IEEE 802.11 Denial of Service Attack Detection and Mitigation Techniques

by

Joseph Soryal

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

2013

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

Date	Professor **Tarek N. Saadawi** Chair of Examining Committee
Date	Dean **Ardie Walser** Executive Officer

Professor **Umit Uyar**
The City University of New York – Electrical Engineering Department

Professor **Kaliappa Ravindran**
The City University of New York – Computer Science Department

Professor **Rosario Gennaro**
The City University of New York – Computer Science Department

Professor **Mehmet Ulema**
Manhattan College - Computer Information Systems

Supervision Committee

The City University of New York

Abstract
IEEE 802.11 Denial of Service Attack Detection and Mitigation Techniques

Joseph Soryal
Advisor: Professor Tarek N. Saadawi

The dissertation presents comprehensive detection and mitigation techniques to combat intelligent denial of service (DoS) attacks targeting the Media Access Control Layer operating on IEEE 802.11, which is the most widely used standard in the wireless technology.

My research stems from the following: (1) the attack is fairly easy to implement but difficult to detect by traditional methods, and (2) impact is severe to a point that it would completely disrupt communication in the network. The goal of my research is to effectively detect the attacker without having false positive results and to find a methodology to reduce the negative impact on the network in different environments.

The negative impact of the DoS attack is investigated and analyzed using network simulators. The designed algorithms are validated using mathematical modeling and network simulators to prove the effectiveness and feasibility of the introduced solutions. The solution is successfully applied on three different physical layer technologies; Direct-Sequence Spread Spectrum, Frequency Hopping Spread Spectrum, and Orthogonal frequency-division multiplexing.

The detection technique applied the solution of the two-dimensional Markov Chain model to determine the detection thresholds in fixed and mobile environments. Markov Chain model is extended to cover multiple wireless domains to include the hidden nodes inside the extended network.

Two mitigation techniques using channel hopping are developed to combat the DoS attack in both fixed and mobile environments. The first mitigation technique is designed to operate in a mixed trusted environment, where two groups exist and one group poses a higher trust level than the other one. When a user of the less-trusted group implements the DoS attack, the algorithm will isolate the attacker along with the less-trusted group. The concept of the "Victim" is implemented, which gives the attacker the false feeling that the attack is successful and continuing as opposed to the reality. The second mitigation technique is designed to operate in environments where all users are authenticated and trusted, however one user turned rouge. The algorithm will isolate the attacker via disseminating the new channel by using asymmetric cryptography.

Acknowledgements

First of all, I would like to thank God for giving me wisdom and courage to complete the PhD journey. I would like to thank my advisor and mentor, Professor Tarek Saadawi, for his continuous professional support, technical guidance, and mentoring during my PhD studies and research. Also, I would like to thank all my committee members: Prof. Uyar, Prof Ravi, Prof. Gennaro, Prof. Ulema, and Dean Walser for their guidance and directions.

Finally, I would like to thank my family and friends for their continuous support.

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

1.1 Challenge and Research Area

Denial of Service (DoS) attacks in wireless networks could be implemented by multiple ways [8]. This research addresses a very specific and sophisticated type of DoS attack. The DoS attacker here disguises itself as a legitimate user and pretends to be following the IEEE 802.11 standards and actually following the communication protocol to some extent. That generates properly formatted control and data packets, which make all other innocent nodes in the network consider him as a peer legitimate node. In addition, the attacker presents himself to the network as having valid information to share with everyone. The goal of this research is to present a complete algorithm to detect, identify, isolate, and react to the DoS attacker to mitigate negative impact on the network. The DoS attacker in this research manipulates the back-off timer to illegally maximize the successful chances of transmitting packets that appear to be legitimate data packets but, in fact, do not contain any valid data. The MAC layer protocol used in this research is IEEE 802.11 DCF (Distributed Coordination Function), which is based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) that specifies two methods for packet transmission. The basic method is a two-way handshaking called "Basic Access", in which every node attempts to send a data packet, and if the transmission is successful, the destination node will send an ACK (Acknowledgement) packet to the source node; otherwise the source will keep attempting to retransmit. The second method applies a four-way handshaking mechanism "Request-to-Send/Clear-to-Send (RTS/CTS)." Each node has a packet to transmit will sense the medium first to apply a collision avoidance mechanism. If the node senses the medium and it is

clear, the node will transmit a Request-To-Send (RTS) packet, and that will be followed by a Clear-To-Send (CTS) packet originating from the destination node. A data packet will be sent to the destination node and will send to the source an Acknowledgement (ACK) packet to complete the four-way handshaking mechanism. If the medium was sensed “busy”, each node will back off for a random period and will count down to zero. Once the counter reaches zero, the node will attempt to retransmit the packet.

The focus of this research is the RTS/CTS mechanism. The proposed algorithm consists of a detection module that applies two-dimensional Markov chain modeling to obtain the base detection threshold for the number of successful packet transmissions. The second module of the proposed algorithm, which consists of the DoS attack mitigation technique, applies a novel method for dynamic channel allocation. It fools the attacker to make him believe that he is still attacking the whole network, so the attacker does not get alerted and change his attacking methodology. The presented algorithm would provide the capability to group of trusted nodes to allow foreign nodes to join the group for valid data exchange and at the same time will enable the trusted group to detect and combat the DoS attack if the allowed foreign node is an attacker.

The algorithms developed as part of this research can be easily integrated into commercial wireless routers and all other wireless devices operating on IEEE 802.11 standards to provide an extra security layer and resilience against DoS attacks.

1.2 Technical Background

1.2.1 IEEE 802.11 DCF

IEEE 802.11 DCF [1][2][3] employs Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) methodology to reduce the probability of the collisions of packets in the wireless network. Networks using CSMA/CA experience higher throughput because fewer packets collide. Four-way handshake mechanisms are implemented using four different types of packets (RTS – CTS – ACK – Data). Interframe space (IFS) intervals are the durations specified to be between the packets transmissions. Distributed Interframe Space (DIFS) is the time a node has to wait to attempt to transmit a packet after sensing a busy medium then it calculates a random back-off time. The random back-off time is an integer value that corresponds to a number of time slots (Sigma), which varies depending on the physical layer technology (PHY) used. A Contention Window (CW) is the idle period that follows the DIFS. Since the time is slotted, each station transmits at the beginning of each slot (Sigma). The slot time varies between PHY layer technologies: the smaller the slot time, the higher the transmission rate. The basis of choosing a slot time is the adequate time needed by any node in the network to detect any packet transmission from any other node inside the wireless network. It accounts for the propagation delay, to time the switching from the receiving to the transmitting states and the time needed to signal to the MAC layer the state of the channel, which is Busy Detect Time (BDT).

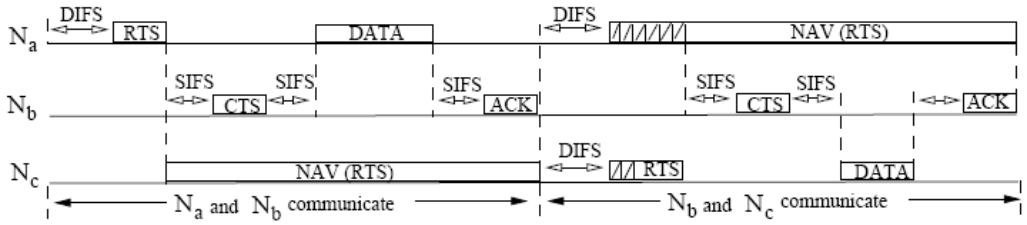

Figure 1-1: IEEE 802.11 DCF

The example in Figure (1-1) shows two nodes Node (a) and Node (c) are contending to reserve the medium for a chance to send a data packet to Node (b). Node (a) waits for DIFS period and senses the channel and finds it idle, and then sends RTS to Node (b). RTS is heard by all stations and those stations that this RTS packet not destined to them would implement a Network Vector Allocation (NAV) value to stay idle to prevent possible collisions. Node (b) waits for SIFS and responds with CTS which is heard by all stations. It signals permission to the sender to send its data packet after a SIFS period and updates the NAV to the other stations in the range. After successful transmission Node (b) confirms receipt by sending ACK to Node (a) to complete the four-way handshake. Node (a) and Node (c) wait for a DIFS period plus the back-off time, which depends on a randomly generated value using the Contention Window doubling mechanism Node (c)’s timer reaches zero first and it succeeds in transmitting RTS to Node (b) to start a data packet transmission.

1.2.2 Binary Exponential Back off (BEB) algorithm

Binary Exponential Back off (BEB) algorithm [1][2][3] is used to regulate the back-off time for each node before attempting to transmit packets to apply the collision avoidance mechanism. Nodes that have packets to transmit would select a back-off value based on the CW: [Back off = int (CW × rand × slot time)]. "Rand" is a randomly generated value uniformly distributed between 0 and 1. Slot-time is "Sigma", and CWmin< CW <CWmax, are the minimum and maximum values for the contention window.

The node calculates a back-off time in the range [0, CWmin - 1]. When the medium becomes idle, after an additional DIFS period, nodes decrement their back-off timers until the medium becomes busy again or until the timer value reaches zero. If the timer has not reached zero and the medium becomes busy, the node freezes its own timer. The process continues until the timer is decremented to reach the value of zero so the node transmits the packet that is waiting in the queue to be transmitted. In case of a successful transmission, the receiving node will acknowledge the packet by sending the ACK packet to the sending node. The sending node will set its CW to CWmin - 1, the initialization state. If two or more nodes decrement their timers and reach zero at the same time, then a collision will take place in the medium, and each sender will have to generate a new back-off time by doubling the CW value [2* CWmin]. During the kth retransmission attempt, the Contention Window will have the form [2k * CWmin] and will be doubled till it reaches CWmax. The MAC parameter values (Slot Time, SIFS, DIFS, ACK, CTS, RTS and CW) are dependent on the physical layer being used by the MAC protocol.

1.2.3 PHY Layer Technologies

My research presents the Detection and Mitigation algorithm and applies it to three different PHY technologies; FHSS, DSSS, and OFDM. Basic information [1][2][3] about the Channels and the Frequency bands used will be briefly discussed to clarify our proposed algorithm. The PHY parameters for each type will be presented in table (1-2) at the end of the section.

1. IEEE 802.11 – Frequency-Hopping Spread Spectrum (FHSS)

FHSS operates in the 2.4 GHz band with a range starting from 2.402 GHz to 2.480 GHz. Each channel has a width of 1MHz and supports 1Mbps and 2Mbps rates. There are 78 hopping sequences and each sequence would use 79 hops. Fifteen systems could be collocated and work independently with minimal amount of collisions [31].

2. IEEE 802.11b – Direct-Sequence Spread Spectrum (DSSS)

DSSS operates in the 2.4 GHz band. Each channel has a width of 22. The rates defined in IEEE 802.11 are 1 Mbps and 2 Mbps, and the rates in IEEE 802.11.b standard are 5.5 Mbps and 11 Mbps.

The channel to frequency mapping is shown in Figure (1-2). Only the first 11 channels are used in the United States, and we assume this is true when running the simulation.

3. IEEE 802.11g - Orthogonal Frequency-Division Multiplexing (OFDM)

OFDM operates in the 2.4 GHz band. IEEE 802.11g supported rates are: 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. The channel to frequency mapping is shown in Figure (1-2). Please note that Figure (1-2) is just a schematic for visualizing the channels and it does not reflect the reality of the OFDM sides which are

sharper than the DSSS sides to reduce the interference between the channels. Only the first 11 channels are used in the United States, and we assume this to be true when running the simulation.

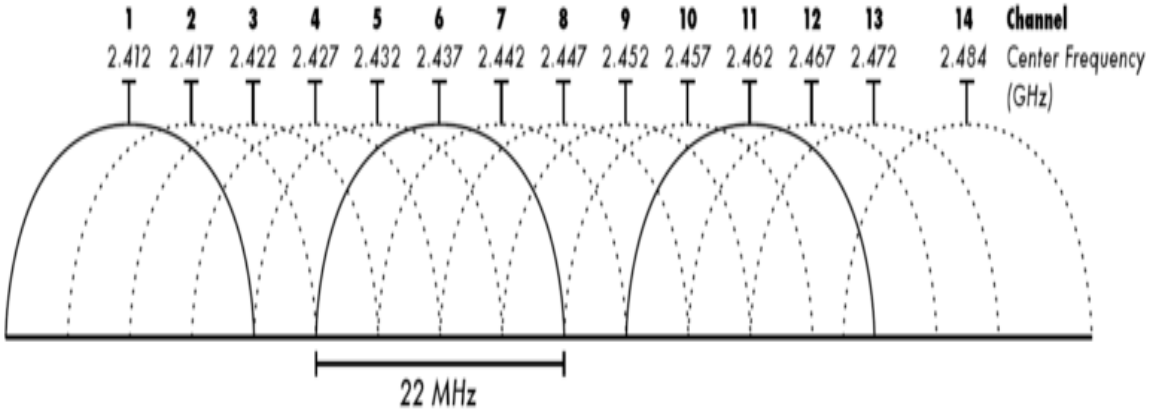

Figure 1-2: IEEE 802.11b/g Channel Distribution [30]

Table 1-1: Frequency to Channel mapping for IEEE 802.11b/g

Channel	Center Frequency	Channel Width (GHz)	Overlaps Channels
1	2.412 GHz	2.401 - 2.423	2,3,4,5
2	2.417 GHz	2.406 - 2.428	1,3,4,5,6
3	2.422 GHz	2.411 - 2.433	1,2,4,5,6,7
4	2.427 GHz	2.416 - 2.438	1,2,3,5,6,7,8
5	2.432 GHz	2.421 - 2.443	1,2,3,4,6,7,8,9
6	2.437 GHz	2.426 - 2.448	2,3,4,5,7,8,9,10
7	2.442 GHz	2.431 - 2.453	3,4,5,6,8,9,10,11
8	2.447 GHz	2.436 - 2.458	4,5,6,7,9,10,11
9	2.452 GHz	2.441 - 2.463	5,6,7,8,10,11
10	2.457 GHz	2.446 - 2.468	6,7,8,9,11
11	2.462 GHz	2.451 - 2.473	7,8,9,10

Each IEEE 802.11 standard [1][2][3] has specific set of parameters in Table (1-3) that dictates the design of each system.

Table 1-2: PHY Layer Parameters

Parameter	FHSS	DSSS	OFDM
Slot Time "σ"	50 us	20 us	9 us
SIFS	28 us	10 us	10 us
DIFS	128 us	50 us	28 us
PHY Header	128 bits	192 & 96 (us)	60 us
MAC Header	272 bits	28 bytes	246 bits
ACK	112 bits	14 bytes	134 bits
CTS	112 bits	14 bytes	134 bits
RTS	160 bits	20 bytes	182 bits
Channel Bit Rate	1 Mbps	11 Mbps	24 Mbps
CWmin, CWmax	15, 1023	31, 1023	15, 1023
Packet Size	8000 bits	8000 bits	10000 bits
Signal Extension	N/A	N/A	6 us

1.3 DoS Attack Impact

The DoS attacker in this research study is an intelligent attacker, who partially follows IEEE 802.11 standards to appear as a legitimate node that wants to join a group of nodes so it could start a valid exchanging of data packets in a distributed network topology with no centralized authority to validate and authenticate non-member nodes that want to join. The attacker will follow control and data packet formats (RTS, CTS, ACK, and Data) and all the PHY layer parameters (SIFS, DIFS, Slot Time, PHY and MAC headers) to further deceive legitimate nodes inside the network. The attacker's behavior is intelligent. It does not use a simple signal jammer, in which a pulsing or continuous signal is transmitted with high power to jam a frequency band. Instead the attacker follows the appropriate traffic exchange four-way handshake model. The attacker will start communicating normally with the nodes and shortly after will activate the DoS attack mechanism and pick a node and bombard it with RTS and data packets to achieve channel capturing. As long as the attacker is getting ACK messages from the victim, it will keep the attack going. The packets received by the victim do not contain useful data but follows the format of a legitimate packet. The attacker could implement his mechanism by modifying the Network Interface Card (NIC) on his communication machine (i.e., laptop). The DoS attack will result in bandwidth starvation, excessive power consumption, and exhaustive CPU processing. The worst impact is felt in an ad-hoc wireless environment, where all the resources (i.e., BW, Power, and CPU Processing) are scarce due to the nature of mobility and lack of infrastructure. The attacker manipulates the Binary Exponential Back-off (BEB) algorithm and does not back off according to the BEB algorithm, but instead it always backs off 1 slot and retries to transmit.

The impact was simulated in fixed and mobile environments. The simulated mobility throughout this research consisted of nodes moving randomly with a constant speed of 60 Km/h with a "Default Random Waypoint" mobility profile. Each case was simulated with a different number of nodes to show the impact in different topologies. All nodes, including the attackers', were configured to generate the same rate of data packets in the upper layer (i.e., application), which is considered the load.

Figure (1-3) shows the difference between data sent by the attacker and a normal node. Although the load is the same, the graph shows that that attacker is transmitting packets 500 times more than the normal node.

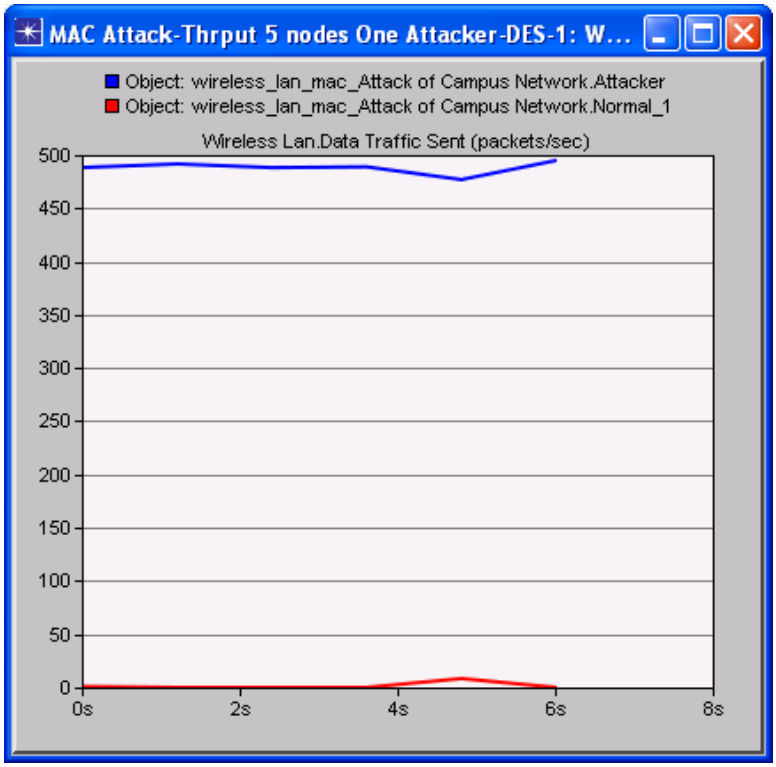

Figure 1-3: Fixed – 5 nodes

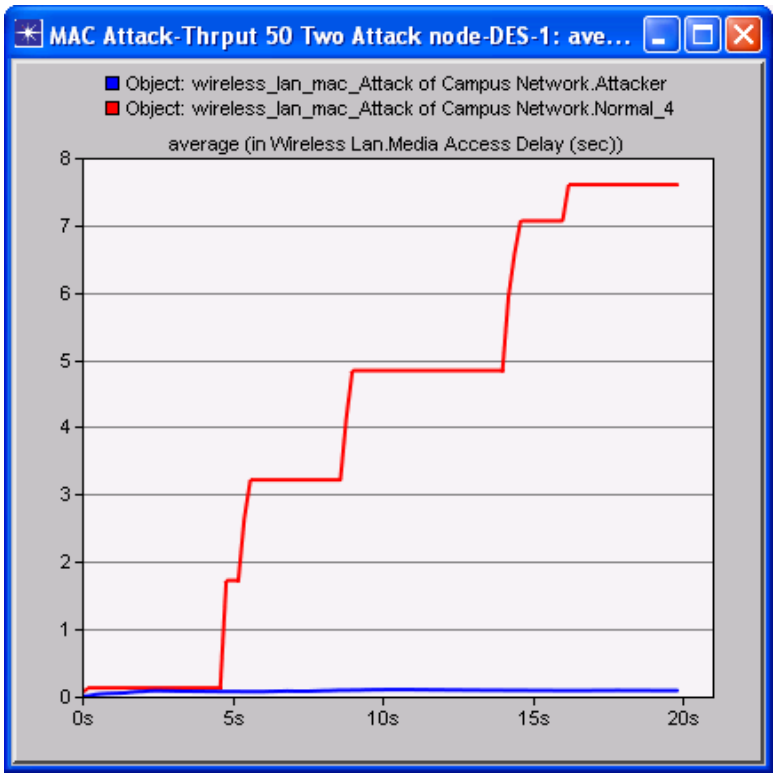

Figure 1-4: Fixed – 50 nodes

Figure (1-4) shows the delay difference between the attacker (blue line) and the innocent node (Red line) and how the delay for the innocent node is building up while the attacker's delay is steadily low.

1.4 Related Work

Because of the randomness in choosing a back-off value for each node, detecting the back-off manipulation is a very challenging task [9], [10], and [11]. The majority of the detection techniques discussed in the literature assume that the network topology with an access point which are considered trusted nodes [10][12] which are tasked to monitor all the nodes in the network to detect any misbehavior. Our approach in this chapter assumes that the network is a distributed network with no centralized authority to monitor and that each node functions as a police node without cooperation from any other nodes inside the network. Other methods of detection mechanisms that were proposed in the literature assume that the attacker will cooperate not that the attacker is trying to disrupt the network [8]. Many methods were proposed in the literature to combat and prevent MAC layer misbehavior and back-off timer manipulation. In [13] the authors proposed a method to force all the nodes to use known back-off timers by appending some values to the existing data packet structure to combat this misbehavior. But this method assumes that the misbehaving node or the attacker will cooperate, not the case in a real organized attack. Channel-hopping techniques [14][15][16] were described in the literature to avoid DoS targeting the physical layer, which constitutes signal jamming. The presented techniques primarily targeted a jamming attack with no intelligence, in which attackers transmit a continuous signal with a sensible power level on a certain frequency band as opposed to our case in this chapter, where the attackers are intelligence and strive to appear as legitimate nodes and to some extent follow IEEE 802.11 DCF protocol and comply with the control and data packets construction and format. The authors in [17] presented a channel-hopping concept that takes place in the MAC layer based on sending a beacon on a non-jammed channel when a node senses a jamming signal and other nodes will follow to the other channel to detect the beacon and resume the communication. Again this example describes simple jammers, not intelligent ones that can adapt and move to the new channel to jam it if it senses that all nodes moved away from the jammed channel. In [18] the authors proposed a channel-

hopping mechanism to countermeasure the DoS impact on the network, but they built their algorithm on a multi-radio network and did not deal with a single-radio network. In [19] [20] [21] and [22] the concept of frequent channel hopping was introduced so the active communication channel would change every fixed interval that ranges from few milli-seconds to a few seconds.

In [23] the authors proposed a channel-hopping scheme based on trusted AP that would coordinate the hopping inside the network whereas our proposed algorithm is completely distributed and does not require a centralized authority like a trusted AP.

1.5 Dissertation Motivation and Goal

- Several factors motivated the research in this area:

1. The importance of wireless communication

   Most people use cellular phones, wireless routers, and long backhaul wireless equipment either in a direct way or in an indirect way.

2. The sensitivity of information and data being transferred using wireless communication.

   People use wireless communication to see their bank accounts, medical and educational records. Governments use wireless communication to exchange secrets and time-sensitive materials. Financial institutions conduct massive transactions worth billions of dollars using communication devices that include wireless equipment.

3. The implementation of a DoS attack is relatively easy and feasible.

   This DoS attack can be implemented by literally changing a few lines in the code of the NIC (Network Interface Card) on any laptop or a communication device that has a NIC.

4. The destructive impact of the DoS attack that will paralyze all the network communication

   As seen in the previous section, the impact is disastrous. Once the attack is launched no legitimate user will be able to communicate.

- Several goals were set for the research that will be achieved by the final product:

1. The ultimate goal for this research is to find an effective method to detect the DoS attacker and identify him by a commonly known and unique ID (MAC Address).

2. Mitigate the negative impact on the network so legitimate users can continue communicating in the presence of the attack with the least amount of losses.
3. The proposed algorithm should be easy to implement without the need of sophisticated hardware modifications.
4. The proposed algorithm should be compatible with all MAC layer protocols that do not require a total communication stack modification.

In the next sections, multiple detection and mitigation algorithms will be presented for single and multiple wireless domains where there are hidden nodes. The algorithm will be examined in a mobile environment. Also presented will be the concept of a network that operates with a single access point where typical WiFi networks are utilized.

Chapter 2 : Statistical Detection of DoS Attack in IEEE 802.11 Wireless Networks

2.1 Abstract

This chapter presents a novel technique to detect the DoS behavior applied by malicious nodes in wireless networks employing the widely used IEEE 802.11 DCF protocols. Malicious nodes manipulate the IEEE 802.11 DCF standards to illegally gain extra throughput and increase the probability of having a successful packet transmission at the expense of the honest nodes that follow protocol standards. The theoretical network throughput will be derived using two-dimensional Markov Chain to determine the network capacity. Results obtained by theoretical computations will be validated by network simulation to determine the baseline for the maximum achievable throughput in the network under fair conditions, where all nodes follow the standards. An approach is proposed to equip all the nodes in a IEEE 802.11 network with a mechanism to detect and identify the malicious nodes in a distributed environment. Results are presented to prove the effectiveness and feasibility of the proposed algorithm in both fixed and mobile environments.

2.2. Introduction

IEEE 802.11 DCF specifies two mechanisms to perform packet transmission. The default mechanism is a two-way handshaking method referred to as "Basic Access". This mechanism employs immediate transmission of an acknowledgement (ACK) packet by the destination node after a successful reception of a packet transmitted by the sender. In this chapter the term "selfish node" implies a DoS attack.

The second mechanism employs a four-way handshaking procedure "Request-to-Send/Clear-to-Send (RTS/CTS)." Prior to transmitting a packet, a node set by using RTS/CTS mode "reserves" the channel by

sending a special Request-To-Send short frame. The available destination node responds to an RTS frame by sending back a Clear-To-Send frame, and then a data packet transmission and ACK response will follow. The RTS/CTS mechanism increases the network throughput by reducing the duration of a collision when long messages are transmitted. In this chapter, our focus is on misbehavior detection in the four-way handshaking mechanism "RTS/CTS scheme". Selfish nodes in wireless networks employ several techniques to illegally increase their throughputs at the expense of other fair-behaving nodes [8]. In IEEE 802.11 based networks, selfish nodes will manipulate the back-off timer to increase the probabilities of having successful transmissions. They decrease the back-off timer value instead of following the back-off timer generation method that all other nodes in the network are using. A node is considered malicious or misbehaving (attacking) when it does not follow the IEEE 802.11 MAC Standard [1]. These selfish nodes will use smaller timeouts than these specified in the protocol standard. With IEEE 802.11, each node is expected to choose a back-off interval prior to initiating a transmission. The back-off interval is to be increased as per a specific set of rules prior to retransmission attempts that are invoked upon failed transmission attempts. A malicious node may choose a small and/or a constant back-off interval prior to the transmission of a data packet or follow a completely different retransmission strategy upon experiencing failed transmissions that do not conform to the standard IEEE 802.11 protocol. Due to the randomness in choosing a back-off value, detecting back-off manipulation is most challenging [27], [9], [10], and [11]. The goal of this chapter is to detect malicious nodes. MAC layer vulnerabilities were discussed in the literature over the years [52] [53] [54] [55] [56] [57] [58] [59] [60] [61].

Several researches were conducted regarding the detection of the manipulation of the back-off timer in wireless networks where there are trusted access points (AP) [12][10], where a trusted AP will regulate the senders back-off timer values and detect the misbehaving nodes. Due to the nature of ad-hoc wireless networks, where there is no centralized authority that will assign and monitor each nodes' back-off timer

values, the task is very challenging. Our proposed algorithm is designed to work in a distributed environment where there is no centralized authority or a supervisor node (i.e., Access Point) watching every transaction in the network.

In [8] the authors assume that each node will cooperate and announce the state of its pseudo-random sequence generator so other nodes monitor its behavior. This approach assumes much cooperation from compromised nodes, which is not realistic in most situations. Our proposed algorithm does not expect any cooperation from any node, hence eliminating the chance of receiving wrong information by a malicious node. In [28] the authors introduced a new parameter to indicate the level of cooperation of each node. In [29] a method was proposed to make the APs functions like a watchdog to monitor all nodes' behaviors. This method consumes the resources of the AP node and is not suitable to a total distributed system like IEEE 802.11 wireless networks. Assigning one node or selected nodes to police the network is a very dangerous concept and creates a single point of failure in case the police node is itself compromised. In [32] the author is proposing to analyze the distribution of inter-delivery times between two consecutive successful transmissions. This method is very challenging and requires measurements in microseconds to accurately detect the selfish behavior. Our proposed algorithm does not require any hardware additions or clocks measuring to microseconds. The words "attacker" and "selfish" will be used interchangeably in this chapter.

2.3. Attacker Behavior

The selfish behavior could be implemented by different methods which will alter the standards to illegally gain benefits which could be in the form of increasing data throughput, reducing delay, preserving power for the attacker on the expense of the other honest nodes that follow the standards. Two ways to implement the attack; the first is to modify the hardware design, and the second is to modify the protocol's firmware. The

second way is a lot easier to implement from the feasibility and cost point of view and is far more prevalent. In our chapter, the presented solution is directed towards detecting the manipulation of the protocol's firmware and more specifically detecting the manipulation of the back off timer. A selfish or an attacker node will not follow the Back off mechanism explained above but instead will minimize the back off time. For example, an attacker could only back off one slot every single time it has a packet to transmit or when it experiences a collision while the other nodes which are honest will follow the exponential back off mechanism. We simulated a network with an attacker presence to show the effect on the other honest nodes. The payload size used throughout this chapter is 8000 bits so it could be sent in one time slot without the need of fragmentation. Maximum number of retransmission was configured to be 6 times before the packet is discarded. We configured all the nodes with the same parameters (data rate generated by every node, packet size and traffic generation interval). Different WLAN statistics were collected to show the effect of the misbehavior in the network – Traffic Generation Scheme (Exponential – 1000 Packets/sec generated by the upper layers) – 5 nodes network with two attackers present.

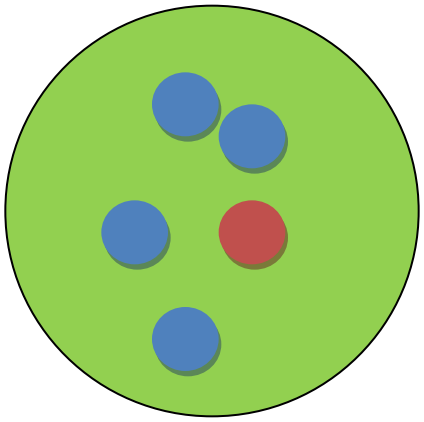

Figure 2-1: Typical Network Configuration

(Blue Nodes are the good ones and red node is an attacker)

Figure (2-1) shows typical network with an attacker present. Figure (2-2) shows the throughput (successful packets/second sent by each node - one attacker and one normal), the difference is very significant. The attacker by simply modifying the MAC protocol effectively increased its throughput more than 20 folds in this scenario. Figure (2-3) exhibits the difference in Media Access Delay in seconds between the attacker and the honest node. So the effect on the honest node is not only the decreased throughput but also the internal resources of the honest node are affected unnecessarily to the extent that it's considered to be Denial of Service (DoS) attack. For example, Figure (2-4) show very high memory and processor consumption; because all these delayed packets will be consuming unnecessarily a significant space in the honest node's buffers and queues. Also, in case the honest node drops these packets because either these packets have reached the maximum retransmission numbers (m = 6) in this case, or the buffer has overflowed, then the honest node will have to regenerate these packets which consume the honest node's power and processing resources.

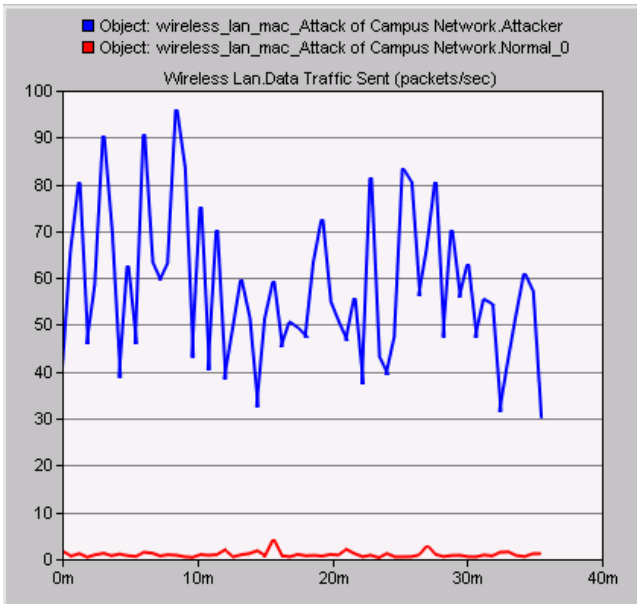

Figure 2-2: Attacker Throughput

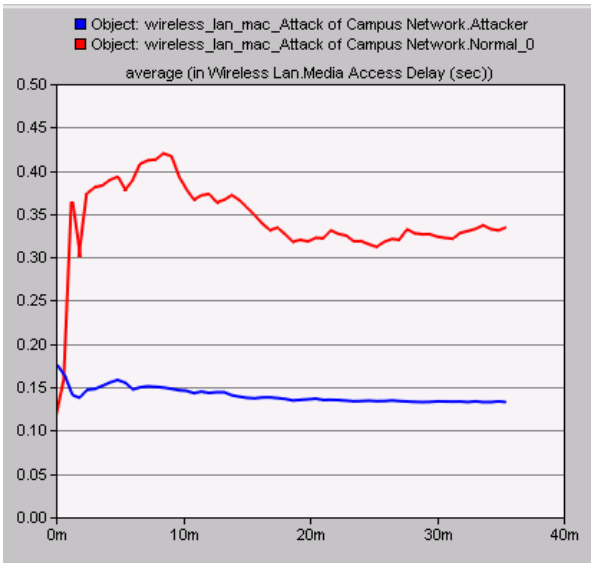

Figure 2-3 Media Access delay

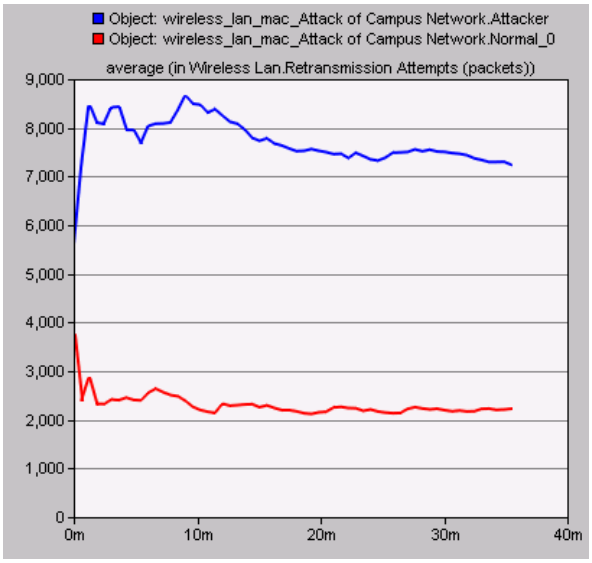

Figure 2-4: Retransmission rate

2.4. Markov Chain

We are analyzing the Wireless Network's throughput using two dimensional Markov Chain to obtain theoretical numerical results that will be used by our algorithm to identify the malicious behavior in the network. We apply and extend Bianchi's Markov Chain model [4] to calculate the individual rate in "Packet per second" values for each node in the network. The chapter will adopt Bianchi's [4] assumption where the calculations assume there are no hidden terminals to eliminate the chances for NAV collisions and no

packets are lost due to poor wireless signal. The first step is to calculate the (Transmission Probability – Tp) then derive the throughput for the whole network as a function of Tp, finally obtain the individual throughput for each node.

Assume that each node has a packet to transmit at any given time (Saturation Condition) and the number of nodes inside the network is constant. b(t): stochastic process representing the back off time counters for a given node. (t and t+1) correspond to the beginning of two consecutive slot times. The back off time counters of each node decrements at the beginning of each slot time. (L) is the maximum number of back off stages. W = CWmin and CWmax = 2n CW. Wi= 2k W, where k $\in$ (0,L), which is a backed off stage. S(t) is the stochastic process representing the back off stage (0,...,L) of the node at time t. At each transmission attempt, and regardless of the number of retransmissions experienced, each packet collides with constant and independent probability (1-S) = p, whereas S is the probability of transmission success by the node itself and (1-S) is the conditional collision probability which is the collision experienced by a packet being transmitted.. By solving b(t) and S(t) using the Markov chain in Figure (5), the only non nullone-step transition probabilities are listed below in Eq (1 – a,b,c,d):

$P\{k,d|\ k, d+1\} = 1 \qquad d \in (0, W_k - 2) \qquad k \in (0,L)$

(At the beginning of each time slot, the back off time is decreased. Eq (1-a)

$P\{0, d|\ k,0\} = (1-p)/W0 \quad d \in (0, W0 - 1) \quad k \in (0,L)$

(A new packet after a successful transmission will start with back off stage 0) Eq (1-b)

$P\{k,d|\ k\text{-}1,0\} = p/W_k \qquad d \in (0, W_k - 1) \quad k \in (1,L)$

(When unsuccessful packet transmission occurs, the back off stage increases and the new back off stage is uniformly chosen from the range of (W_k – 1. Eq (1-c)

$P\{L,d|L,0\} = p/W_L \qquad d \in (0, W_L - 1)$

(Indicates that when the number of back off stages reaches the maximum, it stops increasing).Eq (1-d)

Let $b_{k,d} = \lim t \to \infty\ P\{s(t) = k, b(t) = d\}$, $k \in (0,L)$, and $d \in (0, W_k - 1)$ is the stationary distribution of the chain, then from Markov chain:

$b_{k-1,0} \cdot p = b_{k,0} \to b_{k,0} = p^k b_{0,0} \quad 0<k<L$

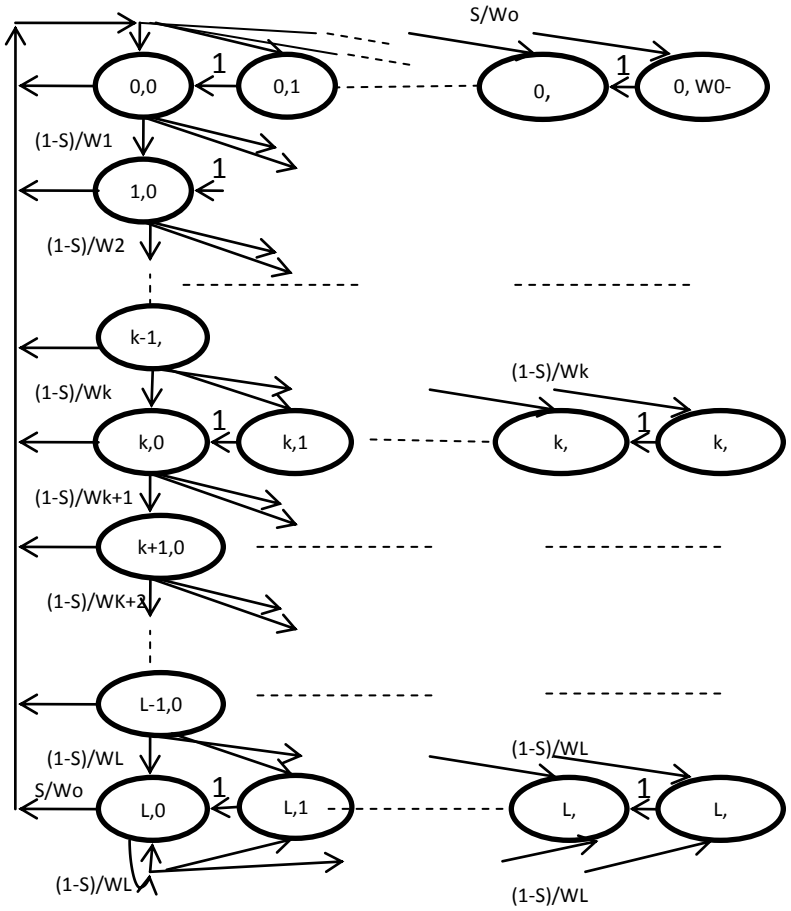

Figure 2-5: Markov Chain

$b_{L-1,0} \cdot p = (1-p)\ b_{L,0} \to b_{L,0} = (p^L/(1-p))\ b_{0,0}$ .....Eq (2)

Due to the chain regularities, for each d $\in$ (1, W_k – 1), then:

$$b_{k,d} = ((W_k - d)/ W_k) \cdot \begin{cases} (1-p) \sum_{k=0}^{L} b_{k,0} & k = 0 \\ p \cdot b_{k-1,0} & 0<k<L \\ p \cdot (b_{L-1,0} + b_{L,0}) & K = L \end{cases}$$

.....Eq (3)

Since, $\sum_{k=0}^{L} b_{k,0} = b_{0,0}/(1-p)$ then,

$b_{k,d} = ((W_k - k)/ W_k) \cdot b_{k,0}$ $k \in (0,L)$, $d \in (0, W_k - 1)$

.....Eq (4)

Relating the previous equations, we find that all the values of $b_{k,d}$ can be expressed as a function of $b_{0,0}$ and p. By imposing the normalization condition and simplifying the result, the equation will have the following form: $b_{0,0} = (2(1-2p)(1-p))/((1-2p)(W+1) + pW(1-(2p^L))$

Since any successful transmission occurs when the back off timer = 0 regardless the number of the back off stage, then the transmission probability:

$$Tp = \sum_{k=0}^{L} b_{k,0} = b_{0,0} / (1-p)$$

$$= (2(1-2p))/((1-2p)(W+1) + pW(1-(2p^L))$$

$$= 2/(1 + W + pW\sum_{k=0}^{L-1} (2p)^k \quad \text{.....Eq (5)}$$

To calculate the throughput in the RTS/CTS model, each node will either transmit successfully or hear one other node (from total n nodes) is transmitting. The collision will only affect the RTS packets assuming there are no hidden nodes as stated in the beginning of the chapter:

Figure 2-6: Successful Transmission Time = Tsuccess

From Fig (2-6): Tsuccess = (RTS + SIFS + Propagation Delay + CTS + SIFS + Propagation Delay + Packet Header + Paylaoad +SIFS + Propagation Delay +ACK + DIFS + Propagation Delay

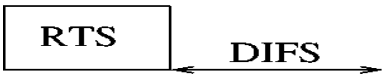

Figure 2-7: Collison Time = $T_{collision}$

From Fig (2-7): Tcollison = RTS + DIFS + Propagation Delay

Ptx is the probability that there is at least one transmission in the slot time.

Psc is the probability that there is at least one packet being successfully transmitted in a slot time

$Ptx = 1 - (1-Tp)^n$, $Psc = (n.Tp. (1-Tp)^{n-1}) / Ptx$

Normalized Throughput =

E (payload transmitted)/E (length of slot time)

Throughput (Packets/Second) =

(Psc.Ptx)/((1-ptx).Sigma + Psc.Ptx. Tsuccess + Ptx (1-Psc)Tsuccess .....Eq (6)

By applying Bianchi's approximation [4] for maximum throughput under saturation condition, the approximate transmission probability Tp = 1/(n * Sqrt (0.5 (Tc/Sigma)).

The interest of this chapter is the individual throughput in terms of transmitted packet rate by each node in the network assuming the wireless channel is perfect and regardless of the individual raw data rate generated by the upper layers of each node. The concept was presented in [6] which show that all nodes equally access the channel. With n nodes each has its own data rate and number of events E at a period of observation Time ($T_{observe}$). Number of empty slots = Empty_Slots. Number of successful transmissions = T_success. Consumed during collisions = $T_{collisions}$. At any given observation period, $T_{observe}$ =

[(Empty_Slots * Sigma) + $T_{collisions}$ + $\sum_{j=0}^{n} M(j)$ * T_success * R(j)] Eq (7), where M(j) is the number of successful transmissions for node j which has data rate of R(j). When P(j) is the packet size for node j, then the throughput for node j is: = M(j)*P(j))/ $T_{observe}$

Since the total throughput for the whole network is:

= (T_success/ $T_{observe}$) * (Average bits per packet for the whole network).

All virtual slots are independent and Empty_Slots = $E(1-Tp)^n$.

The probability finding a slot occupied by a successful transmission by node j is $Tp (1 - Tp)^{n-1}$. The following is concluded from the previous equations: $M(j) = E Tp (1-Tp)^{n-1}$

Since the number of nodes are n, then: T_success = $EnTp(1-Tp)^{n-1}$, then it could be concluded that each node equally accesses the channel [6]. The individual throughput for each node equals the total channel throughput divided by the number of nodes in the network. To validate the theoretical results described

above, we compared the numerical results obtained by solving the Markov Chain. The comparison graphs in Figures (2-8, 2-9 & 2-10) show the differences between theoretical results of the throughputs (presented by Blue lines) which were obtained by solving Markov chain and the simulation throughputs (presented by Green lines) and were obtained by Network simulation using OPNET [5].

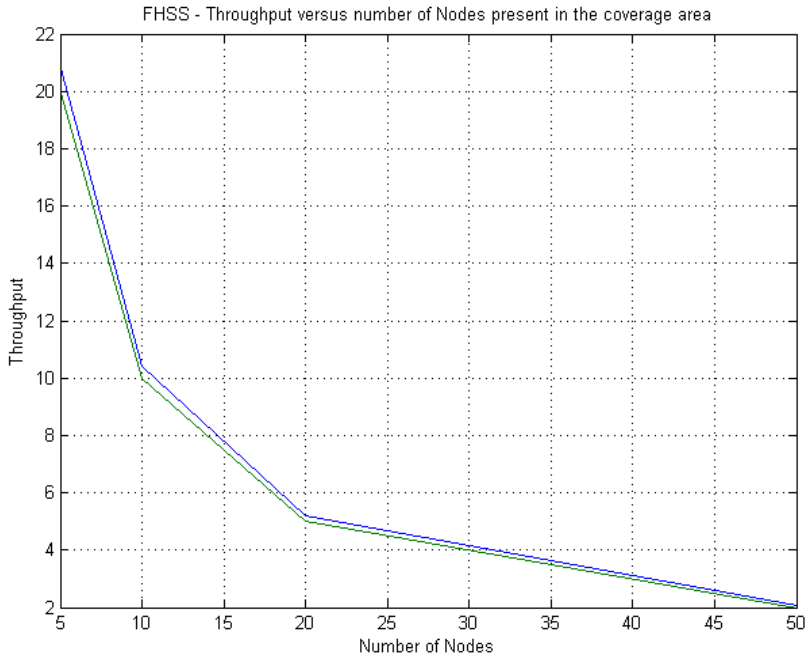

Figure 2-8: Comparison between the Throughputs (packets/Sec) obtained by solving Markov Chain and OPNET simulator. (IEEE 802.11- FHSS)

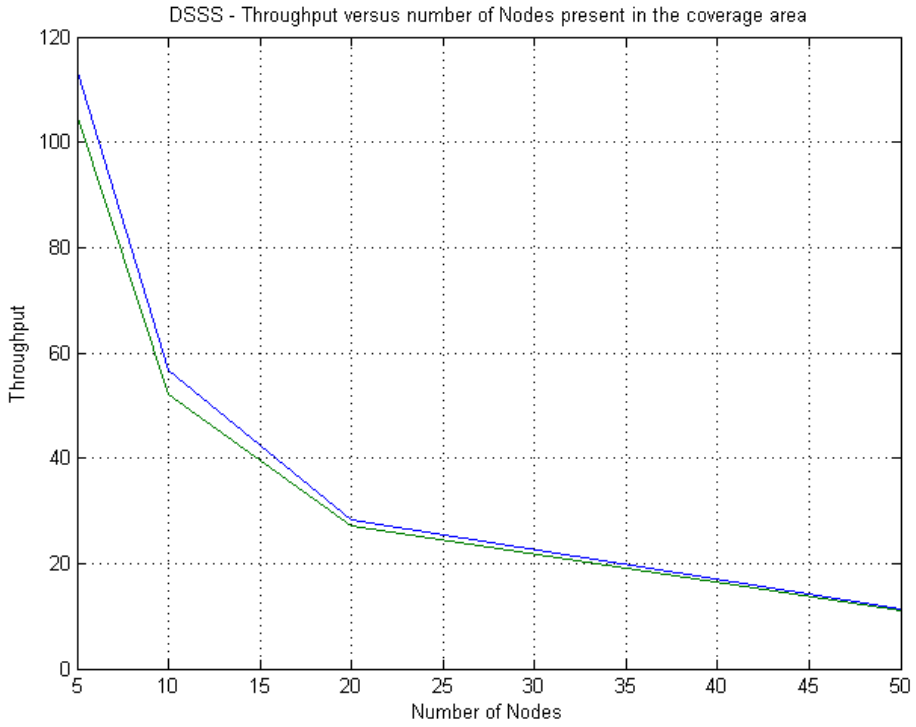

Figure 2-9: Comparison between the Throughputs (packets/Sec) obtained by solving Markov Chain and OPNET simulator. (IEEE 802.11b- DSSS)

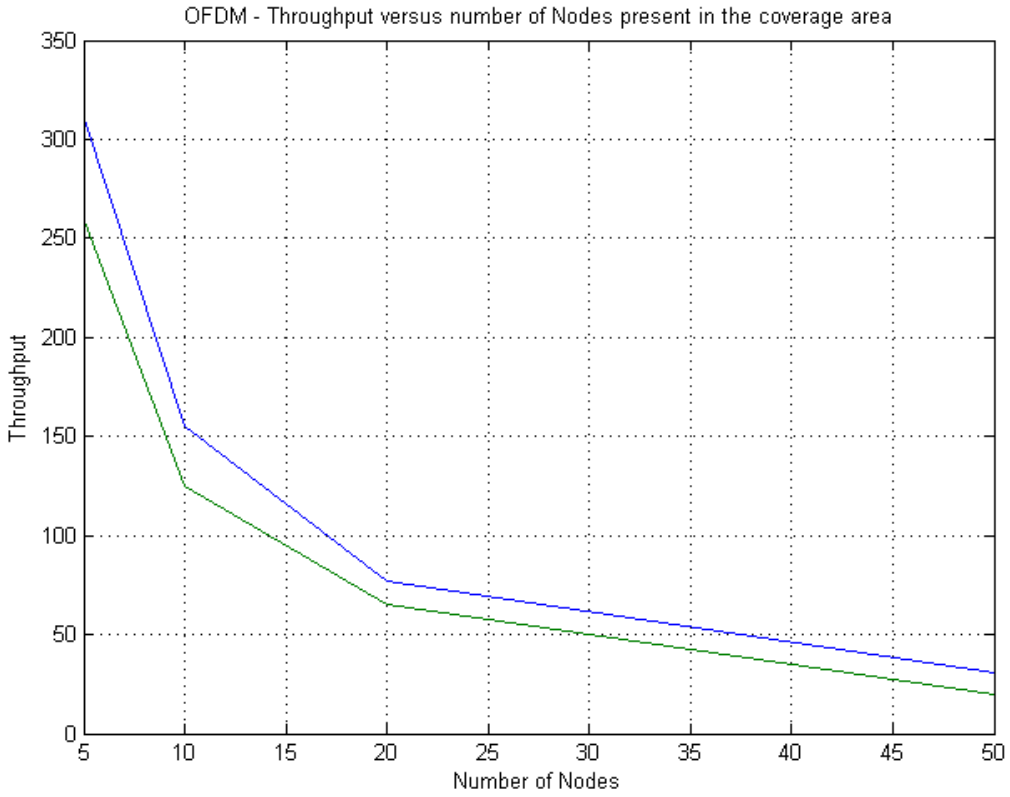

Figure 2-10: Comparison between the Throughputs (packets/Sec) obtained by solving Markov Chain and OPNET simulator. (IEEE 802.11g- OFDM)

.

2.5 Detection Process

From the IEEE 802.11 DCF RTS/CTS operation, it is concluded that the number of successful data packets sent by a given node, given that the wireless channel is perfect and there are no hidden nodes, are equal to the CTS packets received by this specific node with the permission to send data packets. The CTS packets are designed to be heard by every single node in the network.

All the nodes beside the one that the CTS packet is destined will have to update their NAV so they refrain from transmitting any packets during the NAV to eliminate the chances of collisions. OPNET [5] code was modified to hear all CTS packets individually and collect them in separate buckets depending on the destination address. Below is the result from the simulation that the number of received CTS packets is equal to the number of data packets sent. Figure (2-11) shows that the number of CTS received by node_1 is the same number of packets sent by this node to other nodes in the network.

The Algorithm that will reside at each node:

```
Count n          //" Number of Nodes in the network"
Create n Counters
Calculate Maximum Individual Throughput /* obtained from Eq (6) above*/
When Receive CTS
If (Destination Address = My Address)
Do Nothing
Else
{ Update Counter (Destination Address)
        Calculate Rate
/* number of CTS received per second for each Destination Address */          }
        If
        CTS_node_x rate < Maximum Individual Throughput
        Do Nothing
        Else
Announce "node_x is an Attacker" /* it is shown as print command in our OPNET simulation and used it as an output
```

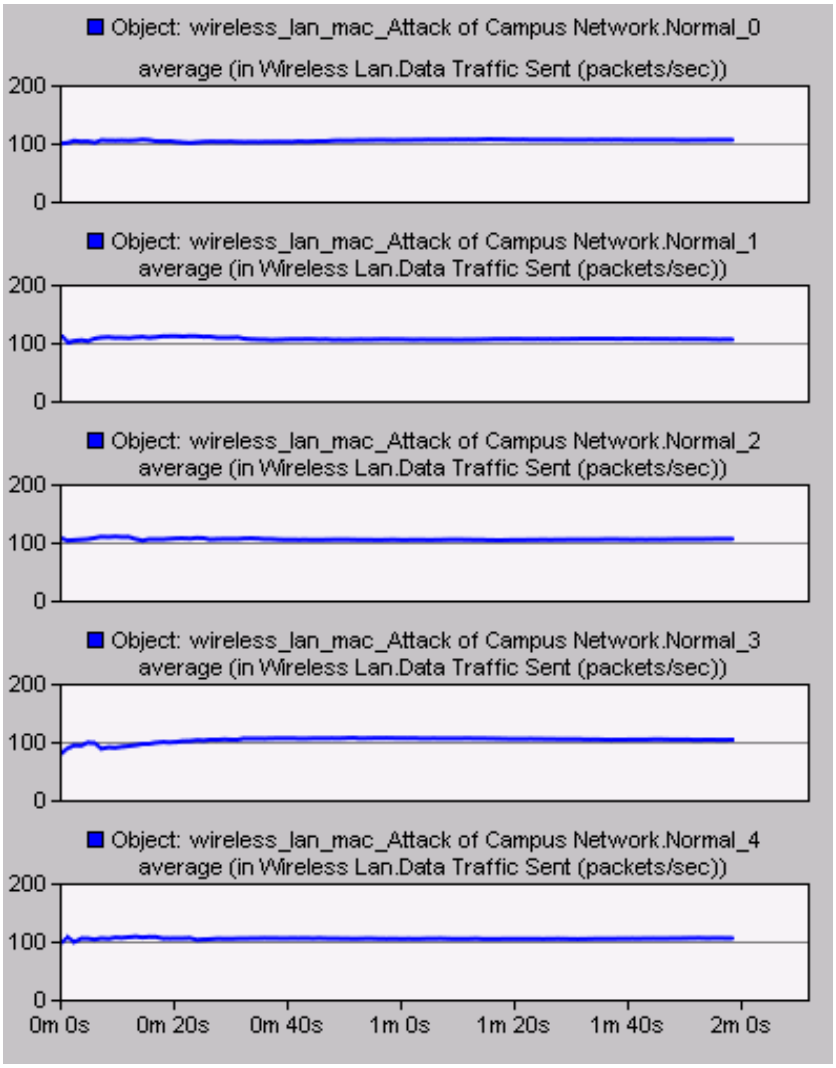

Figure 2-11: Each node has the same average of successful Packets/second Sent to other nodes

Based on that concept, our detection algorithm depends on modifying the IEEE 802.11 DCF code to enable each node to police the network with very low cost (processing and memory consumption wise) solution. Each node does not need any cooperation (information) or any new type of packets sent from any other node to perform the detection process.

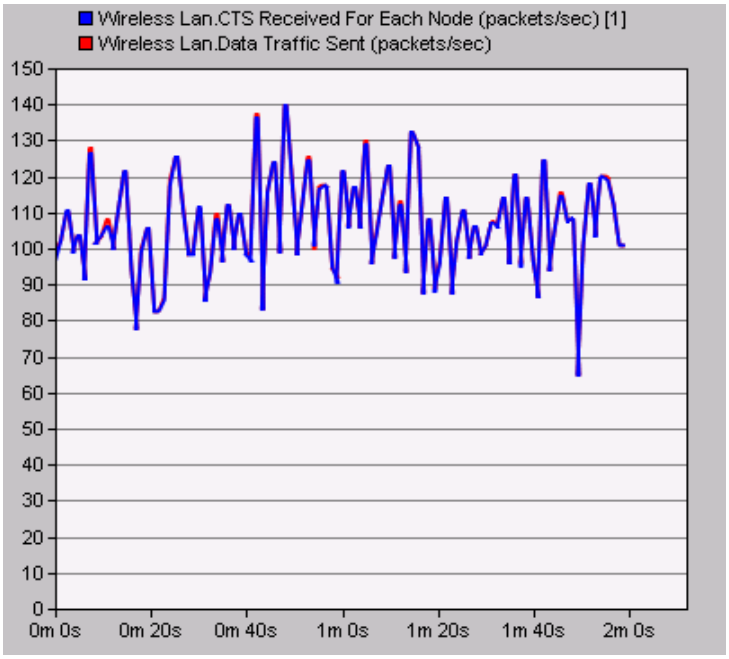

Figure 2-12: Comparison between number of CTS and Data packets for the same node

For our simulation, we used Matlab to resolve the Markov Chain mathematical model and fed the results to OPNET simulator for the detection threshold. The numerical results obtained solving the Markov Chain became the threshold specially that they are very close to the simulation results as shown in Figures (2-8, 2-9 & 2-10). The numerical results are considered the maximum number of packets any node can send in the presence of other number of nodes (as calculated in Eq (6)), so any other node that has more packets successfully sent is not following the IEEE 802.11 DCF standard and manipulating the protocol to illegally increase its throughput.

2.6 Simulation Results

A. Fixed Environment:

To validate the algorithm, we performed a simulation under different network configurations. At each configuration, we checked if the algorithm detected the Attacker/s and also check if there is False detection. (False Detection: where the algorithm flags an honest node as an Attacker). All the simulation rounds were performed using the parameters in Table (1-2) in Chapter 1 for IEEE 802.11 b. We applied our algorithm into three different groups of networks depending on the traffic level generated by the upper layers by each node. We arbitrarily chose three levels of generated traffic:

1. Saturations Level: where generated raw traffic rate is (1000 packets/second). In this group, each node always has a packet to transmit in its queue.
2. Moderate Level: each node generated (100 packets/second)
3. Low Level: each node generated (10 packets/second) and at this level the queues are not always full and do not necessarily have packets to transmit at any given time.

For each level we ran several rounds of simulation for the (5, 10, 20, 50) nodes. Also, with each scenario we ran the simulation twice; first with only one Attacker in the network, and secondly with two Attackers in the network. At each level we are going to show sample graphs only to show the concept.

a. Saturation Level with One Attacker:

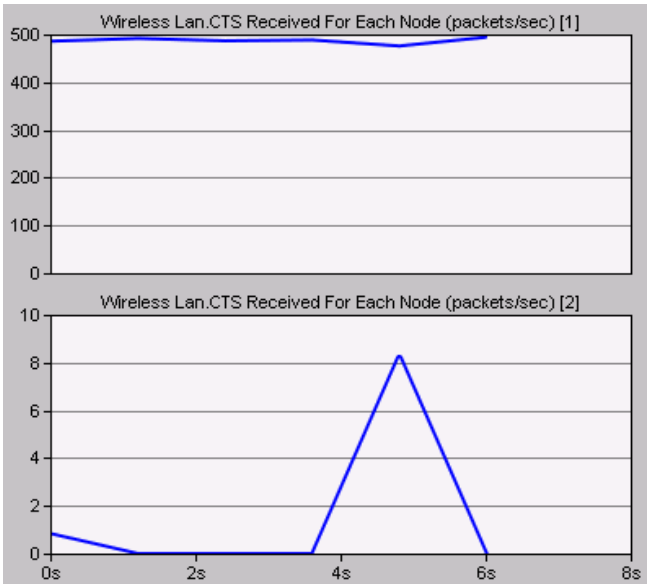

Figure 2-13: 5 Nodes

At this level, the algorithm successfully detected the Attacker in every configuration (5, 10, 20, and 50) nodes. Below in Figures (2-13 & 2-14), the results for 5 and 50 nodes configuration are clear that the Attacker in each case was detected, since the upper portion of the graphs show that the CTS packets destined to the Attacker node went well above the threshold. At this level, the algorithm successfully detected the Attacker in every configuration (5, 10, 20, and 50) nodes. In Figures (2-14 & 2-15), the results for 5 and 50 nodes configuration are clear that the Attacker in each case was detected, since the upper portion of the graphs show that the CTS packets destined to the Attacker node went well above the threshold determined by solving Markov chain. The thresholds for the configurations below were (113 & 11 received CTS packets/second) and the graphs show (500 & 200 received CTS packet/second for the Attacker) which according to our algorithm will trigger the honest nodes to detect the Attacker

[Detection percentage = 100% - False Detection = 0%]

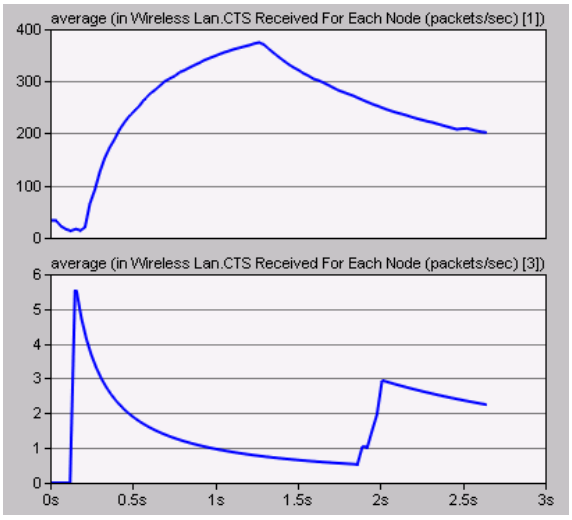

Figure 2-14: 50 Nodes

b. Saturation Level with Two Attackers:

The two graphs in Figures (2-15 & 2-16) are for the CTS packets destined to the Attackers compared to the graphs at the bottom destined to a normal node

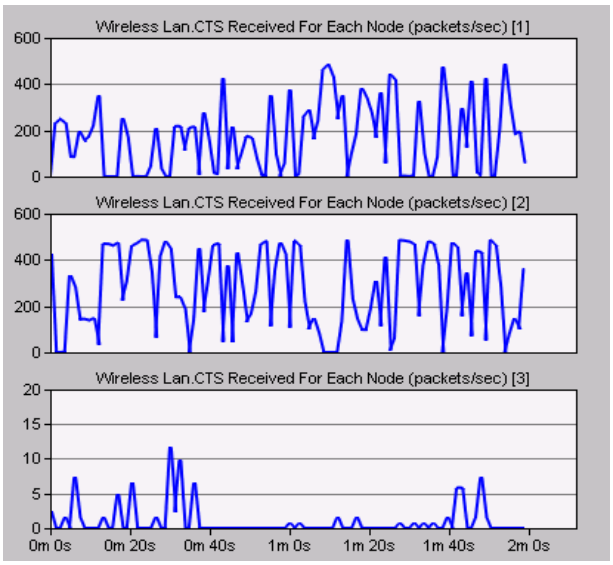

Figure 2-15: 10 Nodes

c. Low level with One Attacker:

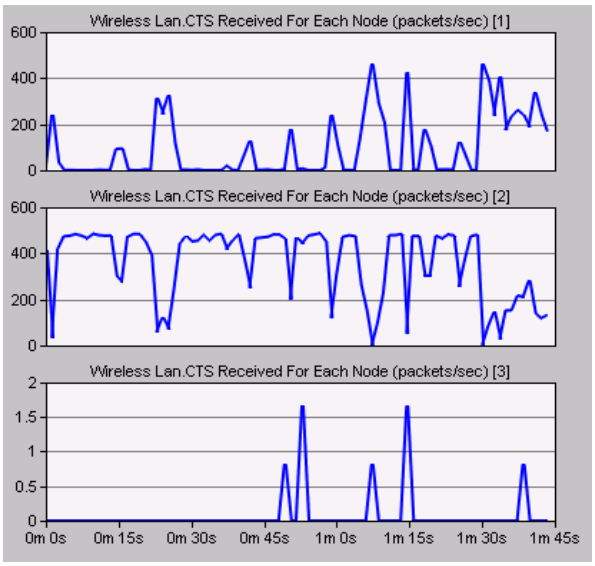

Figure 2-16: 20 Nodes

d. Moderate Level with One Attacker:

The upper graphs in Figures (2-17 & 2-18) are for the CTS packets destined to the Attackers compared to the graphs at the bottom destined to a normal node.

[Detection percentage = 100% - False Detection = 0%]

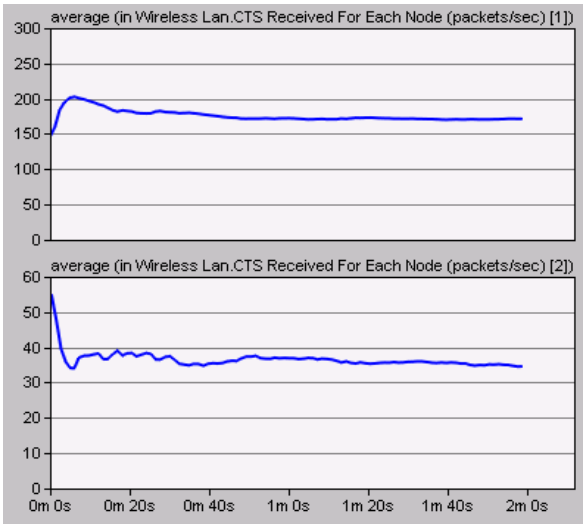

Figure 2-17:10 Nodes

Figure 2-18: 50 Nodes

e. Moderate Level with Two Attackers:

The upper two graphs in Figures (2-19 & 2-20) are for the CTS packets destined to the Attackers compared to the graphs at the bottom destined to a normal node.

[Detection percentage = 100% - False Detection = 0%]

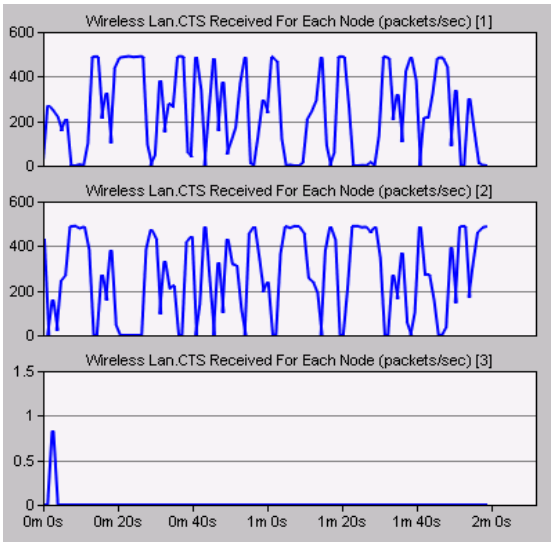

Figure 2-19: 5 Nodes

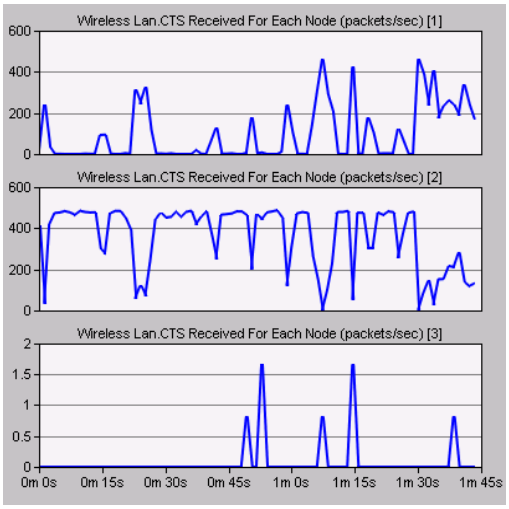

Figure 2-20:20 Nodes

f. Low level with One Attacker:

At this level where the raw traffic generated by each node is low, only the network configuration with 50 nodes achieved the detection (Figure 2-21), the Attacker CTS packet rate is represented by the upper graph compared to an honest node at the bottom.

As expected, other network configurations (5, 10, and 20) nodes, the detection were not achieved (Figures 2-22 & 2-23) due to the availability of the channel bandwidth, the Attackers CTS packet rate are represented by the upper graphs in the Figures (2-22 & 2-23). Figure (2-24) shows that the internal resources of the honest nodes (20 nodes configuration) were not affected and is represented by the rate of the Data dropped because of the buffer flow which is equal to zero. In this scenario, False detection = 0.

It is observed that when the nodes do not suffer from bandwidth shortage or starvation, meaning that each node can send all the data packets without suffering packet drop or significant delay because the channel is not available, the algorithm does not detect the attacker because implementing the attack process does not

negatively affect the honest nodes, so the honest nodes do not care to detect the attacker. Once the honest nodes start feeling the effect of the attacker, the algorithm effectively detects them.

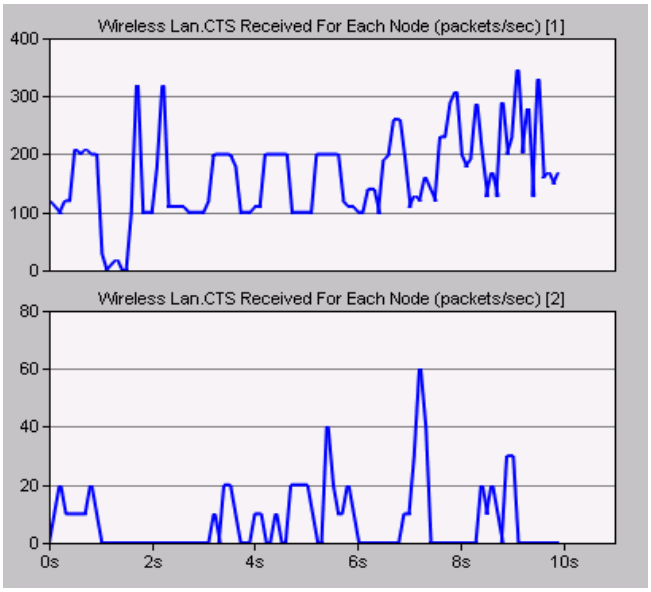

Figure 2-21: 50 Nodes

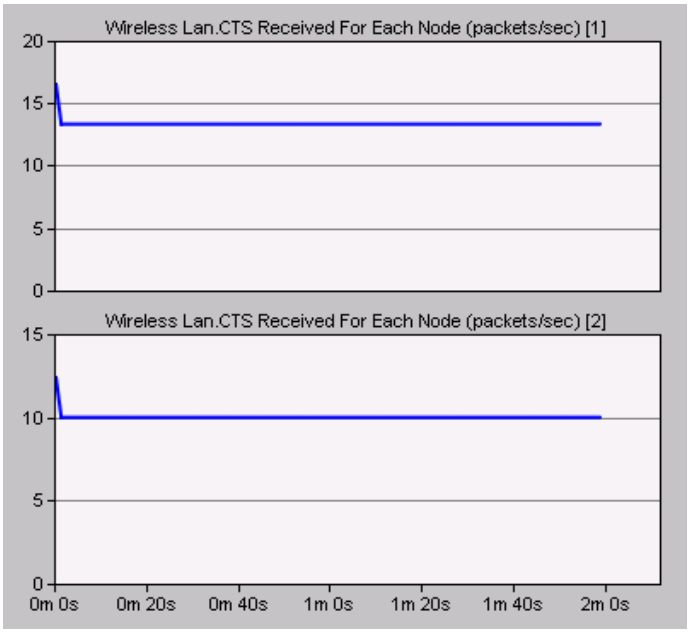

Figure 2-22: 5 Nodes

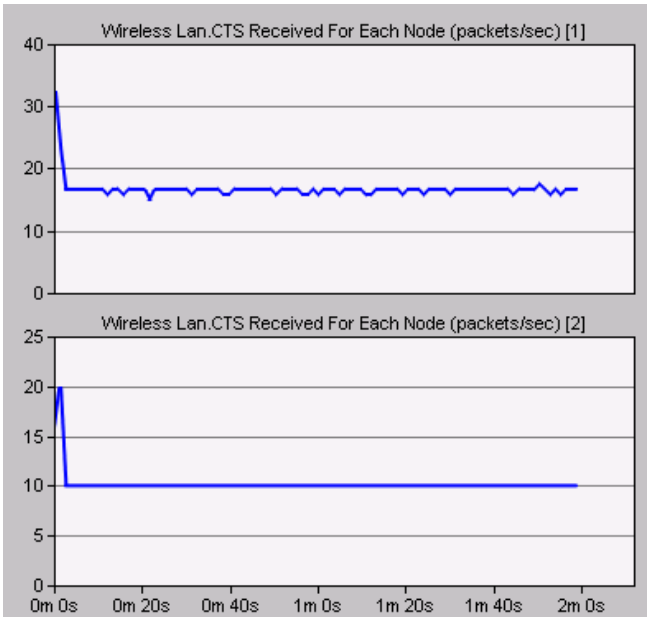

Figure 2-23: 20 Nodes

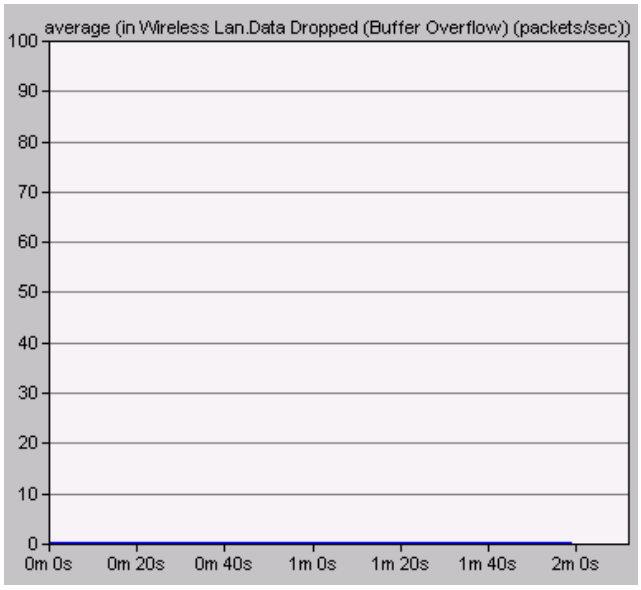

Figure 2-24: 20 Nodes

g. Low level with Two Attackers:

Similar to the previous scenario, level where the raw traffic generated by each node is low, only the network configuration with 50 nodes achieved the detection (Figure 2-25), the two upper graphs are representing the CTS packet rates for the two Attackers compared to an honest node at the bottom. Other network

configurations (5, 10, and 20) nodes, the detection was not achieved (Figure 2-26) due to the availability of the channel bandwidth. In each Figure, we observe that the CTS received rates are almost in the range of each other (the two attackers are on the top and the honest node is on the bottom) Figure (2-27) shows that the internal resources of the honest nodes (10 nodes configuration) were not affected and are represented by the rate of the Data dropped because of the buffer flow which is equal to zero. In this scenario, False detection = 0.

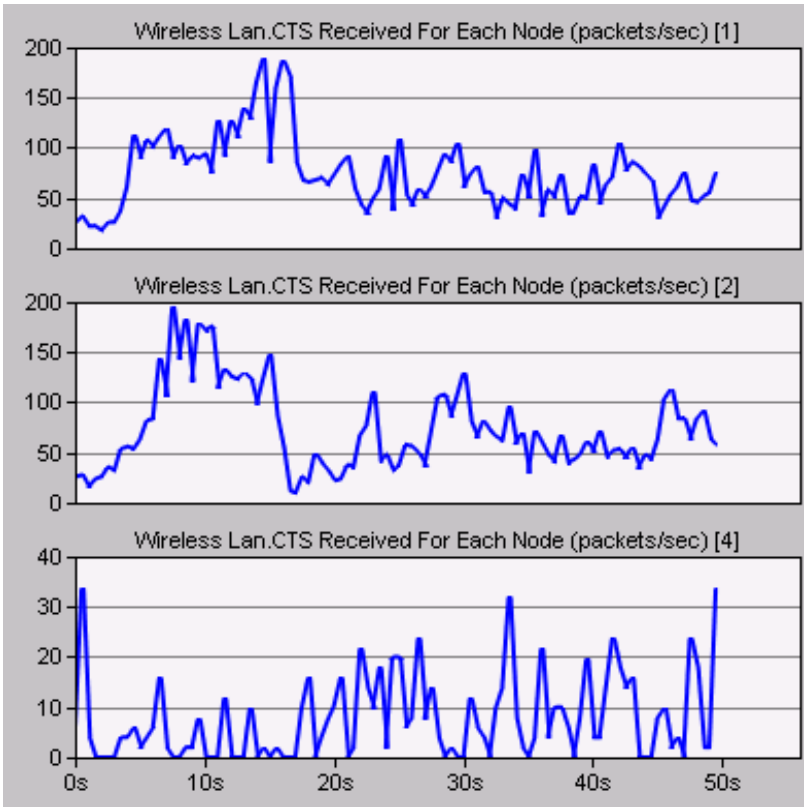

Figure 2-25: 50 Nodes

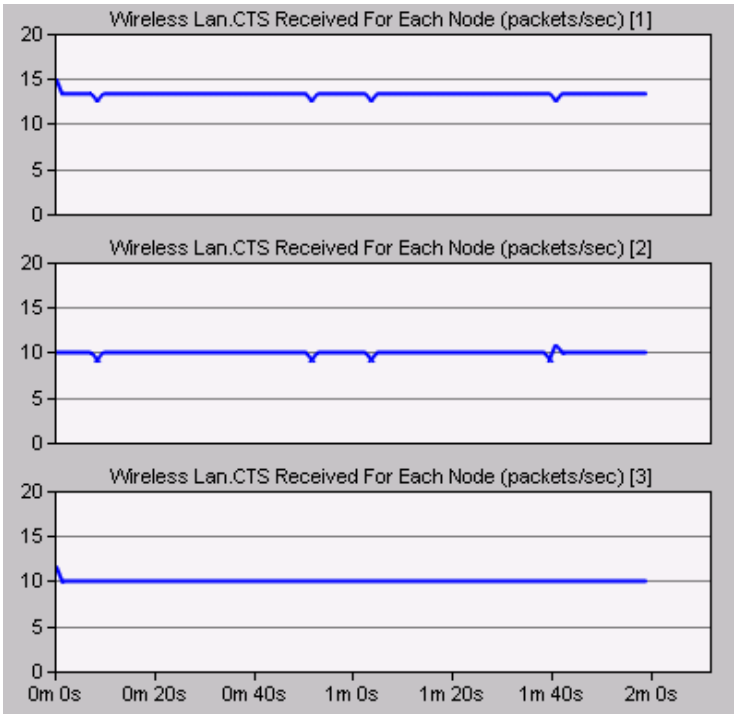

Figure 2-26:5 Nodes

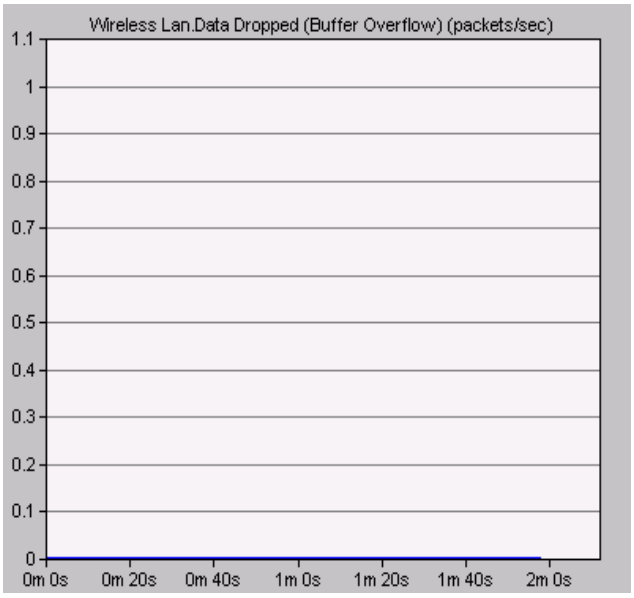

Figure 2-27: 5 Nodes

B. Mobile Environment:

The simulation was performed in a heavy traffic environment which represents saturation condition where every node always has a packet to transmit. All the nodes were moving randomly with a constant speed of 60 Km/h with a "Default Random Waypoint" mobility profile; figure (2-28).

The simulation environment resembles a battlefield where the personnel have their individual communication equipment and constantly exchanging information while aboard a military vehicles with average speed of 60Km/h [13].

The three different IEEE 802.11 protocols were simulated individually so all the nodes in the coverage area operate on the same protocol and rate. The algorithm was tested using four configurations for each IEEE 802.11 protocol: (5, 10, 20, and 50 nodes). Each node becomes a policing node by itself and is capable of making the decision independently. Detection results were chosen from random nodes in the configuration. For the purpose of showing the concept, the simulator had a code added to print "Node XX is an Attacker" whenever an attacker is detected, where XX is the MAC address of the attacker.

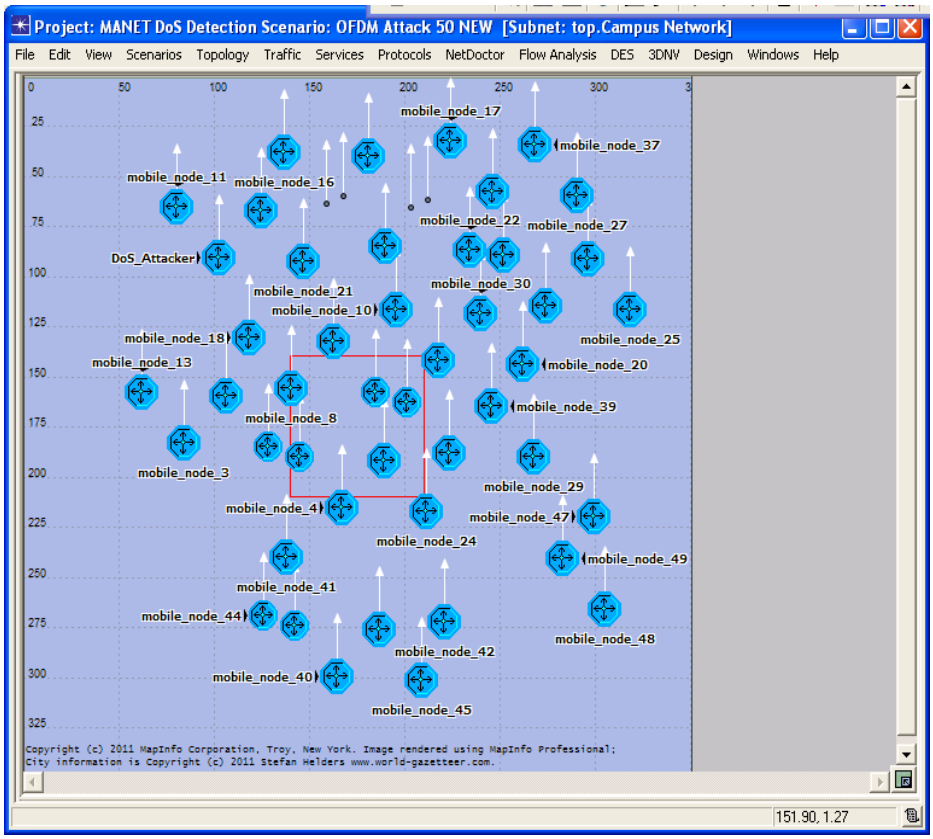

Figure 2-28: 50 nodes configuration

Table 2-1: Detection Thresholds employed in the simulation

Number of Nodes	FHSS (CTS Rate)	DSSS (CTS Rate)	OFDM (CTS Rate)
5	21	115	305
10	10	55	115
20	5	25	60
50	2	25	20

Results displayed below are taken from innocent nodes statistics and recordings to the network. In every graph, the Blue line represents the number of CTS packets heard by this innocent node for a node that was

identified as an attacker when compared the average value of received CTS coming from this suspicious node to the theoretical value computed using Markov Chain model. Some graphs below display other sets of received CTS packets from other innocent nodes (Green and Red lines if where applicable). The difference is noticeable between CTS packets received from innocent nodes and the attacker. The algorithm can display the MAC address of the attacker after identifying an average of received CTS packets that are larger than the average of CTS packets computed theoretically. The simulation did not generate any false positives in any configuration simulated and that goes back to the fact that the simulated results will never be higher than the theoretical results which are being used as a base of detection.

1. Frequency Hopping Spread Spectrum (FHSS) – IEEE 802.11

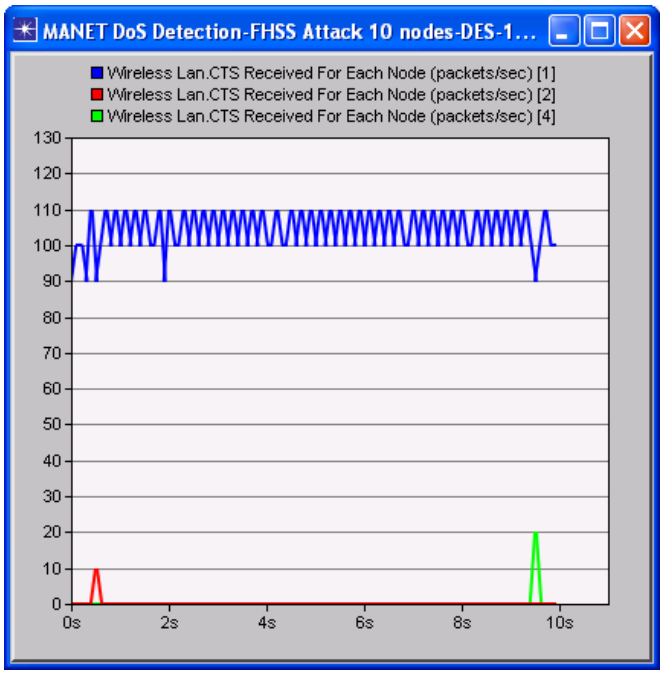

Figure 2-29: FHSS Attack – 10 Nodes

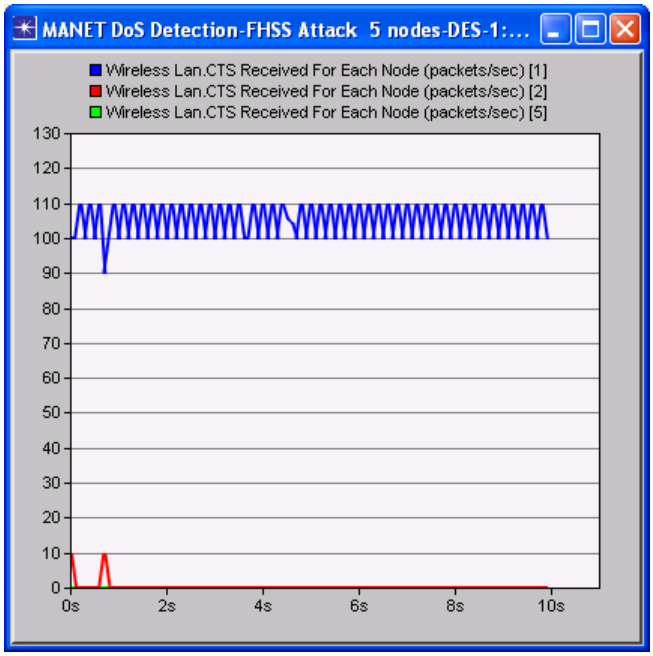

Figure 2-30: FHSS Attack – 5 Nodes

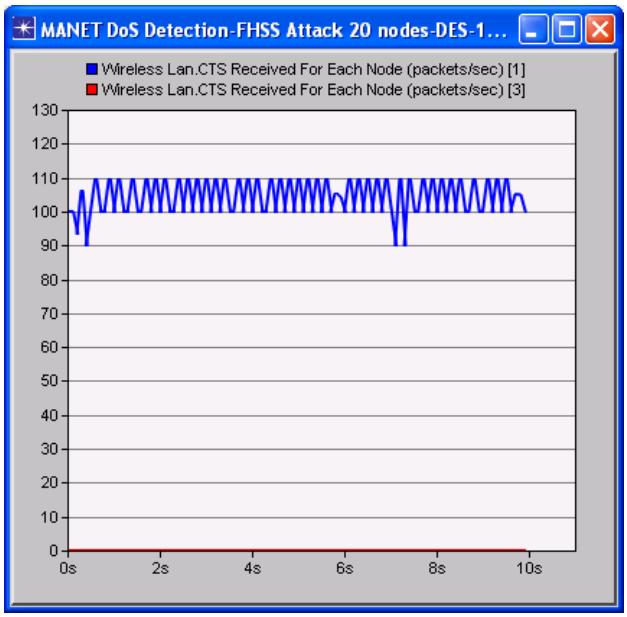

Figure 2-31: FHSS Attack – 20 Nodes

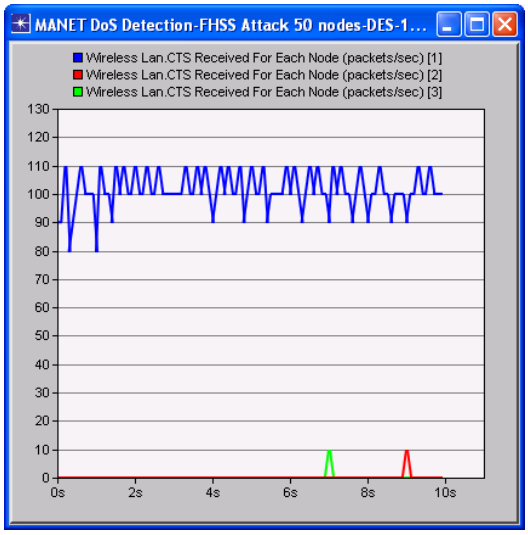

Figure 2-32: FHSS Attack – 50 Nodes

Figures (2-29 to 2-32): FHSS Simulation (5, 10, 20, and 50 nodes)

In every case above a randomly chosen node could successfully sort and detect the DoS attacker while all nodes are in motion.

2. Direct-Sequence Spread Spectrum (DSSS) – IEEE 802.11b

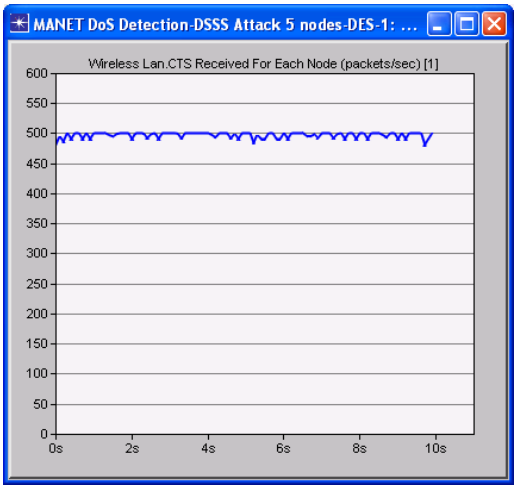

Figure 2-33: DSSS Attack – 5 Nodes

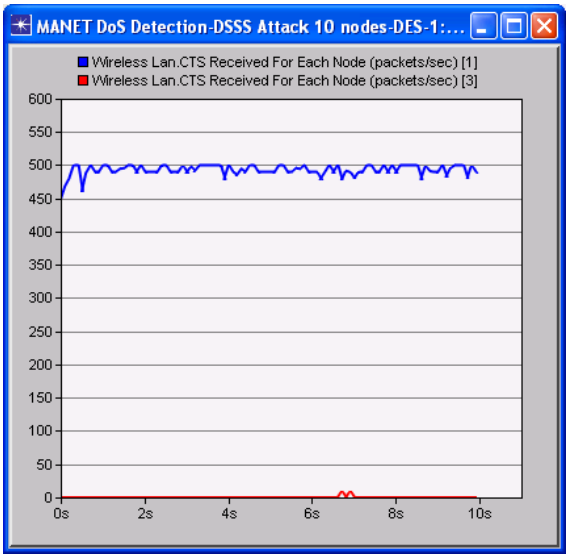

Figure 2-34: DSSS Attack – 10 Nodes

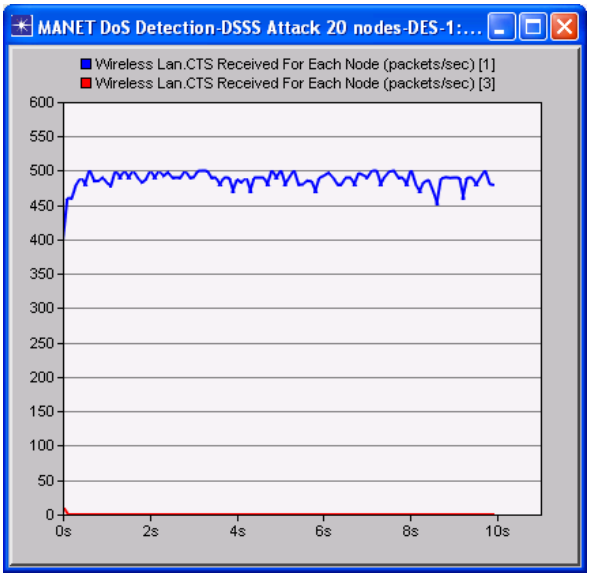

Figure 2-35: DSSS Attack – 20 Nodes

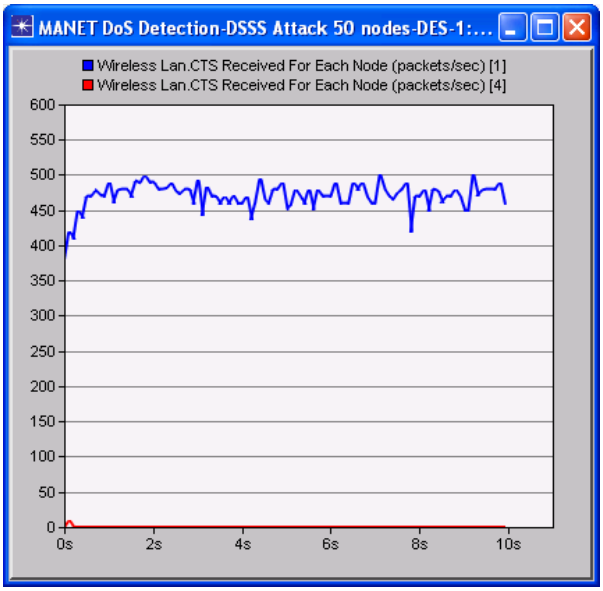

Figure 2-36: DSSS Attack – 50 Nodes

Figures (2-33 to 2-36): DSSS Simulation (5, 10, 20, and 50 nodes)

In every case above a randomly chosen node could successfully sort and detect the DoS attacker while all node s are in motion.

3. Orthogonal Frequency Division Multiplex (OFDM) – IEEE 802.11g

In every case below a randomly chosen node could successfully sort and detect the DoS attacker while all node s are in motion.

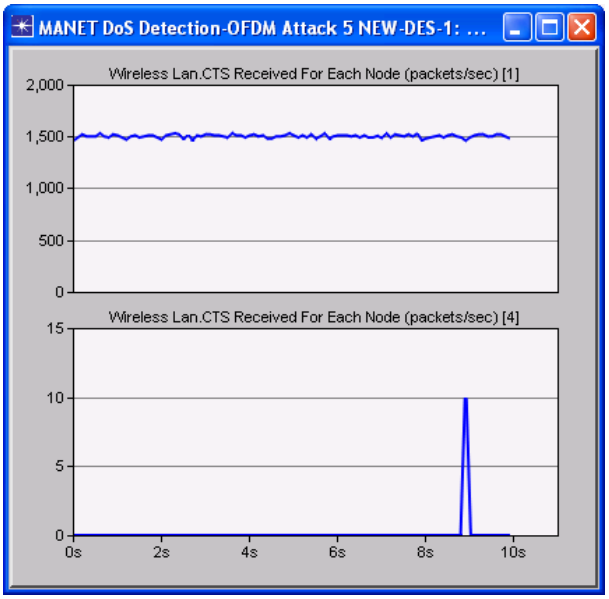

Figure 2-37: OFDM Attack – 5 Nodes

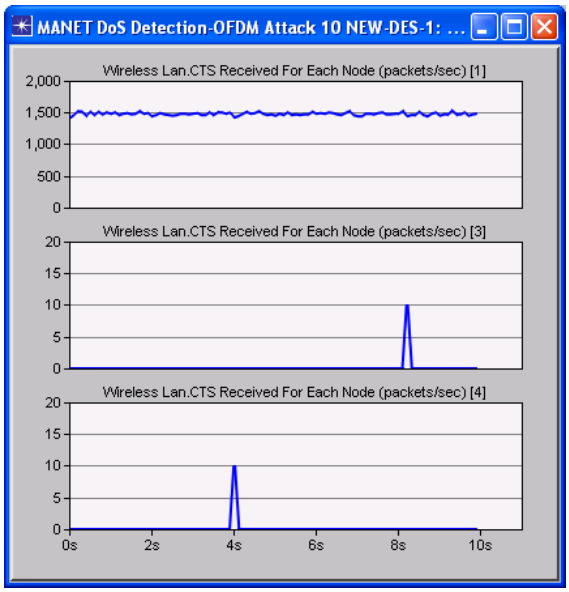

Figure 2-38: OFDM Attack – 10 Nodes

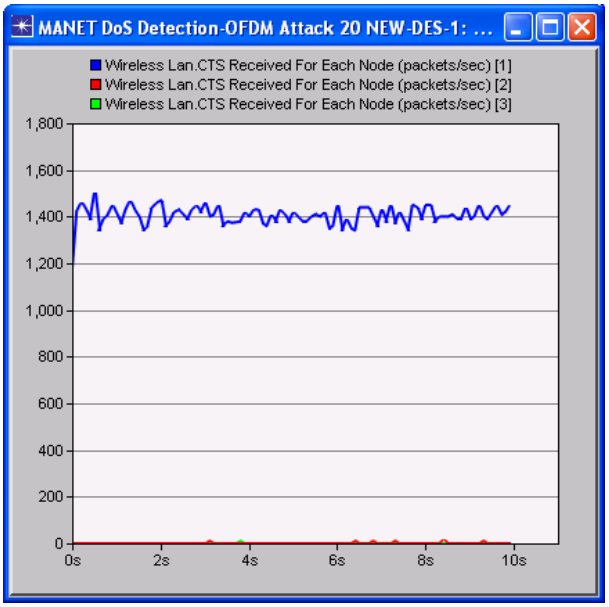

Figure 2-39: OFDM Attack – 20 Nodes

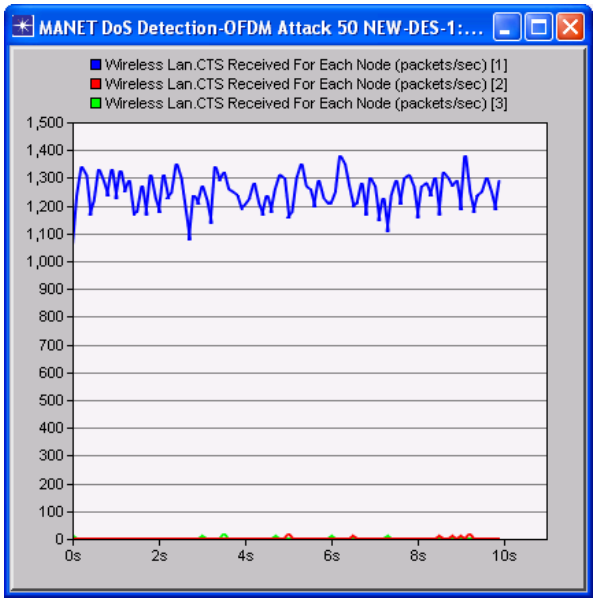

Figure 2-40: OFDM Attack – 20 Nodes

Figures (2-37 to 2-40): OFDM Simulation (5, 10, 20, and 50 nodes)

The algorithm detected all the attacking nodes in every scenario without reporting any positive results.

2.7 Conclusion

The chapter presented a novel approach [45] to detect a cheating node that is trying to illegally increase its throughput on the expense of the honest nodes present in the network. The approach was based in utilizing the numerical results obtained by solving the Markov Chain. Combining the numerical results with the specifications of the IEEE 802.11 DCF RTS/CTS protocol, IEEE 802.11 code was modified to enable each node to detect the attacker. The simulation results proved our concept with very high accuracy. The algorithm was validated using fixed and mobile environments and three different modulation technologies (DSSS – FHSS – OFDM).

Chapter 3 : Wi-Fi Channel Attack Detection and Avoidance with Single Access Point

3.1 Introduction

Wireless Fidelity (Wi-Fi) employing IEEE 802.11 DCF protocols are the most widely used wireless access technique. The goal of this chapter is to introduce the problem of channel capturing by a hacker in a single access point network. A detection technique and an avoidance strategy to reduce the impact of the hacker on the network is presented. IEEE 802.11 DCF employs CSMA/CA mechanism to reduce the probability of collisions inside the network. When a node desires to transmit a packet, it will send a broadcast RTS (Request to Send) packet; if the destination is ready to receive, it will send a CTS (Clear to Send) packet. All nodes inside the network will hear the RTS and CTS and will adjust their NAV (Network Allocation Vector) accordingly so they refrain from transmitting during the NAV set period to avoid collision. If collision happens during the RTS transmission, the node will back off a number of time slots and will try again, the process will be repeated till the packet is transmitted successfully or the window reaches its maximum value and the packet is discarded. Upon successful transmission, the receiver sends an ACK packet to the sender to complete the four-way handshake. The time is divided into slots of equal time duration. The duration of the slots depends on the hardware and IEEE 802.11- Frequency Hopping Spread Spectrum (FHSS) standard used in the network and is set equal to the time needed at any node to detect the transmission of a packet from any other node within the network. Slot time accounts for the propagation delay, the time needed to switch from the receiving to the transmitting state and for the time to signal to the MAC layer the state of the channel (Busy Detect Time). Slots define the interframe space (IFS) intervals, which are the minimum duration between any two packets transmitted by the same node. When a node gets a packet from the upper layers to transmit into the wireless medium, it first senses the channel and if the

channel is busy, the node waits until the channel becomes idle for a distributed interframe space (DIFS) period, and then computes a random back-off time. The random back-off time is specified by an integer value that corresponds to a number of time slots. The idle period after a DIFS period is the contention window (CW). Transmission only occurs at the beginning of each time slot. Nodes that have data packets ready for transmission select a back off value based on the contention window as the following [Back off = int (CW×rand×slot time)], where "rand" is a random number uniformly distributed between 0 and 1, and CWmin < CW < CWmax, where CWmin is the minimum CW, and CWmax is the maximum CW and their values depend on the hardware and IEEE standard used in the network. Initially, the node under discussion computes a back-off time in the range [0, CWmin - 1], where CWmin is the minimum contention window size. When the medium becomes idle, after an additional DIFS period, nodes decrement their back-off timers until the medium becomes busy again or until the timer value reaches zero and transmits the packet.

A network hacker will modify the Network Card Interface (NIC) firmware to deviate from the normal back-off mechanism mandated by the IEEE standard. This substantially increases the successful probabilities of its packet transmissions. The hacker node will only back off one slot every time it has a packet to transmit or when it experiences a collision. As a result, the hacker will capture the channel (bandwidth) and prevent others from sharing it. We simulated the hacker using OPNET and studied the effect. Figure (4-1) shows the difference between the amount of traffic sent by the hacker (blue line) and any other node in the network (red line). All the traffic loads used in this chapter are saturation loads meaning that every node, including the hacker, always has a packet in its buffer ready to be transmitted into the channel. Figure (4-2) shows the relationship between the load and the traffic dropped at the buffer of an innocent node of a network with 10 nodes using the channel. The graph shows that all innocent nodes will be prevented from communicating with the WiFi Access Point (AP).

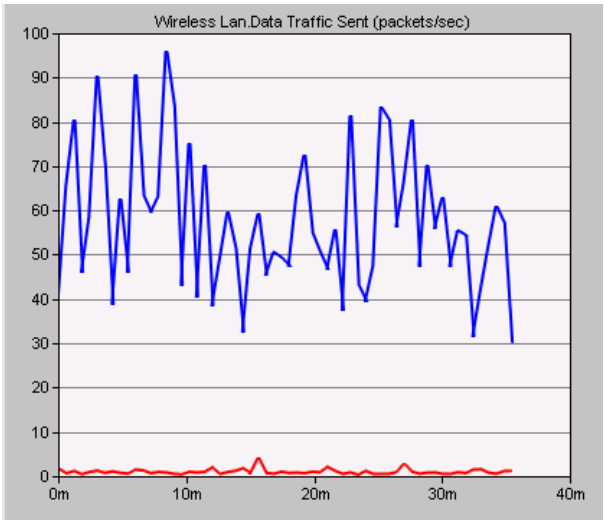

Figure 3-1: Packets Sent Rate

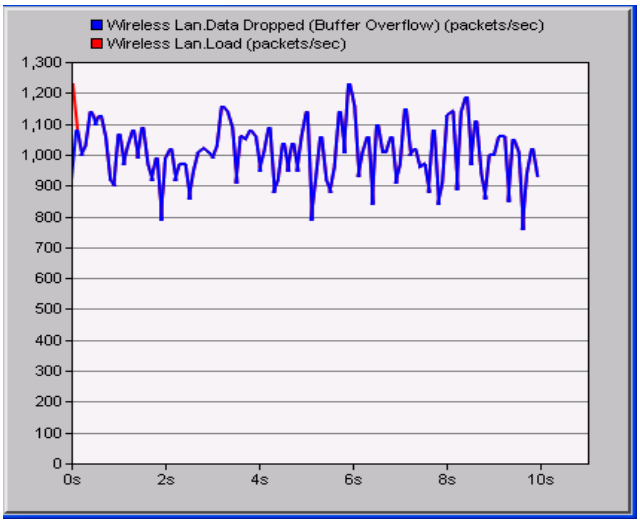

Figure 3-2: Attack Impact on the WiFi Channel

3.2 Simulation

The presented solution [42] is to mathematically calculate the maximum achievable throughput using Markov Chain modeling and monitor each node to detect if any node is capturing the channel and utilizing all the bandwidth between that node and the AP (access point) with the presence of other innocent nodes. Bianchis model is used to determine the maximum achievable throughput. The first step is to calculate the

(Transmission Probability – Tp) then derive the throughput for the whole network as a function of Tp, finally obtain the individual throughput for each node. the number of nodes inside the network is constant. b(t): stochastic process representing the back off time counters for a given node. (t and t+1) correspond to the beginning of two consecutive slot times. The back off time counters of each node decrements at the beginning of each slot time. (L) is the maximum number of back off stages. W = CWmin and CWmax = 2^n CW. Wi= 2^k W, where k $\in$ (0,L), which is a backed off stage. S(t) is the stochastic process representing the back off stage (0,...,L) of the node at time t. At each transmission attempt, and regardless of the number of retransmissions experienced, each packet collides with constant and independent probability (1-S) = p, whereas S is the probability of transmission success by the node itself and (1-S) is the conditional collision probability which is the collision experienced by a packet being transmitted.. By solving b(t) and S(t) using the Markov chain , we obtain the Throughput.

Ptx is the probability that there is at least one transmission in the slot time.

Psc is the probability that there is at least one packet being successfully transmitted in a slot time

Throughput (Packets/Second) = (Psc.Ptx)/((1-ptx).Sigma + Psc.Ptx. Tsuccess + Ptx (1-Psc)Tsuccess

Markov Chain results were modeled using MATLAB to determine the maximum theoretical throughput achievable under saturation condition and compared with the results obtained via the network simulation as shown in Figure (3-3).

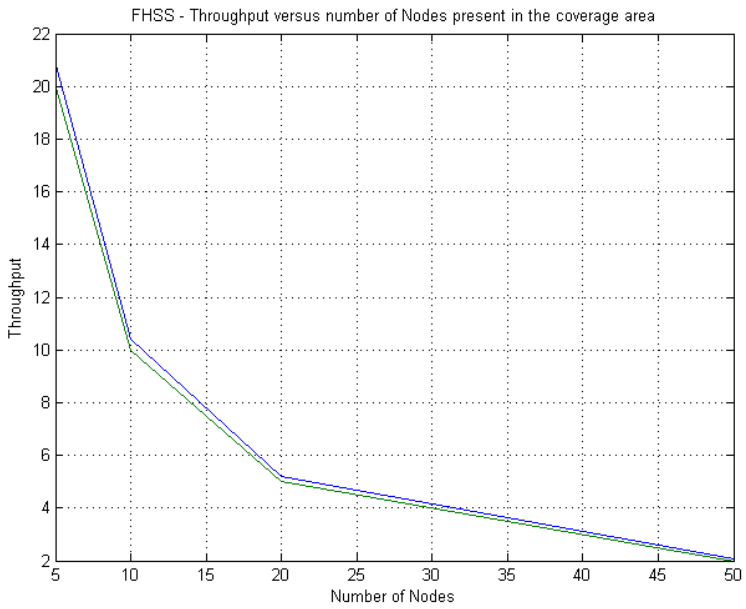

Figure 3-3: Comparison between the Throughputs (packets/Sec) obtained by solving Markov Chain and OPNET simulator. (IEEE 802.11- FHSS)

The theoretical results will be used as the maximum number of packets sent by an innocent node to the AP. Every node will monitor the network via listening to the CTS (Clear to Send) packets by the hacker and will compare the rate of CTS packets sent per second to the theoretical rate and if the sent rate is larger than the theoretical rate then this node that sends packets more than the theoretical rate will be marked as a Hacker and will be identified via its MAC address. The CTS rate is a function of the number of nodes in the channel.

Table 3-1: FHSS Detection Baselines

Number of Nodes	FHSS (CTS Rate)
5	21
10	10
20	5
50	2

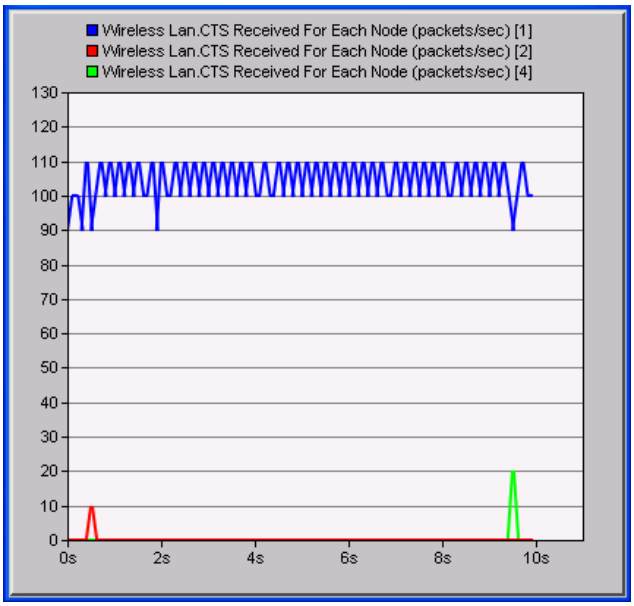

Figure 3-4: Traffic with 10 nodes

Figure (3-4) represents the number of CTS packets heard by an innocent node. For this instance the innocent node heard three different nodes and one of them is the Hacker which is presented by the blue line in the graph which passed the number of packets indicated in Table (3-1) along the other two innocent nodes (red and green lines).

Once a node detects the attacker Figure (3-5) , it will command the whole network to shift to another communication channel based on a preshared sequence for channel hopping. The preshared sequence can be disseminated only to the trusted nodes and allow other nodes to temporarily join the network. If one of those that temporarily join the network is a hacker it will be isolated and marked via its MAC address.

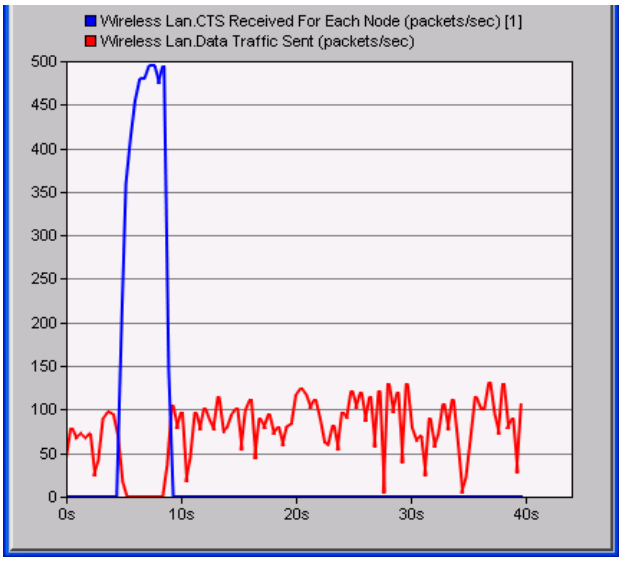

Figure 3-5: Attack Isolation

Chapter 4 : IEEE 802.11 DoS Attack Detection and Mitigation Utilizing Cross Layer Design

4.1 Abstract

Denial of Service (DoS) attack is a powerful attack that disrupts the network and deprives the legitimate users from utilizing the network resources. DoS attacks could be implemented to target different OSI layers, in this chapter we are focusing on DoS attacks that target the MAC layer in wireless networks. In this chapter we present a complete solution using Cross Layer Design techniques to detect and identify the attackers and to mitigate the attack by minimizing the negative impact on the network. DoS attacks sophistication could range from plain attacks which does not require any adaptation or intelligence during the attack like the signal jamming to a sophisticated attacks where the attacker is intelligent and aware of its surroundings and constantly modifying its behavior during the attack to appear as a legitimate node to avoid detection. In this chapter we are focusing on the sophisticated DoS attack in wireless networks using IEEE 802.11 DCF protocols [1][2][3], where the attacker is striving to appear as a legitimate member of the network and fully joined the network group and posses for instance the spread sequence or the channel coding scheme. The algorithm will be examined in a fixed and mobile environments with multiple PHY layer techniques (DSSS, FHSS, and OFDM) using different MAC layer protocols (IEEE 802.11, IEEE 802.11b, and IEEE 802.11g). DoS attackers illegally modify the IEEE 802.11 DCF standards and modify the MAC code on their communication equipment to capture the channel by maximizing the packet transmission success to a degree where all other legitimate node will have zero percent success for their packet transmissions. This type of DoS attack will generally result in bandwidth starvation and extreme power and CPU processing consumption to the legitimate nodes in the network. Two dimensional Markov Chain will be modeled to obtain the maximum throughput to identify the DoS attackers and the rest of the

presented algorithm will mitigate the impact of the attackers while deceiving the attackers and make them falsely believe that the attacks are still disrupting the network so they do not resort to modifying the attacking techniques. The algorithm will be validated using network simulations under different condition using different technologies.

4.2 Introduction

Denial of Service attack in wireless networks could be implemented by multiple ways [8]. This chapter addresses a very specific and sophisticated DoS attack type. The DoS attacker here disguises himself to be a legitimate user and pretends to be following the IEEE 802.11 standards and actually following the communication protocol to some limit, so it generates properly formatted control and data packets which make all other innocent nodes in the network consider him as a peer legitimate node. In addition, the attacker presents himself to the network to be having valid information to share with everyone. The goal of this chapter is to present a complete algorithm to detect, identify, isolate, and react to the DoS attacker to mitigate its negative impact on the network. The DoS attacker in this chapter manipulates the back off timer to illegally maximize the successful chances of transmitting packets which appear to be legitimate data packets but in fact it does not contain any valid data. The MAC layer protocol used in this chapter is IEEE 802.11 DCF which is based on CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) that specifies two methods for packet transmission. The basic method is a two-way handshaking method called "Basic Access " where every node attempts to send data packet and if the transmission is successful, the destination node will send an ACK (Acknowledgement) packet to the source node, otherwise the source will keep attempting to retransmit. The second method applies a four way handshaking mechanism "Request-to-Send/Clear-to-Send (RTS/CTS)." Each node has a packet to transmit will sense the medium

first to apply collision avoidance mechanism. If the node senses the medium and it is clear, the node will transmit a RTS (Request-To-Send) packet, and then it will be followed by CTS (Clear-To-Send) packet originating from the destination node. Data packet will be sent to the destination node and will send to the source an ACK packet to complete the four way handshaking mechanism. If the medium was sensed busy, each n ode will back off for a random period and will count down to zero, once the counter reach zero, the node will attempt to retransmit the packet. The focus of this chapter is the RTS/CTS mechanism. The presented algorithm consists of detection module which applies two dimensional Markov Chain modeling to obtain the base detection threshold for the number of successful packet transmissions. The second module of the presented algorithm which consists of the DoS attack mitigation technique applies a novel method for dynamic channel allocation while fooling the attacker to make him believe that he is still attacking the whole network, so the attacker does not get alerted and change the attacking methodology. The presented algorithm would provide the capability to group of trusted nodes to allow foreign nodes to join the group for valid data exchange and at the same time will enable the trusted group to detect and combat the DoS attack in the event that the allowed foreign node is an attacker. Because of the randomness in choosing a back off value for each node, detecting the back off manipulation is a very challenging task [9], [10], and [11]. The majority of the detection techniques discussed in the literature assume that the network topology with an access point which are considered trusted nodes [10][12] which are tasked to monitor all the nodes in the network to detect any misbehavior. Markov chain modeling provides accurate theoretical measure of the network throughput [4][6][8][34]. Our approach in this chapter assumes that network is distributed network with no centralized authority to monitor and each node functions as a police node without the cooperation from any other nodes inside the network. Other methods of detection mechanisms that were proposed in the literature are assuming that the attacker will cooperate which is not coincide with the reality when the attacker is trying to disrupt the network [8]. Many methods were proposed in the literature to combat and

prevent the MAC layer misbehavior and back off timer manipulation. In [13] the authors proposed a method to force all the nodes to use a known back off timers by appending some values to the existing data packet structure to combat this misbehavior but this method assumes that the misbehaving node or the attacker will cooperate which is not the case in a real organized attack. Channel hopping techniques [14][15][16] were described in the literature to avoid DoS targeting the physical layer which constitutes signal jamming but the presented techniques were primarily targeting a jamming attack with no intelligence where the attackers just continuously transmitting a continuous signal with a sensible power level on a certain frequency band as opposed to our case in this chapter where the attackers are intelligence and strive to appear as legitimate node and to some extent follow the IEEE 802.11 DCF protocol and comply with the control and data packets construction and format. The authors in [17] presented a channel hopping concept that takes place in the MAC layer based on sending a beacon on a non-jammed channel when a node senses a jamming signal and other nodes will follow to the other channel to detect the beacon and resume the communication, but again this concept is dealing with simple jammers not intelligent ones that can adapt and move to the new channel to jam it if it senses that all nodes moved away from the jammed channel. In [18] the authors proposed a channel hopping mechanism to countermeasure the DoS impact on the network but they built their algorithm on a multi-radio networks and did not deal with a single radio networks. In [19] [20] [21] and [22] the concept of frequent channel hopping was introduced so the active communication channel would change every fixed interval that ranges from few milli-seconds to few seconds. In [23] the authors proposed a channel hopping scheme based on trusted AP that would coordinate the frequency hopping sequence inside the network whereas our presented algorithm is completely distributed and does not require a centralized authority like a trusted AP.

4.3 PHY Layer Modulation Techniques

This chapter presents the Detection and Mitigation algorithm and applies it into three different PHY technologies; FHSS, DSSS, and OFDM. Basic information [1][2][3] about the Channels and the Frequency bands used will be briefly discussed to better help understanding our presented algorithm.

1. IEE 802.11 - Frequency Hopping Spread Spectrum (FHSS)

FHSS operates in the 2.4 GHz band with a range starting from 2.402 GHz to 2.480 GHz. Each channel has a width of 1MHz. Support two rates of 1Mbps and 2Mbps. There are 78 hopping sequences and each sequence would use 79 hops. 15 systems could be collocated and work independently with minimal amount of collisions [29].

2. IEEE 802.11b - Direct Sequence Spread Spectrum (DSSS)

DSSS operates in the 2.4 GHz band. Each channel has a width of 22. The rates defined in IEEE 802.11 are 1 Mbps and 2 Mbps and the rates in IEEE 802.11.b standard are 5.5 Mbps and 11 Mbps.

The channel to frequency mapping is shown in Figure (4-1). Only the first 11 channels are used in the United States and this is our assumption when running the simulation.

3. IEEE 802.11g - Orthogonal Frequency-Division Multiplexing (OFDM)

OFDM operates in the 2.4 GHz band. IEEE 802.11g supported rates are: 6, 9, 12, 18, 24, 36, 48, and 54 Mbps. The channel to frequency mapping is shown in Figure (4-1). Please note that Figure (5-2) is just a schematic for visualizing the channels and it does not reflect the reality of the OFDM sides which are sharper than the DSSS sides to reduce the interference between the channels. Only the first 11 channels are used in the United States and this is our assumption when running the simulation.

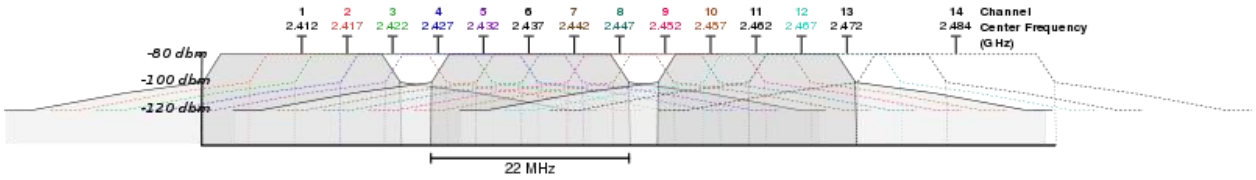

Figure 4-1: IEEE 802.11g Channel Distribution [49]

Each IEEE 802.11 standard [1][2][3] has specific set of design parameters hat dictates the design of each system.

The simulation disabled the "CTS-To-Self" option in simulating IEEE 802.11g since it is not accounted for in modeling the Markov chain. It is worth noting that the difference between the theoretical and simulation results in the OFDM technology is little higher than the difference in FHSS and DSSS cases, however the values will be valid for the detection process because the DoS attacker will have much higher data packet transmission rate as will be shown in the simulation results.

The following three figures (4-2, 4-3, and 4-4) will show the comparison between results obtained by modeling Markov chain and results obtained by the simulation to validate our models and assumption. These values in will be used as the detection threshold, if any node has data transmission rate that the values shown in the graphs will be marked as an attacker.

The three PHY layer technologies that were modeled are:

a. Frequency Hopping Spread Spectrum (FHSS) – IEEE 802.11
b. Direct-Sequence Spread Spectrum (DSSS) – IEEE 802.11b
c. Orthogonal Frequency Division Multiplex (OFDM) – IEEE 802.11g

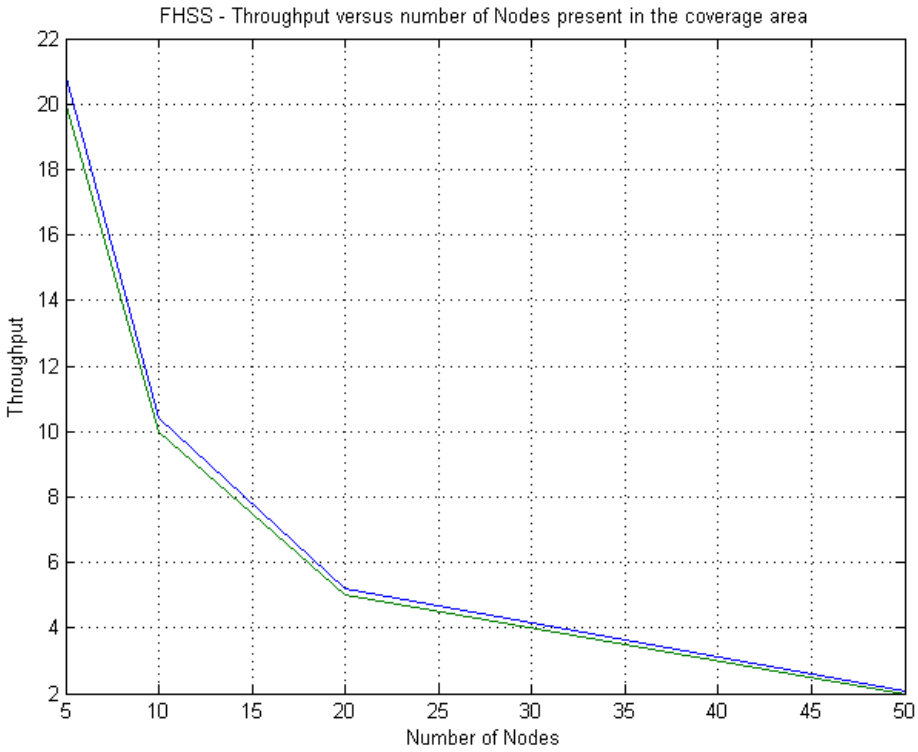

Figure 4-2: IEEE 802.11- FHSS

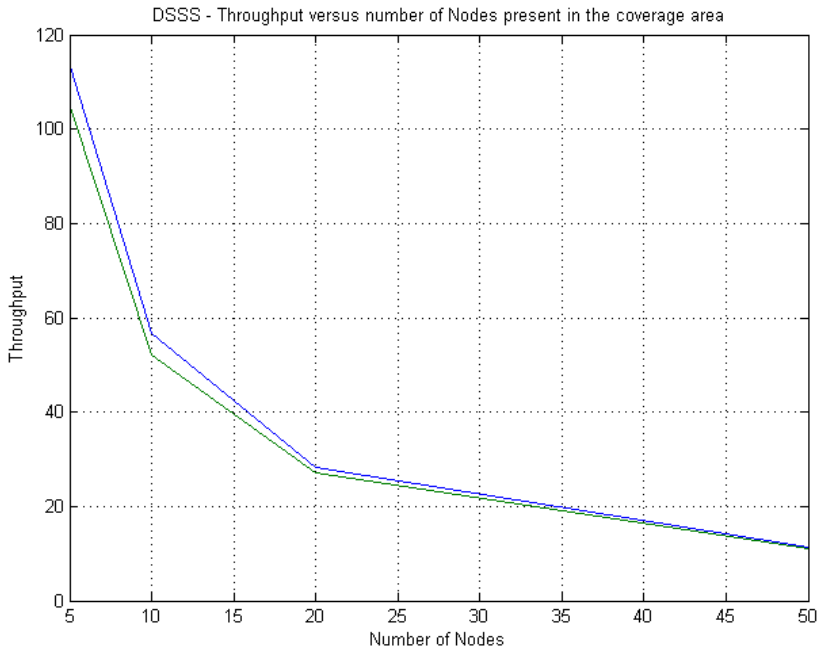

Figure 4-3: IEEE 802.11b- DSSS

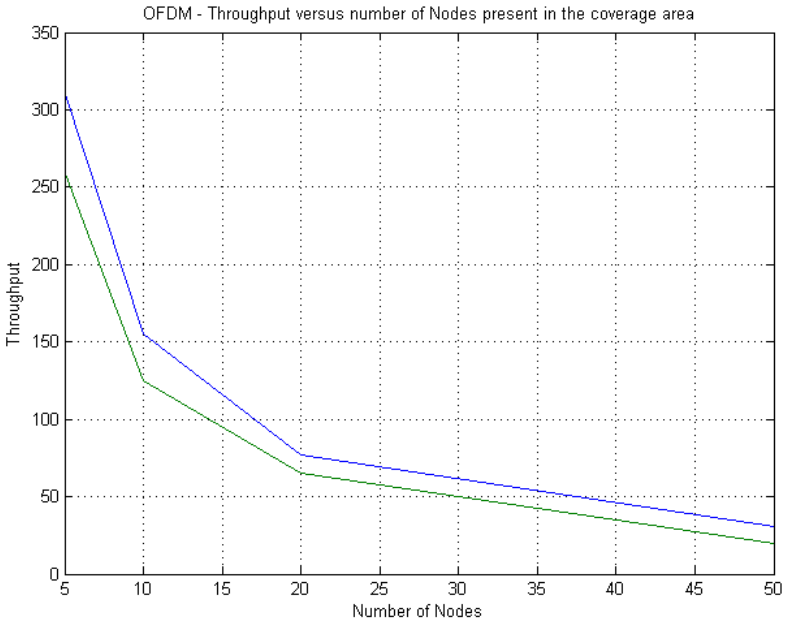

Figure 4-4: IEEE 802.11g- OFDM

4.4 DoS Attack model and Impact

The DoS attacker in this chapter is an intelligent attacker who partially follows the IEEE 802.11 standards has and presents the appearance of a legitimate node that wants to join a group of nodes so it can start a valid exchanging of data packets in a distributed network topology with no centralized authority to validate and authenticate non member nodes that want to join. The attacker will follow the control and data packet formats (RTS, CTS, ACK, and Data) and all the PHY layer parameters (SIFS, DIFS, Slot Time, PHY and MAC headers) to further deceive the legitimate nodes inside the network. The attacker's behavior is intelligent and is not a simple signal jammer where a pulsing or continuous signal is transmitted with high power to jam a frequency band but the attacker follows the appropriate traffic exchange four-way handshake model. The attacker will start communicating normally with the nodes and shortly after will activate the DoS attack mechanism and targets a node and bombards it with RTS and Data packets to a

achieve channel capturing. As long as the attacker is getting ACK messages from the victim, it will keep the attack going. The packets received by the victim do not contain useful data but follows the format of a legitimate packet. The attacker could implement his mechanism by modifying the (Network Interface Card) NIC on his communication machine (i.e. Laptop). DoS attack will result in bandwidth starvation, excessive power consumption, and exhaustive CPU processing. The worst impact is felt in an ad-hoc wireless environment where all the resources (BW, Power, and CPU Processing) are scarce due to the nature of mobility and the lack of infrastructure. The attacker manipulates the Binary Exponential Back off (BEB) algorithm and does not backoff according to the BEB algorithm but instead it always backs off 1 slot and retries to transmit.

The impact was simulated in fixed and mobile environment with all three standards (IEEE 802.11, IEEE 802.11b, and IEEE 802.11g). The simulated mobility throughout this chapter consisted of nodes moving randomly with a constant speed of 60 Km/h with a "Default Random Waypoint" mobility profile. Each case was simulated with different number of nodes to show the impact in different topologies. All nodes including the attackers were configured to generate the same rate of data packets in the upper layer (Application) which is considered the Load.

Figure (4-5) shows the difference between data sent by the attacker and a normal node. Although the Load is the same, the graph shows that the attacker is transmitting packets 500 times more than the normal node.

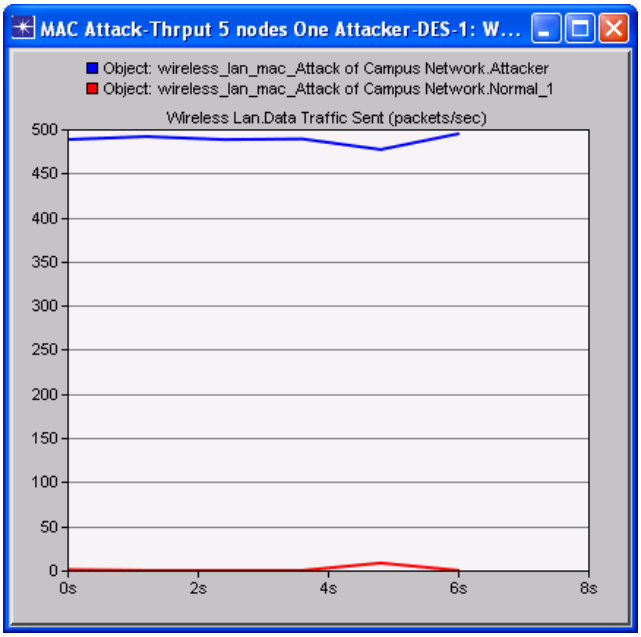

Figure 4-5: IEEE 802.11b- Fixed – 5 nodes

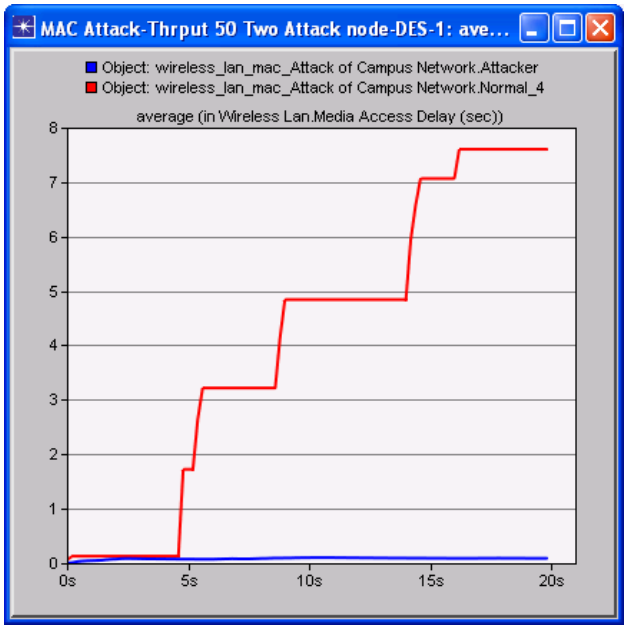

Figure 4-6: IEEE 802.11b- Fixed – 50 nodes

In Figure (4-6), we find the Media Access Delay encountered by the normal node which is more than 70 times the delay encountered by the attacker.

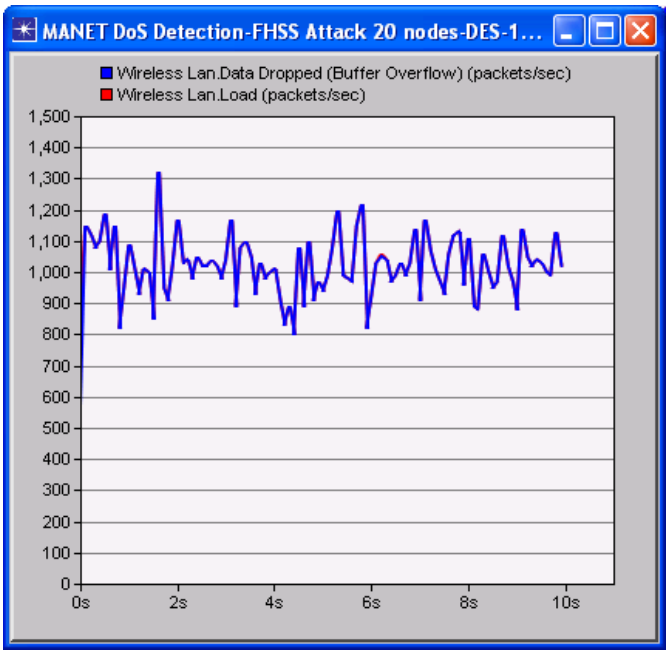

Figure 4-7: IEEE 802.11- Mobile – 20 nodes

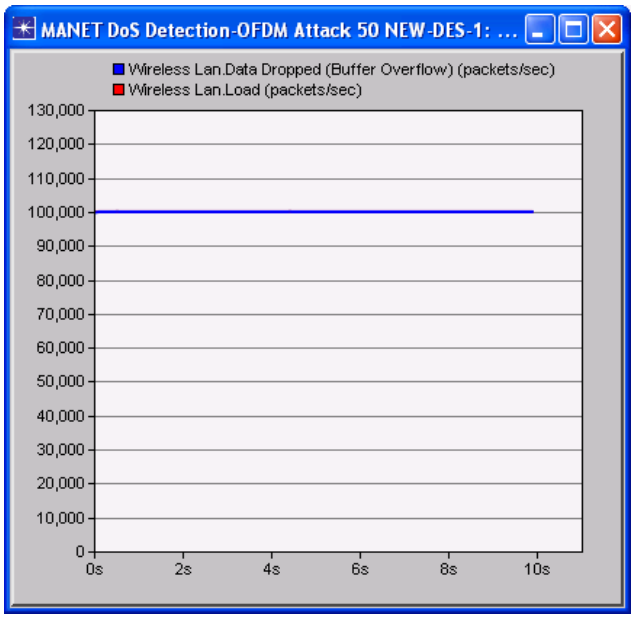

Figure 4-8: IEEE 802.11g- Mobile – 50 nodes

In Figures (4-7 and 4-8), the graphs show that 100% of the Load is dropped before it was sent because of Buffer Overflow which means this node could not send out any data. Note that the two lines coincide.

From the above graphs, the impact of the DoS attack is very obvious and affected all three IEEE 802.11 protocols with different sizes of the network and the results were shown in fixed and mobile environments.

4.5 The Algorithm

The algorithm consists of two modules; first is the detection algorithm and the second is the mitigation algorithm that utilizes Cross layer Design. The algorithm functions as one protocol that is added to the IEEE 802.11 MAC layer code and once it is activated will automatically detect the attackers and will react without any manual assistance from the users. The algorithm will work over the span of the MAC and PHY layers and will control some aspects in both layers but will primarily reside in the MAC layer. The presented code will be only implemented on the trusted nodes' MAC layers and will not be shared with others outside the trusted group. The modified MAC layer code will still be 100% interoperable with any other IEEE 802.11 MAC that follow the regular standards. The trusted members of the group will have the option either to enable or disable the algorithm.

1. The Detection module of the Algorithm:

The detection algorithm utilizes the IEEE 802.11 DCF standards [1][2][3] to perform the detection with some modification to the IEEE 802.11 MAC layer code. The algorithm is totally designed for a distributed architecture, so every node will act on its own to reach the same conclusion about the detection. The concept depends on the fact that number of successful packets sent by a certain node is almost equivalent to the CTS packets received by this particular node in a IEEE 802.11 DCF RTS/CTS operation – Figure (5-13). The algorithm modifies the MAC layer firmware to enable each node to police the network within the receiving range and enables each node to detect a DoS attacker by utilizing the baselines obtained by solving the

Markov Chain which are shown in Chapter 2 Each node listens and sorts the received CTS packets by the destination MAC address, then the moving average is constantly calculated and matched by the theoretical results obtained by Markov chain modeling and once the moving average rises above the theoretical value, then this node will declare the attacking node as an attacker and will be identified by its MAC address.

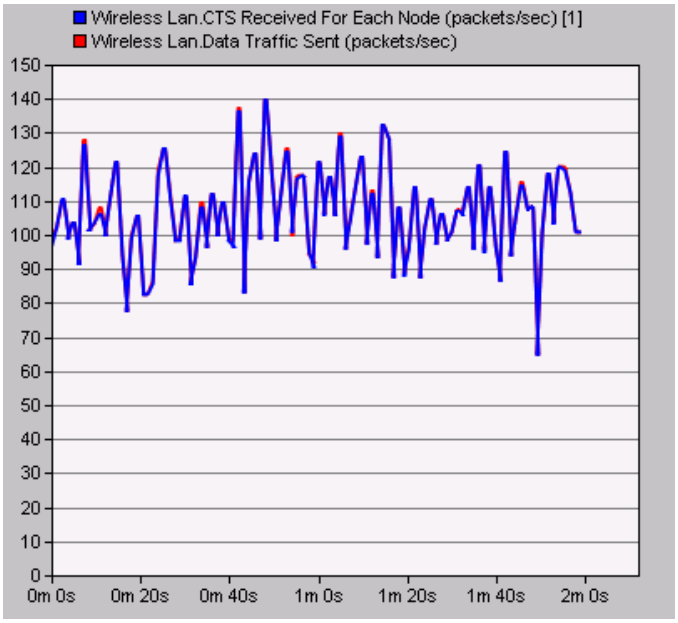

Figure 4-9: Comparison between number of CTS and Data packets for the same node

2. The Mitigation module of the Algorithm:

Once the Detection module determines the presence of the attacker and identifies the attacking node based on its MAC address, the second module kicks in and starting to change the communication channel based on a Pre-Shared-Sequence (PSS). In this chapter we assume that the PSS will be disseminated manually by the users. All trusted nodes in a group will possess a Pre-Shared-Sequence (PSS) and will not share the PSS with other nodes that want to join the group to exchange valid data for a short period of time. The PSS is

basically a variable function that will tell each node what is the next secure channel to go to resume the communication away from the DoS attacker. Since our algorithm is dealing with intelligent attackers, which will target a single node (victim node) in the network to establish communication with and will send massive amount of RTS and Data packets to block the whole channel for all other nodes, the algorithm will force all nodes in the network to change the communication's channel to another secure except the victim node. So after all innocent nodes will resume the communication on the new safe channel, the victim node will still be on the old channel exchanging communication with the DoS attacker and sending CTS and ACK packets so the attacker is deceived to think that the whole channel is captured and no other node can send or receive any information. Since this is a distributed environment every innocent node will come to the same conclusion applying our algorithm.

PSS Design: the PSS is a randomly generated function to determine the next safe channel when the PHY layer is DSSS or OFDM or next hopping sequence when the PHY layer is FHSS that all nodes except for the victim nodes would move to. The function design should take into consideration the interference factor. For DSSS and OFDM, the next safe channel should be a non-overlapping channel with the previous selected channel to avoid unnecessary interference. For FHSS, the next sequence should be a non-interfering sequence.

Number of innocent Nodes that can survive DoS attack:

N: Total Number of Innocent Nodes

M: Number of DoS Attackers

V: Number of Victim Nodes

According to the algorithm: $M <= V$

At any given time, the number of communicating nodes that survived the DoS attack = $N - V$

The presented algorithm will only function when [$(N - V) >= 2$]

3. The Algorithm:

The code will reside in the IEEE 802.11 MAC layer by the normal nodes in one group and will have the same PSS and it is activated.

```
START
/* Start Detection Module*/
Count n          /*" Number of Nodes in the network"*/
Create n Counters
Calculate Maximum Individual Throuput /* obtained from Modeling Markov Chain*/
When Receive CTS
If (Destination Address = My Address)
Do Nothing
Else
{
Update Counter (Destination Address)
 Calculate Rate
    /* number of CTS received per second for each Destination Address */
  }
  If
CTS_node_x rate < Maximum Individual Throuput
Do Nothing
Else


"node_x is an Attacker" /* Start DoS Mitigation Module*/
```

{

If

Source Address (CTS_node_x) = My Address

Do Nothing

*/*I am a Victim and communicating with the DoS attacker, then do not change the channel and keep talking to the Attacker to save the rest of the nodes */*

Else

*Invoke **PSS** function /* Run the PSS equation*/*

*Get **PSS**_result*

/ Number of next safe channel for DSSS and OFDM, and number of the next Hopping Sequence for FHSS */*

*Send **PSS**_result to PHY Layer*

/ In the PHY layer*/*

*Change Rx and Tx Channel/Sequence to **PSS**_result*

/ Resume communication on the new Channel/Sequence */*

}

END

4.6 Simulation Results

The algorithm was simulated with multiple scenarios to cover the assumptions that were presented in this chapter. All three PHY layer technologies were simulated separately under fixed and mobile conditions. Also, the simulation will show the presence of one and two attackers where they are consecutive to each other with few seconds between DoS activation in each node. The effects of the attacks without activating the algorithmare shown in in the next sections. The presented graphs will be a sample representation of the configurations and results and not presenting every configuration for the limited space in the chapter.

Data Rates used for the simulation for 10 nodes:

- IEEE 802.11: 1 Mbps
- IEEE 802.11b: 11 Mbps
- IEEE 802.11g: 24 Mbps

Figures (4-10, 4-11 and 4-12) show the traffic sent by the innocent nodes (Red) and the CTS packets heard by the innocent nodes that were destined for the attackers (Blue). When the attackers activated their attacking mode which is the rising of the Blue lines in the graphs (almost at the 5th second), the innocent nodes were deprived from using the bandwidths and the traffic sent was zero for IEEE 802.11 and IEEE 802.11b and the traffic dipped severely but did not reach zero for IEEE 802.11g. the graphs show that IEEE 802.11g handled the DoS attack with more resilience than IEEE 802.11/b. when the innocent nodes heard the CTS packets which surpassed the thresholds destined to a node, they declared that hacking node as an attacker and reacted and changed their channels to continue communicating.

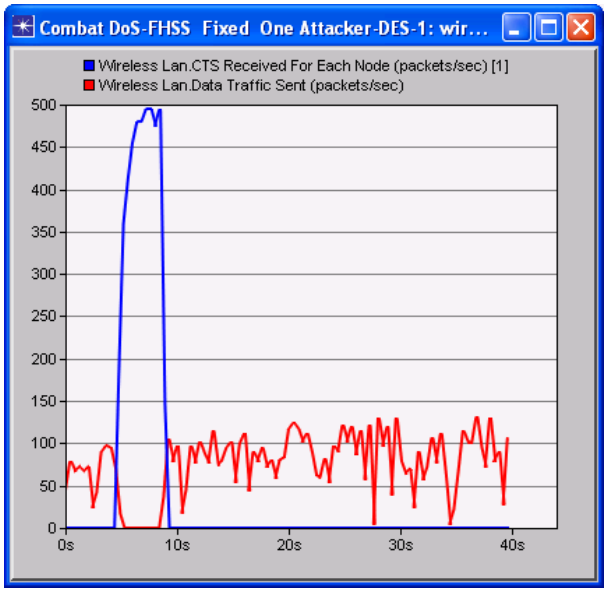

Figure 4-10: IEEE 802.11- Fixed –One Attacker

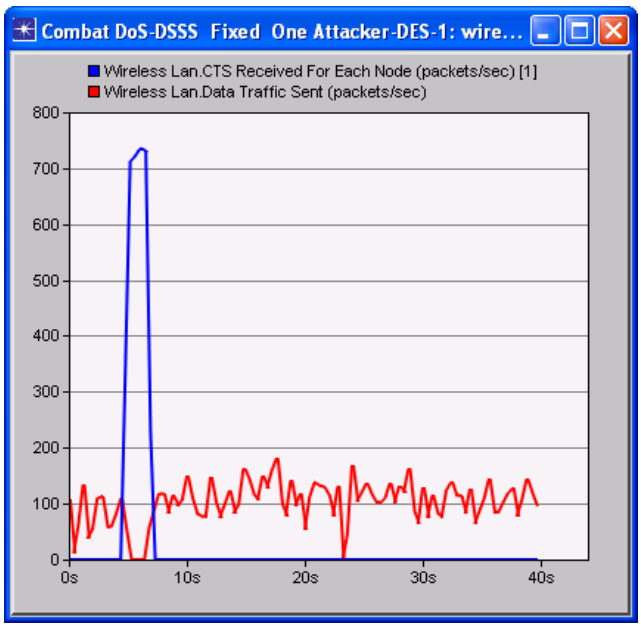

Figure 4-11: IEEE 802.11b- Fixed –One Attacker

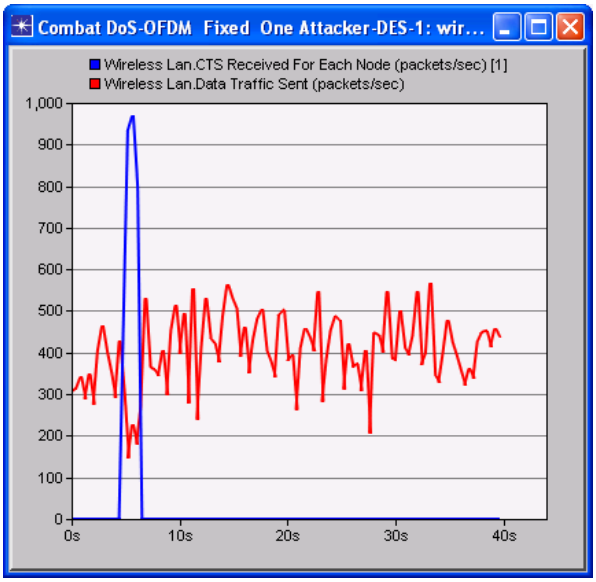

Figure 4-12: IEEE 802.11g- Fixed –One Attacker

Figure (4-13) shows the victim node which is the node that was picked by the attacker to communicate with. The victim node did not change the channel and kept the communication channel open with the attacker as a sacrifice to allow its peers in the network to have proper communication and not to alert the attacker that all nodes had left the channel to a safer channel picked by PSS. The graph shows that before the attacker was

activated, it had some data to send but once the attacker activated the DoS attack around the 5th second of the simulation, the victim's output traffic reached zero and all the Load traffic has been discarded because of buffer overflow.

Figure (4-14) shows the recovery time from the DoS attack, which is marked by the dip in the data sent for each MAC protocol type. It is very apparent that IEEE 802.11g using OFDM performed the best with a very short time to recover followed by IEEE 802.11b using DSSS and IEEE 802.11 using FHSS came last with the longest time to recover from the DoS attack.

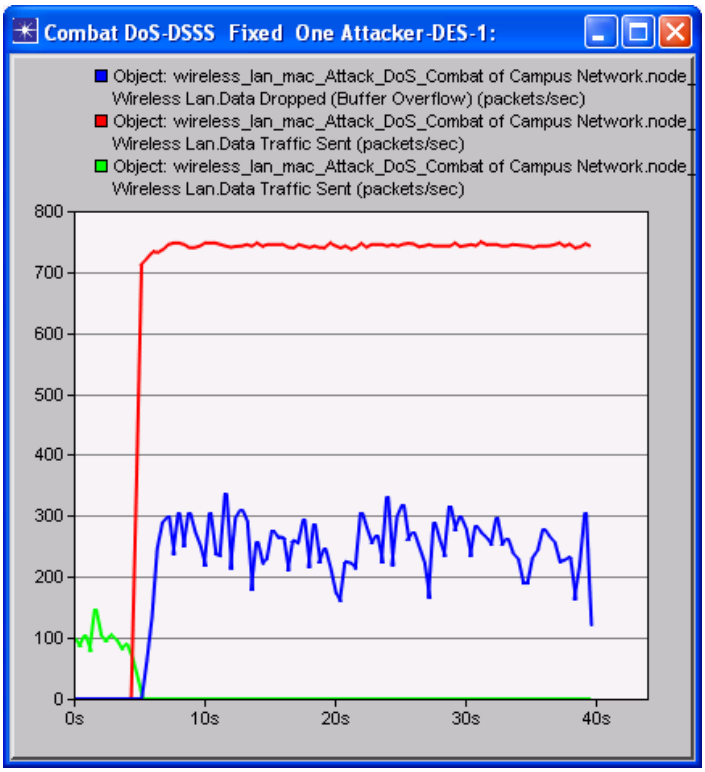

Figure 4-13: Victim Node and the attacker

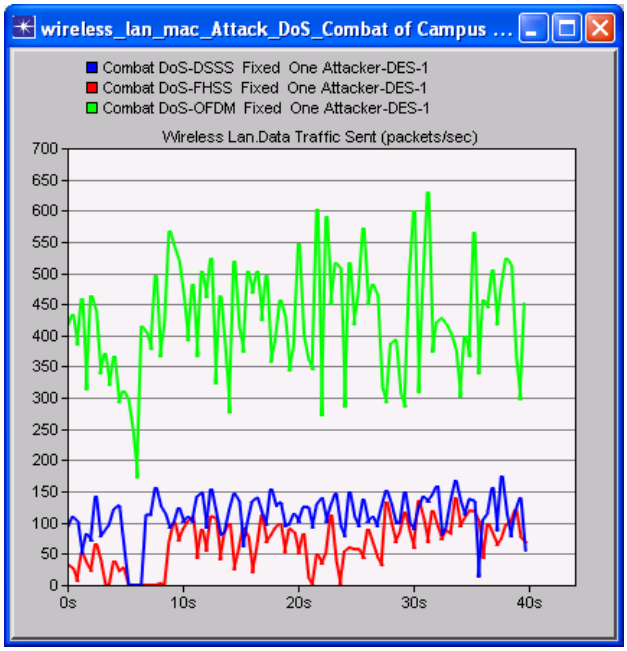

Figure 4-14: Recovery Times

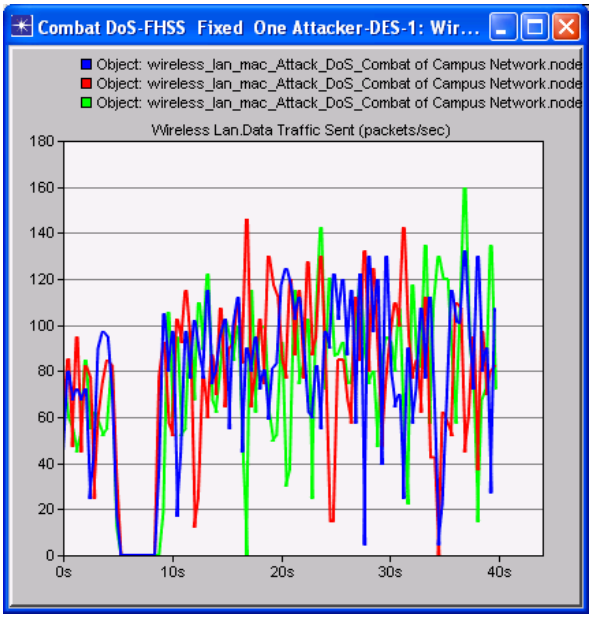

Figure 4-15: Recovery among different nodes

Figure (4-15) shows three different innocent nodes in the network and how each node individually detected and changed the communication channel without the coordination with other nodes. The graphs show that

all three nodes recovered and changed the channel around the same time (around the 8th second) which further proves our algorithm in a distributed environment.

Figure (4-16) shows the presence of two attackers that started attacking the network at two different times. The blue and red graphs represent the attackers destined CTS packets, and the green graph shows the traffic sent by the innocent node and how it was disrupted twice and reached zero when the attackers were present on the channels before the algorithm residing in the innocent node reacted twice. It is apparent that the second recovery took longer than the first one due the detection process which collects the average CTS packets over time that are sent to the attacker before reacting to avoid falsely declaring an innocent node as an attacker.

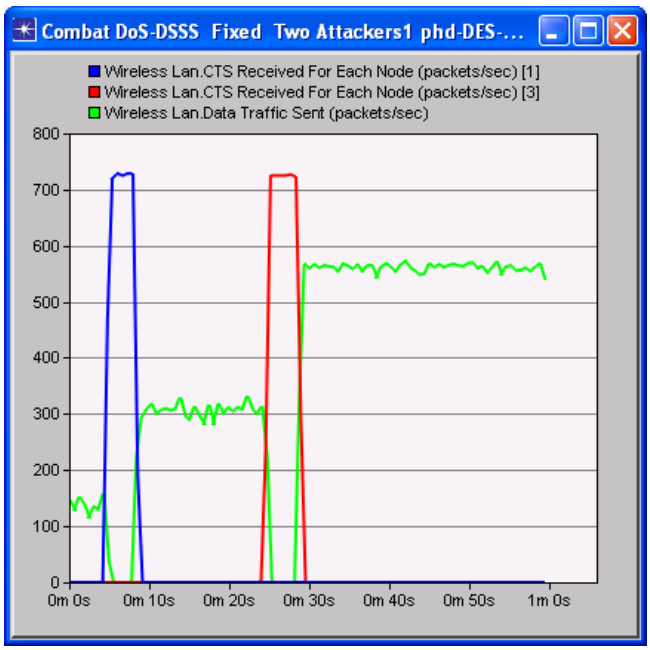

Figure 4-16: Two Attackers

Figure (4-17) shows the mobile environment for all three IEEE 802.11/b/g protocols, each in a different simulation scenario. The graphs show that the algorithm was successful in detecting the attackers and changed from the attacked communication channel to a safe one so it detected the attack that started around the 10th second and recovered for each scenario. The results in the mobile environments were similar to the

fixed environment. IEEE 802.11 g was first to recover with a significant dip in the rate of the traffic sent but did not reach zero, however if the attack continued without proper reaction, the rate of the sent traffic would have reached zero Figure (4-8). IEEE 802.11b recovered faster than IEEE 802.11 and both had their traffic dipped to zero for few seconds until the full recovery was achieved.

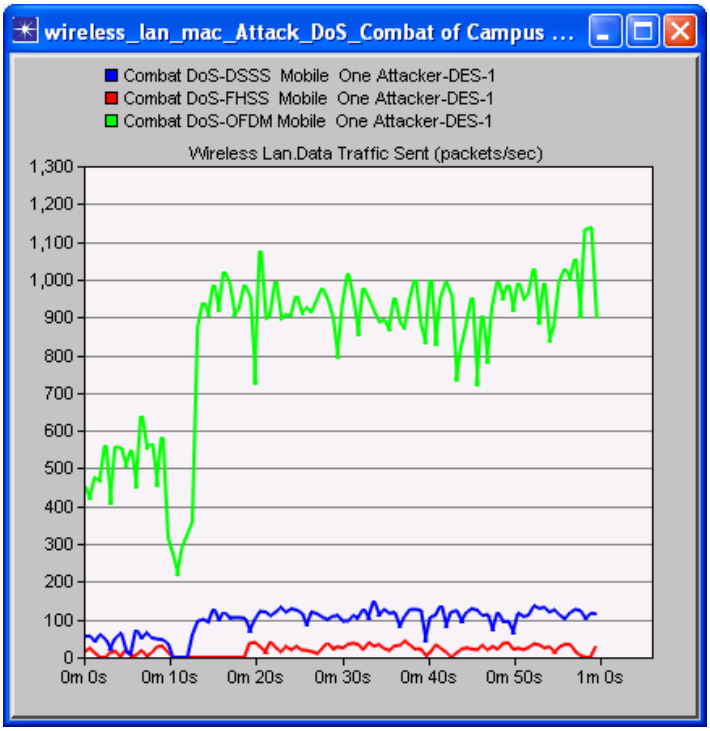

Figure 4-17: Mobile Environment

4.7 Conclusion

The chapter presented an end-to-end Cross Layer Design protocol [44] in a totally distributed environment to detect and react to DoS attacks targeting IEEE 802.11 MAC layers. The algorithm which utilizes Cross Layer Design techniques was successfully applied to different PHY technologies and it was effective in both fixed and mobile environment. The algorithm will enable any trusted group of wireless users to allow stranger users to join their networks without the fear of them being DoS malicious attackers since our

protocol will effectively detect and deal with them with minor communication interruption. The attackers were modeled as intelligent attackers where they will pose as legitimate users that have valid data to share with the group and will follow the packets formats to appear as complying node. The algorithm was validated by network simulations and proved successful and effective. Results obtained show that our algorithm had superior performance especially with IEEE 802.11g. Future research will be geared towards developing more robust mechanisms on the PHY layer technologies for faster channel switching. New methodologies for Cross Layer Design that will involve more layers will be investigated to increase the resiliency of the network against DoS attacks.

Chapter 5 : Byzantine Attack Isolation in IEEE 802.11 Wireless Networks

5.1 Abstract

This chapter introduces an effective solution against Denial of Service (DoS) implemented by byzantine attack in a fully distributed wireless network employing IEEE 802.11. Byzantine attack is the attack performed by a fully trusted node that's turned rogue and already has passed all the authentication and verification processes. When a trusted node is turned rogue, it can easily perform DoS attack on the media access control (MAC) layer to prevent other nodes from communicating. DoS attack is an easy and effective method to disrupt the communications. The byzantine attacker will alter the implementation of the IEEE 802.11 DCF standards to illegally increase the probability of having a successful packet transmitted into the channel on the expense of the other nodes that follow the protocol standards. The solution presented in this chapter depends on three stages. First stage is to identify the attacker using mathematical modeling. The second stage utilizes asymmetric cryptography to allow the good nodes communicate to agree on communicating on another frequency and excluding the byzantine attacker, and finally the third stage where the good nodes change the frequency via controlling their transmitters and receivers. The theoretical throughput will be generated using two dimensional Markov Chain to determine the network capacity. Results obtained by the theoretical computations will be used to constantly monitor the network and identify an attacker if present. A cross layer technique will allow the MAC layer to control the Physical layer to change the frequency of the communication session based on the MAC's decision of identifying an attacker.

5.2 Introduction

Byzantine attacks in general are very serious attacks because all the efforts to secure communication networks are directed towards outside threats. Nodes are authenticated as part of a group using (Wired Equivalent Privacy) WEP or (WiFi Protected Access) WPA and are trusted by the rest of the group members. When a trusted node is compromised and started to act maliciously to disrupt the communication in the network, this is called Byzantine Attack. The effect of Byzantine Attack is fatal on the network since the other nodes do not normally employ any further authentication techniques after joining the group. Byzantine attacks could happen in multiple situations like a college campus when students collaborate via peer to peer communication to transfer files and lectures. All students nodes in this case have already passed the authentication process by providing multiple authentication tokens (i.e., college network password, specific class password, and specific collaboration group password), yet one user turns rogue and start DoS attack by simply sending packets continuously into the channel and these packets do not carry any useful data. Another example is a hostile environment in the battlefield where a physical unit (vehicle) is captured and the enemy uses the wireless communication equipment to perform DoS attack on other units. This chapter deals with an intelligent and sophisticated DoS attack not simply jamming the signal on the physical level. The DoS attacker described in this chapter pretends to have valid information to send and appears to be following the IEEE standards (i.e., sending RTS, CTS, and ACK packets).

IEEE 802.11 DCF [1] specifies two mechanisms to perform packet transmission. The default mechanism is a two-way handshaking method referred to as “Basic Access “. This mechanism employs immediate transmission of an acknowledgement (ACK) packet by the destination node after a successful reception of a packet transmitted by the sender.

The second mechanism employs a four way handshaking procedure "Request-to-Send/Clear-to-Send (RTS/CTS)." Prior to transmitting a packet, a node set by using RTS/CTS mode "reserves" the channel by sending a special Request-To-Send short frame. The available destination node responds to an RTS frame by sending back a Clear-To-Send frame, and then a data packet transmission and ACK response will follow. The RTS/CTS mechanism increases the network throughput by reducing the duration of a collision when long messages are transmitted. In this chapter, our focus is on misbehavior detection in the four way handshaking mechanism "RTS/CTS scheme". Attackers employ several techniques to illegally increase their throughputs on the expense of other fair behaving nodes [8]. A compromised Peer to peer node using IEEE 802.11 will manipulate the Back-off timer to increase their probabilities in having successful transmissions by decreasing the Back-off timer value instead of following the Back-off timer generation method that all other nodes in the network are using. A node is considered malicious or misbehaving (attacking) when it does not follow the IEEE 802.11 MAC Standard [1]. The back-off interval is to be increased as per a specific set of rules prior to retransmission attempts that are invoked upon failed transmission attempts. A malicious node may choose a small and/or a constant back-off interval prior to the transmission of a data packet or follow a completely different retransmission strategy upon experiencing failed transmissions that does not conform to the standard IEEE 802.11 protocol rules. Due to the randomness in choosing a back off value, it is considered that detecting the back off manipulation to be one of the most challenging topics [27], [9], [10], and [11]. The goal of this chapter is to detect and isolate any compromised node.

Most researches that dealt with the Byzantine attack problem in Peer to peer networks focused on the Network layer [35][36][37][62][63][64][65][66][67][68][69][70]. Many others dealt with the detection of the back-off timer deviation in wireless networks where there are trusted Access Points [12][10], where a trusted AP will regulate the senders back-off timer values and detect the misbehaving nodes. Due to the

nature of peer to peer, where there is no centralized authority that will assign and monitor each node's back-off timer values, the task is very challenging. The presented algorithm is designed to work in a distributed environment where there is no centralized authority or a supervisor node that is monitoring every communication event in the network.

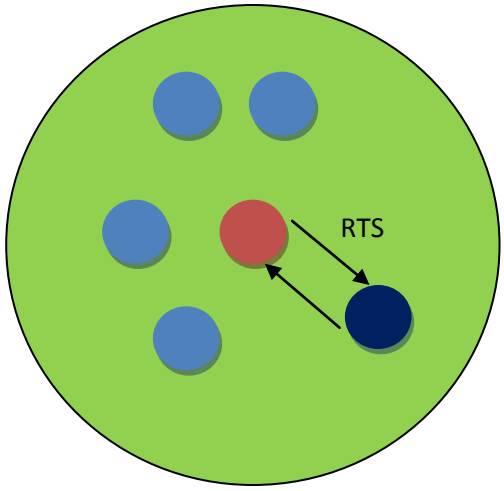

Figure 5-1: Byzantine Attack

In Figure (5-1), the compromised node (dark blue) picks one node randomly (red) to perform the attack and capture the channel. The rest of the nodes (blue) won't be able to communicate.

In [8] the authors assume that every node will cooperate and announce the state of its pseudo-random sequence generator so other nodes would monitor its behavior. This approach assumes much cooperation from compromised nodes which is not realistic in most situations. The presented algorithm does not expect any cooperation from any node hence eliminating the chance of receiving wrong information by a malicious node. In [28] the authors introduced a new parameter to indicate the level of cooperation of each node. In [29] a method was proposed to make the Access Point functions as a watchdog to monitor all nodes' behaviors. Assigning one node or selected nodes to police the network is a very dangerous concept and

creates a single point of failure in case the police node is compromised itself. In [32] the author is proposing to analyze the distribution of inter-delivery times between two consecutive successful transmissions. This method is very challenging and requires very accurate measuring clocks in the order of micro seconds to accurately detect the selfish behavior.

5.3 Byzantine DoS Attack

The DoS attacker will update the firmware on its Network Interface Card and the updated firmware contains the malicious code that would deviate from the IEEE standard when it performs the back-off calculations. At this point the attacker will appear to acting normally and follows the protocol since the attacker will still use the same control messages (RTS, CTS, and ACK) but in reality the attacker is capturing the channel, and is only backing off one slot every single time it has a packet to transmit or when it experiences a collision while the other nodes which are honest will follow the exponential back off mechanism.. The rest of the nodes in the network will be clueless about the attack since the current IEEE 802.11 standard does not provide such detection method. The modulation technique used in this chapter is direct Sequence Spread Spectrum (DSSS). The attack is simulated using OPNET [5] to show the effect on the rest of the nodes. The payload size used throughout this chapter is 8000 bits so it could be sent in one time slot without the need of fragmentation. Maximum number of retransmission was configured to be 6 times before the packet is discarded. We configured all the nodes with the same parameters (data rate generated by every node, packet size and traffic generation interval). Multiple WLAN statistics were collected to show the effect of the misbehavior in the network – Traffic Generation Scheme (Exponential – 1000 Packets/sec generated by the upper layers). Figure (5-2) shows the throughput (successful packets/second sent by each node – attacker (blue line) and a normal node (red line), the difference is very significant. The attacker by simply modifying the MAC protocol effectively increased its throughput more than 20 folds in this scenario. Figure (5-3)

exhibits the difference in Media Access Delay in seconds between the attacker and the honest node. So the effect on the honest node is not only the decreased throughput but also the internal resources of the honest node are affected.

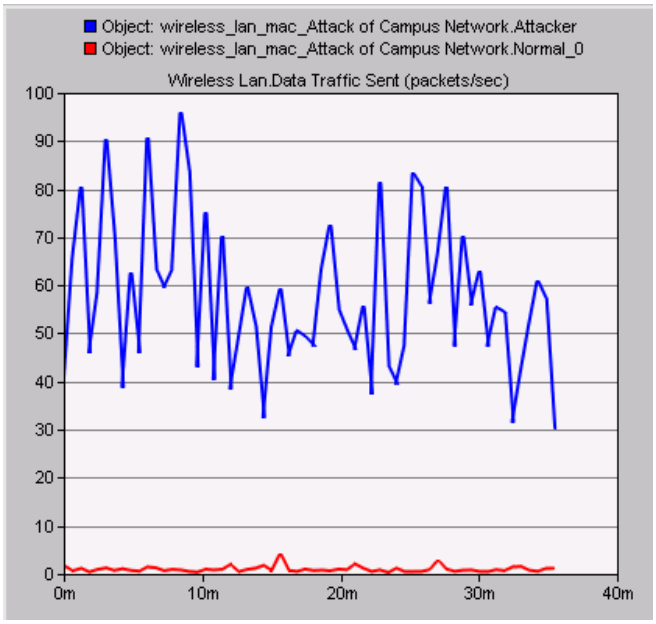

Figure 5-2: Attack-Innocent nodes Traffic Comparison

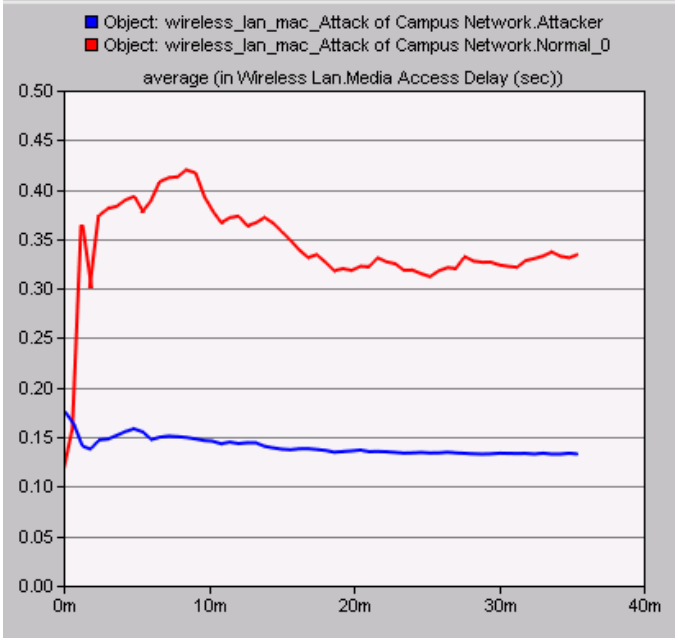

Figure 5-3: Delay Conditions

5.4 Detection and Isolation Process

From the IEEE 802.11 DCF RTS/CTS operation, it is concluded that the number of successful data packets sent by a given node, given that the wireless channel is perfect and there are no hidden nodes, are equal to the CTS packets received by this specific node with the permission to send data packets. The CTS packets are designed to be heard by every single node as presented in Chapter 2. The presented algorithm will utilize the number of CTS packets to detect the attacker. If a certain node has a number of CTS packets more than the threshold value determined by Markov Chain calculations destined to it that would mean that this packet is an attacker. The CTS packets are chosen to be our monitoring packet because it is meant for every node to receive it as per the standard and it is a small packet to process and contains the destination MAC address which could be a potential attacker. The algorithm will reside in every node and only the nodes that have the detection and isolation algorithm will be allowed into the communicating group. Asymmetric encryption technique [38][39][71][72][73][74] will be used by every node. So every node will send out its public encryption key as part of the modified RTS packets and will keep its private encryption key.

The RTS packet will be modified for the algorithm and it will contain two new fields. The first field will contain the public encryption key of the sender. The second field will contain the new frequency encrypted in case of an attacker present. Each node will construct a table (see an example in Table (5-1)) containing the MAC addresses in the domain and the public key for each node and both pieces of information is available by listening to the exchange of the RTS packets in the network. Once an attacker is detected, the node with the highest MAC address will send an RTS packet directed to the node that has the second highest MAC address. The RTS packet will include a new frequency channel that is randomly chosen by the node with the highest MAC address and the channel number is encrypted using the public key of the node with second highest MAC number. The node with second highest MAC number would read the value

of the new channel and encrypt it using he public key of the following node and will send an RTS packet to the following node and so on until a full circle is achieved. After sending out the new channel number, this node will shift its own channel frequency. Once every node sends the RTS packet to the following node, it shifts its transmitter and receiver to tune to the new frequency selected by the first node with the highest MAC address. In DSSS, there are 12 different channel frequencies [40], so the algorithm will have 11 choices every time an attacker is detected. Every node gets the modified RTS packet that commands the channel frequency shifting will verify the sender to make sure it is not the attacker sending false command.

OPNET [5] code was modified to hear all CTS packets individually and collect them in separate buckets depending on the destination address.

Table 5-1: Example of the table constructed by every node

MAC Addresses	Public Encryption Key
MAC_1	Key_1
MAC_2	Key_2

The Algorithm:

First Stage:

Begin

Hear RTS Packets

Construct Table (MAC Addresses, Public Encryption Key)

Count n //" Number of Nodes in the network"

Create n Counters

Calculate Maximum Individual Throughput /* obtained from Eq (6) above*/

When Receive CTS

If (Destination Address = My Address)

Do Nothing

Else

{ Update Counter (Destination Address)

Calculate Rate

/* number of CTS received per second for each Destination Address */ }

If

CTS_node_x rate < Maximum Individual Throughput

Do Nothing

Else

"node_x is an Attacker (Then GOTO Second Stage)

Second Stage:

Look up the table

If "My Address is the highest MAC address"

False

{Wait for an RTS packet from a node with higher MAC address to shift the frequency}

If RTS packet received with encrypted frequency channel number

Then Decrypt using my private key & Encrypt and Send to the next node in a new RTS packet

GOTO Stage Three

Else

Randomly select a new frequency channel number

Encrypt the number using the public key of the next node

{**If** next node MAC = Attacker

Then Skip to the following node}

Send RTS packet to the following node (new frequency channel encrypted)

Third Stage:

Command my own physical layer to communicate on the new frequency channel.

End

Based on that concept, our detection algorithm depends on modifying the IEEE 802.11 DCF code to enable each node to police the network with very low cost (processing and memory consumption wise) solution. Each node does not need any cooperation (information).

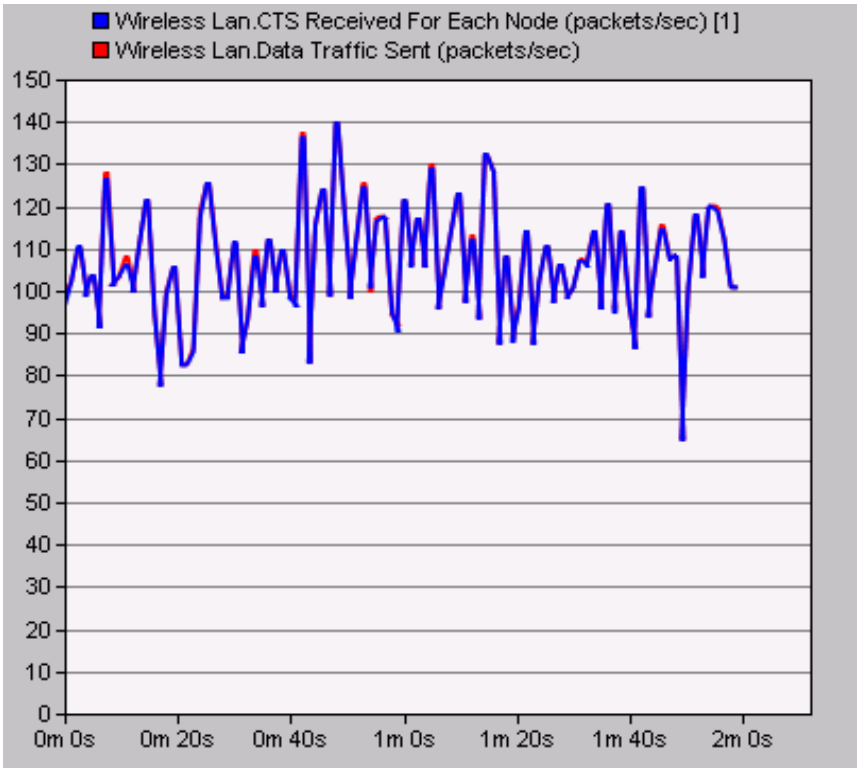

Figure 5-4: Comparison between number of CTS and Data packets for the same node

For our simulation, we used Matlab to resolve the Markov Chain mathematical model and fed the results to OPNET simulator for the detection threshold. The numerical results obtained solving the Markov Chain became the threshold for attacker detection.

For the simulation, each node generated raw traffic rate is (1000 packets/second) which is the Traffic Load In this group, each node always has a packet to transmit in its queue. Also, the assumption is the number of nodes in the group is fixed and all nodes can hear each other.

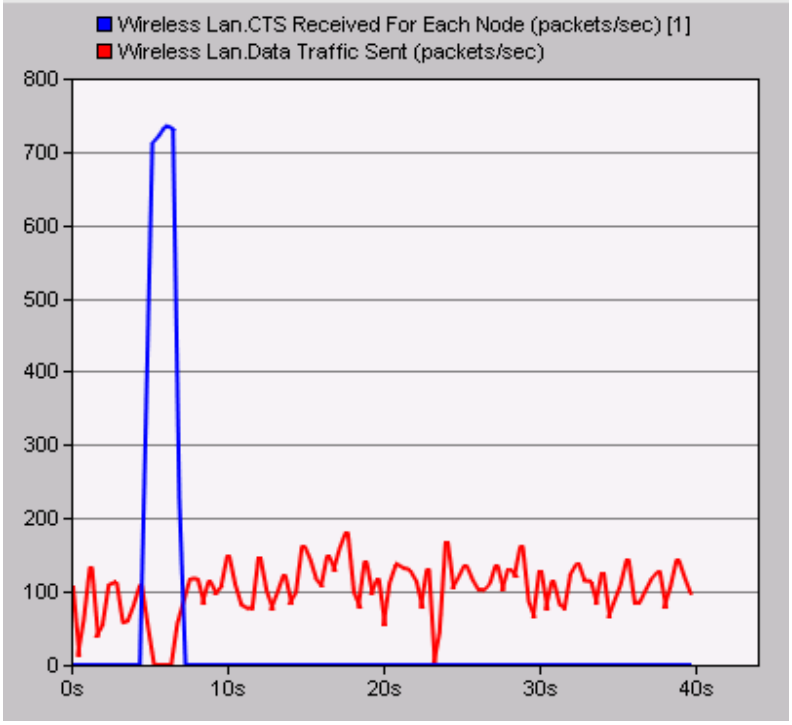

Figure 5-5: Attacker vs. innocent node sent traffic

In Figure (5-5) the red line is the traffic sent by an innocent node while the blue line is the attacker's traffic. This data is recorded at the innocent node's network interface. After few seconds from the start of the communication session the attacker started to capture the channel. Once the attacker is detected and the algorithm is applied, the node changed the DSSS channel frequency and recovered. This simulated network has ten good nodes and one compromised node. It took the network about three seconds to shift to the new frequency leaving the attacker alone in the abandoned frequency.

5.5 Conclusion

The chapter presented a new technique [43] to deal with the Byzantine attack on the MAC layer. The algorithm detects and isolates a compromised node that had passed all the initial authentication steps and is

trying to perform DoS attack on the honest nodes present in the network. The approach is based on utilizing the numerical results obtained by solving the Markov Chain to detect the attacker and the following step is a mechanism to change the frequency to escape the attack by distributing the new frequency using asymmetric encryption technique. The simulation results proved the concept with very high accuracy. This solution is scalable and good for autonomous environment.

Chapter 6 : DoS Detection in IEEE 802.11 with the Presence of Hidden Nodes

6.1 Abstract

The chapter presents a novel technique to detect Denial of Service (DoS) attack applied by malicious nodes in wireless networks with the presence of hidden nodes employing the widely used IEEE 802.11 DCF protocols [1]. Attacker nodes alter the IEEE 802.11 DCF standards to illicitly capture the channel and increase the probability of successful packet transmission on the expense of innocent nodes that follow the protocol standards. Innocent nodes will be deprived from using the channels bandwidth. The theoretical network throughput will be derived using two dimensional Markov Chain [4][34]to determine the network capacity. Results obtained by the theoretical computations will be validated by network simulation [5] to determine the baseline for the maximum achievable throughput in the network under standard conditions where all nodes follow the standards. The main goal of the DoS attacker is to prevent the innocent nodes from accessing the channel and communicating by capturing the channel's bandwidth. The presented protocol will reside in every node to enable each node to police its immediate wireless coverage area. All innocent nodes will be able to detect and identify the DoS attacker in its wireless coverage area. The protocol will be applied to two Physical Layer technologies (DSSS and FHSS) and the results will be presented.

6.2 Introduction

IEEE 802.11 DCF standard specifies two mechanisms to transmit a packet. The basic mechanism is a two-way handshaking method “Basic Access” which employs immediate transmission of a positive acknowledgement (ACK) by the destination node after a successful reception of a packet. ACK packets are required since the sender is unable to determine if each packet was successfully transmitted by listening to its own transmission. The second mechanism uses a four way handshaking scheme “Request-to-Send/Clear-to-Send” (RTS/CTS) before transmitting any packet, a node configured to use RTS/CTS mode “reserves” the channel by sending a special Request-To-Send short frame. The available receiver node responds to an RTS frame by sending back a CTS frame, and then a data packet and ACK response will follow. Since collision may occur only on the RTS frame and it is detected by the lack of CTS response. RTS/CTS mechanism increases the channel’s performance by reducing the duration of a collision when long messages are transmitted. In this chapter, our focus is on DoS detection in the four way handshaking scheme “RTS/CTS”.

Malicious nodes employ several techniques to illegally increase their throughputs and capture the channel on the expense of other fair behaving nodes [8][76][77][78][79][80][81][82][83] [84][85][86][87][88][89][90] [91]. In IEEE 802.11, selfish nodes will manipulate the Back-off timer to increase their probabilities in having successful transmissions by simply decreasing the Back-off timer value instead of following the Back-off timer generation method that all nodes in the network are using. X node is considered malicious or misbehaving when it does not follow the IEEE 802.11 MAC Standard [1]. Attackers will use smaller timeouts than these specified in the protocol standard. With IEEE 802.11, each node is expected to choose a back-off interval

prior to initiating a transmission. The back-off interval is to be increased as per a specific set of rules before all retransmission attempts that are invoked upon failed transmission attempts. An attacker may choose a small and/or a constant back-off interval prior to the transmission of a data packet or follow a completely different retransmission strategy upon experiencing failed transmissions that does not conform to the standard IEEE 802.11 protocol rules. Due to the randomness in choosing a back off value, it is considered that detecting the back off manipulation to be one of the most challenging topics [9] [10] [11]. The purpose of the presented algorithm in this chapter is to detect DoS attackers.

One major contribution in this chapter is that the presented algorithm works in a wireless network with the presence of hidden nodes utilizing the mathematical results of Markov Chain modeling as baseline, in addition to a network mapping algorithm. Several researches were performed to detect the manipulation of the back-off timer in wireless networks where there are trusted Access Points [12][10], where a trusted AP will regulate the senders back-off timer values and detect the misbehaving nodes. Due to the nature of autonomous wireless networks, where there is no centralized authority that will assign and monitor each nodes' back-off timer values, the task is very challenging. The presented algorithm can be applied to a autonomous environment where there is no centralized authority or a supervisor node (i.e. Access Point) that is supervising every transaction takes place between different users. In [8] the authors assume that nodes will cooperate and announce the state of their pseudo-random sequences so the nodes would monitor each other's behaviors. This approach assumes the cooperation from an attacker which is not realistic. Our presented algorithm does not expect or wait for any cooperation from any node hence eliminating the chance of getting fed wrong information by a malicious node. In [28] the authors introduced a new parameter to indicate the level of cooperation of each node

which increases the complexity of each transaction throughout the whole communication session. The presented algorithm utilizes the already-used CTS packets in IEEE 802.11 to perform the detection process by further processing the CTS packets and appending a new field to the existing Hallo packets only once during the communication session. In [29] a method was proposed to make the Access Point functions as a watchdog to monitor all nodes' behaviors. This method consumes the resources of the AP node and is not suitable to a total autonomous system like ad-hoc networks. Assigning one node or selected nodes to police the network is a very dangerous concept and creates a single point of failure in case the police node is compromised itself. In [32] the author is proposing to analyze the distribution of inter-delivery times between two consecutive successful transmissions. This method is very challenging and requires very accurate measuring clocks in the order of micro seconds to accurately detect the selfish behavior. The presented algorithm does not require any hardware additions or clocks. The majority of researches that were performed on backoff timer manipulation detection assumed that there are no hidden nodes [41]. Few chapters presented the concept of detection with the presence of hidden nodes [8][48]. In [8] the chapter assumes cooperation among nodes which is not realistically applicable to a DoS attacker. In [10] the authors propose new messages to the existing packets used by IEEE 802.11 which increases the network overhead unnecessarily.

6.3 IEEE 802.11 and DoS Behavior

IEEE 802.11 DCF [1] uses Carrier Sense Multiple Access / Collision Avoidance CSMA/CA mechanism to reduce the probability of collisions in a wireless network to enhance the throughput. Time is divided into slots and is used to define the inter-frame-space (IFS) intervals and to determine the back-off times for competing nodes in the wireless network. When a node has a packet in its queue to transmit, first it senses the channel and if the channel is busy, the node waits until the channel becomes idle for a Distributed Inter Frame Space (DIFS) period, and then calculates a random back-off time. The random back-off time is specified by an integer value that is equivalent to a number of time slots. The idle period after a DIFS period is called the contention window (CW).

Nodes are allowed to transmit only at the beginning of each Slot Time. The Slot Time size, (*Sigma*), is set equal to the time needed at any node to detect the transmission of a packet from any other node inside its coverage network. Slot Time values are determined by the physical layer used by the MAC protocol, and it takes into consideration the propagation delay, for the time needed to switch from the receiving to the transmitting state and for the time to signal to the MAC layer the state of the channel (Busy Detect Time). Nodes with packets to transmit select a back off value based on the Contention Window as the following [Back off = int (CW×rand×slot time)]. Where “rand” is a random number uniformly distributed between 0 and 1, and CWmin < CW < CWmax, where CWmin is the minimum CW, and CWmax is the maximum CW. Firstly, the node that has a packet to transmit computes a back-off time in the range [0, CWmin - 1], where CWmin is the minimum contention window size. When the medium becomes idle, after

an additional DIFS period, nodes decrement their back-off timers until the medium becomes busy again or until the timer value goes to zero.

If the timer has not reached zero and the medium becomes busy, the node freezes its timer. This procedure continues until the timer is finally decremented to zero. Then, the node transmits its packet. If the sender receives an ACK from the destination, the transmission is assumed to be successful, and the station sets its CW back to CWmin – 1. If two or more nodes decrement their timers to zero at the same time, a collision will occur, and each node will have to generate a new back-off time by doubling the Contention Window value [2* CWmin]. During the kth retransmission attempt the Contention Window will have the form [0.2k* CWmin] and will be doubled till it reaches CWmax. This provides a mean of avoiding repeated collisions when there is congestion.

The MAC parameter values (Slot Time, SIFS, DIFS, ACK, CTS, RTS and CW) are dependent on the physical layer being used by the MAC protocol. In this chapter, we are applying the developed algorithm on two different systems, the first is using Frequency Hopping Spread Spectrum (FHSS) and the second is using direct sequence spread spectrum (DSSS) - (Table 7-1).

1. IEEE 802.11 - Frequency Hopping Spread Spectrum (FHSS)

FHSS operates in the 2.4 GHz band with a range starting from 2.402 GHz to 2.480 GHz. Each channel has a width of 1MHz. Support two rates of 1Mbps and 2Mbps. There are 78 hopping sequences and each sequence would use 79 hops. 15 systems could be collocated and work independently with minimal amount of collisions.

2. IEEE 802.11b - Direct Sequence Spread Spectrum (DSSS)

DSSS operates in the 2.4 GHz band. Each channel has a width of 22. The rates defined in IEEE 802.11 are 1 Mbps and 2 Mbps and the rates in IEEE 802.11.b standard are 5.5 Mbps and 11 Mbps. Only the first 11 channels are used in the United States.

B. Network Configuration and DoS Attack Impact

The network configuration simulated is presented in Figure (6-1) where there are three areas X, Y, and Z. Nodes located in area Y can hear all other nodes located in area Y and Z. Nodes located in area X can hear all other nodes located in area X and Z. Nodes in area Y cannot hear nodes in area X and vice versa.

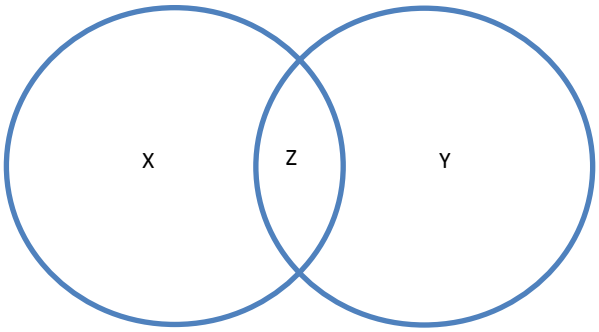

Figure 6-1: Coverage Areas

The presented algorithm is scalable and deals with the number of nodes in each area as an independent variable and performing the calculations accordingly. For the sake of simplicity in

presenting this chapter and conducting the simulations, we will assume that the number of nodes in each area is constant, although the Markov Chain modeling will present those numbers as variables for generalization.

The DoS attacker can implement the attack by several methods by altering the firmware code on the Network Interface Card (NIC) or by modifying the hardware. The first method is a lot easier to implement from the feasibility and cost point of view and is far more prevalent. In our chapter, the presented solution is directed towards detecting the manipulation of the protocol's firmware and more specifically detecting the manipulation of the back off timer. In this case the DoS attacker will keep transmitting packets that does not contain any useful just to occupy the channel. The attacker will only back off one slot every single time it has a packet to transmit or when it experiences a collision while the other innocent nodes will follow the exponential back off mechanism.

We simulated a network with an attacker present to show the effect on the other innocent nodes. The payload size used throughout this chapter is 8000 bits so it could be sent in one time slot without the need of fragmentation.

For the simplicity, we will assume the following constant number of nodes in each area throughout the entire chapter – these numbers will be used for the simulations and solving the Markov Chain:

[Area X = 2 Nodes] - [Area Y = 3 Nodes] - [Area Z = 2 Nodes]

In Figures (6-2 & 6-3) the simulation shows the comparisons between traffic sent by innocent nodes under fair conditions without the attacker (red lines) and the traffic sent with the attacker present (blue lines). Figure (6-2) shows a network using FHSS and Figure (6-3) shows a network using DSSS. The effect of the DoS attack on the innocent nodes in both cases is very clear that once the attacker existed the innocent nodes are deprived from accessing the channel to send anything. In both cases, before and after the attack, the loads are the same.

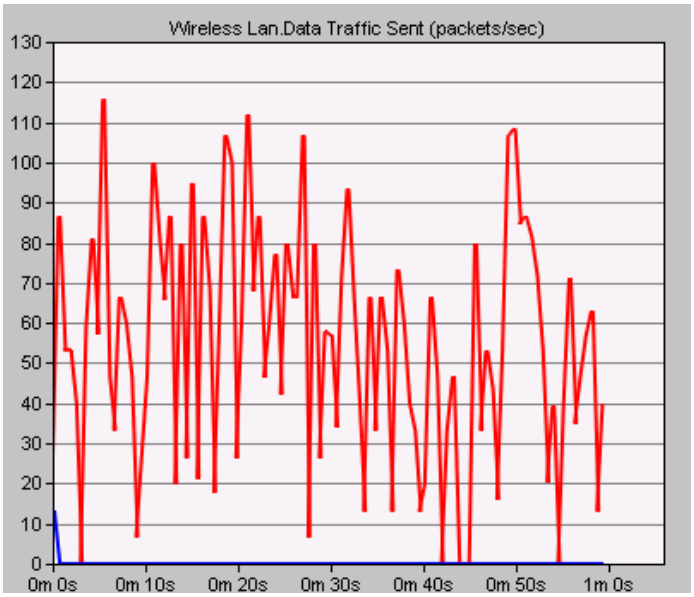

Figure 6-2: FHSS Attacker Traffic

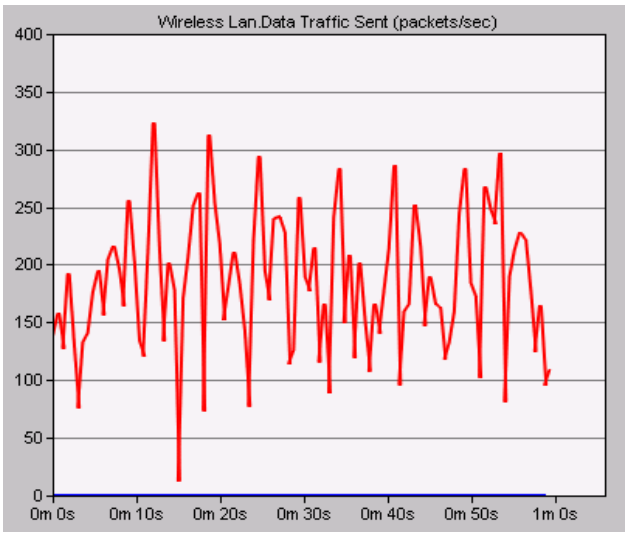

Figure 6-3: DSSS Attacker Traffic

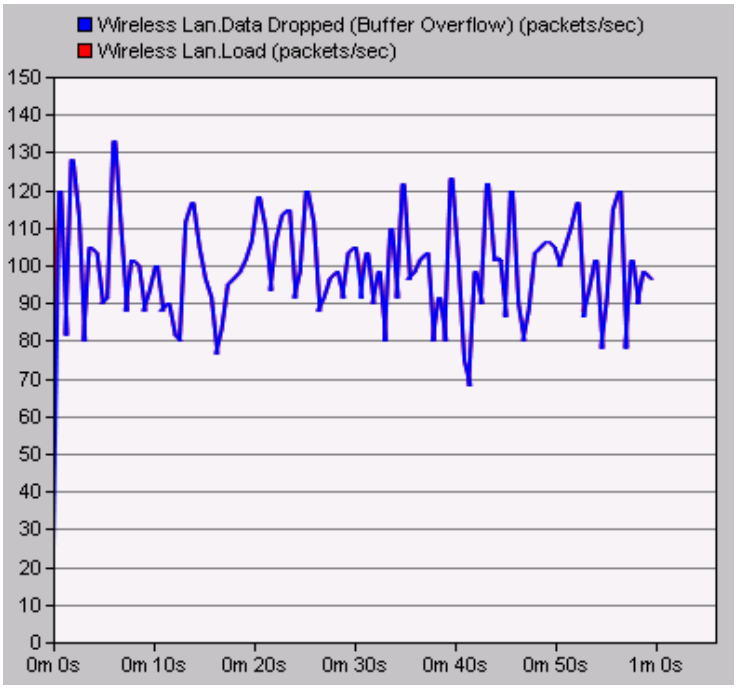

Figure 6-4: Load Dropped

Figure (6-4) shows an innocent node's load and dropped traffic, the two lines are identical which means all the packets generated by the upper layers were discarded because the innocent node could not access the channel with the attacker present which shows the impact of the DoS attack.

6.4 Markov Chain Analysis

The Wireless Network's throughput will be derived using two dimensional Markov Chain presented in Figure (6-7) for each area in Figure (6-6) to obtain theoretical numerical results that will be used by our algorithm to identify the DoS attacker in the network. Markov Chain will be modeled to include a more generic configuration (See Figure (6-7)) than what is presented in Figure (6-3) and this generic configuration could theoretically have infinite number of overlapped coverage areas and is suitable for a continuously aligned coverage areas like a highway covered with equally spaced IEEE 802.11 networks to provide coverage to vehicles for a full stretch of the road. The simulation of this chapter will only deal with three areas as presented in Figure (6-1). Bianchi's Markov Chain model [4] is extended to calculate the individual rate in "Packet per second" values for each node in each area. One of our contributions here is extending Bianchi's model which is only applicable to wireless networks without hidden nodes to be able to calculate the throughput with the presence of hidden nodes.

Assume that each node has a packet to transmit at any given time (Saturation Condition) and the number of nodes inside the each area is constant during the calculation.

The first step is to calculate the Transmission Probability for each area then derive the throughput for this area and finally obtain the individual throughput for each node in each area.

b(t): stochastic process representing the back off time counter for a given station. (t and t+1) correspond to the beginning of two consecutive slot times.

, , and are the number of nodes in areas X, Y, and Z respectively and the number of nodes in overlap 1 and 2 are

And respectively.

(1)

(2)

in the different area

(3)

According to the given topology, in the different area

(4)

Then, from (3) and (4) obtain the and .

Throughput in the different area:

We define as the probability that there is at least one transmission within station 's coverage area in a random time slot.

(5)

We denote by the probability that station successfully transmits its packet to its neighbors, which equals the probability that exactly only one station transmits on the channel covered by station in a given time slot, and no hidden station transmits either. Hence the formulas for and are given by

(6)

Let be the normalized capacity of station ,

(7)

Let be the average length of a slotted time and be the average packet payload size. The average amount of payload information successfully transmitted in a time slot is . will be .

is the duration of a time slot. Here the term accounts for an idle time slot with probability ; the term stands for successful transmissions of station with successful probability ; and the term deals with the collision duration. is the average time for a successful transmission, and is the average time duration for the collision. and can be derived for both the basic and the RTS/CTS access mechanisms. Obtaining the throughputs for RTS/CTS access mechanism:

Figure 6-5: Successful Transmission Time

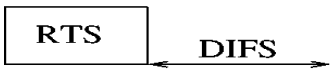

Figure 6-6: Collision Time

Then we are able to obtain the and

(8)

(9)

(10)

(11)

(12)

(13)

(14)

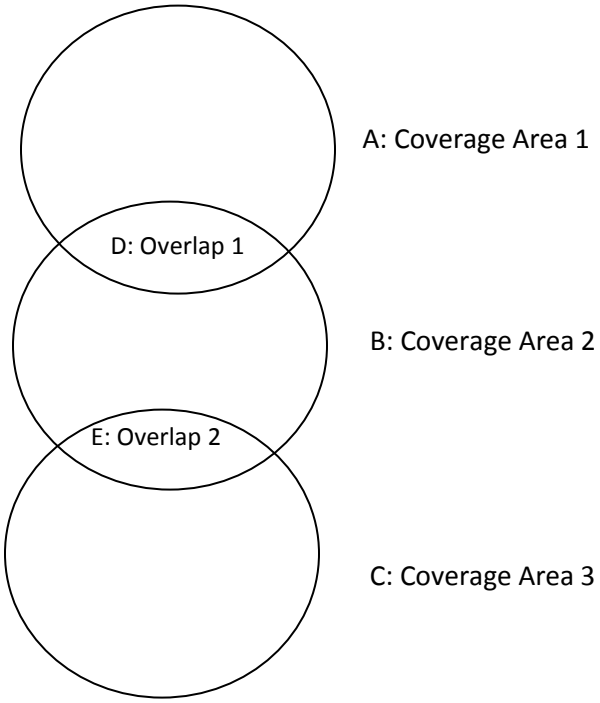

Figure 6-7: Three Coverage Areas

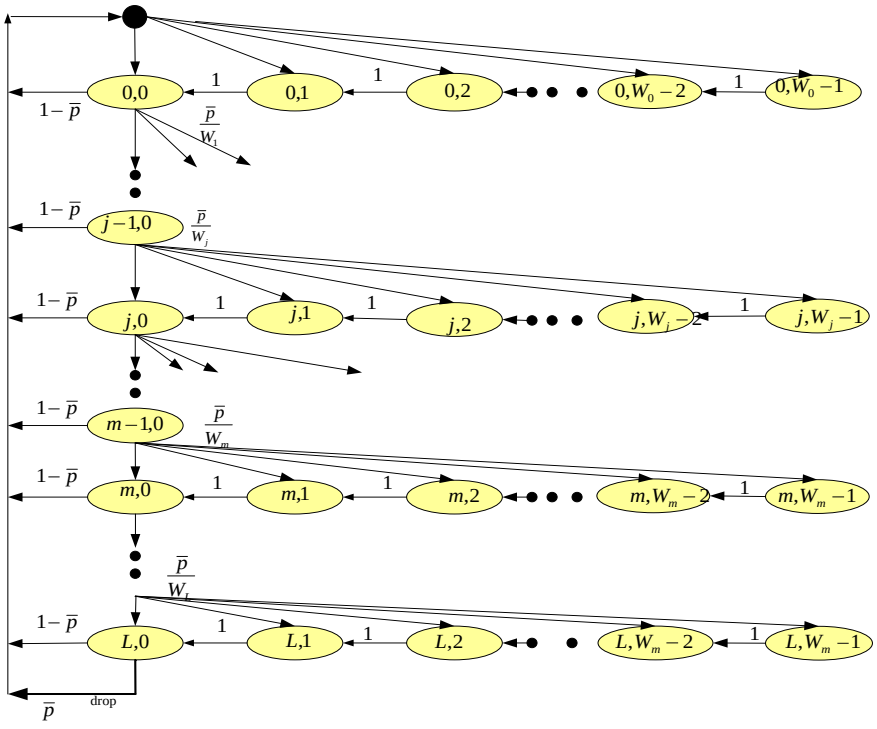

Figure 6-8: Markov Chain

To validate the theortical results described above, we compared the numerical results obtained by solving the markov Chain using parameters with the OPNET [5] simulation under the saturation

condition where every node has a packet to transmit in its queue at any given time. Matlab [7] was used to solve the markov chain numerical results.

The network Configuration presented in Figure (6-1) is used in the simulation and to verify the numerical results obtained by modelling Markov Chain. The following table shows the values obtained from Markov Chain modelling and from OPNET simulation to show the maximum achievable throughput (packets/second) for each area for both FHSS and DSSS under saturation conditions where every node always has a packet to transmit. Since all nodes have the same condition, then every node has the same probabilkity in accessing the channel which is translated to same average number of packets transmitted into the channel over time. This table bridges the value of the theortical calculations and emperical results an dshows the significance of the detection thresholds accuracy. It is noted that the results for areas X & Y are slightly different in the simulation results because of the imperfection of wireless nature. It is also noted in Table (6-2) that the theortical rsults are generally higher than the calculations due to the imperfections in the environment that would negatively affect the throughput, and the simulator used takes into account such imperfections to simulate real environemnts. One benefit of using the theortical results as opposed to emperical results that the theortical results will generate higher values of thresholds that will help eliminating false postives. As shown in the previous section that the number of the CTS packets received is equal to number of data packets transmitted.

Table 6-1: Comparison between Maximum Throughput (Packets/Second) for each Area

PHY Technology	**Area X**	**Area Y**	**Area Z**
FHSS (Simulation)	100	110	100
FHSS (Theoretical)	105	110	105
DSSS (Simulation)	360	360	270
DSSS (Theoretical)	510	520	510

6.5 Detection Process and Simulation Results

IEEE 802.11 DCF RTS/CTS operation has an inherited feature that the number of successful data packets sent by a given node are equal to the CTS packets received by this specific node with the permission to send data packets. The CTS packets are designed to be heard by every single node within its coverage area. All the nodes beside the one that the CTS packet is destined will have to update their NAV so they refrain from transmitting any packets during the NAV to eliminate the chances of collisions. We modified the OPNET [5] code to hear all CTS packets individually and collect them in separate queues depending on the destination address. Below is the result from the simulation to prove that the number of recived CTS packets is equal to the number of data packets sent. Figure (6-9) shows that the number of CTS received by node_1 is the same number of packets sent by this node to other nodes in the network. Based on that

concept, our detection algorithm depends on modifying the IEEE 802.11 DCF code to enable each node to police the network with very low cost (processing and memory consupmtion wise) solution withourt introducing new types of messages or altering the existing messages. Basically, the algorithm that resides in each node further processes the received CTS packets before dicarding it. Upon network communication initialization, which includes the initial exchange of Hello packets, every node will map out which nodes it can sense in its range and compile a list of MAC addresses that it can communicate with. This list will be broadcasted by all the nodes. Then each node will compare its list to other nodes' lists. If the two lists (its own and the other node) match then both nodes belong to the same domain and will mark that domain for node count (Areas X or Y in Fig (6-1)).

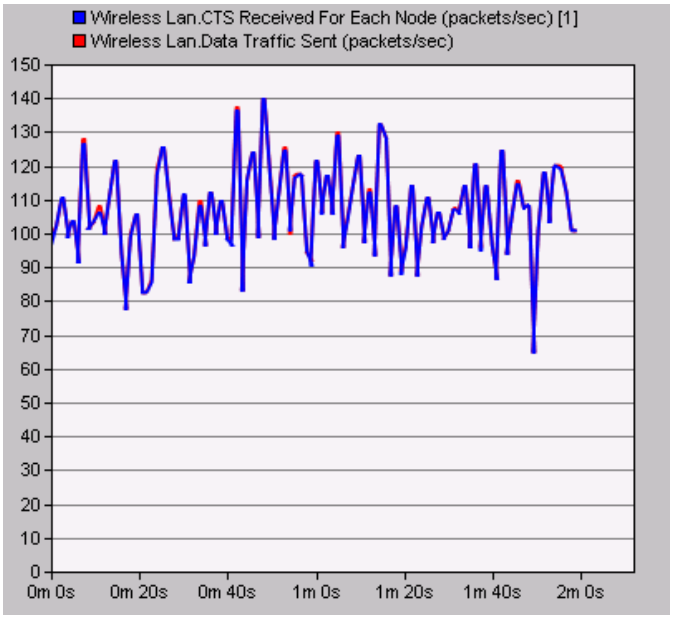

Figure 6-9: Comparison between number of CTS and Data packets for the same node

If the two lists do not match then, this node will identify itself as an overlapping node that shares two domains (Area Z in Fig (6-1)). The lack of cooperation from the attacker will not impact the

results because the detection threshold will have enough tolerance to account for a missing count from a node. The algorithm will have to phases that will run in series. The first phase is the network mapping where all the nodes detrmine their coverage area to decide which Markov Chain Throughput equation will be used, either an exclusive domain (X& Y) or an overlapping area (Z). Accordingly each node wil choose the approperiate Markov Chain equation. Lists creates during the network mapping phase will be appended to the Hello packets and will only exchanged once among the network after the intialization of the network. Each node will further process each received List to derive the number of the nodes in each area.

Example to explain the Network Mapping technique – Using Figure (6-1):

Area X has 2 nodes: X1, X2,

Area Y has 2 nodes: Y1, Y2

Area Z has 2 nodes: Z1, Z2

After the exchange of the List which includes all the MAC addresses heard by those nodes, each node will have the following on its own list:

X1: (X2, Z1, Z2) X2: (X1, Z1, Z2)

Y1: (B2, Z1, Z2) Y2: ((B2, Z1, Z2)

Z1: (X1,X2,Y1,Y2,Z2) Z2: (X1,X2,Y1,Y2,Z1)

Now, for instance node X1 will compare its own list with the others and it will find that the list from X2 is identical (beside itself), then it will decide that X1 and X2 belong to the same region and the number of nodes in this region is two nodes for Markov Chain Throughput calculations as to which equation to use. The same will happen with all nodes. When it is Z1' turn to compare the lists, it will find that Z2 has the same number of nodes, which will give the ide to Z1 that Z1 nad Z2 belong to the same region. In addition, Z1 will find its list is longer than the others heard (X1,Y1,X2,Y2), then node Z1 will releaize that its location is the overlapping area in Figure (6-1) and will use these numbers for the calculation of the Throughput.

Phase I will be triggered after the exhange of the first Hello packets and the Lists will be included in the second round of Hello packets. The assumption is the number of nodes are fixed in each area throughout the communication session and all nodes are not mobile. Following Phase I, Phase II will be triggered to detect cheaters based on thre network topology formed in phase I.

The Algorithm that will reside at each node:

Phase I: Network Mapping:

Each node will map the network to know its own coverage area, number of nodes in each area and to determine which throughput equation generated by Markov Chain modelling will be used:

Start

Create List_x */* List_x is the MAC addresses that node x can hear in its domain: x = 1 to* n_k *, where* n_k *is the number of nodes in each coverage area, k = X,Y, or Z*/*

Broadcast List_x

Revceive List_1 through List_ $\boldsymbol{n_k}$ */* (excluding List_x which is my list of MAC addresses) */*

Compare Rcvd (List_1 to List_n_k) /*(all received lists from all other nodes*/)* ***to List_x /**** *(my generated list) */*

If List_n_k */*Matches my List (Same number of nodes and same nodes can be heard) */*

Then */* (We are neighbors in the same area) */*

Update Node Count */* (For the same area) */*

Else */* **(***We do not belong to same area or I belong to an overlapping area) */*

Update Node Count */* (For the those areas) */*

If (number of Nodes in my area >Numberof Nodes in others)

Then (I am in an overlapping area)

*/*This function to determine if a node is in an overlapping area*/*

*/*At the end of this phase each node will know how many nodes in its immediate area and other areas – also, the nodes in overlapping area will know themselves)*/*

Phase II: Detection:

Each node will implement the detection algorithm

Count n_k */*" Number of Nodes in the immediate area and other areas"*/*

Create n_k Counters

Calculate Maximum Individual Throuput / obtained from Markov Chain modeling above for each area*/*

When *Receive CTS*

If *(Destination Address = My Address)*

Do Nothing

Else

{

Update Counter *(Destination Address)*

Calculate Rate

/ number of CTS received per second for each Destination Address */ }*

If

CTS_node_x rate < Maximum Individual Throuput

Do Nothing

Else

Anounce *"node_x is an Attacker" /* it is shown as print command in our OPNET simulation and used it as output */*

End

For the simulation, we used Matlab to solve the Markov Chain mathmatical model and fed the results to OPNET simulator for the detection threshold. The numerical results obtained solving the Markov Chain became the threshold specially that they are very close to the simulation results as shown in figure (6-10). The numerical results are considered the maximum number of packets any node can send in the presense of other number of nodes (as calculated in Markov Chain modelling, so any other node that has more packets succssfully sent is not following the IEEE 802.11 DCF standard and manipulating the protocol to illegally increase its throughput to attack the network.

A. Simulation Results

The simulation is conducted to show that innocent nodes in multiple areas can detect the attacker via monitoring the number of CTS packets sent by all reachable nodes inside the network. The simulation will show that the thresholds shown in Table (6-2) are exceeded whenever an attacker is present in the network which will enable the innocent nodes to detect the attacker using the theoretical results obtained via solving Markov Chain and divided on the number of the nodes in each area since all the channels operate under saturation condition. To avoid false positives where an innocent node is falsely marked as an attacker, the algorithm will not react to instantaneous spike but rather will look for a moving average over time to ensure that any spike by an innocent node will not be mistaken for an attacker.

In Figure (6-10) – FHSS and Figure (6-11) - DSSS, an innocent node in Area Z was listening to the CTS packets sent in the medium and found that one node in Area X is exceeding the threshold calculated for the channel in this area divided by the number of nodes in this area. The blue line is for the attacker and the red is for another innocent node and the difference is very significant (more than 80 times for FHSS and more than 270 times for DSSS). According to the thresholds calculated in Area Z, the channel capacity is 105 Packets/sec (57 Packets/sec per node) for FHSS and for DSSS is 510 Packets/sec (250 Packets/Sec per node) with the existence of two nodes in each type, the attacker achieved number of transmitted packets well over the threshold and is detected by this innocent node and marked as an attacker.

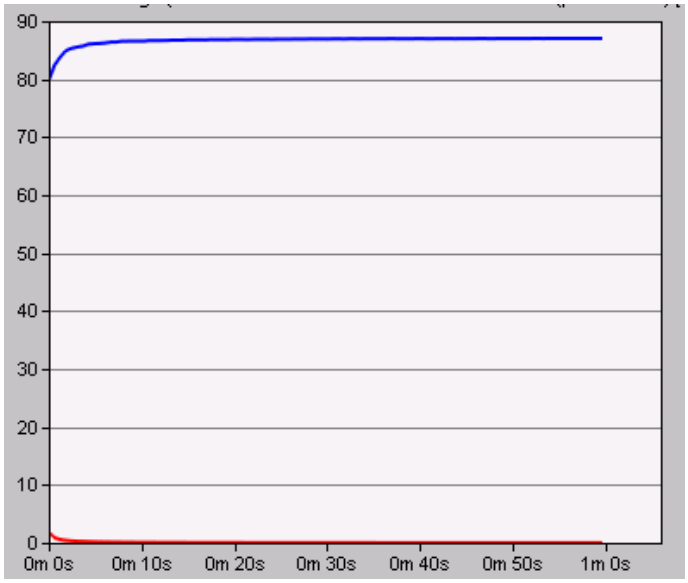

Figure 6-10: FHSS - Node Z - Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker

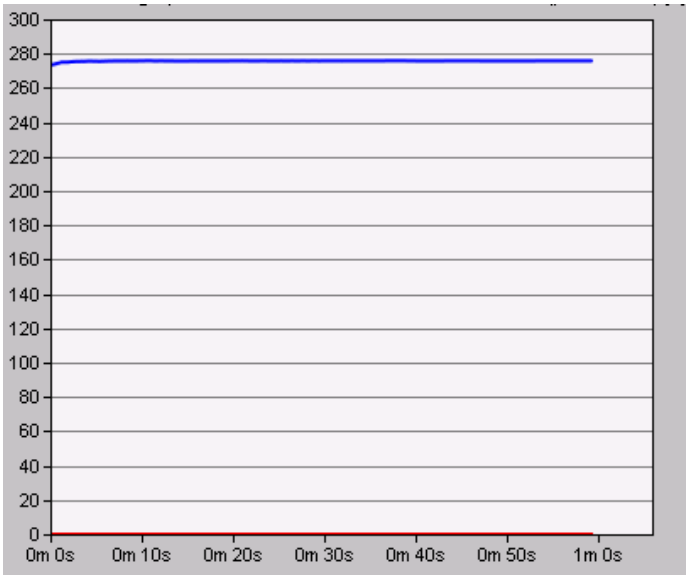

Figure 6-11: DSSS - Node Z - Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker

In Figure (6-12) – FHSS and Figure (6-13) - DSSS, an innocent node in Area X was listening to the CTS packets sent in the medium and found that one node in Area Z is exceeding the threshold calculated for the channel in this area divided by the number of nodes in this area. According to the thresholds calculate – in Area Z, the channel capacity is 105 Packets/sec (57 Packets/sec per node) for FHSS and for DSSS is 510 Packets/sec (250 Packets/Sec per node)with the existence of two nodes in each type, the attacker achieved number of transmitted packets well over the threshold and is detected by this innocent node and marked as an attacker.

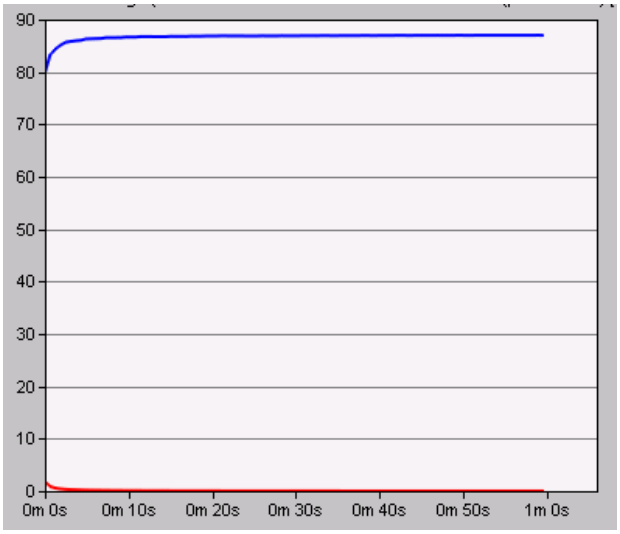

Figure 6-12: FHSS - Node X - Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker

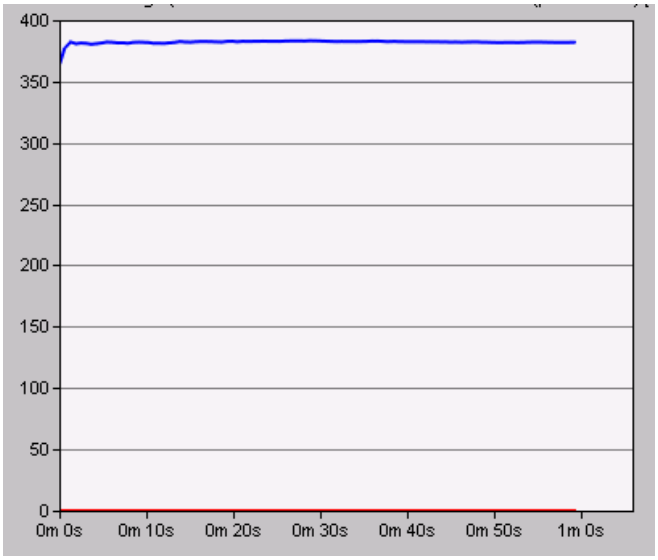

Figure 6-13: DSSS - Node X - Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker

6.6 Conclusion

A novel approach [46][47] to detect a node employing DoS attack in the IEEE 802.11 wireless network with the presence of hidden nodes was presented and the algorithm proved to be effective as verified by the simulation. The approach is based on utlizing the numerical results obtaind by solving the Markov Chain. Combining the numerical results with the specifications of the IEEE 802.11 DCF RTS/CTS protocol, a develpoed code was impedded into IEEE 802.11 code to enable each node to police the network and detect the attacker. The simulation results proved our concept with very high accuracy without any false postives recorded and this in part caused by taking advantage of the higher values of the throrticaI results generated by solving Markov Model. This solution is scalable and applicable for autonomous environment where there is no centralized authority overseeing the communication process and transaction among the nodes.

Chapter 7 : Conclusion and Future Work

7.1 Conclusion

My research focused on the security aspect of wireless networks. Wireless technology is being used by everyone for a wide spectrum of needs that range from personal communication to banking, education, and medical, to name a few. Most of us run our lives—personal, institutional, and governmental—on wireless communication by using cellular phones, wireless routers, and wireless microwave backhauls. Sensitive information is being transferred every second across the globe on the wireless medium using cutting-edge technologies that are vulnerable to a wide range of attacks. This PhD dissertation investigated the impact, detection, and mitigation of Denial of Service attacks on a fully distributed ad-hoc wireless network utilizing IEEE 802.11 DCF [1][2][3] in the Media Access Layer. The goal of my research was to effectively detect the attacker and find a way to ease the negative impact on the network. OPNET [5] simulator and MATLAB [7] software were used during the course of the research to validate the proposed solutions along with some mathematical modeling, including Markov chain modeling. The research was challenging because of the autonomous and distributed nature of the ad-hoc networks where there is no centralized authority to monitor the security aspect of the network and the capability of the nodes to be mobile and move in and out of the coverage area. I showed how to detect DoS attacks for single and multiple domains. Also, the detection algorithm was applied successfully in a mobile environment. Two mitigation techniques were developed to restore the network's operation after detecting an attack. The results were presented in the previous chapters that show the effectiveness of the algorithms developed.

7.2 Future Work

In the future, the Byzantine attack mitigation algorithm will be further developed to include a Public Key management scheme to reduce the number of key exchange transactions and the convergence time of the network when all nodes move to a different communication channel. Also, a Lemma will be introduced to formulate the proof of Markov chain results obtained in the previous chapters.

The developed algorithms will be further advanced to be compatible with IEEE 802.11i standards [50] [51] for more security assurance enhancements.

The WiMax IEEE 802.16 MAC Layer Attack problem will be investigated and a solution will be proposed to detect the attacker. The vulnerability of using Ranging Request-Response (RNG-REQ, RNG-RSP) messages, which are used in the initial ranging process. The RNG-REQ message is sent by a SS trying to join a network to propose a request for transmission timing, power, and frequency and burst profile information. Then, the BS responds by sending a RNG-RSP message to fine-tune the setting of the transmission link. After that, the RNG-RSP can be used to change the uplink and downlink channel of the SS. There are several threats related to these messages including the following threat where the attacker can intercept the RNG-REQ to change the most preferred burst profile of SS to the [15][16].

Chapter 8 : Bibliography

[1] IEEE Standard 802.11 – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: 2003

[2] IEEE Standard 802.11 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: 1999

[3] IEEE Standard 802.11 - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: 2007

[4] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE Journal on Selected Areas in Communications, 18(3):535–547, March 2000.

[5] www.OPNET.com

[6] Mustafa Ergen and PravinVaraiya, "Formulation of Distributed Coordination Function of IEEE 802.11 for Asynchronous Networks: Mixed data Rate and packet Size", 2007

[7] www.mathworks.com

[8] Venkata Nishanth Lolla , Lap Kong Law , Srikanth V. Krishnamurthy , Chinya Raishankar and Dharmaiah Manjunath. "Detecting MAC Layer Back-off Timer Violations in Mobile Ad Hoc Networks"

[9] J. Bellardo and S. Savage, "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", in Proc. of the USENIX Security Symposium, Washington, DC, August 2003.

[10] M. Raya, J. Hubaux, and I. Aad. DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots. In Proceedings of MOBISYS, 2004

[11] Svetlana Radosavac, Alvaro A. C´ardenas, John S. Baras, George V. Moustakides Detecting IEEE 802.11 MAC Layer Misbehavior in Ad HocNetworks: Robust Strategies Against Individual and Colluding Attackers

[12] P. Kyasanur and N. Vaidya.Detection and Handling of MAC Layer Misbehavior inWireless Networks.In Proceedings of Dependable Systems and Networks, 2003.

[13] Lin Chen, Khaled Aslan Almoubayed, Jean Leneutre. "Detection And Prevention Of Greedy Behavior In Ad Hoc Networks. International Conference on Risks and Security of Internet and Systems (CRISIS 2007), Marrakech, Maroc, July 2007

[14] Kemal Bicakci, BulentTavli “Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks” Computer Standards & Interfaces 31 (2009)

[15] E. Bayraktaroglu , C. King, X. Liu, G. Noubir, R. Rajaraman, B. Thapa, “On the Performance of IEEE 802.11 under Jamming”, INFOCOM 2008.

[16] Ali Hamieh, Jalel Ben-Othman, Lynda Mokdad, “Detection of Radio Interference Attacks in VANET” IEEE "GLOBECOM" 2009 proceedings.

[17] W. Xu, T. Wood, W.Trappe, and Y. Zhang.Channel Surfing andSpatial Retreats: Defenses against Wireless Denial of Service.In Proceedings of the ACM Workshop on Wireless Security (WiSe), 2004.

[18] Khattab, S.M., Mossé, D., Melhem, R.G.: Modeling of the channel-hopping anti-jamming defense in multi-radio wireless networks. In MobiQuitous(2008)

[19] A. D. Wood, J. A. Stankovic, and G. Zhou, \DEEJAM: Defeating Energy-E_cient Jamming in IEEE 802.15.4-based Wireless Networks," in SECON'07, June 2007.

[20] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan, Understanding and mitigating the impact of RF interference on 802.11 networks," in IGCOMM '07, 2007, pp. 385{396.

[21] P. Bahl, R. Chandra, and J. Dunagan, \SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in MobiCom '04, 2004.

[22] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly, Distributed channel management in uncoordinated wireless environments," in MobiCom '06,2006.

[23]V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, .Using channel hopping to increase 802.11 resilience to jamming attacks,.in IEEE INFOCOM Minisymposium, 2007.

[24] Wireless Networking in the Developing World, “File:Wireless Networking in the Developing World.pdf (page 25/15) uploaded by Kozuch”, 2007 - http://commons.wikimedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).png

[25] Andreas Deininger, ShinsakuKiyomoto, Jun Kurihara, Toshiaki Tanaka, Securit y Vulnerabilities and Solutions in Mobile WiMAX, International Journal of Computer Science and Network Security, VOL.7 No.11, November 2007.

[26] Sheraz Naseer, Dr. Muhammad Younus, Attiq Ahmed, Vulnerabilities Exposing IEEE 802.16e Networks To DoS Attacks: A Survey, Proceedings of the 2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008.

[27] Jalaa Hoblos, “SELFISH NODE MISBEHAVING STATISTICAL DETECTION WITH ACTIVE MAC LAYER NAV ATTACK IN WIRELESS NETWORKS” Kent State University

[28] Revoti Prasad Bora, Dheeraj Harihar and Saurabh Sehrawat, Detection, penalization and handling of misbehavior in ad hoc wireless networks

[29] Ali Mohammed Alsahag and Mohamed Othman, Enhancing Wireless Medium Access Control Layer Misbehavior Detection System in IEEE 802.11 Network

[30] Wireless Networking in the Developing World, “File:Wireless Networking in the Developing World.pdf (page 25/15) uploaded by Kozuch”, 2007http://commons.wikimedia.org/wiki/File:2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).png

[31] sorin m. Schwartz, “ Frequency Hopping Spread Spectrum (FHSS) vs. Direct Sequence Spread Spectrum (DSSS) in Broadband Wireless Access (BWA) and Wireless LAN (WLAN)

[32] Yanxia Rong, DETECTING MAC LAYER MISBEHAVIOR AND RATE ADAPTATION IN IEEE 802.11 NETWORKS: MODELING AND SPRT ALGORITHMS, 2002

[33] Pablo Serrano, Member, IEEE, Albert Banchs, Member, IEEE, Valerio Targon, and Jos´e F´elix Kukielka, “Detecting Selfish Configurations in 802.11 WLANs” FEBRUARY 2010

[34] Xijie Liu, Tarek N. Saadawi, "Throughput analysis of IEEE 802.11 multihop ad hoc wireless networks under saturation condition," ISCC, pp.245-248, The IEEE symposium on Computers and Communications, 2010

[35] Abhay Kumar Rai, Rajiv Ranjan Tewari & Saurabh Kant Upadhyay,” International Journal of Computer Science and Security (IJCSS) Volume (4): Issue (3)” 2012

[36] Ming Yu; Mengchu Zhou; Wei Su; , "A Secure Routing Protocol Against Byzantine Attacks for MANETs in Adversarial Environments," Vehicular Technology, IEEE Transactions on , vol.58, no.1, pp.449-460, Jan. 2009

[37] S. Albert Raebra and S.Vijayalakshmi “BYZANTINE BEHAVIOUR (B2) –MITIGATING MIDWAY MULTICAST MISBEHAVIOUR (M4) IN ADHOC NETWORK”International Journal of Network Security & Its Applications (IJNSA), Vol.2, No.3, July 2010

[38] N. Ferguson; B. Schneier (2003). Practical Cryptography. Wiley. ISBN 0-471-22357-3.

[39] J. Katz; Y. Lindell (2007). Introduction to Modern Cryptography. CRC Press. ISBN 1-58488-551-3.

[40] 802.11b-1999 Higher Speed Physical Layer Extension in the 2.4 GHz band" 1999-02-11. http://standards.ieee.org/getieee802/download/802.11b-1999.pdf. Retrieved 2007-09-24.

[41] Joseph Soryal and Tarek Saadawi, "IEEE 802.11 Denial of Service Attack Detection in MANET" - "Wireless Telecommunications Symposium 2012 (WTS 2012)", “London, England, United Kingdom”, April 2012

[42] Joseph Soryal and Tarek Saadawi,”WiFi Channel Attack Detection and Avoidance” - "35th IEEE Sarnoff Symposium 2012 (Sarnoff 2012)","Newark, NJ, USA", May 2012

[43] Joseph Soryal and Tarek Saadawi,"Byzantine Attack Isolation in IEEE 802.11 Wireless Ad-Hoc Networks",“The Eighth IEEE International Workshop on Wireless and Sensor NetworksSecurity in conjunction with MASS2012 (WSNS'12),USA,October 2012

[44] Joseph Soryal and Tarek Saadawi, "IEEE 802.11 DoS Attack Detection and Mitigation Utilizing Cross Layer Design "- ADHOC-D-12-84- Elsevier Ad-hoc Journal - Under Review

[45] Joseph Soryal and Tarek Saadawi, "Statistical Detection of Selfish Behavior in IEEE 802.11 Wireless Networks" – to be submitted.

[46] Joseph Soryal, Xijie Liu and Tarek Saadawi, “Misbehavior Detection in IEEE 802.11Wireless Networks with the Presence of Hidden Nodes" – to be submitted.

[47] Joseph Soryal, Xijie Liu and Tarek Saadawi,"DoS Detection in IEEE 802.11 with the Presence of Hidden Nodes Utilizing Network Mapping" – to be submitted.

[48] Alvaro X. C´ardenas, Svetlana Radosavac and John S. Baras, “Detection and Prevention of MAC Layer Misbehavior in Ad Hoc Networks” SASN 2004 Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks

[49] 2.4_GHz_Wi-Fi_channels_(802.11b,g_WLAN).svg: Michael Gauthier, Wireless Networking in the Developing World

[50] "IEEE 802.11-2007: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications". IEEE. 2007-03-08. http://standards.ieee.org/getieee802/802.11.html.

[51] "The Evolution of 802.11 Wireless Security". ITFFROC. 2010-04-18. http://itffroc.org/pubs/benton_wireless.pdf.

[52] Nicole R. Nagel, Ruzbeh Shokranian, Jacir L. Bordim, K. Nakano” MAC Layer Misbehavior on Ad Hoc Networks” EUC '08 Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing - Volume 02 Pages 538-542

[53] S. Bansal and M. Baker, “Observation-based cooperation enforcement in ad hoc networks,” Stanford University, CA, Tech. Rep., July 2003.

[54] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovac, Mobile Ad Hoc Networking. Wiley-IEEE Press, 2003.

[55] S. Buchegger and J. L. Boudec, "Performance analysis of the CONFIDANT protocol: Cooperation of nodes — fairness in dynamic ad-hoc networks," in Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC). Lausanne, CH: IEEE, June 2002.

[56] S. S. Frank Kargl, Andreas Klenk and M. Weber, "Advanced detection of selfish or malicious nodes in ad hoc networks," University of Ulm, Dep. of Multimedia Computing, Ulm, Germany, Tech. Rep., 2004.

[57] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputationbased incentive scheme for ad-hoc networks," in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC2004), vol. 2. IEEE, March 2004, pp. 825–830.

[58] X. Li, A. Nayak, I. Ryl, D. Simplot, and I. Stojmenovic, "Secure mobile ad hoc routing," Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on, vol. 2, pp. 737–742, May 2007.

[59] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in Proceedings ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM), vol. 2, August 2000, pp. 255–265.

[60] P. Michiardi and R.Molva, "CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," Institut Eurecom, France, Tech. Rep. EURECOM+816, December 2001.

[61] S. Zhong, Y. Yang, and J. Chen, "Sprite: A simple, cheat-proof, credit based system for mobile ad hoc networks," in In Proceedings of IEEE INFOCOM'03, vol. 3, San Francisco, CA, 30 March–3 April 2003, pp. 1987–1997.

[62] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru and Herbert Rubens "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures"

[63] Baruch Awerbuch - Reza Curtmola - David Holmer - Cristina Nita-Rotaru – Herbert Rubens "Mitigating Byzantine Attacks in Ad Hoc Wireless Networks (Technical Report Version1) March 2004"

[64] W. Lou, W. Liu, and Y. Fang. SPREAD: enhancing data confidentiality in mobile ad hoc networks. In Proceedings of the IEEE computer and communications societies (INFOCOM), 2004.

[65] P. Papadimitratos and Z. J. Haas. Secure data transmission in mobile ad hoc networks. In Proceedings of the 2003 ACM workshop on wireless security (WiSe), pages 41–50, New York, NY, USA, 2003. ACM Press.

[66] R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya. On designing MAC protocols for wireless networks using directional antennas. IEEE transactions on mobile computing, 5(5):477–491, 2006.

[67] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes:A defense against wormhole attacks in wireless ad hoc networks,"in Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), April 2003.

[68] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in ACM Workshop on Wireless Security (WiSe), 2003.

[69] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in The 6th ACM International Conference on Mobile Computing and Networking, August 2000.

[70] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)," IEEE Journal on Selected Areas in Communication, vol. 18, no. 4, 2000.

[71] A. J. Menezes; P. C. van Oorschot; S. A. Vanstone (1997). Handbook of Applied Cryptography. ISBN 0-8493-8523-7. http://www.cacr.math.uwaterloo.ca/hac/.

[72] Christof Paar, Jan Pelzl, "Introduction to Public-Key Cryptography", Chapter 6 of "Understanding Cryptography, A Textbook for Students and Practitioners". (companion web site contains online cryptography course that covers public-key cryptography), Springer, 2009.

[73] IEEE 1363: Standard Specifications for Public-Key Cryptography

[74] http://cseweb.ucsd.edu/~mihir/cse207/w-asym.pdf

[76] Lei Guang, Chadi Assi, "Vulnerabilities Assessment of Ad Hoc Networks to MAC Layer Misbehavior" Wireless Communications and Mobile Computing, Wiley, (Accepted January 2006).

[77] M. Cagalj et al., "On Selfish Behavior in CSMA/CA Networks," Proc. IEEE INFOCOM, Mar. 2005.

[78] L. Guang, C. Assi, and A. Benslimane, "Interlayer Attacks in Mobile Ad Hoc Networks," Proc. MSN, Springer-Verlag LNCS, Dec. 2006.

[79] J. Konorski. MAC Contention in a Wireless LAN with Noncooperative Anonymous Stations," J. Networks, vol. 1, no. 2, Dec. 2006, pp. 27–36.

[78] S.-K. L. Yanxia Rong and H.-A. Choi, "Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis," Proc. IEEE INFOCOM, Apr. 2006.

[79] C. He and J. C. Mitchell, “Analyzing and Improving the IEEE 802.11 MAC Protocol for Wireless LANs,” Proc. NDSS, Feb. 2005.

[80] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the Key Scheduling Algorithm of RC4,” 8th Annual Wksp. Sel. Areas in Cryptography, Aug. 2001.

[81] Archana Bharathidasan, Vijay Anand Sai Ponduru “Sensor Networks: An Overview” Potentials, IEEE Proceedings, vol. 22, no. 2, pp: 20- 23, May 2003, Doi:10.1109/MP.2003.1197877.

[82] Chris Karlof, Naveen Sastry, David Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks", in proceedings of 2nd International Conference on Embedded networked sensor systems, pp: 162-175, November 2004, ISBN: 1-58113-879-2.
[83] Manzur Ashraf, Aruna Jayasuriya, Sylvie Perreau, “On Load Regulated CSMA”, The 28th ICDCS Workshops 2008.

[84] W. M. Kiesel and P. J. Kuehn, “A New CSMA/CD Protocol for Local Area Networks with Dynamic Priorities and Low Collision Probability,” IEEE Journal on Selected Areas in Communications, vol. 1, no. 5, pp. 869-876, November 1983.

[85] Younggoo Kwon, Yuguang Fang and Haniph Latchman, “A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs”, IEEE INFOCOM 2003.

[86] Roberto Verdure, Flavia Fabbri, Chiara Buratti, “Area Throughput for CSMA Based Wireless Sensor Networks”, IEEE EW2008, 22-25 June 2008, Prague, Czech Republic.

[87] Anis KOUBAA, Mário ALVES, Eduardo TOVAR, “A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks”, Technical Report, TR-060601, Version: 1.0, IPP-HURRAY, Portugal, June, 2006.

[88] T.R. Park, T.H. Kim, G.Y. Choi, S.Choi and W.H. Kwon, “Throughput and Energy Consumption Analysis of IEEE 802.15.4 slotted CSMA/CA”, Electronic Letters Online No. 20051662, IEE, 2005.

[89] D. J. Aldous, “Ultimate Instability of Exponential Back-Off Protocol for Acknowledgement-Based Transmission Control of Random Access Communication Channels,” IEEE Transactions on Information Theory, vol. IT-33, no. 2, pp. 219-223, March 1987.

[90] Y. Zhou, D. Wu, and S. Nettles. Analyzing and preventing MAC-layer denial of service attacks for stock 802.11 systems. In Workshop on BWSA, BROADNETS, October 2004.

[91] Jianliang Zheng, Myung J. Lee, “A Comprehensive Performance Study of IEEE 802.15.4”, Proc. IEEE journal on selected areas in communications, vol. 21, pp. 229-239, Feb. 2003