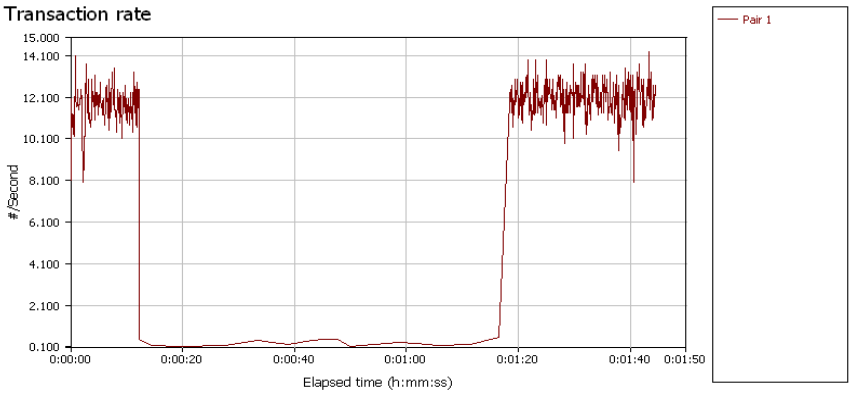

Figure 2.17 Chariot transaction rate

Figure 2.17 shows the transaction rate which is high to 12.100. While the attack station is turned on, the transaction rate will decrease to 0.1 immediately.

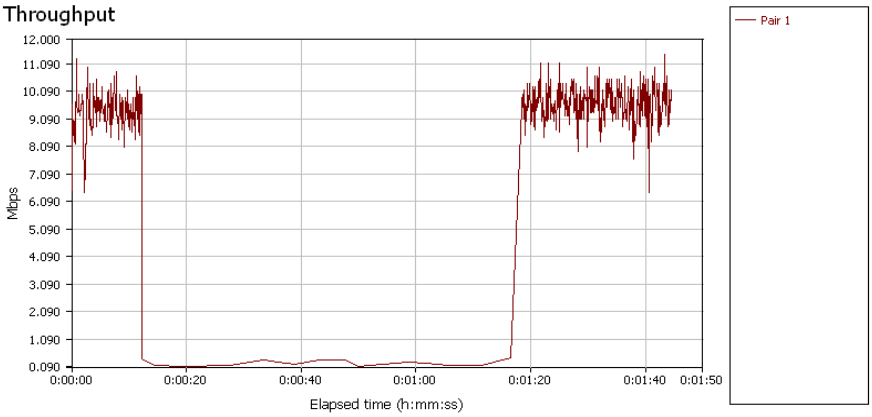

Figure 2.18 Chariot throughput

Figure 2.18 shows the throughput during the whole test process. In a common communication, the throughput between two stations may reach to 10 Mbps. While in the attacking period, the throughput can only reach to 0.1 Mbps.

2.5 Attacker Implementation

The real Wi-Fi environment will be used for our research. In this part, we talk about how to setup our wireless implementation platform from choosing wireless adapters, platform topology, configuration, and simple attacker setup.

The most common jammers are performed by utilizing the normal 802.11 wireless devices. The jammers may be simply implemented by continuously transmitting large amounts of packets with small time interval. However, there are some limitations of such kind of jammers. As we know, the 802.11 devices use the CSMA/CA mechanisms to sense the wireless media and then transmit packets if the media is idle. The jammers which are implemented by the normal 802.11 devices still have such mechanisms. They also need to sense the media and transmit the jamming packets only when the media is idle.

There are two kinds of carrier sense mechanisms, the physic carrier sense mechanism and the virtual carrier sense mechanism. The virtual carrier sense mechanism is used to reduce the probability of two stations colliding because they cannot hear each other. The virtual carrier sense is implemented by the RTS/CTS (Request to Send/Clear to Send) packets. It is easy to disable the RTS/CTS packets since it is an option in 802.11 wireless networks and can be controlled through the interface provided for the application.

According to the layers in which the 802.11 MAC function is implemented, the wireless adapters can be classified into three categories: FullMAC, SoftMAC and HalfMAC. Based on the definition, in the FullMAC wireless adapters all the 802.11 functions are put into the

hardware or firmware. While in the SoftMAC wireless adapters, the 802.11 MAC functions are implemented in the wireless driver or the operating system. The HalfMAC is situated between the FullMAC and SoftMAC with part of the 802.11 functions implemented in the hardware and other functions by the software.

There are many jammer devices available. One of the Wifi/Bluetooth jammer called WLJ100 has been developed by phantom technologies LTD [36]. The jammer has been designed to cut-off wireless LAN networks in a radius of 20 meters. The device transmit white noise signal in Wifi frequencies, 2400 – 2483 MHz, and cut-off all wireless network in such frequencies. This part will talk about our attacker platform setup and implementation.

2.5.1 Choose Wireless Adapters

The wireless adapters with Atheros chipset are selected to implement our attacker. TL-WN651G and TL-WN510G are two types of TP-LINK wireless adapters where TL-WN651G (with cardbus interface) may be used in laptop computers and TL-WN510G (with PCI interface) may be used in desktop computers.

There are three kinds of drivers for the Atheros wireless adapters: Madwifi, ath5k and ath9k. MadWifi is one of the most advanced WLAN drivers available for Atheros wireless adapters in Linux today. It is stable and has an established userbase. The driver itself is open source but depends on the proprietary Hardware Abstraction Layer (HAL) that is available in binary form only. Ath5k is a relatively new and emerging driver and does not depend on the HAL. It is intended to replace MadWifi in the long run and exceed it feature-wise. Ath9k is the youngest of the three drivers. We will perform out implementation based on the Madwifi wireless drivers.

2.5.2 Jammer Implementation

2.5.2.1 Packets Flooding Method

In the previous method, we used Harris as the DoS attack tool to flood packets under Windows operating system. We can also write our own program to flood UDP packets by creating sockets and transmitting large amounts of UDP packets with multiple threads.

There are some limitations of this application layer packets flooding method. As we know, when the network adapter wants to send the packets, it will check the connection before the transmission. That's to say, the network adapters need to create connection by ad-hoc or infrastructure mode before flooding the packets. The behaviors of 802.11 MAC layer avoid implementing the jamming attack by using only one wireless adapter through this method. As a result, the MAC layer needs to be modified to implement an effective jammer.

2.5.2.2 Disable Back-off

In the normal 802.11 device, it is necessary that all devices follow the back off protocol – waits for DIFS before transmission when no one else is transmitting, back off a random amount when other transmissions are detected. Against the rule by changing the back off is not a good idea for the total wireless network because doing so is unfair to all the other participants in the network. However, as the attackers, disable back off time and sending out packets continuously is a very effective method to augment the attack impact.

The Madwifi driver has an attribute called “continuous transmission” which may disable the back off and allow continuous transmission of scrambled packets. The Madwifi driver also has a function called “txcont_configure_radio” which can setup the Atheros adapters to enable the “continuous transmission” attribute. The command “iwpriv” may be used to turn on the “continuous transmission” by calling the “txcont_on” function through ioctl.

We use Archlinux (with kernel 2.6) as our operating system which has the default wireless network driver “ath5k” for the Atheros adapters. In order to disable the back off, we should remove the default wireless driver and install the required “Madwifi” driver. Follow the commands to remove “ath5k” and load the Madwifi drivers.

```
# rmmod ath5k
# pacman -S madwifi
# modprobe ath_pci
# ifconfig ath0 up
```

The Atheros adapter will be identified as the “ath0” device (for example “ath0”) after installing the Madwifi driver. Then the channel number should be set for the attacker. One adapter can only implement attack in one channel at one time. The following commands set the attacker’s channel at channel 1. After that the “iwpriv” command is used to enable the “continuous transmission”. Then use the command to check whether the “continuous transmission” has been turned on:

```
# iwconfig ath0 channel 1.
# iwpriv ath0 txcont 1
# iwpriv ath0 get_txcont
```

Actually, the command of “iwpriv ath0 txcont 1” not only disables the backoff time, it also floods frames continuously. As a result, we can simply use one wireless adapter together with the above commands to achieve the jamming attack.

2.6 Jamming Attack Evaluation

2.6.1 Evaluation Setup

In order to verify the effectiveness of our wireless attacker, we setup out platform to test the jamming attack influence. Four computers and one access point are used to setup the test environment which is shown in figure 2.19. One computer acts as the normal wireless device

which communicates with the server through the access point. Two wireless devices are used as the attackers by running the Packets flooding application. All the three wireless devices are using the Atheros wireless adapters and Madwifi drivers. They have the same channel with the access point. The User1 and the Server will use the “iperf” tool to measure the bandwidth between them. The bandwidth is the maximum data rate the devices can transmit and can represent the level of suffering attacks. All the computers are running the operating system of Archlinux with kernel of 2.6.30.

We can implement the attack by running the iperf application to transmit large amounts of packets instead of using the UDP flooding application. We also can use one wireless device to achieve the attack through the “iwpriv ath0 txcont 1” command. Both of the two kinds of attacks are tested in the test bed. All the wireless devices including the normal devices and the attackers are put near each other within the range of 5 meters.

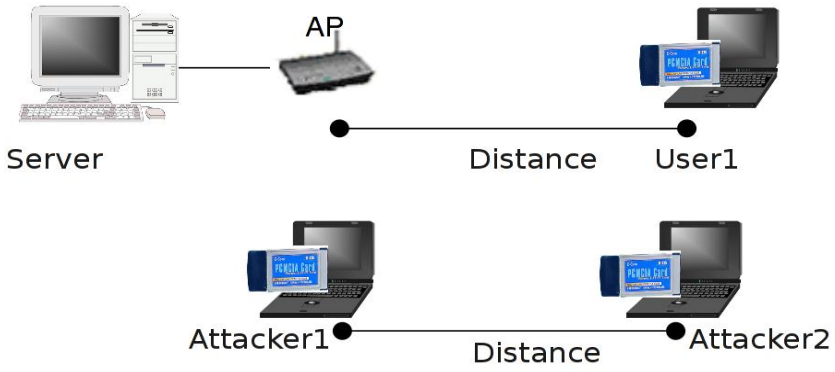

Figure 2.19 Jamming attack setup

2.6.1.1 User and server setup

The server computer is connected to the access point with Ethernet cable and the user1 is connected to the access point through infrastructure mode at channel 1. The user1 and server use iperf to perform network throughput between each other. We only specify the server or client

mode for the iperf command and leave the other parameters as the default setting. Configure the server with IP address of 192.168.0.2 and the client with IP address of 192.168.0.1.

For the ad-hoc mode, the user and the server are connected with each other directly by ad-hoc mode at channel 1. The tool of iperf is also used between the user and server to test the throughput.

Use the following command on the server side to run in the server mode:

```
# iperf -s
```

Use the command on the user side to run in the client mode which will connect to the server (192.168.0.2).

```
# iperf -c 192.168.0.2
```

2.6.1.2 Attacker setup

The attacker1 and attacker2 connect with each other with ad-hoc mode at channel 1. The attackers and other wireless devices are in different networks. We use the following command to setup the ad-hoc network of the attacker1.

```
# wlanconfig ath0 destroy
# wlanconfig ath0 create wlandev wifi0 wlanmode adhoc
# rmmod ath5k
# rmmod ath_pci
# modprobe ath_pci
# ifconfig ath0 up
# iwconfig ath0 essid attacker
# iwconfig ath0 channel 1
# ifconfig ath0 192.168.0.3
# perl flood.pl 192.168.0.4
```

For the attacker2, we only change the setup of IP address to 192.168.0.4 and use “perl flood.pl 192.168.0.3” to flood packets to the peer attacker.

We can either use iperf to make large amounts of packets traffic or use the command “iwpriv ath0 txcont 1” to achieve the attack effect. When using iperf to increase the network traffic to cause to the attack effect, the wireless network cannot be blocked completely. The legal devices and the attackers have equally opportunity to transmit and receive packets. The attacking result shows that the throughput of legal devices is obviously decreased. While the attacker which uses the “iwpriv ath0 txcont 1” may block the channel completely, and the legal devices cannot send out any packets at all.

2.6.2 Test result Ad-hoc

We use one Linksys Compact Wireless-G USB adapter and one Atheros PCI wireless adapter with the model of TL-WN510G as the wireless clients. The two adapters are configured of ad-hoc mode and communicate with each other at channel 1.

Two Atheros cardbus wireless adapters with the model of TL-WN651G are used as the attackers. We can also use one wireless adapter together with the “iwpriv” command to implement the attack. First of all we configure the attackers at the same channel as the two clients (channel 1) and measure the co-channel interference of the attack. Then we move the attackers to other channels besides the user and server channel and measure the adjacent channel interference.

2.6.2.1 No attack

First of all, we test the performance of the wireless network without the start-up of the attackers. The performance measurement tool of “iperf” is used to measure the bandwidth of the network. One client is used as the server and another client as the client. The client will try its

best to transmit TCP packets with the server. The total test lasts 30 seconds. Figure 2.20 shows that the bandwidth between the server and the client is 10.3Mbits/sec.

```
kmeng@w4net2:~$ iperf -c 192.168.0.3
------------------------------------------------------------
Client connecting to 192.168.0.3, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.4 port 37620 connected with 192.168.0.3 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.6 sec  13.1 MBytes  10.3 Mbits/sec
```

Figure 2.20 iperf measurement without attack

We also use jperf to measure the network bandwidth which may give us a GUI curve as in figure 2.21.

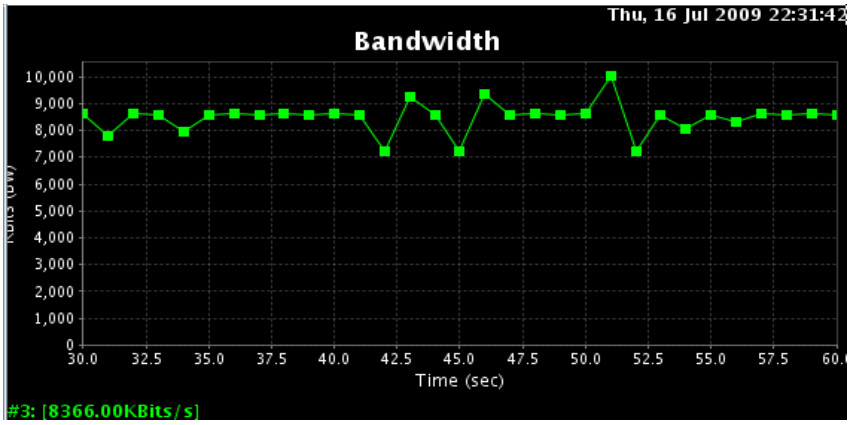

Figure 2.21 Jperf measurement without attack

2.6.2.2 Normal packets flood

After measurement of non-attack network, the influence on the wireless devices under normal network traffic scenario will be tested. The tools of iperf/jperf are used to measure the network performance. Two pairs of wireless devices were used in the test. The first pair of wireless devices was acted as the normal wireless devices and started at first. Then the second pair of wireless devices, which performs the packets flood attackers, was started in the 30th

seconds. The whole test lasted 60 seconds. iperf/jperf were used to measure the network performance in the 60 seconds.

Figure 2.22 shows the iperf network performance

```
kmeng@w4net2:~$ iperf -c 192.168.0.3
------------------------------------------------------------
Client connecting to 192.168.0.3, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.4 port 37621 connected with 192.168.0.3 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.6 sec  9.89 MBytes  7.81 Mbits/sec
```

Figure 2.22 iperf measurement with normal packet flood

Figure 2.23 shows the throughput measurement result of Jperf. Since the Jperf only can show the figure of 30 seconds, we select the middle 30 seconds which is from 18th to 48th seconds. We can see an obvious throughput drop at the 30th second.

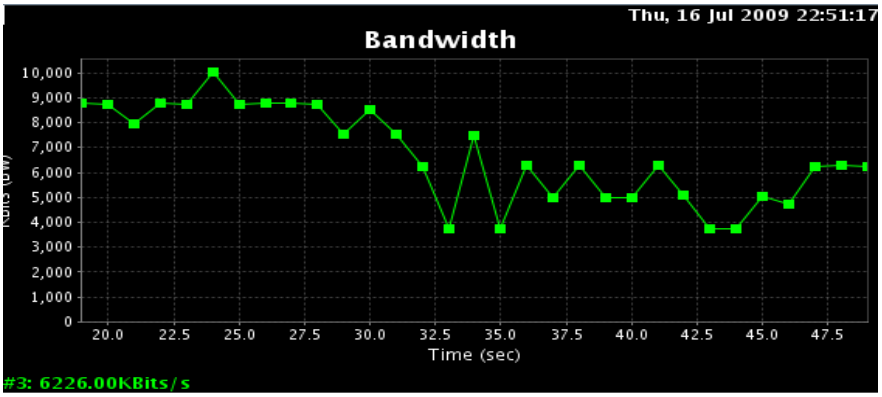

Figure 2.23 Jperf measurement with normal packet flood

2.6.2.3 Disable backoff

In this scenario, we disable the backoff time of the attackers by the command depicted earlier and use Iperf and Jperf to measure the performance. First of all, let's see the connection information shown by the "iwconfig ath0" command before starting the attack.

```
ath0      IEEE 802.11g  ESSID:"atta"  Nickname:""
          Mode:Ad-Hoc  Frequency:2.437 GHz  Cell: 06:21:27:EE:C8:CD
          Bit Rate:0 kb/s   Tx-Power:16 dBm   Sensitivity=1/1
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=60/70  Signal level=-36 dBm  Noise level=-96 dBm
          Rx invalid nwid:3717  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Figure 2.24 Wireless device information

Then we started the attacker and use Iperf tool to measure the network performance between the wireless devices. However, Iperf cannot connect to the server. The connection between the two nodes is broken by the attacker and the device cannot find the route the host. We also use "Ping" command to check the connection and it still failed after the attack as shown in the figure 2.25.

```
kmeng@w4net2:~$ iperf -c 192.168.0.3
connect failed: No route to host
```

Figure 2.25 Iperf measurement with disabled backoff attacker

Figure 2.26 shows the Jperf test results. The attacker was started at the 10th second and ended at the 25th second. The throughput is dropped to zero during this attacking period and resumes once the attacker is stopped.

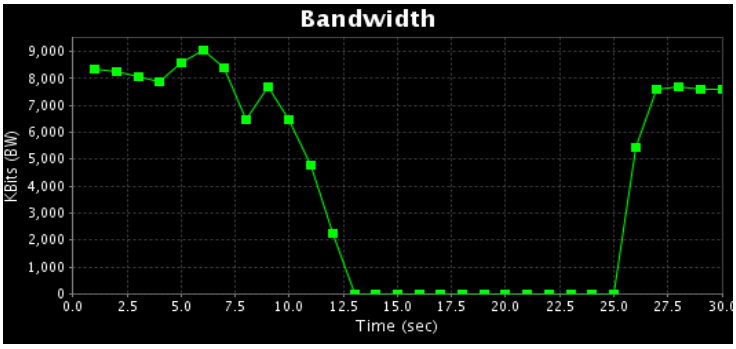

Figure 2.26 . Gperf measurement with disabled backoff attacker

2.7 Summary

In this chapter we first reviewed some DoS attack schemes. Then we used OPNET to do 802.11 attack simulations which included our wireless attack module and several attack scenarios. We also setup our 802.11 wireless attack platform and did the wireless attack evaluation. The experiments showed that the 802.11 wireless attacks effectively influenced the throughput of legal wireless network. The influence was not only limited to the current channel but extended to adjacent channels. The attacks also have distance limitations where the wireless stations closed to the attacker will be influenced the most.

CHAPTER 3

IEEE 802.11 DOS ATTACK DETECTION

3.1 Introduction

Jamming attack detection is the prerequisite of jamming attack mitigation method. It is so important that the operation of jamming attack mitigation cannot be performed unless the jamming attack has been detected. It is a big challenge to detect the jammers because there are different kinds of jammers and even the same jammer can switch between different jamming models or jamming powers. There are also lots of network conditions, such as low throughput, normal communication, congestion, and so on, which have similarity with the jammed network, making it difficult to distinguish the jamming situations from legitimate ones. The jamming attacks should also be differentiated from the special circumstances, such as system power off, operating system hung up, antenna problems, communicating distance and so on, which may also lead to the similar results as the jamming attacking.

The behaviors in wireless networks under jamming attacks can be used to identify the attackers or jammers. However, the methods used to detect the attacks may not always be effective for all kinds of jamming attacks [1] [4] since the characteristics of some attackers or jammers may not particularly owned by other jammed network. Here we will first discuss some basic jamming attack detection methods which may be successful for constant jammer and deceptive jammer. However, for the reactive jammer and random jammer, the basic detection methods will be useless. We also review some advanced jamming detection scheme, which

combine two or more basic methods together, may distinguish all kinds of wireless jammers. Then we will propose and implement our jamming detection models.

3.2 Related Work

3.2.1 Basic Jamming Detection Methods

Wireless networks have many behaviors which can be used to identify the jamming attacks. For example, when packets are transmitting in the air, the signal level of the current channel will become high. When there are more packets transmitting in a period, the average signal level will be higher. However, the signal strength cannot be always high, since there are DIFS, SIFS and other waiting times between packets transmission. When the jammers use their best efforts to attack the network, the signal level in the current channel will keep high all the time. As a result, signal strength may be considered as one of the wireless network measurement for detecting the simple jamming attack which constantly transmits packets without interval.

The authors in [1] use the higher order crossings (HOC) spectral discrimination mechanism to try separating the jamming attack signal strength from other normal network status. As we talked in section II about the behaviors of the jamming attack, the signal strength level of CBA and Max Traffic are fluctuated up and down between -90 and -70 dBm while the constant jammer and deceptive jammer keep the signal strength high all the time. As a result, it is easy to distinguish the constant and deceptive jammer from the normal traffic scenarios by the HOC spectral discrimination mechanism. However, we cannot distinguish the normal traffic scenarios from the reactive and random jammers because they have similar characteristic with the normal network traffic. Carrier sensing time and packets delivery ratio are other methods which are frequently used to determine whether the wireless network is under attack or not. However, they still cannot be used to distinguish the reactive jammer and random jammer from the normal

network traffic (the method is specifically elaborated in [1]). There are also other research works about the wireless network detection. Based on received signal strength indicator (RSSI), the authors in [41] present a robust and lightweight solution for attack problem in wireless sensor networks. In [42], a real-time attack detection tools is presented to detect local and distributed attacks within its radio range by monitoring network packets for AODV-based Ad hoc wireless networks.

3.2.2 Advanced Jamming Detection Methods

In [1], the authors also proposed advanced detection strategies which are called multi-modal strategies. In one of those advanced strategies, the (Packet Delivery Ratio) PDR and signal strength are combined together in order to separate the normal scenarios from the four jammed ones. If there are huge packets transmitted in the normal network, both the PDR and signal are high. If there is little packet transmission, the normal network will have low PDR and low signal strength. There should not be both high signal strength and low PDR in a normal wireless network. Whereas the jammed wireless networks have low PDR and high signal strength which is different from the normal wireless networks.

Another advanced detection strategy is location consistency check which combines PDR and the distance between sender and receiver together. In the wireless network, the low PDR may also be caused by the long distance between the sender and receiver and the jammed networks should have low PDR value and short sender-receiver distance. Considering the special distance conditions, the location consistency check also has the ability to separate the jammed scenarios from the normal network scenarios.

3.3 System Model

3.3.1 Infrastructure Mode Attack Detection

3.3.1.1 Overview

The infrastructure wireless network is the wireless distributed network with the center device of access point. In the infrastructure mode, we use a scheme which is similar to the PDR method to implement attack detection. Each node connecting to the access point will send “ping” message to the access point and expect to get response message very soon if the current network is fine. The PDR is calculated by the amount of “ping” message and response message.

3.3.1.2 Algorithm Description

In the infrastructure network, the access point sends out beacons to the stations in the current channel. The stations may determine whether the signal in the current channel is good enough for the communication by receiving the beacons. At the same time, the stations may get the RSSI (received signal strength indication), noise level and know whether the noise is big. Stations may also analyze the current channel by the PDR (packet delivery rate).

The signal level is the signal power or intensity at a specified point and with respect to a specified reference level. Noise levels are usually viewed in opposition to signal levels. The signal level and noise level are usually measured in decibels (dB) for relative power or kilowatts for absolute power. dB is an abbreviation for "decibel". The measurement quoted in dB describes the ratio (10 log power difference, 20 log voltage differences, etc.) between the quantity of two levels, the level being measured and a reference. To describe an absolute value, the reference point must be known. There are a number of different reference points defined. A suffix is added to denote a particular reference base or specific qualities of the measurement. For example, the normally used measurement unit is dBm. dBm represents the power level compared to 1 mWatt

[6]. This is a level compared to 0.775 Volts RMS across 600 Ohm load impedance. Note that this is a measurement of power, not a measurement of voltage. We use “dBm” for most of measurement in this project.

In our algorithm, we will use the beacons, RSSI, noise level and PDR to determine whether the current channel is under attack or not. RSSI and noise level can be read from the network driver. The beacons can also be received in the network driver (MAC layer). PDR can be calculated by the sequence number of received packets including both management packets and data packets. The PDR, together with the beacons, are used to determine the communication quality of current channel.

3.3.2 Ad-hoc Mode Attack Detection

3.3.2.1 Overview

In an 802.11 wireless Ad-hoc network, the stations connect with each other directly without the center controlled access points (or wireless routers). The wireless Ad-hoc network is a coordinated peer to peer network which is different from the infrastructure mode. In the ad-hoc network, the wireless devices don't have the center device and cannot use the “ping” method to perform attack detection. However, it is possible to fully control the ad-hoc nodes for attack detection with the open source 802.11 wireless network adapters.

The channel of Ad-hoc network is determined by the first station that creates the SSID and Channel. Then the following stations which have the same SSID will join the Ad-hoc network by setup the same channel. In this Ad-hoc network, we should find a new configuration method for the station to detect attacks.

3.3.2.2 Algorithm Description

In a common ad-hoc wireless network, all the nodes have the equal position during the communication. In this algorithm, every station uses the UDP packets to keep the connection and collaborate with other stations in the wireless network. We also use UDP packets to detect the attack.

We define 3 types of packets: “KEEPALIVE”, “RESPALIVE” and “CHANGECHL”. Every 5 seconds (the default value is 5) each station sends “KEEPALIVE” packets to the peer stations and waits for the reply packets of “RESPALIVE” from those stations. Each station will calculate the delay time between sending the “KEEPALIVE” packet and receiving the “RESPALIVE” packet. The station also has a counter (initialized as 0) for each peer station. The counter will be added by one when the delay is longer than the threshold value or even the “RESPALIVE” packet is lost. If the station receives “RESPALIVE” within the delay threshold, it will reset the counter to 0.

3.3.3 Jamming Attack Detection Implementation

The main purpose of this development is to detect jamming attacks in Madwifi driver and return relevant results to application layer through IOCTL. The attack detection is done by the low level Madwifi driver in the MAC layer. The MAC layer knows more information than the application layer and can communicate with the hardware quickly. After attack detection, the Madwifi driver will notify the application layer to take relevant action to handle the jamming attack. As a result, we also need to add the communication interface between the MAC layer and application layer.

3.3.3.1 Communication between application layer and MAC layer

The command of “iwpriv” will be used to communicate between the application layer and the MAC layer. “iwpriv” command is the capsulation of “ioctl” which can operate the driver parameter easily (include both setting and getting information). Twelve parameters are defined to control the jamming detection for “iwpriv” command. Application notifies MAC layer to detect attack, checks the noise status in each channel and switches channel if possible.

3.3.3.2 MAC layer detection

The first step of attack detection is to get the value of RSSI and PDR. The value should be the average value in a certain time. One link list table is added to the Madwifi driver to collect the information and calculate the average RSSI and PDR as in the following code.

```
struct msgInfo {
	int rssi;//RSSI value
	u_int16_t seqNum;//sequnce number
	struct timespec timestamp;//time of kernel when reciving the msg
	struct msgInfo *nextMsg;//next cached msg info
};

struct nodeStat {
	u_int8_t macAddr[IEEE80211_ADDR_LEN];//mac address for this node
	struct timespec lastTimestamp;//the timestamp of latest observed msg
	struct msgInfo *cachedMsgInfoHead;//the cached msgInfo list header
	struct msgInfo *cachedMsgInfoTail;//the cached msgInfo list tail
	struct nodeStat *nextNode;//the link for the next node information
};
```

Madwifi driver will analyze the received packets and organize them to link lists in order to save packets information. All the wireless stations which send the packets will be organized to one link list as the vertical link list in the figure 3.1. Each station has one link list which contains the information of packets sent recently as the horizontal link lists in the figure. The information of the wireless devices link list include the MAC address of the device, the time stamp of latest

observed message and the pointers to the relative message list. The information in the message lists includes the value of RSSI, sequence number and time stamp. The Madwifi driver (MAC layer) analyzes the information in the link lists and determines whether the current channel is under attack. If current channel is under attack, the driver will tell application layer through "iwpriv" IOCTL. The value of average "rssi" is calculated by the "rssi" in each package:

```
rssi = (rssi1+rssi2+rssi3+……+rssin)/n
```

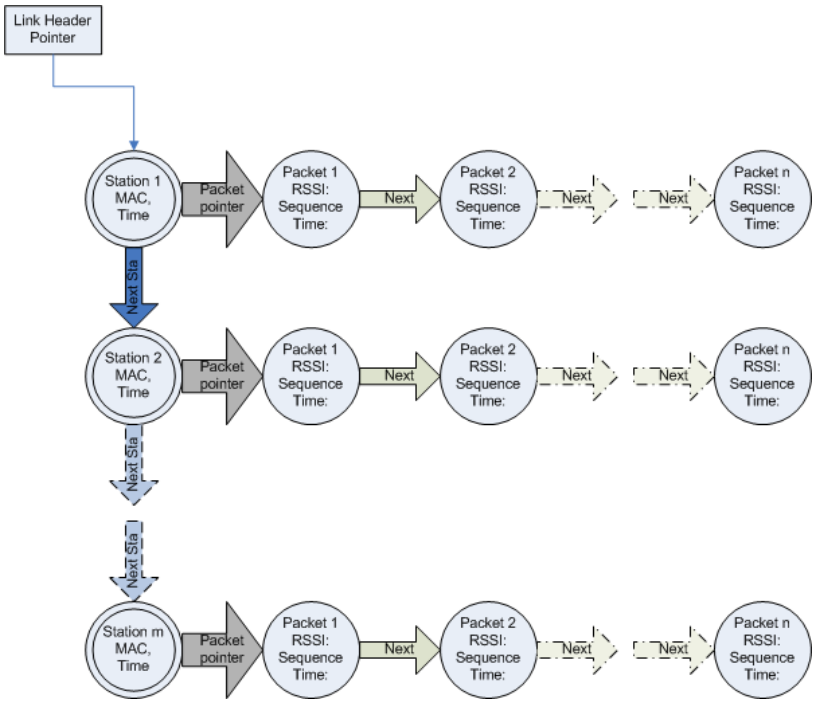

Figure 3.1 2-dimension link list structure

In order to get the PDR, we calculate the missed packets of each wireless station. The number of missed packets is got by the difference of sequence numbers in any two continuous packets. For example, if the sequence number of one packet is 2313 and the next one is 2318, then the number of missed packets between the two packets is 2318-2313-1 = 4. We can get the number of all the missed packets and calculate the PDR value:

PDR = (received packet number n)/(received packet number n + number of total missed packets)

The value of noise can be read directly from the Madwifi driver. If RSSI is larger than the rssi threshold, PDR is less than PDR threshold and noise is larger than noise threshold, the current channel will be considered to be under jamming attack. The application layer can get the noise level of each channel by calling the command of "# iwpriv ath0 get_noise" in each channel and control wireless devices switch to the best channel.

3.4 Simulations Results

3.4.1 Jamming Detection Time

Different jamming detection method determines different detection time. For example in the application layer jamming detection, the network clients transmit packets with the frequency of 5 seconds to keep connection with each other. If the detection threshold is set to 60 seconds, the network clients will detect the attacker when they cannot receive the response message from other clients for 60 seconds. In that case, the detection response time is 60 seconds. We cannot get an exact detection response time.

This jamming detection method still works when we try to reduce the value of packet transmitting frequency and detecting threshold. The wireless devices may detect attack successfully when we set both the frequency and threshold as 1 second which means that the detection response time here is in 1 second.

We also can use beacon frames (which will be explained in the following part) to detect attacks. According to my test, the beacon missing function will only be called at about 250 jiffies (about 820 milliseconds) after attack. As a result, the attack detection time may be about 820 milliseconds when using the beacon frames.

3.4.2 False Alarm Rate

We use beacon frame to detect the attacker and hereby measure the false alarm rate. Different numbers of beacon missing are used to determine whether the network is attacked and different thresholds are used to calculate false alarm rate.

When the wireless clients lost the beacons from the AP, the client will trigger the function of ath_bmiss_tasklet(). If the number of consecutive missed beacons is larger than 9, then the function of ieee80211_beacon_miss() will be called to re-associate or scan. Otherwise, the missed beacons will be ignored. The beacon interval is 100 milliseconds by default and the time period of missing 9 beacons is 900 milliseconds. In this wireless adapter, it is 276 jiffies. If the clients keep missing beacons for a long time, it will trigger the ath_bmiss_tasklet every 276 jiffies.

We use RF cable and adjust the TX power (1 - 18 dBm) on both access point and wireless client for the measurement. When the TX power is larger than 3 dBm, there is no beacon missing. In this case, once the attacker is started we can detect it immediately and the false alarm rate is 0%. However, the association between clients and access point is not stable when we use TX power of 1 or 2 dBm. Most of the times, the connection is normal and there is no beacon missing. There exist the cases that the association is disconnected and the clients cannot receive beacon from AP. In this case, the ath_bmiss_tasklet will be triggered every 276 jiffies or 900 milliseconds. So the ath_bmiss_tasklet may be triggered at most 66 (60*1000/900) times in one minute. Any of the 66 times may be caused by an attacker or not.

RF cable are used in the tests to connect the wireless device together and avoid interfere from other wireless networks. We know from the table that the false alarm rate is very small when the connection between clients and AP are good enough. In the normal wireless

environment, when there are large numbers of wireless devices and huge wireless communication data, the wireless clients may encounter the case of beacon missing. In these cases, the lost beacons are intermittent. They are not like the attack cases which will block all the packets including the beacons. As a result, we can use multiple beacon miss triggers to detect the attacker.

Table 3.1 Beacon Miss False Alarm Rate

TX power	Bmiss tasklet in	False Alarm Rate	Memo
18	0	0%	No beacon missing
10	0	0%	
5	0	0%	
3	0	0%	
2	0 - 66	0%- 100%	Sometimes beacon missing
1	0 - 66	0% - 100%	

3.4.3 Attack detection scenarios

Our algorithms are implemented in Linux operating system. The wireless network adapters with the Atheros chipset are used. We select Madwifi as the network driver for the network adapters. The MAC layer in the wireless network driver is modified to detect the attacker accurately. The following scenarios are considered in our implementation.

- The attacker is very effective and can block the current channel completely.
- The attacker cannot destroy the wireless network completely and only leads to some delay and packet lost.
- Move the device far away from each to see whether the program can distinguish it from attacks.
- Turn off power of one device to see whether the program on the other device can distinguish it from attacks.

- Start the attacker to disturb the communication.
- Place the attacker in different distance from the device to see the whether the program can detect the attack

According to our tests in the real 802.11 environment, more than 95% of the attackers can be detected. At the same time, the false alarm rate is about 25% due to the complication of the wireless network. However, even though the false alarm appears and the mitigation strategy will be applied, there are small influences to the overall network performance.

3.5 Summary

In this chapter, we talked about jamming attack detection methods which were the prerequisite of jamming attack mitigation strategy. Both of the signal strength and packet delivery ratio are used for the detection. The jamming attack detection scheme was implemented in the MAC layer and returned the result to the application layer. The application layer might take action once it detected the attacks. The attack mitigation schemes can be implemented effectively based on the results of jamming attack detection. In next chapter, we will propose and implement jamming attack mitigation schemes elaborately.

CHAPTER 4

IEEE 802.11 DOS ATTACK MITIGATION

4.1 Related Work

Theoretically, a powerful jammer may attack all the channels in the range closed to the device. However most of the times, the jammer may be restricted by the hardware configuration and cannot use high emit power to jam all the channels, making it feasible to mitigate the jamming attack influence. This chapter will review the schemes to mitigate jamming attacks.

We have defined the jamming attack in the introduction part as decreasing signal to noise ratio (SNR) by the attacker's radio signals and know the SNR is the key point of wireless jamming. Now, let's look at the two elements of SNR. $SNR=P_{signal}/P_{noise}$. The attackers decrease the SNR by increasing the value of P_{noise}. In order to mitigate the attacking effects, we should increase the value of SNR. Of course, we could increase the value of P_{signal}, decrease the value of P_{noise} or both. During our survey, most of the jamming attack mitigation techniques may be classified into changing the value of either P_{signal} or P_{noise}.

P_{signal} is the useful receiving power legitimate wireless signals. The measures to increasing the P_{signal} include increasing the send power on the legitimate wireless devices or shorten the distance of wireless devices to reduce the signal attenuation.

P_{noise} is the unwanted signal that may be sent by the attackers or other nearby wireless devices. In order to reduce the noise signal, we can move the legitimate wireless devices far

away from the attackers, switch to another channel which is less influenced by the attackers, or even optimize the hardware or MAC layer wireless protocol.

There are some methods presented in [28] to deal with the wireless jamming. First of all, we know that higher transmitted power may achieve higher resistance against jamming which may be inducted from the formula of SNR. Second, the FHSS (frequency-hopping spread spectrum), DSSS (direct sequence spread spectrum) or hybrid FHSS/DSSS spread spectrum methods may be used to minimize unauthorized interception and jamming of radio transmission. Third, Ultra Wide Band (UWB) technology, which is based on transmitting short pulses on a large spectrum of a frequency band, causes the transmitted signal more resistant to the jamming. In the end, adjusting the polarization of antenna or using the directional antennas to transmit signals may help to improve jamming tolerance.

The authors in [28] also classify the countermeasures against jamming attacks into three categories: proactive, reactive and mobile agent-based. Instead of reacting after detecting the occurrence of jamming, the proactive countermeasures make the wireless network immune to the jamming attacks. The reactive countermeasures enable reaction only upon detecting the jamming attacks, thus leading to low energy cost compared to proactive jamming countermeasures. According to the paper, both proactive and reactive countermeasures can be implemented as software based countermeasures or software-hardware combined countermeasures

4.1.1 Channel Hopping

4.1.1.1 Constraints of 802.11 channels

Channel hopping is the most used countermeasures to mitigate the jamming attack influence in 802.11 wireless networks at current. However, there are some constraints in 802.11 b/g wireless networks when using the channel hopping methods [19].

It is well known that 802.11 b/g network provides 11 channels from channel 1 to channel 11 for the FCC domain where each channel has the bandwidth of 22 MHz and the center frequency separation of adjacent channels is only 5 MHz. As a result the adjacent channels will overlap with each other and be interfered from the signals of adjacent channels, causing that there are only three non-overlapping (orthogonal) channels (channel 1, 6 and 11) available in the 802.11 b/g wireless networks as in figure 1.1 of first chapter [14].

4.1.1.2 Reactive channel switch

In the reactive channel switch scheme, the wireless network only changes channel after detecting the wireless attack as in figure 4.1. The next channel should be orthogonal with the old channel. There are nearly no research about the reactive channel switch right now. We will propose our own attack detection and mitigation method in this research work.

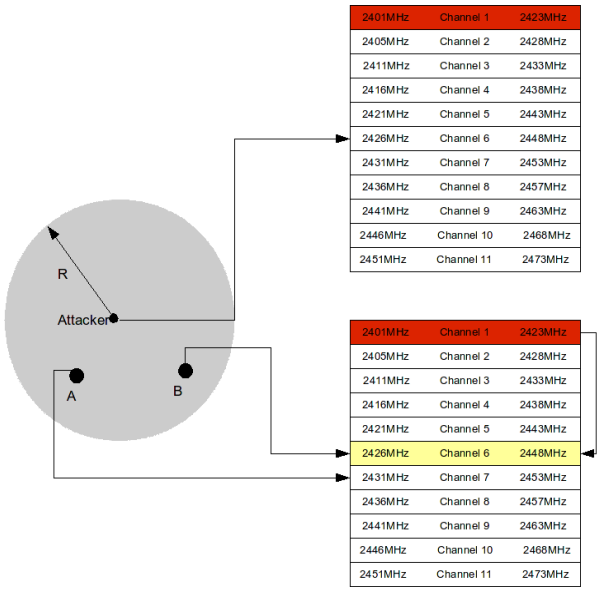

Figure 4.1 Reactive channel switch

4.1.1.3 Proactive Channel Hopping

The above DoS mitigation schemes are passive DoS mitigation schemes. The wireless network only switches channel after the coordinators detect the DoS attacks. The advantage of the passive schemes is the less average switching cost. The wireless network only needs to switch channel when the attackers exist around the wireless stations. During the period of switching channel, the network is disconnected and cannot transmit data packets. When there are no attackers near the wireless network, the network doesn't need to switch channel. The passive channel hoping scheme reduces the operational cost caused by switching channel.

However, there are drawbacks of the passive channel switch scheme. The wireless network needs to detect the attackers before switching channel. The process of detecting attackers may cost certain time. As a result, the wireless network will use more time to detect the attackers even there are no DoS attack, since the attack detection scheme needs to measure the current network status, such as the signal strength, throughputs, and so on. Here we present another DoS mitigation scheme called “always-on prevention” scheme.

The proactive channel hopping method is an always-on DoS jamming prevention mechanism. The always-on DoS prevention is an active channel switch mechanism which will switch the wireless network channel no matter whether there exists attackers or not. In this mechanism, the wireless network will switch channel very frequently and get rid of the attack by switching to new channel before the attackers detect the current transmitting channel of the network.

There are also some researches performed the proactive channel hopping in 802.11 wireless networks [22] [13] [26]. In [22], the authors did some tests on the 802.11a wireless networks with 5GHz band. The purpose of the tests is to evaluate the methods of channel

hopping during the jamming attack. During the period of jamming attacks, the nodes with channel hopping keep in the high throughput (about 17 Mbps). While the wireless nodes without channel hopping have a low throughput (about 2 Mbps). From the test we can see that, the channel hopping methods may increase the network throughput during the period of jamming attack. When there is no jamming attack, the channel hopping method will not decrease the throughput obviously when selecting proper parameters such as channel residence time.

4.1.2 Spatial Defense

Mobile agent solutions may improve the survivability of wireless network with a cluster of mobile hosts during attack by moving to another place to escape from the attacker's influence range as in figure 4.2. The ant system with evolutionary algorithm in [30] is an example to using mobile agent solutions to detect and avoid jamming attacks.

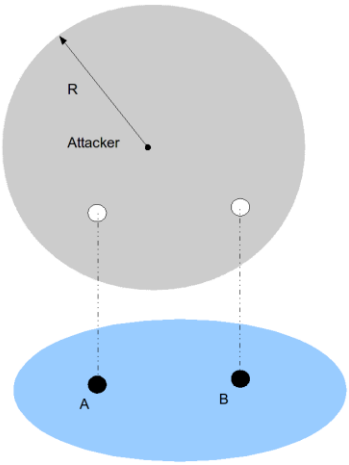

Figure 4.2 Spatial defense

In [18], the authors depict a better spatial defense scenario where the wireless nodes can adjust their position to defeat the attacker. When the attacker comes into the network, the wireless nodes around the attacker will move away from the attacker and organize the new

wireless network automatically. As the attacker moves away, the nodes will resume their old topology for optimal performance.

4.1.3 DEEJAM

Wood et al. described four jamming attack classes and proposed four related defense strategies in 802.15.4-based wireless networks [17] which may also be transplanted to the 802.11 wireless networks.

4.1.3.1 Interrupt Jamming vs. frame Masking

The interrupt jammer starts to jam the network only when valid radio activity is signaled from radio hardware. The Start Frame Delimiter (SFD) is used by the jammer to detect the radio activity. Once detecting the SFD, the jammer will start emitting the jamming radio to disturb the payload after a delay of $T_{txdelay}$. Figure 4.3 shows the interrupt jamming scheme. [17]

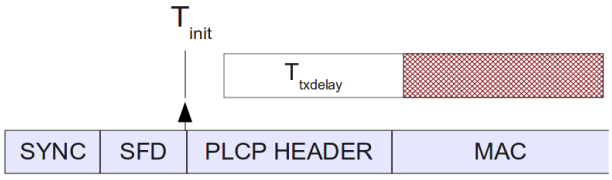

Figure 4.3 Interrupt jamming scheme

In the Frame Masking defense, the sender and receiver agree on a secret pseudo-random sequence for the SFD in each packet which make the SFD transmit time unable to detect for the attacker. The pseudo-random number is generated by a pairwise shared key. The Frame Masking can effectively avoid the interrupt. However, it is vulnerable to the Activity Jamming.

4.1.3.2 Activity Jamming vs. Channel Hopping

The attacker cannot continue the jamming since it cannot detect the SFDs. However, the attacker may detect the activity of transmitting packets by sampling the radio signal strength

indicator (RSSI). If the RSSI is above a certain threshold, the attacker will consider there exists packets transmission and will initiate the attack. As in the figure 4.4, the attacker samples the RSSI periodically and begins jamming upon packet detection.

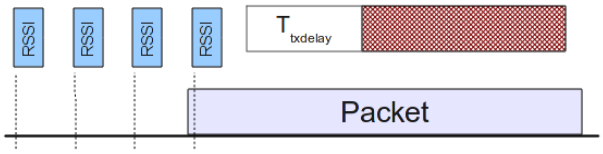

Figure 4.4 Activity Jamming

Since the attacker can only sample RSSI for one channel on which it is listening, in the Channel Hopping defense scheme, the sender and receiver may change channels to avoid the jamming. The pairs use their shared key to create a channel key which generates a pseudo-random channel sequence. Each time the sender will use a new channel to transmit the packet.

4.1.3.3 Scan Jamming vs. Packet Fragmentation

The attacker can scan all the channels in the entire band to search for the current transmitted packet and try to jam the packet immediately once find the packet. As in figure 4.5, the attacker scans channel C_{j-1}, C_j and C_{j+1} then detects the packet by sample the RSSI at channel C_{j+1}. Then after initiating for time T_{init} and delay of $T_{txdelay}$ it will start the attack.

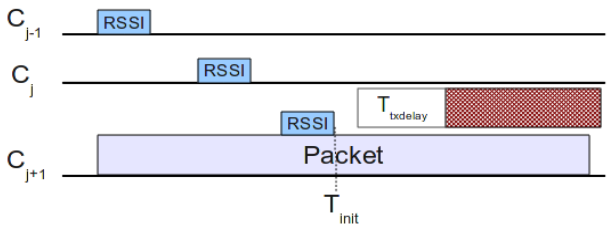

Figure 4.5 Scan Jamming

The Scan Jamming attack requires the attacker complete scanning the entire channel within the time of transmitting of one packet. So the attacker needs to sample each channel as briefly as possible. As a result, we can use Packet Fragmentation defense (as shown in figure 4.6) to defeat the Scan Jamming attack.

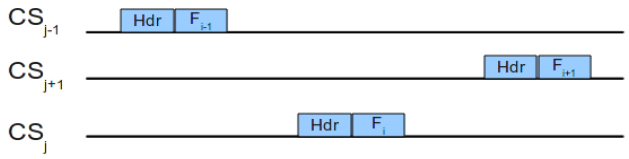

Figure 4.6 Scan jamming attack detection

The sender and receiver will break the payload into fragments to transmit separately in different channel. The time period used to transmit one fragment is smaller than the time of sampling the packet and initiating the attack by the attacker. The scheme of packet Fragment is valid to evade the jamming attack of scan jammers.

4.1.3.4 Pulse Jamming vs. Redundant Encoding

The best remaining strategy for the attacker seems to be jamming continuously on a single channel. In the pulse jamming attack, the jammer remains on a single channel and jams the fragments transmitted there. As a result, as long as one fragment is lost, the whole packet will be invalid. The Redundant Encoding is used to defeat the Pulse Jamming attack. This scheme allows the receiver to recover from one or more corrupted fragments with duplicated fragment and discard other repeated fragments.

4.1.4 Directional Antenna

Most of today's wireless devices use Omni-directional antennas which emit the signals to all sides from the antennas and cause considerable signal influence to other nearby wireless

devices. The use of directional antennas could obviously improve the attack tolerance in wireless networks [28]. The directional transmission also offers better protection to cope with eavesdropping, detection and jamming attack than the common Omni-directional transmission [31] [32] [33]. As in figure 4.7 from [34], one wireless node with multiple directional antennas may have multiple connections with other neighbor node and have multiple transmissions at the same time.

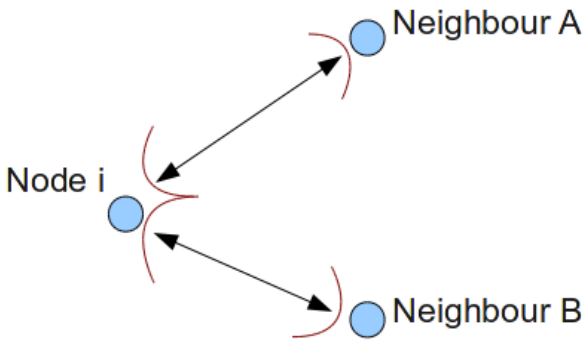

Figure 4.7 Directional antenna

Directional antenna is a good choice to reduce the wireless interference. At the same time, it is also a strategy to avoid the jamming attacks. As we mentioned earlier, the jammers need to scan the channel and find out the current channel used by the legal devices and then launch the attack. The directional transmission can make the scan process difficult, since the transmission path may not pass through the attacker, thus making the attacking process difficult.

4.1.5 Other Defense Strategy

There are other countermeasures that may deal with jamming attacks. For example, the use of low transmitted power decreases the discovery probability from the attacker. High transmitted power may also improve the resistance against jamming attacks [35]. Ultra wide band technology (UWB), which is based on transmitting short pulses on large spectrum of a

frequency band simultaneously, may also help to guarantee the wireless network communication. Spread spectrum techniques [35] are used to spread the energy level of signal across a wide bandwidth.

4.2 802.11 ATTACK MITIGATION STRATEGY

The basic idea for the always hopping scheme is to keep switching channel no matter there are existing attackers or not. The scheme is based on the assumption that the attackers scan all the channels, find out the current channel used by legal devices and then start jamming. During the jamming process, the attackers need a period to scan channel and jump to the channel before the real jamming. The always-hopping scheme utilizes the period to avoid being attacked by jumping to the new channel before the attackers find the current channel used by current wireless network.

The successive channels always hopping scheme is a simple implementation which completes the basic function of channel hopping. For example, the hopping method can be simply implemented by controlling the whole wireless network to switch channel from the channel 1 to channel 11 then go back to channel 1 as shown in figure 4.8.

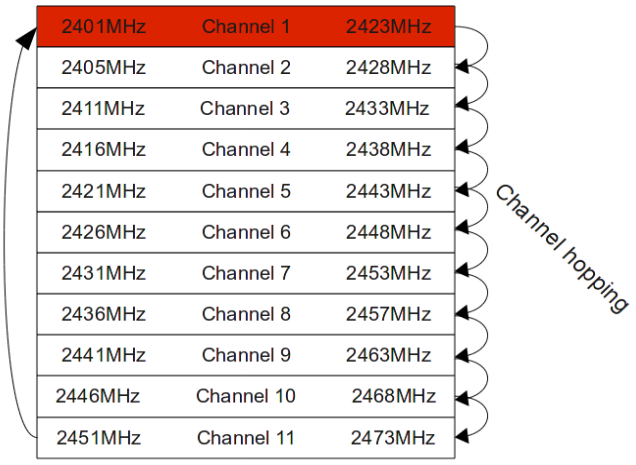

Figure 4.8 Successive channel hopping strategy

The simple channel hopping method is easy for the attacker to find out the channel hoping pattern and perform effective jamming according to the pattern. We will propose and implement new attack mitigation strategies for different kinds of attackers in this part. Our mitigation strategies include the random channel generation, the algorithm of channel hopping, new MAC layer header format and so on.

4.2.1 Always-on DoS prevention mechanisms

The above DoS mitigation schemes are passive DoS mitigation schemes. The wireless network only switches channel when the coordinators detect the DoS attacks. The advantage of the passive schemes is the less switching cost. The wireless network only needs to switch channel when the attackers exist around the stations. During the period of switching channel, the network is disconnected and cannot transmit data packets. When there are no attackers in near the wireless network, the network doesn't need to switch channel, thus reducing the operational cost caused by switching channel.

However, there are drawbacks of the passive channel switch scheme. The wireless network needs to detect the attackers before switching channel. The process of detecting attackers may spend time. As a result, the wireless network will use more time to detect the attackers even there are no DoS attack, since the attack detection scheme needs to measure the current network status, such as the signal strength, throughputs, and so on. Here we present another DoS mitigation scheme called “always-on prevention” scheme.

The always-on DoS prevention is an active channel switch mechanism which will switch the wireless network channel no matter whether there exists attackers or not. In this mechanism, the wireless network will switch channel very frequently and get rid of the attack by switching to new channel before the attackers detect the current transmitting channel of the network.

It is easier to implement the always-on DoS prevention in the application layer. However, the channel switch in application layer needs long time (several seconds) which is not acceptable for a frequently channel switch network. In this part we will design and implement channel switch in MAC layer which will short the channel switch time to one hundred milliseconds (estimate value). As a result, the always-on DoS prevention will be effective in mitigate the DoS attack.

The always-on DoS prevention algorithm is simple. The whole wireless network switches to a new channel every fixed time interval. The time interval will be determined later in the implementation. The next switch channel number may be decided by the coordinators and then broadcast to other wireless nodes. The channel number may also be determined by a special encryption function which is run on all the wireless stations and gives each wireless station the same channel number each time.

The drawback of the always-on DoS prevention algorithm is wasting time during the process of switching channel. However, when using the MAC layer channel switch the drawback may become small since the channel switch time is small compared to the time period between each channel switch.

4.2.2 Based Random Channel Generation

The Linear Congruential Generator (LCG) algorithm may be used to generate the pseudorandom numbers:

$$X(n) = [a * X(n-1) + b] \, mod \, m \quad (1)$$

where a is the multiplier ($0 < a < m$), b is the increment ($0 \leq b < m$), and m is the modulus ($m=11$ for 802.11b/g networks). X(n) and X(n-1) are the n-th and (n-1)-th numbers, respectively,

in the sequence generated by the LCG. X(0) is called the seed of the LCG ($0 \leq X(0) < m$). X0, a, and b are the parameters of the LCG. The LCG will have a full period for all seed values if and only if:

1, c and m are relatively prime,

2, a - 1 is divisible by all prime factors of m,

3, a - 1 is a multiple of 4 if m is a multiple of 4.

In 802.11b/g wireless network (where m = 11), it is difficult to select the appropriate for a full random number distribution from 1 to 11.For example, if we select a = 4, b = 5 and X(0) = 1, the random number generated by the formula can only be 1, 9, 8, 4 and 10.

We design our own pseudorandom number generator by improving the LCG algorithm:

$$X(n) = \begin{cases} (X(n-1)/a)*b - (X(n-1)\%a)*c), & result > 0 \\ (X(n-1)/a)*b - (X(n-1)\%a)*c) + m, & lastresult \leq 0 \end{cases} \quad (2)$$

C(n) = [X(n) mod (m+1)] mod 11 + 1

In this algorithm we separate the seeds from the random numbers. X(i) is the i-th seed for the random numbers and for the next seed. C(i) is the i-th random number. X(0) is the initial seed for the whole random number sequence. a, b, c are constant parameters and m is the modulus.

We select X(0) as 2 and have the test. The channel numbers may cover the full period from channel 1 to channel 11. In the evaluation, the random channel number will be limited between 1 and 11 and generated 10000 times. Figure 4.9 shows the channels are well-distributed between channel 1 and channel 11.

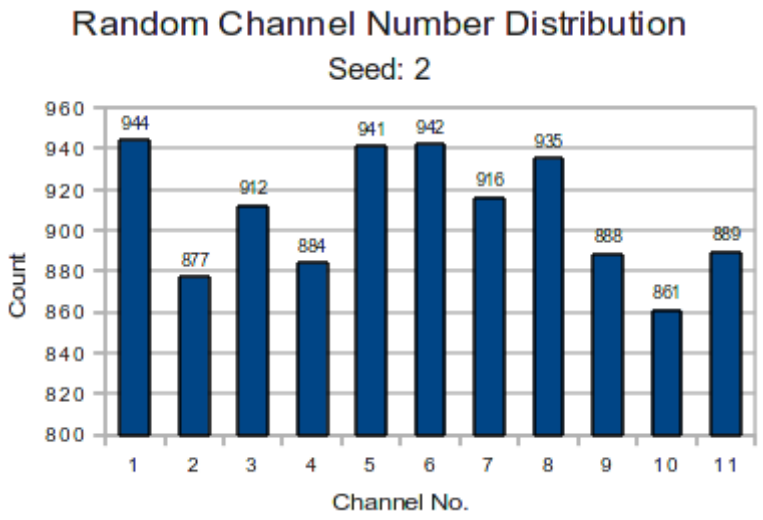

Figure 4.9 Random channel number distribution

If a station plans to jump at the current time is t, it tries to find the value n, and then uses algorithm 1 with guard for obtaining attempting channel. Assume that initial time is t0, and a fixed and predefined constant is T. We have,

$$n = (t - t0)/T \bmod m \quad (3)$$

4.2.3 Algorithm with a guard

There are possibilities that the new channel number generated by the function is the same or near the old channel number. As a result we added a 'guard' to the channel generation function. The 'guard' guaranteed that the next channel should not be the same as or closed to the previous channel and decreases the possibility of being detected and attacked by the jammer. For example, setting the 'guard' to 0 means we don't care about the original channels and the new channel can be the same as the previous channel. When the 'guard' is set to 1, the channel of next hop should not be same as the current channel.

The algorithm with a guard is presented by the following in order to generate the next channel (C(next)):

Do
{

$$X(next) = \begin{cases} (X(seed)/a)*b - (X(seed)\%a)*c), & result > 0 \\ (X(seed)/a)*b - (X(seed)\%a)*c) + m, & lastresult \le 0 \end{cases}$$

C(next) = [X(next) mod (m+1)] mod 11 + 1
X(seed)=X(next)
}
while (|C(next) – C(current)| < guard)

(4)

We first set the guard to 1 in the evaluation, limit the random channel number between 1 and 11 and generate 10000 times. Figure 4.10 shows the channel distribution with/without the guard.

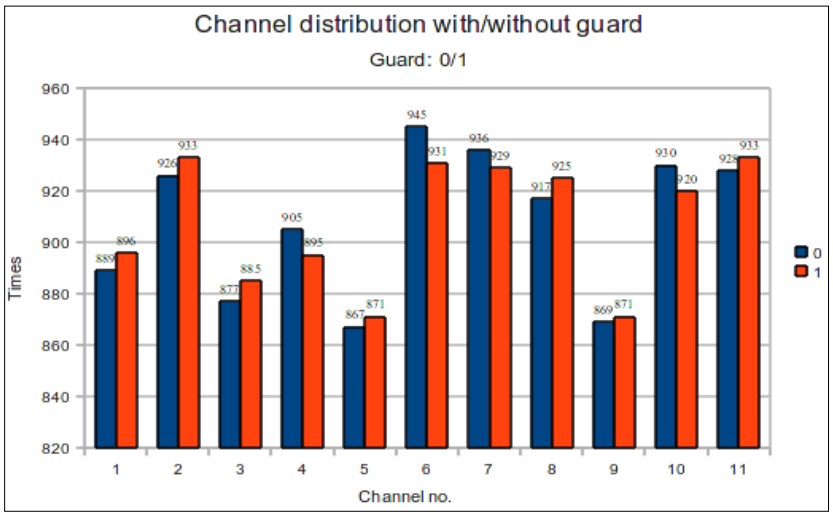

Figure 4.10 Channel distribution with/without guard

In 802.11 protocol, the channel is taken up 22M frequency. There are only 3 orthogonal channels (1, 6 and 11) in the 802.11 b/g networks (FCC domain).The smart jammer may detect the channel of the legal devices and switch to that channel to continue jamming. As a result, even if the legal devices jump to a closed channel, it can still be affected by the jammer. In order to

avoid or mitigate the adjacent channel influence, the 'guard' should be set to a bigger number other than just set to 1 to avoid switch to the same channel. Figure 4.11 shows the channel distribution for different guard values.

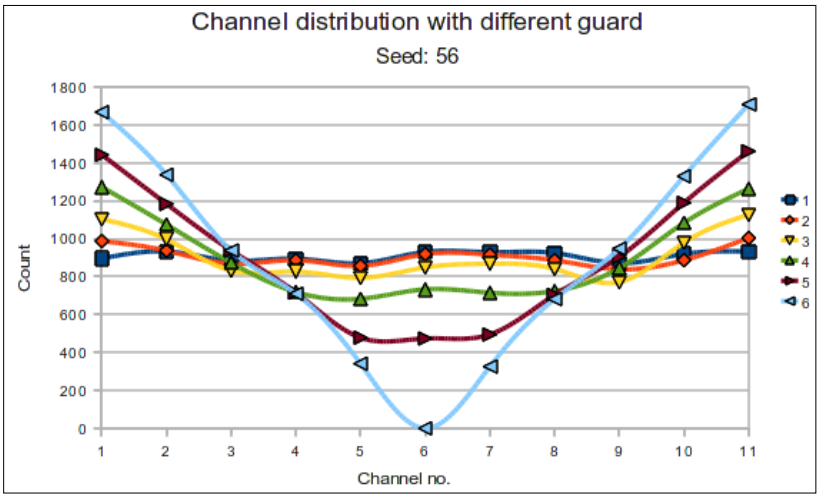

Figure 4.11 Channel distribution with different guard

We can see from the figure that as we increase the guard value, there are more channel numbers generated near the both sides of the channels (channel 1 and channel 11) and less channel near the middle (channel 6). When we set the guard value to 6, there are nearly 0 channels in channel 6. That's because even when the old channel is 1, the new channel number should be larger than 6 when applying the guard to generate the new channel.

4.2.4 Randomized Pattern Channel Generation

Sometimes we need to switch channel by following some channel number patterns in a loop. In order to avoid the attackers from guessing the pattern, we need generating the channel pattern by re-numbering the old channel numbers. The RANDOMIZE-IN-PLACE algorithm [15] is modified to generate the pattern.

RANDOMIZE-CHANNEL-PATTERN(A)

Input. An array A[1..11] of 11 channel numbers.
Output. A rearrangement of the channel numbers of array A, with every permutation of the numbers 1-11.

```
for i := 1 to 11 do
swap A[i] and A[Random(i, 11)]
rof
```

The function of Random(i,11) returns a random number between i and 11. This function can be implemented by utilizing the method of Based Random Channel Generation talked in above parts.

RANDOM (p, q)

Input. Two numbers of p and 1 (p≤q).
Output. Return a random number between p and q

$$X(n) = \begin{cases} (X(n-1)/a)*b - (X(n-1)\%a)*c), & result > 0 \\ (X(n-1)/a)*b - (X(n-1)\%a)*c) + m, & lastresult \le 0 \end{cases}$$

Return [X(n) mod (m+1)] mod (q-p+1) + p

(5)

We select X(0) as 1024 and have the test. The result of above algorithm is a sequence of 11 channel numbers after re-numbering. We can convert it to the table which shows the mapping between the old channel and new channel to switch as in table 4.

The table can be converted to an array A[11] with A[1]=6; A[2]=11, and so on. As a result,the new channel number can be got in constant time by the formula 6:

```
new channel = A[old channel]
```

(6)

Table 4.1 Old channel to New channel

Old channel	New channel
1	6
2	11
3	9
4	8
5	4
6	7
7	5
8	2
9	1
10	3
11	10

4.2.5 Channel switch synchronization

Synchronization is needed for the wireless nodes to switch channel together. Normally, when one node decides to switch channel, it has the responsibility to notify other nodes to switch channel together. In our developing and test platform, by using the Atheros chipset adapter and Madwifi driver, the node can automatically send out the packets to notify other nodes at the time of switching channel. This method is enough during normal use to switch channel. However, it is too simple to synchronize the nodes together when the communication between the nodes is blocked under the Deny of Service attack.

During the period of DoS attack, the communication between wireless nodes is blocked in the attacked channel. As a result, the channel switch notification may not be received by other wireless nodes and the connection between the nodes may be broken. The idea of synchronization before channel switching as in the figure 4.12 is used to solve the problem.

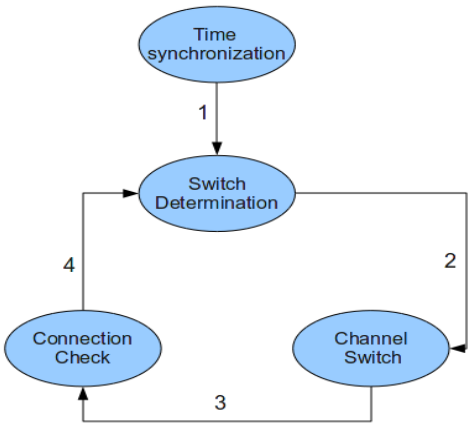

Figure 4.12 Channel switch synchronization

In time synchronization period, the access point and stations switch packets to synchronize their time which is used to prepare for the channel switch together. In this period, the seeds for generating next random channel number is also synchronized in this period. After the synchronization, each node knows the seeds and the time they can receive the channel switch notification. Therefore, each node may generate the next channel number by itself and switch channel in case the communication is blocked by the attacker.

Usually, it is the access point that determines when to switch channel. After time synchronization, the access point will send frames to notify the stations to switch channel at the designated time. After receiving the notification, the stations may switch channel according to the information from access point. However, if the channel of the wireless network is the same as the attacker, the stations cannot receive the message from access point and cannot switch to the new channel. A timer is set for the wireless station in case of the block of communication. When the timer is time out, the station will generate the next channel number by itself and switch to the new channel and keep its connection with access point.

After receiving the channel switch notification or timeout of the timer, the stations will switch to the new channel by directly calling the channel switch function. The hardware will be reset, and related configuration will be set to fit the new channel. At the same time, the access point will switch to the channel to keep connection with the stations.

In the connection check period, Access Point checks the connection in the new channel. If the connection is still established, the next time channel switch will be prepared after a short time of data packets transmission. At the same time, the stations will reset the timer and wait for the switch channel notification or time out.

The synchronization and channel switch need the new proposed MAC layer packet format which will be explained in the next part.

4.3 New Management Frame

We proposed new management frame format for the 802.11 MAC layer. We modified the DoS frame content and put the DoS subtype control field in the MAC header. We also defined the new MAC header as following:

```
struct ieee80211_dosframe {
      u_int8_t i_fc[2];
      __le16 i_dur;
      u_int8_t i_addr1[IEEE80211_ADDR_LEN];
      u_int8_t i_addr2[IEEE80211_ADDR_LEN];
      u_int8_t i_addr3[IEEE80211_ADDR_LEN];
      u_int8_t i_seq[2];
+      u_int8_t i_dos[2];
} __packed;
```

The DoS MAC header uses the three addresses format which is shown in figure 4.13.

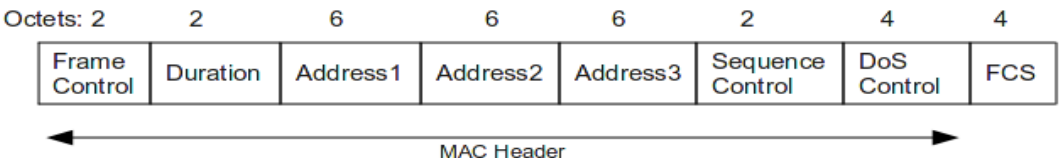

Figure 4.13 DoS MAC Header structure

In this 802.11 MAC header structure, the Address1 is the Destination MAC Address, the Address2 is the Source MAC Address and the Address3 is the BSSID. We add the four bits DoS control between the Sequence Control and the FCS field.

The Frame Control field has the value of 0x0060 for the DoS frame and its structure is shown in figure 4.14:

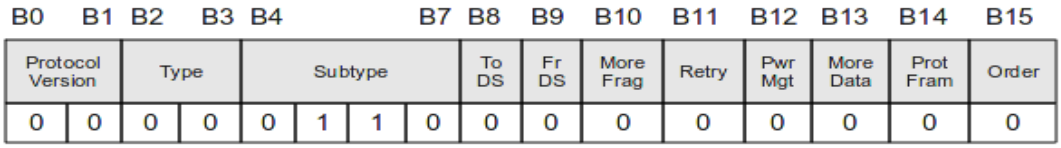

Figure 4.14 Frame Control field for DoS MAC header

The type field “00” means the new frame is belong to the Management Frame. The new subtype field is “11”. Four types of DoS sub types were defined and specified in the DoS Control field. The first is type is the KeepConnection frame with the definition and MAC header format in the figure 4.15:

#define IEEE80211_DOS_KEEPCONNECTION 0x0000

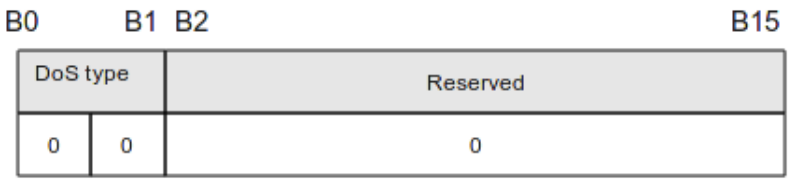

Figure 4.15 KeepConnection frame

The second DoS frame subtype is the ConnectionACK frame with the definition and MAC header format in the figure 4.16:

#define IEEE80211_DOS_CONNECTIONACK 0x0001

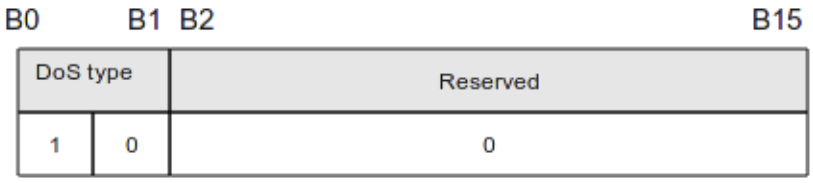

Figure 4.16 ConnectionACK frame

The third DoS frame subtype is defined for the ChannelSwitch with the definition and MAC header format in figure 4.17:

#define IEEE80211_DOS_CH_SWITCH 0x0002

Figure 4.17 Channel Switch frame

The forth DoS frame subtype is defined for the AlwaysHopping with the definition and MAC header format in the figure 4.18:

#define IEEE80211_DOS_ALWAYS_HOPPING 0x***3

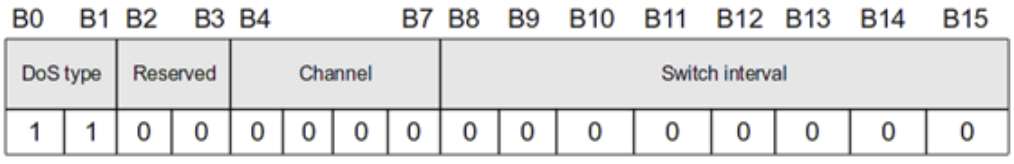

Figure 4.18 Always Hopping frame

There are 8 bits for the switch interval, which only allow inputting the value from 0 to 255. However, the unit used in timer function is millisecond. In order to meet our needs, we modify the unit here to 100 milliseconds which give the range of switch interval from 0 to 25500 milliseconds.

The channel field has 4 bits and suit for the channel from 0 to 15. We only need channel 1 to 11 for the IEEE802.11b/g wireless network in the FCC domain. By the way, the channel format is only for the 802.11b/g network.

4.4 Experiment Result

4.4.1 Performance without channel hopping and no attacking

This experiment measures the performance of wireless network without jammers. The parameters used to measure the performance are throughput and network delay/jitters. We use iperf to have the test in this experiment. The server side uses the iperf commands which are similar to the following example command:

```
# iperf -s -u -P 0 -i 1 -p 5001 -f k
```

The above command means that the iperf runs in the server mode (-s), listened on the port of 5001 (-p) and enables UDP (-u) connections from unlimited client threads (-P). Iperf will periodically report the test result with the format of kbps (-f k) every 1 second (-i).

The client side uses the iperf commands which are similar to the following example command:

```
# iperf -c 192.168.0.2 -u -P 1 -i 1 -p 5001 -f k -b 20.0M -t 60 -T 1
```

The client command means that the iperf runs in the client mode (-c) which connect to the server with the IP address of "192.168.0.2" and the UDP (-u) port of 5001 (-p). The client will use one thread (-P) with the bandwidth of 20 Mbps (-b) to send packets. Iperf will periodically report the test result with the format of kbps (-f k) every 1 second (-i). The test time is 60 seconds (-t).

We selected different UDP bandwidth (40M, 20M, 10M, 5M and 2M bps) and applied them to different test cases. The 802.11g wireless network operates at a maximum physical layer bit rate of 54 Mbps. However, due to the packet conflict and packet check, the real world throughput of the 802.11g network may be only as high as 27 Mbps. The real world throughput can also be proved through our tests. The throughput and jitter in the server side and the throughput in the client side are recorded and shown in figure 4.19.

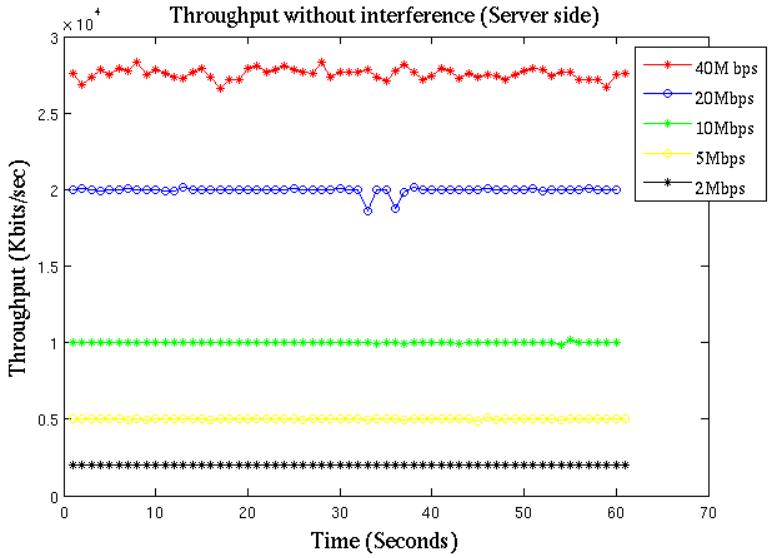

Figure 4.19 Throughput wighout interference (server)

The figure shows the throughput measurement in the server side. When the client used the bandwidth of 2M, 5M, 10M or 20M bps to transmit UDP data, the throughput on the server

side has the full bandwidth. When the client used the bandwidth of 40M bps, the average throughput on the server side can only reached to about 27.6 Mbps which is the limit of throughput of 802.11g wireless network.

We can also see from the figure that the curve with higher bandwidth has bigger fluctuation. The red curve with the bandwidth of 40M bps has obvious fluctuation through the whole 60 seconds. The blue curve with the bandwidth of 20M bps has the fluctuation in the middle. The green curve with the bandwidth of 10M bps has the fluctuation in the end. While the yellow and black curves with the lower bandwidth of 5M and 2M bps are very stable and don't have any obvious fluctuation through the 60 seconds in the figure.

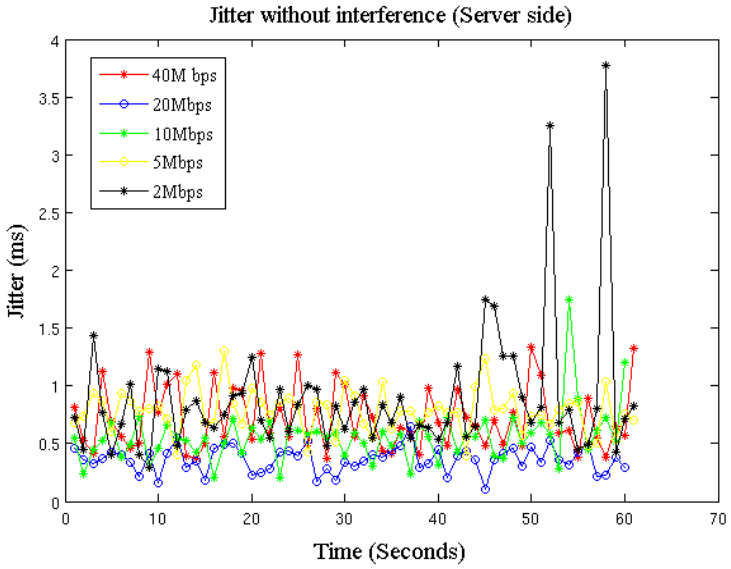

Figure 4.20 Jitter without interference (server)

The jitter/network delay on the server side is stable for any of the five curves in this test without the interference as shown in figure 4.20. For most of the time the jitter is smaller than 1.5 milliseconds which match the stable throughput in the figure 4.19.

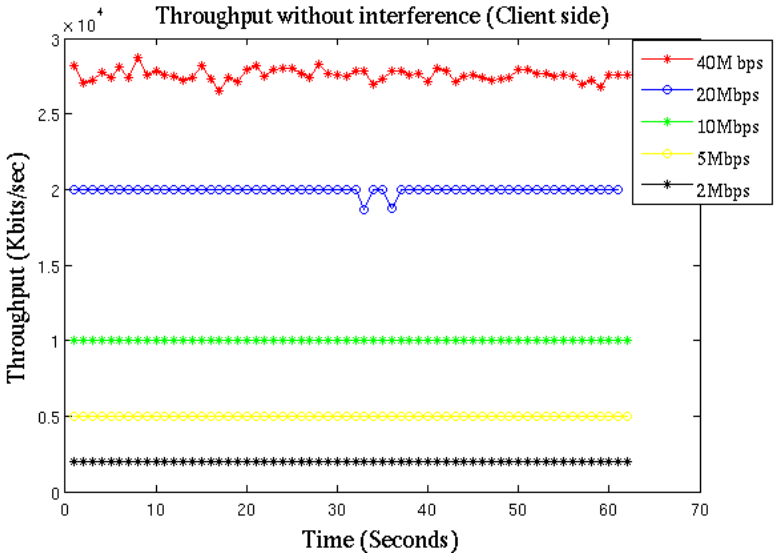

Figure 4.21 Throughput without interference (Client)

The throughputs on the client side are very similar to the server side besides some small differences which may not be distinguished in figure 4.21. The throughputs are still fluctuated up and down for the large bandwidth and more stable for the low bandwidth.

4.4.2 Performance with channel hopping and no attacking

We measure the network throughput and jitters when the wireless devices keep switching channel without being attacked by the jammer in this part. We still selected different UDP bandwidth (40M, 20M, 10M, 5M and 2M bps) and applied them to different test cases. The throughput and jitter in the server side and the throughput in the client side are recorded in figure 4.22, which shows the throughput measurement in the server side with different bandwidths in the client side. When the client used the bandwidth of 2M, 5M, 10M or 20M bps to transmit UDP data, the throughput on the server side almost has the full bandwidth. When the client used the bandwidth of 40M bps, the average throughput on the server side can only reached to about 23 Mbps which is the lower than the no channel hopping case.

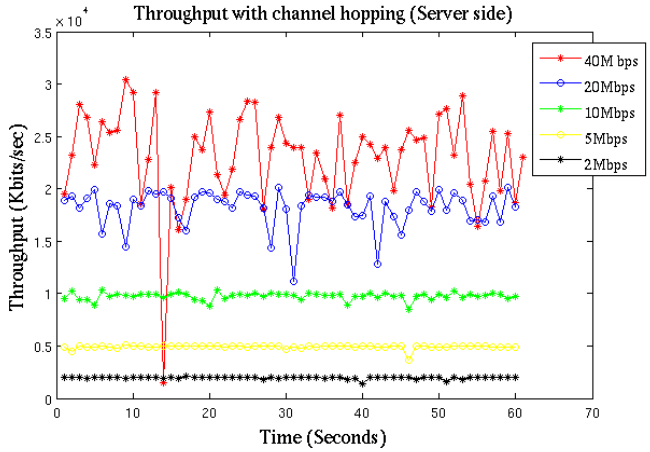

Figure 4.22 Throughput with channel hopping (Server)

Like the no channel hopping case, we can also see from the figure that the curve with higher bandwidth has bigger fluctuation. The red and the blue curves with the bandwidth of 40M and 20M bps have obvious fluctuation through the whole 60 seconds. The other three curves (green, yellow and black) with the bandwidth of 10M, 5M and 2M bps are very stable and don't have any obvious fluctuation through the 60 seconds in figure.

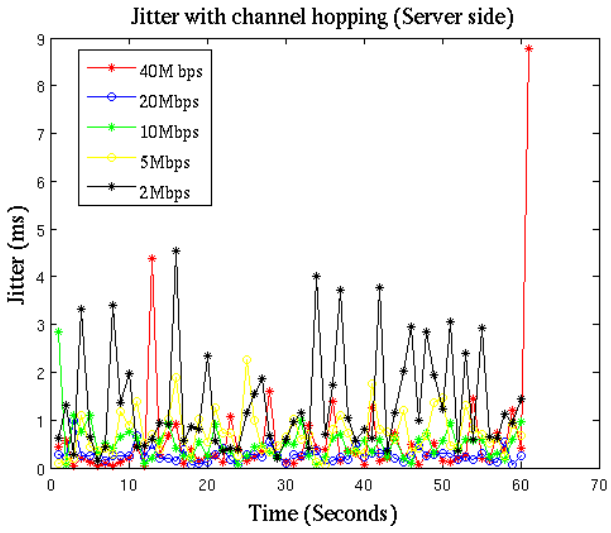

Figure 4.23 Jitter with channel hopping (Server)

The average jitter/network delay shown in the figure 4.23 is longer than in the no channel hopping case. It is stable for most of the period. The throughput with channel hopping in the client side (figure 4.24) is similar to the throughput in the server side.

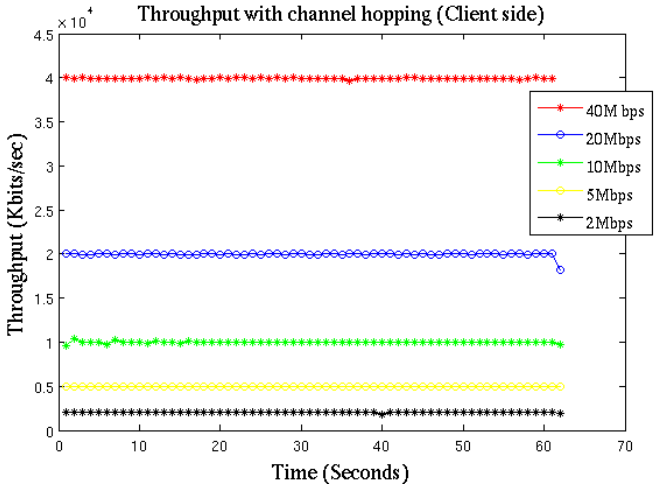

Figure 4.24 Throughput with channel hopping (Client)

The performance of the wireless networks which have the behavior of channel hopping could be affected by the channel switch interval (**I**) and the traffic rate (**R**). The channel switch interval is consisted of two parts: the channel residing time (**T**) and channel switch time (**S**). The communication and channel hopping process can be presented like this: the wireless devices use the traffic rate of **R** to transmit packets. After the transmitting time of **R**, the wireless devices spend **S** time to switch channel and then continue **R** time's packets transmission. The interval between two channel hopping is **I** and **I = T+ S**. The analytical throughput can be expressed by the following equation:

$$g(I, S, R) = \frac{I-S}{I} R \qquad (7)$$

The channel switch time is a fixed value and channel switch interval can be changed during our implementation. The channel residing time is difficult to get directly and can be presented by **I – S** as in the above equation. The equation shows that the throughput should be increased as the value of I be increased. The maximum value of throughput is the transmission rate when **I** is big enough (which means the wireless network seldom switches channel). We still selected different UDP bandwidth (20M bps) and use different channel switch interval (1000ms, 500ms and 250ms) to test the performance of wireless network during channel hopping. The throughput and jitter in the server side are recorded and shown in figure 4.25 and 4.26: The figures show that when the channel switch interval is decreased, the throughput in the wireless network drops and the jitter becomes high. The over frequently channel hopping will degrade the performance of the wireless network. We need to balance between the performance of wireless network and attacker's influence.

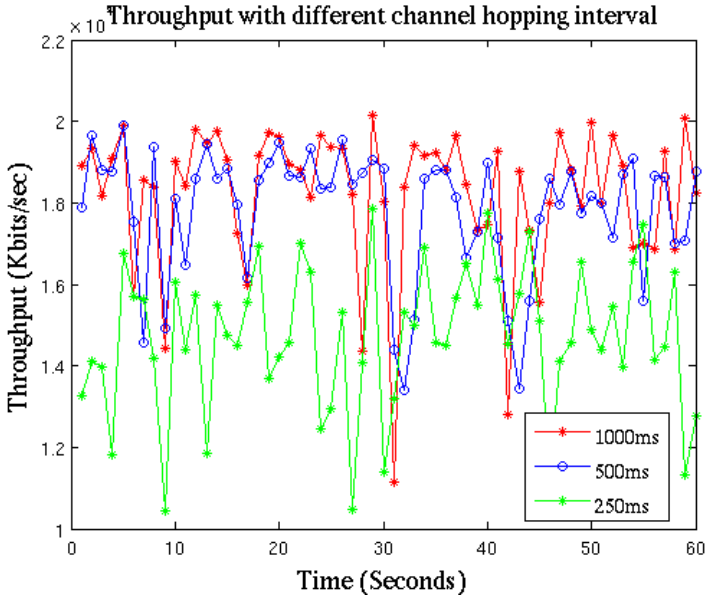

Figure 4.25 Throughput with different channel hopping interval

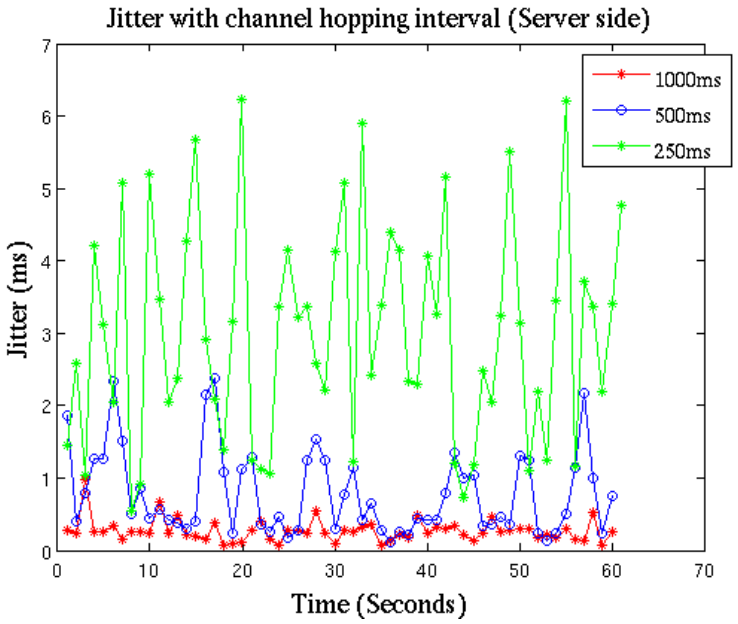

Figure 4.26 Jitter with different channel switch interval

4.4.3 Performance without channel hopping and under attacking

This part measures the network throughput when the wireless devices are attacked by the jammer. We selected UDP bandwidth of 10M bps and applied them to different test cases. Two different scenarios will be considered: the jammer is fixed in one channel and the jammer switches in random channels.

4.4.3.1 Adjacent channel jamming interference

The two adapters are configured as ad-hoc mode and communicate with each other from channel 1 to channel 11. The channel switch interval of the wireless network is set to 1000ms. The attacker device will disable backoff time and start attack at channel 1. We will have a 30 seconds throughput test at each channel and record the data every second.

For the channel 1 attack, the user and server cannot communicate with each other and the throughput drops to zero. As a result, we start the attack in channel 1 between the 10th second and the 25th second. From the figure 4.27 we can see that the channel 1 throughput drops to zero

immediately around the 10th second and resumes to normal at the 25th second when we turn off the attacker. The throughput in channel 2 also has obvious drop during the attacking period. For the channel 3 to channel 7, the throughputs don't change too much when start the attacker. Another strange curve is channel 6 which is lower than channel 3, 4, 5 and 7 even the attacker is stopped. The reasonable explanation is that most people are accustomed to use the three non-overlapping channels of 1, 6 and 11. Although channel 6 is not attacked by the attacker, it is affected by other wireless devices in channel 1 and leads to the low throughput.

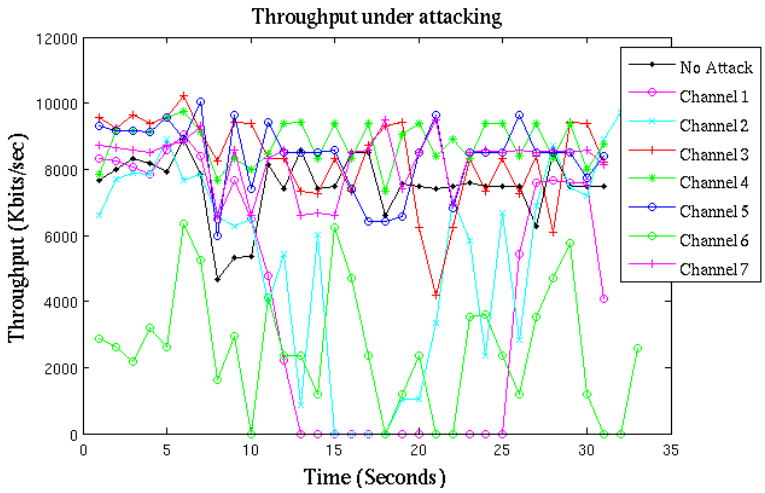

Figure 4.27 Throughput under attacking

4.4.4 Performance with channel hopping and under attacking

4.4.4.1 Fixed Channel jamming

In this part we consider the scenario of fixed channel jamming in which the wireless networks will use channel hopping scheme and jammer will be fixed into one channel. The previous environment will be used in this test. The wireless network will use the scheme of random channel hopping with different hopping frequency. The channel switch interval will be set larger (1 second) in order to see the hopping effect of different scenarios.

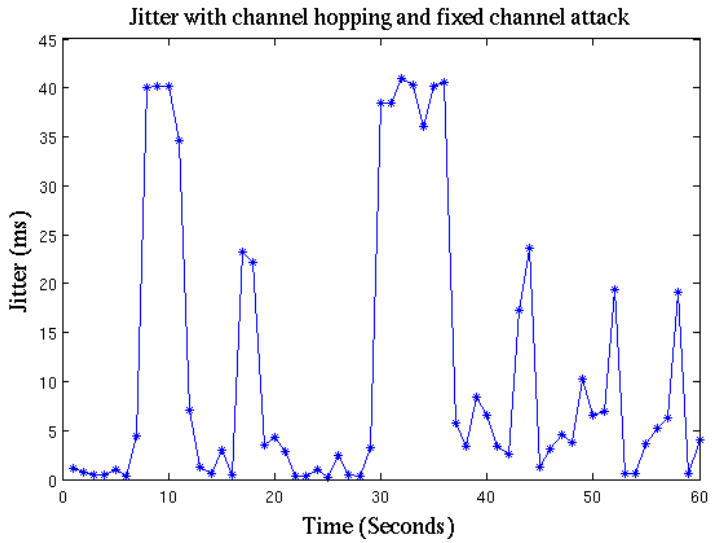

Figure 4.28 Jitter with channel hopping and fixed channel attack

The figure 4.28 shows the jitters of the always hopping wireless network under the fixed channel attacker. When the wireless network switched to the same channel as the attacker, the jitter may be as long as 40 milliseconds. There are also some times where the jitter is up to 20 milliseconds. The reason is the wireless networks switched to neighbor channels of the attacker. In this case, the wireless can still communicate with each other with longer delay.

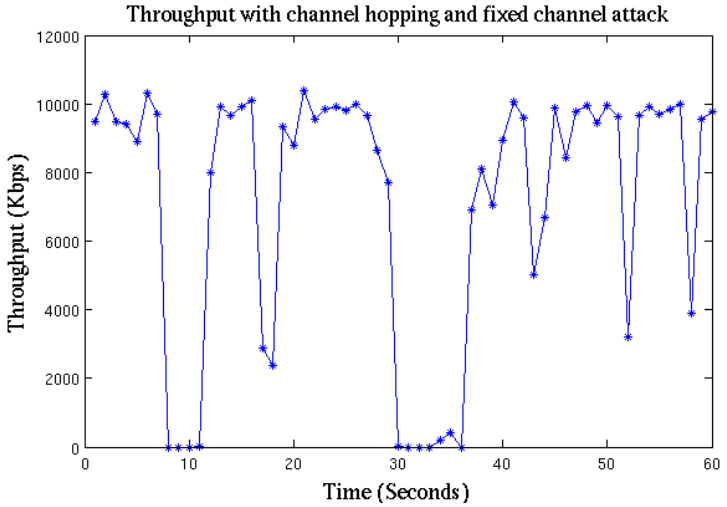

Figure 4.29 Throughput with channel hopping and fixed channel attack

The explanation in of the throughput curve in figure 4.29 is similar to the jitter curve. When the wireless network switched to the same channel as the attacker, the throughput dropped to zero. While the wireless network switched to the neighbor channel of the attacker, the throughput had an obvious decrease.

4.4.4.2 Attack detection and fixed channel jamming

Through the analysis of the above fixed channel attacker, it is effective to use a better strategy other than the always channel hopping one. Once finding out the attacked channel, the wireless network may avoid this channel and use the uninfluenced channels for the following communication. In the test, the UDP data rate of 10 Mbps is used to transmit data from sender to receiver. Attacker is fixed in the channel 1 all over the test to prevent any communication in the channel. The sender and receiver use the channel hopping method before detecting the attacker's channel. In this period, the throughout and jitter delay are not stable. When the channel of sender and receiver is closed to the attacker's channel, the throughput will drop and the delay will be longer. However, once the sender and receiver jump to the attacker's channel, they will get a long delay and zero throughput. At this time, the network will find a safe channel which is far away from the attacked channel. In our test, the network switched to channel 11 which had the longest distance from channel 1, and kept the channel unchanging. After that time, the network throughput is stable as long as the attacker doesn't change its channel. The jitter and throughput over the whole process are shown in the figure 4.30 and 4.31. The sender and receiver switched to the attacker's channel in the 7^{th} second and got zero throughputs. After 9^{th} second, the network switched to and stayed at channel 11 and got unaffected.

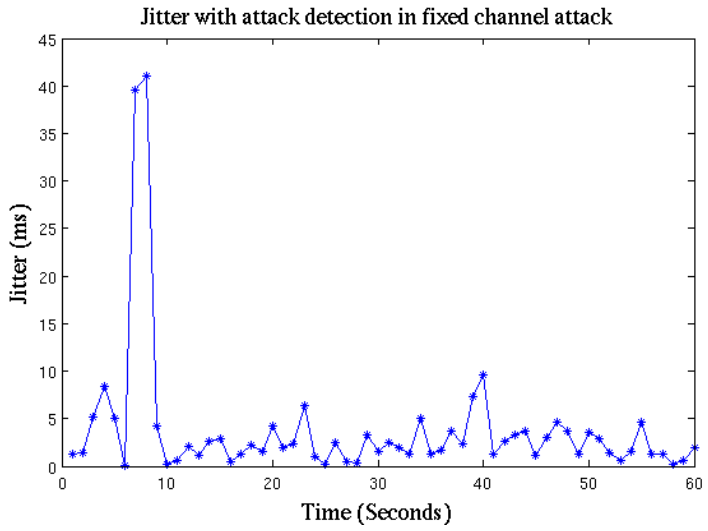

Figure 4.30 Jitter with attack detection in fixed channel attack

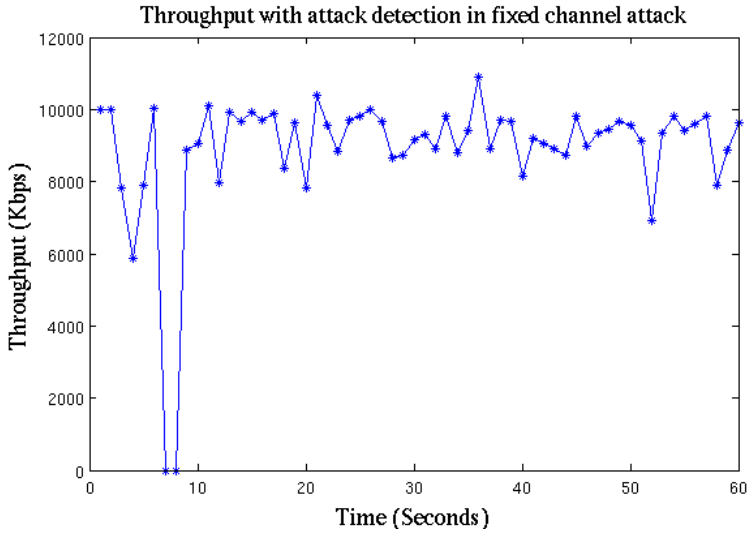

Figure 4.31 Throughput with attack detectio in fixed channel attack

4.4.4.3 Smart jamming attack

In this scenario, the jammer scans for the AP for a period of S seconds (via AP's beacons) and jams the network for J seconds after find the AP. When AP is admitting a channel hopping strategy, it hops channels every U seconds. U is the channel residence time. We also compare the performance between the upper layer implementation and our MAC layer implementation.

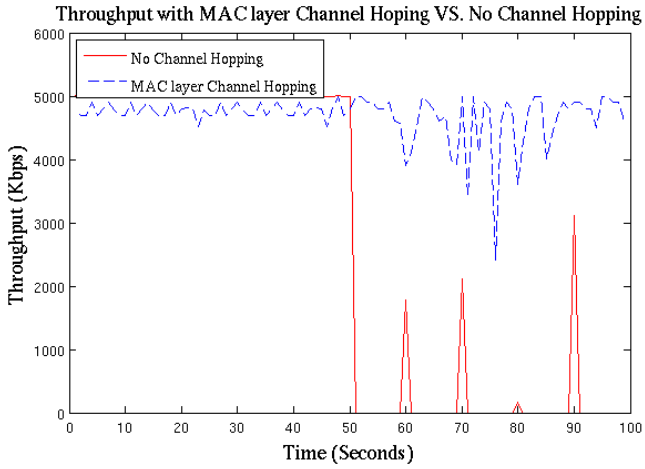

Figure 4.32 Throughput with MAC layer channel hopping VS. No channel hopping

Figure 4.32 shows the comparison of throughput between No Channel Hopping and MAC Layer Channel Hopping under the smart jammer attack with S=400ms and J=10s. Once the jammer is started, the throughput in the network with no channel hopping drops to zero very quickly. The spike every ten seconds is due to the scanning period of the jammer. While the throughput in the MAC layer channel hopping network does have some decrease which is caused by the channel hopping. The throughput is good enough for the communication.

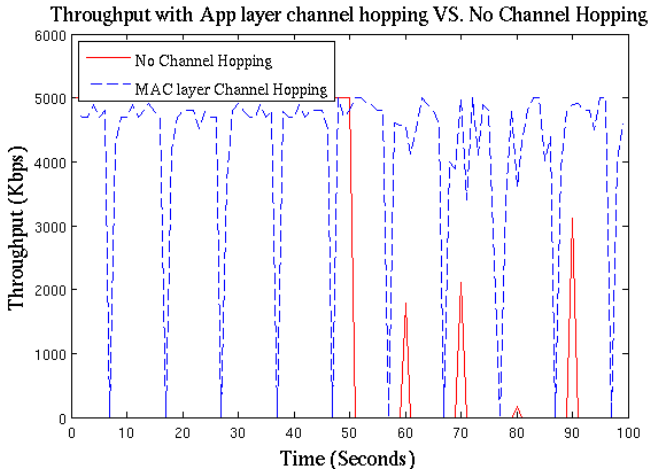

Figure 4.33 Throughput with App layer channel hopping VS. No channel hopping

Figure 4.33 shows the comparison of throughput between No Channel Hopping and Application Layer Channel Hopping under the smart jammer attack with S=400ms and J=10s. The Application Layer Channel Hopping can also mitigate the jamming attacks effectively. However, it cost more time on hopping from one channel to another compared to the MAC layer channel hopping scheme.

During our tests, we also noticed that different parameter selections play a role to the system performance. We further tested the system performance under different parameter selections when the network is under no attack and under attack. We selected channel residence time U from 200 ms to 1000 ms in a step of 200ms. The corresponding 100-second time-averages are presented in the following tables (R = 5 Mb/s). The throughput is calculated by the ratio of throughput with channel hopping and the maximum throughput without channel hopping and jamming attack.

Table 4.2 Normalized throughput with MAC channel hoppingwithout Jammer

Channel Resident Time (ms)	200	400	600	800	1000
Throughput	90%	85%	96%	98%	99%

Normalized Throughput with MAC Channel Hopping without Jammer ()

Table 4.3 Normalized Throughput with MAC Channel Hopping with Jammer

Channel Resident Time (ms)	200	400	600	800	1000
Throughput	60%	71%	96%	98%	99%

Table 4.2 and 4.3 show the throughput of MAC layer channel hopping with/without the jammer. When the channel resident time is around 1 second, the throughput may reach to the maximum one which is closed to the throughput without channel hopping and attacks.

Table 4.4 Normalized Throughput for Upper Layer Channel Hopping without Jammer

Channel Resident Time(s)	Data rate			
	10	5	2	1
10	80%	78%	75%	84%
7	55%	66%	66%	78%
5	65%	62%	58%	64%
3	46%	50%	43%	56%

Table 4.5 Normalized Throughput for Upper Layer Channel Hopping with Jammer

Channel Resident Time (s)	Data rate			
	10	5	2	1
10	55%	68%	70%	76%
7	50%	65%	65%	73%
5	48%	60%	60%	62%
3	35%	44%	46%	47%

Table 4.4 and 4.5 show the throughput of Application layer channel hopping with/without the jammer. The throughput may only reach to the 80% of the throughput without channel hopping and attacks even the channel resident time is set to 10 seconds.

4.5 Summary

In this chapter, some related attack mitigation schemes were reviewed. After that, our own 802.11 wireless network attack mitigation schemes were proposed. The New Management Frames were depicted and applied to our schemes. We also implemented our new schemes and put them into the experiments which showed their attack mitigation effects and advantages compared to the application attack mitigation.

CHAPTER 5

ADVANCED WI-FI DOS ATTACK MITIGATION STRATEGY

The basic Wi-Fi DoS attack mitigation strategy may decrease the attacking influence and increase the throughput. However, the basic strategies cannot work perfectly in special network conditions, for example, when the network channel is completely broken or the multi-hop wireless network is under attack. In this chapter, we will consider the network conditions and propose our advanced attack mitigation strategies.

5.1 Advanced Channel Hopping Strategy

In previous part the basic channel hopping scheme are proposed and implemented to mitigate wireless attack. The advanced channel hopping strategies which are based on the normal channel hopping strategy are analyzed and proposed in this part.

5.1.1 Two-Phase switch

5.1.1.1 Two-phase commit and two-phase channel switch

There is a problem in our original scheme of switching channel. The collaboration among the wireless nodes is not enough. The jamming attacks may cause the wireless networks unstable, either high delay or completely disconnected. As a result, there is no guarantee that the nodes may effectively communicate with each other and switch channel together.

“Two-phase Switch” protocol will be used to enhance the wireless nodes collaboration in the wireless network. The “Two-phase switch” protocol is inspired by the “Two-phase Commit”

protocol which is used to make all nodes in a distributed system agree to commit a transaction. “Two-phase Commit” protocol results in all nodes either commit the transaction or abort committing, even in the case of network failures or node failures, thus achieving ACID properties in distributed transactions.

In our previous scheme, the nodes in the wireless network will switch channel once one wireless node has detected the attackers and broadcast the message. There are some drawbacks of the scheme. For example, the wireless nodes who received the message will switch to the new channel and other nodes which didn't receive the under jamming message due to network problem or other reasons were not able to perform the switch channel action. As a result, there are possibly that only part of the wireless nodes in the network switch to the new channel and other nodes still keep in the original channel and lose communication with other nodes.

5.1.1.2 Two-phase Switch algorithm

For the two-phase switch protocol, we will add another node status, which is called “ready to switch”. The wireless nodes which are in the status of “ready to switch” may switch to the new channel once they get the “switch channel” command. They also can cancel the switch if the command they get is “cancel switch”.

Also, a special wireless node called coordinator will appear in the new protocol. In the 802.11 infrastructure network, the access point is selected to be the coordinator because the infrastructure wireless network is a center controlled network with the center of the access point. It is easier to use the access point to control the wireless network. In the 802.11 ad-hoc network, each node is equivalent. The coordinator is assigned manually at the network initial stage or elected automatically among the wireless nodes through proper communication. There are

several election algorithm may be adopted, for example, Bully Algorithm, ring algorithm, and so on. The detailed methods will be analyzed later.

(1) Successful channel switch

The coordinator will be in charge of detecting the jamming attack and controlling the whole process of channel switch, including whether switch channel or cancel the switch. The two-phase channel switch protocol works as in figure.5.1

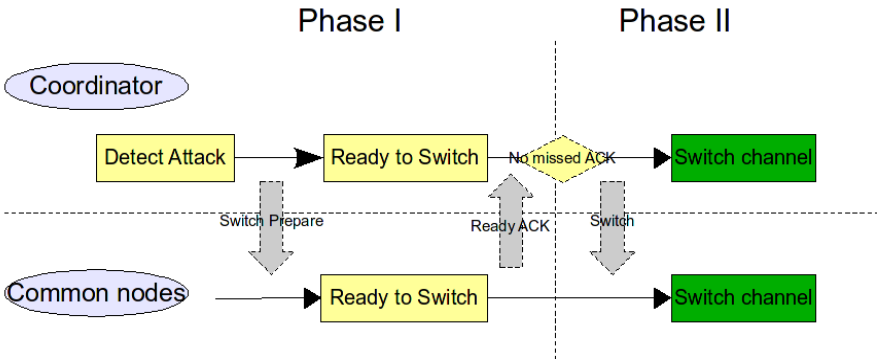

Figure 5.1 Two phase channel switch

An integrated wireless network includes the coordinator and the common wireless nodes which can have common data communication before the appearance of the attackers. The coordinator takes charge of detecting the jamming attack when the attackers come into the wireless network. After detecting the attack, the coordinator notifies other common nodes the attack by broadcasting the "Switch Prepare" message. Then the coordinator goes into the status of "ready to switch". Once the common wireless nodes receive the message of "Switch Prepare", they will know the current network is under attack. Then the common nodes will go to the "ready to switch" status and reply the "Ready ACK" message to the coordinator immediately. This is the Phase I (the phase of "Ready to Switch").

The coordinator will count the number of “Ready ACK” message received from other wireless nodes. When the “Ready ACK” message from all other wireless nodes are received by the coordinator in a fixed time period, the coordinator will send the channel “switch” command to all the other nodes and start the channel switching operation. Once other nodes receive the channel switch command, they will switch channel immediately. This way, the whole wireless network will communicate in a new channel with less jamming attack effects. This is Phase II (the real channel switch).

(2) Fail channel switch

On the other hand, if the coordinator cannot receive all the “Ready ACK” message and encounter a time out, the coordinator will cancel the channel switch operation by broadcasting the notify message, “Cancel Switch”, to other wireless nodes. Other nodes will cancel the channel switch and go to normal status when receiving the notify message. In this case, all the wireless network nodes still keep in the old channel and wait for the next schedule. Figure 5.2 shows this “fail channel switch” due to the lost “Ready ACK”.

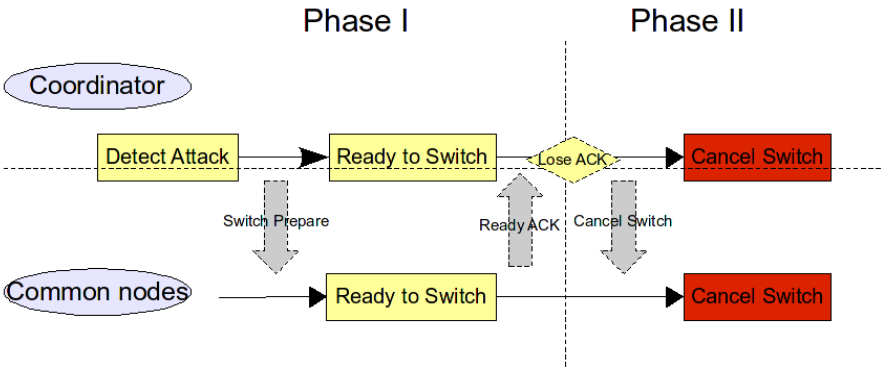

Figure 5.2 Fail channel switch

5.1.1.3 Two-phase Switch assumption

We have some assumptions in the two-phase channel switch algorithm. First of all, the coordinator knows the total number of the wireless nodes and can identify each node by communication. As a result, the coordinator may count the number of received ACKs from wireless nodes and decide whether all the wireless nodes can receive and reply the channel switch message. Actually, this assumption is easy to satisfy. For the wireless infrastructure network, each wireless station will exchange its information with the access point during the authentication and association stage. The access point will save the information of stations in the wireless network. As a result, the coordinator/AP may easily count the number of wireless stations connecting to it and identify each station by their specific information (e.g., MAC address). For the normal wireless ad-hoc network, each wireless node may not know the number of nodes exist in the network. It is more complicated than the Infrastructure network in our algorithm. The coordinator need to be elected manually or assigned automatically. Then other wireless nodes need to send their identification to the coordinator node during the setup period of the network. The specific method or technical skill will be elaborated later.

Second, the coordinator needs the smart update information of the wireless network. One of the advantages in wireless network is the scalability. The extension and reduction of the wireless network is very common. As a result, the coordinator needs the update information of the wireless network timely, which includes the current number of wireless node the specific information of each node, in order to control the whole wireless network switch channel together. This assumption may be satisfied by some specified communication between the coordinator and the wireless nodes. No matter in the Infrastructure wireless network or in the Ad-hoc wireless network, each wireless node should send the frame every small interval in order to keep

connection with the coordinator after join the wireless network. In our previous design and implementation, in fact, we have already implemented the “keep connection” algorithm in the application layer with the purpose of detecting the attacks. This time, we will use the “keep connection” in the MAC layer in which the “keep connection” frame will be analyzed. When the coordinator cannot receive the “keep connection” frame from some node, it will consider the node is out of connection. The detailed method will be explained during the implementation.

Third, the time interval between phase I and phase II is very small. The wireless nodes will send the “Ready ACK” immediately when they receive the “Switch Prepare” message. We ignore the network status change for each node from Phase I to Phase II. This means, if the network connection is OK in Phase I, it should still be OK in Phase II for an specified wireless node, and vice versa, since the interval between Phase I and Phase II is very small. In this way, we assume the common wireless nodes may receive the “Switch” channel message if they can receive the “Switch Prepare” and reply the “Ready ACK” messages. This assumption needs the small interval between phase I and phase II which includes the time period from time of sending the “switch prepare” frame to the time of sending “switch channel” frame. Most of the time period is spent in waiting for the response (Ready ACK) from other nodes. The method to satisfy this assumption is to make nodes reply quickly if they can receive the “switch prepare” frame or determine a proper threshold for the waiting timeout.

5.1.1.4 Three phase committing

In the two-phase commit protocol, the coordinator sends a prepare message to all participants (nodes) and waits for their answers. The coordinator then sends their answers to all other sites. Every participant waits for these answers from the coordinator before committing to or aborting the transaction. If committing, the coordinator records this into a log and sends a

commit message to all participants. If for any reason a participant aborts the process, the coordinator sends a rollback message and the transaction is undone using the log file created earlier. The advantages of this are all participants reach a decision consistently.

However, the two-phase commit protocol also has limitations in that it is a blocking protocol. For example, participants will block resource processes while waiting for a message from the coordinator. If for any reason this fails, the participant will continue to wait and may never resolve its transaction. Therefore the resource could be blocked indefinitely. On the other hand, a coordinator will also block resources while waiting for replies from participants. In this case, a coordinator can also block indefinitely if no acknowledgement is received from the participant.

In our two-phase switch protocol, if the coordinator cannot collect all the "Ready ACK" from the wireless nodes, the switch process may be blocked until we add the waiting timeout to the protocol. Even now, we still cannot solve the channel switch problem, since the channel switch process is canceled and the wireless network is still suffering the attacks.

An alternative way to the two-phase commit protocol used by many database systems is the three-phase commit. The three-phase commit is a non-blocking protocol which was developed to avoid the failures occurred in two-phase commit transactions.

As with the two-phase commit, the three-phase also has a coordinator who initiates and coordinates the transaction. However, the three-phase protocol introduces a third phase called the pre-commit. The aim of this is to 'remove the uncertainty period for participants that have committed and are waiting for the global abort or commit message from the coordinator. When receiving a pre-commit message, participants know that all others have voted to commit. If a pre-

commit message has not been received the participant will abort and release any blocked resources.

Let's consider our wireless network channel switch scheme right now. In the “two-phase switch” scheme, the coordinator needs to wait for the “Ready ACK” from other nodes, and the wireless network may have the “fail channel switch” if the coordinator cannot receive all the “Ready ACK”. So can we use the “three-phase switch” by utilizing the idea of “three-phase commit”?

The advantage of “three-phase commit” is to remove the blocking status of the coordinator. However, it does not suit for our problem in the wireless network. In our “two-phase switch” protocol, the blocking status does not exist. The only question is the wireless network cannot communicate to switch channel together since the network may be heavily destroyed and the wireless nodes can't receive the notify message from the coordinator. The “three-phase” protocol cannot resolve the problem and we will continue the “two-phase switch” scheme. The problem of “fail channel switch” will be solved by the method bellowing.

5.1.2 Scan and join

5.1.2.1 Basic idea of scan and join

The above “two-phase switch” protocol cannot solve the jamming attack problem. It cannot alleviate the attacks or just move part of the nodes to a new channel and keeps others under the attacks if the communication between the coordinator and other wireless nodes is interrupted due to the network problem. Here we propose another algorithm which is called “Scan and Join” scheme.

The “Scan and Join” scheme can be applied to the wireless network when the “two-phase switch” protocol is failed to switch the wireless network to a new channel. The basic idea of

"Scan and Join" is that, after the coordinator switches the new channel the wireless nodes scan all the channels from 1 to 11 to determine the channel of the coordinator and then join the wireless network in the new channel.

5.1.2.2 Apply Scan and Join

The "Scan and Join" scheme can be applied either to work as an independent method or to work together with the "two-phase switch" algorithm. We will explain the use of "Scan and Join" scheme by the following three scenarios.

(1) Scan and join after failed switch

In this scenario, the coordinator and the wireless nodes will use the "two-phase switch" scheme with high priority when the coordinator detected the attack. If the "two-phase switch" scheme works, the wireless network will switch to a new channel. Otherwise, if the "two-phase switch" scheme encounters fail switch, the coordinator will cancel the channel switch.

Then the coordinator may repeat the "two-phase switch" process. If the repeat switch is still failed, the coordinator may have the option to switch channel in advance. Then other wireless nodes will detect the failure of current network, scan the new channel of the coordinator and join the wireless network in the new channel. Figure 5.3 shows the whole process of scan and join after failed channel switch.

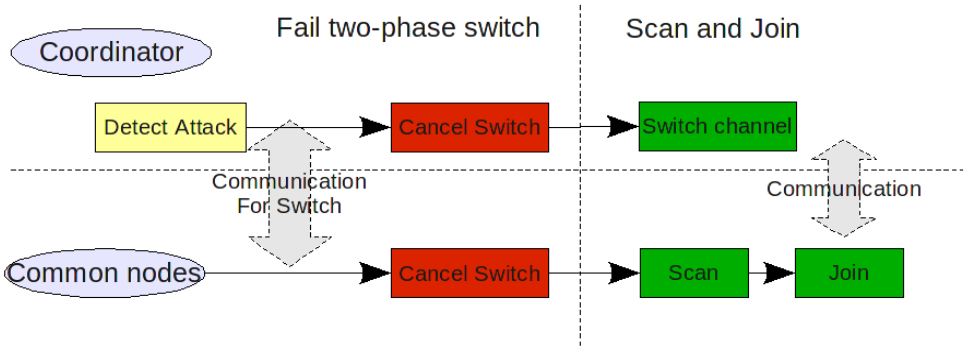

Figure 5.3 Scan and Join process

(2) Scan and join after partly switch

This scenario is similar to the first one. First of all, the coordinator will send the “channel switch” packets to the wireless nodes after detecting the attacks. The difference from the first scenario is that without waiting for the “ACK” packets, the coordinator switch to the new channel directly. The other wireless nodes will not send the “ACK” packets any more. Those nodes who receive the “switch message” will switch to the new channel immediately.

However, there are possibly that part of the wireless nodes cannot receive the “channel switch” message due to the network problem which may be caused by the attack. In this scheme, those wireless nodes who cannot receive the switch message will scan all the channels to find the coordinator in the new channel and join the network after detecting the network failure.

One of the advantages of this method is that we discard the “two-phase switch” method where the coordinator only send one switch message to other nodes and does not wait for the “ACK” message. In this circumstance, the scheme does not need the assumption that the time interval between phase I and phase II should be very small to keep the consistency between the two phases. The second advantage is that the method may guarantee the establishment of new wireless network in the new channel as soon as possible. The wireless nodes who receive the “switch channel” message may join the new network in very short time period. Other nodes may join the network later when they detect the network failure with certain delay. The process of the scenario is depicted in figure 5.4.

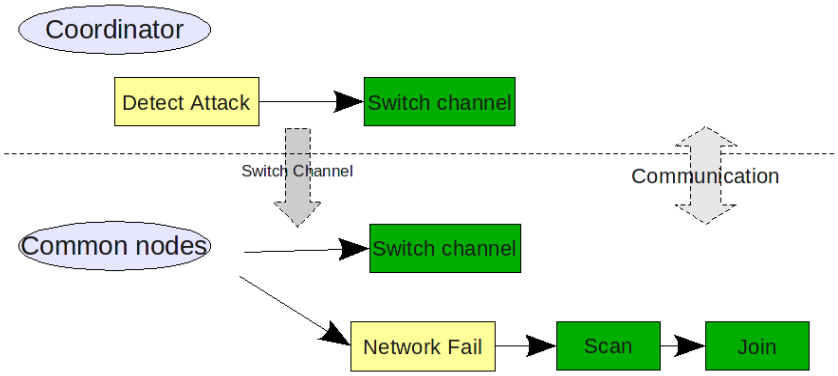

Figure 5.4 Scan and join with partly switch

(3) Independent scan and join

The third scenario, independent scan join, is the simplest channel switch method. The coordinator will switch channel immediately without notifying other wireless nodes once detecting the attacks. All of other nodes will detect the network fail by themselves and then scan the channel of coordinator and join the new network. Figure 5.5 shows the basic idea of independent scanning and joining operations. The advantage of the method is to keep the scheme as simple as possible.

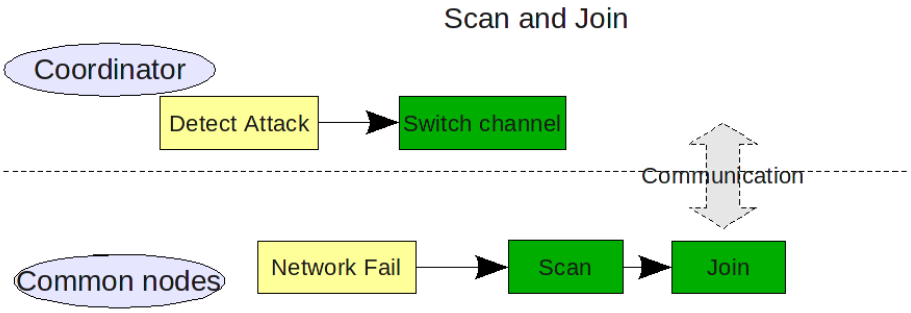

Figure 5.5 Independent Scan and Join

5.1.2.3 Feasibility of scan and join

The “scan and join” scheme is easy to be implemented, since the normal 802.11 network already contains most of the operations used by our scheme. In the normal wireless network, the wireless node will connect to the access point by starting with the scan process. The scan process may be either active or passive.

During the active scan process, the 802.11 device broadcasts an 802.11 probe request on the channel it is scanning on. The driver uses the zero-length broadcast SSID in the probe request. The driver will add any received 802.11 beacons or probe responses to its cached BSSID scan list. During the passive scan process, the 802.11 device does not send any 802.11 probe request. Instead, it dwells on each channel for a period of time and adds any received 802.11 beacons or probe responses to its cached BSSID scan list. The cached BSSID scan list is used for the connections.

After either the active scan or the passive scan, the 802.11 device will get the information of the access point or the wireless network. The information may include the SSID, channel, transmit power, authentication, encryption, and so on. After that, the wireless device may initiate the authentication and association process with the access point and then join the network.

Borrowing the method from the procedure of 802.11 connections, the “scan and join” scheme is easy to be implemented. The simplest method for the wireless stations is to scan the beacon sent from the access point and then switch to the same channel and join the network. The re-association process may be not required after modification to the protocol in order to reduce the channel switch time. The access point may save the association information of the wireless stations and will not need the re-association once the wireless stations have joined the network.

5.1.3 How to determine the new channel

We discussed the “two-phase switch” and “scan and join” schemes in the above sections. However, there is a question in the schemes. How could the network determine the new wireless channels to switch to? The question may be easy to be solved in the “scan and join” scheme. Only the access point or coordinator needs to determine the new channel. Other wireless nodes will scan all the channels and jump to the new channel after finding out the coordinator's channel. The channel determining in the “two-phase switch” scheme is complicated. Both the coordinator and common wireless nodes needs to know the new channel before channel switching.

Let's take the “two-phase switch” scheme for example. The coordinator will determine the next channel to switch once it detects the attacks. The new channel may be calculated by the coordinator according to the current wireless channel or the current distribution around the coordinator. The new channel may also be generated randomly by the coordinator or read from the configuration file which is assigned manually at the initialization of the wireless network.

The coordinator will send the channel information to other wireless nodes. The channel information, which contains the next channel the whole network should jump to, may be comprised as part of the “switch prepare” packet. After receiving the “switch prepare” packets, the wireless nodes will analyze the packets and get the information of next channel in order to switch once receiving the command of “switch”.

Figure 5.6 shows the “two-phase switch” process which includes the channel information exchange.

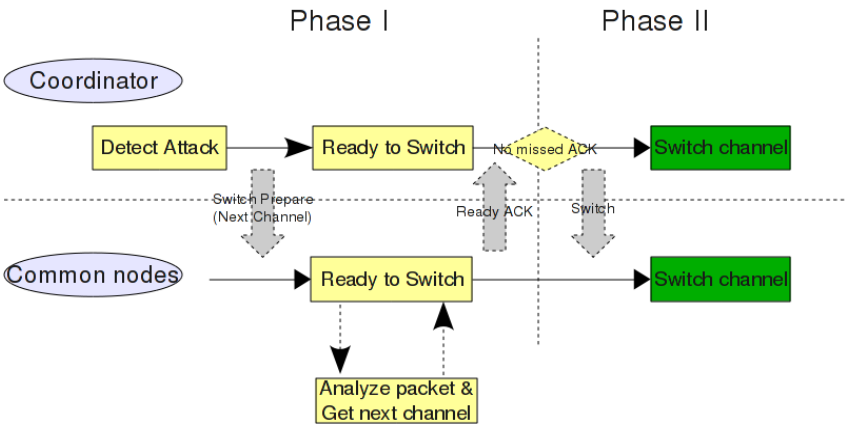

Figure 5.6 Channel information exchange

5.1.4 Effectiveness evaluation

Theoretically, in the wireless domain, a jammer with unlimited resources can always successfully jam any wireless transmission by flooding the entire spectrum that could be used by the client, making the resistance to the jamming impossible.

However, most of the times, the jammer may be restricted by the hardware configuration which is similar to a legitimate client. For instance, in many situations, the equipment used to jam would be visible to the legitimate participants and it should not be special equipment. To remain inconspicuous, the jammer would need to jam with conventional 802.11 wireless devices, such as a laptop or other small computer with one or two wireless interfaces. Under the hardware constraints, the jammer has limited ability to jam the entire spectrum and the feasibility of mitigating the jamming attacks could be increased. We would have the assumption throughout our projects that the jammers have limited resources and limited transmitted powers as common wireless devices.

The successful two-phase channel switch scheme is on the base of the successful message communication. The coordinators should be able to exchange messages with other wireless

stations, although the network delay may be large due to the DoS attacks. We can see from the discussion of above paragraph that most of the jammers cannot destroy the wireless network completely with limited jamming resources and transmitted power except that the network delay may be long. During such circumstance, the wireless devices still can hear each other and the two-phase channel switch scheme should be effective if we select proper timeout threshold or try enough times.

The two-phase switch scheme also has some assumptions, one of which is that the time interval between phase I and phase II should be very small. This assumption cannot be satisfied when the network delay is large due to the DoS attacks. As we mentioned in the first chapter, the coordinator will send the "switch-prepare" message, wait for the "ACK" message from other wireless nodes and then send the "switch channel" message. What we focus on is the time period between the "switch-prepare" message and "switch channel" message sent by the coordinator. We assume this time period is very small and the network status keeps the same during the time interval. As a result, when the network delay is large and we select the large timeout threshold, the two-phase switch scheme assumptions may not be satisfied. The standalone two-phase switch scheme may be ineffective or failed during the heavy DoS attack which may lead to the large network time delay.

The "scan and join" scheme will be successful if the DoS attack leads to large network delay or even disconnection. The coordinator will switch the new channel after detect the attacks. Then other wireless stations will fail to connect to the coordinator and switch to the same new channel according to some algorithm. The only question is that the wireless network cannot switch to the new safe channel if the attackers could jam all the channels which may be used by the wireless network. However in this research, we consider that the jammers are limited to the

resources and power and could not jam all the channels at the same time. We will think that the "scan and join" scheme is effective.

The "always-on prevention" scheme is dependent on the packets communication between the coordinator and other wireless nodes. The purpose of the scheme is to prevent the attacks on the current communicating channel. However, if the attackers can destroy one channel completely, and the channel is coincidently used by the wireless network, the "always-on prevention" scheme will be broken by the attackers. The probability of such failure is high especially in the 802.11 b/g networks which only have 3 orthogonal channels.

5.2 RS Controlled Wi-Fi Attack Mitigation

In this part we talk about the new jamming attack detection and mitigation scheme which needs to add an extra wireless device to the current network. The extra device takes charge of jamming detection and controls the wireless network. The new scheme will not change the configuration of the current wireless network. The overall jamming attack detection and mitigation scheme are described in figure 5.7:

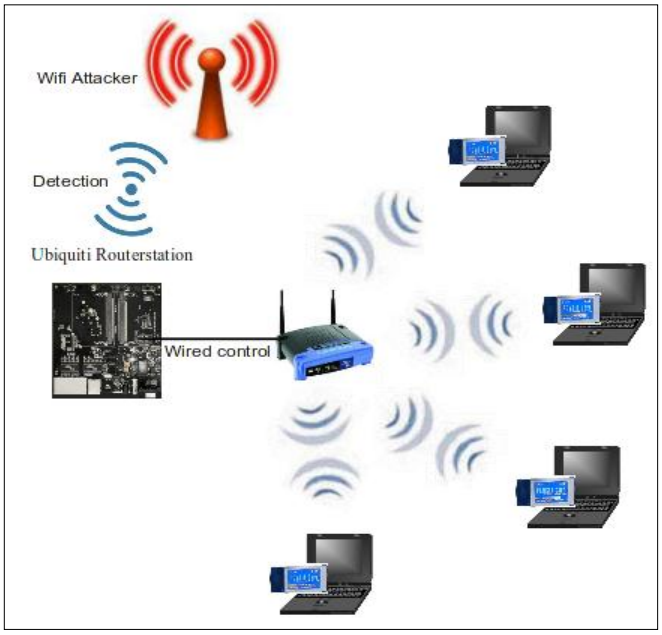

Figure 5.7 RS controlled attack mitigation

The RouterStation runs OpenWRT's Kamikaze 8.09.2 [6] with Linux kernel 2.6.28.9 and MadWifi driver for 802.11 communications. The Ubiquiti Routerstation is the core device which takes charge of the role of jamming attack detection. Our implementation will be focus on the Routerstation. After detecting the attacks, the Routerstation controls the access point in the network to switch to a safe channel through Common Gateway Interface (CGI). The wireless stations in the network don't need to join the attack detection process. Instead, the wireless stations only need to follow the access point's channel to continue their communication.

The advantage of this jamming attack detection and mitigation scheme is easy-to-use. In this scheme, the existing wireless networks don't need to be modified or upgraded to support jamming attack detection and mitigation. The only thing needs to do is adding the Routerstation to the current wireless network by connecting it to the access point through Ethernet interface.

The embedded Linux router implementation requires the router running Linux, which may not be always available to the user. We designed an advanced implementation, which can be

used by the user running any COTS routers, including both Linux and non-Linux routers. To perform the jammer detection and upper-layer implementation automatically and seamlessly, we implemented a Monitoring Box based on a RouterStation. The main functionality of the Monitoring Box is to detect the jammer and change the WiFi channel of the COTS router if necessary. Also, the Box can be configured to periodically change the channel of target Access Point (AP). In this way, a proactive hopping strategy can be achieved. The box does not provide any network services. Instead, the box monitors the quality of target 802.11 network. Once the quality is bad, the box controls the target router to switch channel. The box uses a modified MadWifi device to monitor the quality of target 802.11 network. The box uses the Ethernet interface to connect the target commercial router and controls the target router through the URL interface provided by the commercial router

The main purpose of this development is to detect jamming attacks in Madwifi driver and return relevant results to application layer through IOCTL. Then the application layer will call the command of “iwpriv” to get network information from low level. Then the application layer controls the access point to switch channel through CGI once it detects the attacks. This part will focus on the function of attack detection, sending the attack message through IOCTL, and how to implement the functions in Madwifi driver.

Our test bed consists of 1 Ubiquiti Networks RouterStations, 1 Linksys WRT54G access point, and 2 laptops with wireless adapters connected to the access point. The RouterStation has AR71xx MIPS architecture and is equipped with Atheros 5212 WiFi adapters through their miniPCI slots. The wireless adapter takes charge of detecting the attack. Then the RouterStation will control the access point to switch to the safe channel together with the two laptops. During our test, the whole attack detection and channel switch process can be completed smoothly.

Since the communication between the RouterStation and Access Point is through the CGI and the channel switch in the Access Point and related wireless adapters is controlled by the Application layer, the time duration for the whole process is as long as several seconds. We only proposed idea and proved its possibility in this chapter. The performance study and strategy improvement will be left to the further works.

5.3 DoS Attack Mitigation in Multi-hop Wireless Network

There are many research works about the attack mitigation in the multi-hop wireless networks. Khalil present the LiteWorp [45] method to detect and isolate wormhole attacks in static multihop wireless networks. The MobiWorp countermeasure was presented in [43] to mitigate the wormhole attack in mobile networks. Khalil also proposed the UnMask framework that mitigates the attacks (such as wormholes, rushing, Sybil attacks, etc.) by detecting, diagnosing, and isolating the malicious nodes. The distributed denial of service (DDoS) attack weas studied and an anomaly detection scheme working on each mesh router for congestion-based DDoS attacks in WMNs was proposed in [46].

The multi-hop wireless networks, such as wireless mesh network or wireless multi-hop ad-hoc network, are more complicated than the single hop infrastructure or ad-hoc network. Multi-hop wireless network provide large area network coverage and becomes popular with the rapid growth of wireless communication technologies. The wireless users are allowed to connect to the internet or intranet anywhere in the range of the multi-hop networks in theory. However in practice, the multi-hop wireless networks are easier to suffer the disconnection and DoS attacks due to its wide range distribution and the need of longer distance of packets transmission. Especially in the Mobile wireless network, the wireless nodes may easily lose connections with their neighbors.

Despite the wide distribution of multi-hop ad-hoc network results in the high possibility of suffering attack, it is hard for a single attacker to jam the whole multi-hop network at one time. The attack area changes with the attacker moves from one place to another. When the attackers start, only part of the multi-hop network is influenced and the other parts may have normal network communication. However, in the traditional TCP or UDP network, the packets need to re-transmit from source to destination even with the failure of one hop. The drawbacks are not that obvious in a regular with high throughput and low network delay. But when in the network where the attacker exists, the low efficiency of packet retransmission require a new scheme for the multi-hop ad-hoc network under certain attacks since the frequent network disconnection and long network delay or timeout.

The DTN (Delay-Tolerant Networking or Disruption-Tolerant Networking) network provides an effective scheme to improve the network status in the network with long delay or frequent disruption. In the DTN network, the data packets are encapsulated into bundles and send from source to destination hop by hop instead end-to-end. After the bundles are received in each hop, they will be saved in the permanent storage media and sent to the next hop. Even if the send is failed, TCP is timeout or even the node is rebooted, the bundles may be sent next time when the network resumes. The DTN scheme avoids the packet retransmission from source to destination.

There are several DTN distributions available, e.g., DTN2 (current version 2.7), IBR_DTN (current version 0.6.0), ION, Postellation, DASM and so on. DTN2 is a reference implementation of the bundle protocol by DTNRG. It provides a robust and flexible software framework for experimentation, extension, and real-world deployment. Besides the basic bundle protocol, DTN2 also support a bunch of features, such as SDNV (Self Describing Numeric

Values), reactive bundle fragmentation, bundle fragmentation reassembly, bundle security blocks and so on. DTN2 is the main implementation to demonstrate the basic functionality and robustness. However it is not optimized for the embedded system and doesn't operate very efficiently [37]. The size of daemon application will take 21.9M and the RAM usage (swappable VZS) will take 41.7M. It is not a good choice for embedded systems.

IBR-DTN is a C++ implementation which aims to be portable, light and extensible. It is a small, efficient, but powerful implementation that can be run on both standard Linux system and embedded devices. According to [37], IBR-DTN takes out some features of DTN2, e.g., registration, persistent storage of bundles, reactive bundle fragmentation and bundle security block. At the meantime, IBR-DTN consumes far less memory and storage resources than DTN2. Also, the throughput in IBR-DTN is more stable and has an increase of 100% compared to DTN2.

By introducing DTN strategy in the Multi-hop wireless network, the packets don’t need to retransmit end to end. Instead, the packets will retransmit from the hop of last transmitting fail. The strategy is especially useful in the space wireless network where the network delay is long and network disconnection always happens. Another importance of the strategy is to increase the probability of packet delivery.

The DTN network has the overhead of pack the bundles and hop-by-hop TCP connections. The throughput has dropped 30 - 50% compared to the traditional TCP network without jamming attacks DTN network is not so effective in the traditional Wi-Fi wireless networks. The complicated packet process leads to the lower throughput and longer delay. However, the packet delivery rate may be increased in the network under the jamming attacks.

In the traditional TCP network, the packet may be failed to be delivered to the destination which may be caused by the network disconnection or packet corruption. As a result, the packet retransmission from the source to the destination is very normal. In the DTN network, the packet transmission is divided into hop-by-hop and there are seldom end-to-end packets retransmissions exist. After the attacker moves between the source and destination, there is almost no complete TCP connection from the source node to the destination. As a result, the packet delivery rate over TCP is very low. However, the DTN hop-by-hop transmission is a better scheme to deliver the packet to the destination.

5.4 Summary

Some advanced attack mitigation schemes were proposed in this chapter. We also made simple analysis and test to prove the feasibility of our attack mitigation strategies. However, we didn't have deeper research and designed for the advanced strategies. They will be left for the future study and design.

REFERENCES

[1] W. Xu, W. Trappe, Y. Zhang and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks", ACM MobiHoc, May 2005, pp.46-57

[2] U. Varshney, "The status and future of 802.11-based WLANs", IEEE Computer, vol. 36, no. 6, June 2003

[3] "802.11 Vulnerability in Clear Channel Assessment (CCA) Algorithm"

[4] A. D. Wood, J. A. Stankovic and S. H. Son, "JAM: A Jammed-Area Mapping Service for Sensor Networks", Proc. 24th IEEE Intl. Real-Time System Symposium, 2003, pp. 286-297

[5] http://www.ping127001.com/pingpage.htm

[6] Jim Price, Understanding dB "http://www.jimprice.com/prosound/db.htm"

[7] J. Bellardo, S. Savage. "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions". In Proceedings of the USENIX Security Symposium, pages 15-27, August 2003.

[8] V. Gupta, S.V. Krishnamurthy, and M. Faloutsos. "Denial of Service Attacks at the MAC Layer in Wireless Ad Hoc Networks". In Proceedings of MILCOM, 2002.

[9] Y. Xiao, C. Bandela, X. Du, Y. Pan, and K. Dass, "Security Mechanisms, Attacks, and Security Enhancements for the IEEE 802.11 WLANs," International Journal of Wireless and Mobile Computing, Vol. 1, Nos. 3/4, 2006, pp. 276-288.

[10] D. B. Faria and D. R. Cheriton. "DoS and Authentication in Wireless Public Access Networks". In ACM Wireless Security Workshop (WiSe'02), 2002.

[11] IEEE Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[12] IEEE Part 11g: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 2003.

[13] R. Gummadi, D. Wetherall, B. Greenstein, S. Seshan, "Understanding and mitigating the impacts of RF interference on 802.11 networks," SIGCOMM 2007.

[14] E. G. Villegas, E. Lopez-Aguilera, R. Vidal, j. Paradells, “Effect of adjacent-channel interference in IEEE 802.11 WLANs,” The Proceedings of the 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, CrownCom 2007.

[15] Bluetooth coexistence with 802.11, http://tinyurl.com/2grndv.

[16] H. Haas and S. McLaughlin, “A Derivation of the PDF of Adjacent Channel Interference in a Cellular System”, IEEE LCOMM, 2004, pp. 102-104

[17] A. D. Wood and J. A. Stankovic. “Denial of service in sensor networks”. Computer, 35(10):54--62, 2002.

[18] W. Xu, K. Ma, W. Trappe, Y. Zhang, “Jamming Sensor Networks: Attacks and Defense Strategies,” IEEE Network, May/June 2006.

[19] W. Xu, T. Wood, W. Trappe, Y. Zhang, “Channel Surfing and Spatial Retreats: Defenses against wireless Denial of Service,” Proceedings of the Wise 2004.

[20] A. D. Wood, J. A. Stankovic, G. Zhou, “DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks,” accepted to The 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), San Diego, CA, June 2007.

[21] Y. Xiao, H. Chen, B. Sun, R. Wang, and S. Sethi, “MAC Security and Security Overhead Analysis in the IEEE 802.15.4 Wireless Sensor Networks,” EURASIP Journal on Wireless Communications and Networking, vol. 2006, Article ID 93830, 12 pages, 2006. doi:10.1155/WCN/2006/93830.

[22] V. Navda, A. Bohra, S. Ganguly, R. Izmailov, and D. Rubenstein, “Using channel hopping to increase 802.11 resilience to jamming attacks,” In IEEE INFOCOM Mini-symposium, 2007.

[23] P. Kyasanur, N.H. Vaidya, “Detection and Handling of MAC Layer Misbehavior in Wireless Networks”. In Proceedings of Dependable Systems and Networks, 2003.

[24] M. Raya, I. Aad, J-P.Hubaux, A. El Fawal, “DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots,” Proceedings of ACM MobiSys, Boston (MA), USA, 2004.

[25] I. Korn and B. Seth, “Adjacent-Channel and Quadrature-Channel Interference in Minimum Shift Keying”, IEEE 1983.

[26] M. Pajic and R. Mangharam. "Spatio-Temporal Techniques for Anti-Jamming in Embedded Wireless Networks", submitted to EURASIP Journal on Wireless Communications and Networking

[27] IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

[28] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," Communications Surveys & Tutorials, IEEE, vol. 11, no. 4, pp. 42–56, 2009.

[29] IxChariot Website http://www.ixiacom.com/products/ixchariot

[30] R. Muraleedharan, "Jamming Attack Detection and Countermeasures In Wireless Sensor Network Using Ant System", 2006 SPIE Symposium on Defense and Security, Orlando, FL

[31] R. Ramanathan, "On the Performance of Ad Hoc Networks with Beamforming Antennas", October 2001 ACM International Symposium on Mobile Ad Hoc Networking and Comuting

[32] A. Spyropoulos and C. S. Raghavendra, "Energy Efficient Communications in Ad Hoc Networks Using Directional Antennas", IEEE conference on Computer Communications, NY, USA, June 2002

[33] B. S. Manoj and C. S. Muthy, "Transport Layer and Security Protocols for Ad Hoc Wireless Networks", Ad Hoc Wireless Networks: Architectures and Protocols. Prentice Hall PTR, May 2004

[34] A. R. Harish, S. Garigala, B. Raman, and P. Gupta, "Feasibility study of spatial reuse in an 802.11 access network," XXVIII URSI General Assembly, New Delhi, India, 2005.

[35] A. Viterbi, "Spread spectrum communications–Myths and realities," IEEE Communications Magazine, vol. 17, no. 3, pp. 11–18, 1979.

[36] The new Wi-Fi Jammer: WiFi-Bluetooth-Jammer WLJ100 http://www.phantom.co.il/productpage.asp?id=24

[37] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf, "IBR-DTN: an efficient implementation for embedded systems," in Proceedings of the third ACM workshop on Challenged networks, 2008, p. 117–120.

[38] W. A. Arbaugh, N. Shankar, Y. C. J. Wan, and K. Zhang, "Your 80211 wireless network has no clothes," Wireless Communications, IEEE, vol. 9, no. 6, p. 44–51, 2002.

[39] C. He and J. C. Mitchell, "Security analysis and improvements for IEEE 802.11i," In proceedings of the 12th Annual Network and Distributed System Security Symposium, p. 90--110, 2005.

[40] M. Bernaschi, F. Ferreri, and L. Valcamonici, “Access points vulnerabilities to DoS attacks in 802.11 networks,” Wireless Networks, vol. 14, no. 2, p. 159–169, 2008.

[41] M. Demirbas and Y. Song, “An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks,” in A World of Wireless, Mobile and Multimedia Networks, International Symposium on, Los Alamitos, CA, USA, 2006, vol. 0, pp. 564-570.

[42] G. Vigna, S. Gwalani, K. Srinivasan, E. M. Belding-Royer, and R. A. Kemmerer, “An Intrusion Detection Tool for AODV-Based Ad hoc Wireless Networks,” in Computer Security Applications Conference, Annual, Los Alamitos, CA, USA, 2004, vol. 0, pp. 16-27.

[43] I. Khalil, S. Bagchi, and N. B. Shroff, “MobiWorp: Mitigation of the wormhole attack in mobile multihop wireless networks,” Ad Hoc Networks, vol. 6, no. 3, pp. 344-362, May. 2008.

[44] I. Khalil, S. Bagchi, C. N. Rotaru, and N. B. Shroff, “UnMask: Utilizing neighbor monitoring for attack mitigation in multihop wireless sensor networks,” Ad Hoc Networks, vol. 8, no. 2, pp. 148-164, Mar. 2010.

[45] I. Khalil, S. Bagchi, and N. B. Shroff, “LiteWorp: Detection and isolation of the wormhole attack in static multihop wireless networks,” Computer Networks, vol. 51, no. 13, pp. 3750-3772, Sep. 2007.

[46] Y. H. Liu, “The Study of a Congestion-Based DDoS Attack Detection in Wireless Mesh Networks,” 2007.