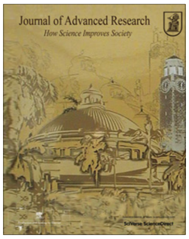
Journal of Advanced Research (2014) 5, 415–422

Cairo University

Journal of Advanced Research



ORIGINAL ARTICLE

DoS detection in IEEE 802.11 with the presence of hidden nodes

Joseph Soryal *, Xijie Liu, Tarek Saadawi

Electrical Engineering Department, The City College of New York, The City University of New York, United States

ARTICLE INFO

Article history:
Received 24 August 2013
Received in revised form 2 November 2013
Accepted 3 November 2013
Available online 9 November 2013

Keywords:
Network security
Wireless networks
IEEE 802.11
Markov Chain
Network mapping

ABSTRACT

The paper presents a novel technique to detect Denial of Service (DoS) attacks applied by misbehaving nodes in wireless networks with the presence of hidden nodes employing the widely used IEEE 802.11 Distributed Coordination Function (DCF) protocols described in the IEEE standard [1]. Attacker nodes alter the IEEE 802.11 DCF firmware to illicitly capture the channel via elevating the probability of the average number of packets transmitted successfully using up the bandwidth share of the innocent nodes that follow the protocol standards. We obtained the theoretical network throughput by solving two-dimensional Markov Chain model as described by Bianchi [2], and Liu and Saadawi [3] to determine the channel capacity. We validated the results obtained via the theoretical computations with the results obtained by OPNET simulator [4] to define the baseline for the average attainable throughput in the channel 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 by capturing the channel's bandwidth. In addition, the attacker strives to appear as an innocent node that follows the standards. The protocol resides in every node to enable each node to police other nodes in its immediate wireless coverage area. All innocent nodes are able to detect and identify the DoS attacker in its wireless coverage area. We applied the protocol to two Physical Layer technologies: Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) and the results are presented to validate the algorithm.

Introduction

IEEE 802.11 DCF specifications list two mechanisms to transmit a packet. The basic mechanism is a two-way handshaking method called "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 is successfully transmitted by listening to its own transmission. The second mechanism uses a four-way handshaking scheme called "Request-to-Send/Clear-to-Send" (RTS/CTS) before transmitting any packet. A node that is configured to use RTS/CTS mode performs channel reservation by sending out RTS short frame. The available receiver node responds to an RTS frame by sending back a CTS frame, and then packets contain data and ACK packet response follows. RTS frames may encounter collisions, which are detected by the absence of CTS responses. RTS/CTS mode increases the performance of the network through decreasing

* Corresponding author. Tel.: +1 646 284 4853.
E-mail address: jsoryal00@ccny.cuny.edu (J. Soryal).

Peer review under responsibility of Cairo University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jare.2013.11.001

the duration of a collision for long messages. In this paper, our focus is on DoS detection in the four-way handshaking scheme using "RTS/CTS" mode.

Malicious nodes employ several techniques to illegally increase their throughputs and capture the channel on the expense of other fair behaving nodes as demonstrated by Lolla et al. [5]. In IEEE 802.11, selfish nodes 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. A node is considered malicious when it deviates from the IEEE 802.11 MAC Standard [1]. Attackers employ shorter timeouts than these specified in the standards. With IEEE 802.11, nodes choose a back-off interval before attempting a transmission. The back-off interval gets to be increased according to set of rules before every retransmission attempt after every failed transmission. Attacker nodes choose a small or a fixed back-off interval before transmission attempts that does not follow the IEEE 802.11 standard. Detecting the back-off manipulation is very challenging due to its randomness as presented by Bellardo and Savage [6], Raya et al. [7], and Radosavac et al. [8]. The purpose of the proposed algorithm in this paper is to detect DoS attackers.

A major contribution in this paper is that the algorithm works in a wireless network with the presence of hidden nodes utilizing the mathematical results of Markov Chain modelling as baseline. Also, a network mapping algorithm is used to detect the network's topology. Several researchers worked to detect the manipulation of the back-off timer in wireless networks where there are trusted Access Points (AP) as presented by Kyasanur and Vaidya [9], and Raya et al. [7], where a trusted AP regulates the senders' back-off timer values and detect the misbehaving nodes. Ad-hoc networks do not have centralized authority that assigns and monitors the back-off timer values for each node, which is a challenging task. The presented algorithm can be applied to a distributed 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. As demonstrated by Lolla et al. [5] where the authors assume that the nodes are cooperating and they announce the state of their pseudo-random sequences so node monitors the behavior of other nodes. This approach assumes the cooperation from an attacker which is not realistic. Our 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. Bora et al. [10] 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. Our 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 "Hello" packets only once during the communication session. Alsahag and Othman [11] proposed a method to make the AP 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 distributed 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. Rong [12] proposes 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. Our algorithm does not require any hardware additions or clocks. The majority of researches that were performed on back-off timer manipulation detection assumed that there are no hidden nodes as presented by Soryal and Saadawi [13]. Few papers presented the concept of detection with the presence of hidden nodes as described by Lolla et al. [5], and Cárdenas et al. [14]. Lolla et al. [5] assume cooperation among nodes, which is not realistically applicable to DoS attacks. Raya et al. [7] propose new messages to the existing packets used by IEEE 802.11 which increases the network overhead unnecessarily. The following sections include description to the IEEE 802.11 DCF CTS/RTS scheme and the DoS impact. The throughput analysis for Markov Chain and the algorithm with the results are presented to prove the concept and the validity of the algorithm.

Methodology

CTS/RTS mode

IEEE 802.11 DCF standards [1] use 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. Each slot defines the inter-frame-space (IFS) intervals and determines the back-off values for nodes inside the network. Whenever a node has a packet to transmit, it senses the medium and if it is busy, the node waits until the medium becomes idle for a period equivalent to the Distributed Inter Frame Space (DIFS) period, and then computes a random back-off time which is specified by an integer value and is equivalent to a number of time slots. The Contention Window (CW) is the idle period after a DIFS period.

Nodes are only allowed to transmit at the beginning of each Slot-Time. The Slot-Time size (*Sigma*) is set equal to the time needed for a node to detect a packet transmission from adjacent nodes 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 which is defined as the time required to switch from the receiving to the transmitting state and also for the time to signal to the MAC the state of the channel defined as (Busy Detect Time). Nodes with packets to transmit select a back-off based on the Contention Window defined as [Back-off = int(CW × rand × slot time)]. The term "rand" is a random number uniformly distributed between 0 and 1, and $CW_{min} < CW < CW_{max}$, where CW_{min} is the minimum CW, and CW_{max} is the maximum CW. Firstly, the node that has a packet to transmit selects a back-off time in the range $[0, CW_{min} - 1]$, where CW_{min} is the minimum Contention Window size. When the channel gets to idle state, after another DIFS period, nodes decrement the 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 timer. This process continues until the timer reaches zero then the node transmits the packet. If the sender receives an ACK from the destination, the transmission is assumed to be successful and the node sets its CW back to $CW_{min} - 1$. If two or more nodes decrement their timers to reach zero simultaneously, the packets will collide, and each node will have to start over and selects a new back-off time by doubling the Contention Window value [2^{*} CW$_{min}$]. During the kth retransmission attempt the Contention Window will have the form [$0.2k^{*}$ CW$_{min}$] and will be doubled until it reaches CW$_{max}$.

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 paper, 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) as shown in Table 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 1 MHz. FHSS supports two rates of 1 Mbps and 2 Mbps. There are seventy-eight hopping sequences and each sequence would use seventy-nine hops. Fifteen 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.

Network configuration and DoS attack impact

The network configuration is presented in Fig. 1 where there are three areas A, B, and C. Nodes located in area B can hear all other nodes located in areas B and C. Nodes located in area A can hear all other nodes located in areas A and C. Nodes in area B cannot hear nodes in area A and vice versa.

The algorithm is scalable and deals with the number of nodes in each area as an independent variable and performs the calculations accordingly. For the sake of simplicity in presenting this paper and conducting the simulations, we assume that the number of nodes in each area is constant, although the Markov Chain model handles any variable number of nodes in general.

Table 1 PHY layer parameters.

Parameter	FHSS	DSSS
Slot Time "**σ**"	50 us	20 us
SIFS	28 us	10 us
DIFS	128 us	50 us
PHY header	128 bits	192 and 96 (us)
MAC header	272 bits	28 bytes
ACK	112 bits	14 bytes
CTS	112 bits	14 bytes
RTS	160 bits	20 bytes
Channel bit rate	1 Mbps	11 Mbps
CW$_{min}$, CW$_{max}$	15, 1023	31, 1023
Packet size	8000 bits	8000 bits
Signal extension	N/A	N/A

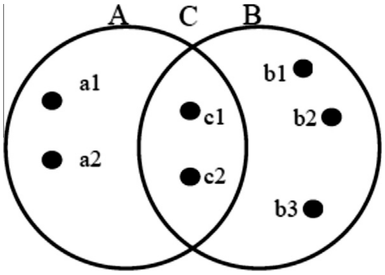


Fig. 1 Network configuration.

The DoS attacker can implement the attack by several methods. The most prevalent method is altering the firmware code on the Network Interface Card (NIC). Also, in some instances attackers modify the hardware. The first method is a much easier to implement from the feasibility and cost point of view. In our paper, the solution is directed toward 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 keeps transmitting packets that do not contain any useful information just to occupy the channel. The attacker backs off only one slot every time a packet is ready to be sent out or when it encounters a collision while the other innocent nodes 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 paper is 8000 bits so it can be sent in one time slot without the need of fragmentation.

For the simplicity, we assume the following constant number of nodes in each area throughout the paper – these numbers are used for the simulations and solving the Markov Chain: area A has 2 nodes, area B has 3 nodes, and area C has 2 nodes as shown in Fig. 1.

In Fig. 2 the simulation shows the comparisons between traffic sent by innocent nodes under fair conditions without the attacker (red line) and the traffic sent with the attacker present (blue line) for a network using DSSS technology. The effect of the DoS attack on the innocent nodes is very clear that once the attacker existed the innocent nodes are deprived from accessing the channel to send anything.

Markov Chain

Fig. 3 shows a two-state Markov Chain model that models the IEEE 802.11 wireless network. Such model is extracted for each of the three areas (A, B and C) as shown in Fig. 1. This allows obtaining each node's throughput values for the purpose of identify the attack. Bianchi's Markov Chain model [2] and Liu and Saadawi [3] is extended to calculate the individual rate in "Packets 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.

The assumption is that all nodes have packets to transmit all the time (saturation condition) and the number of nodes is fixed during the communication session.

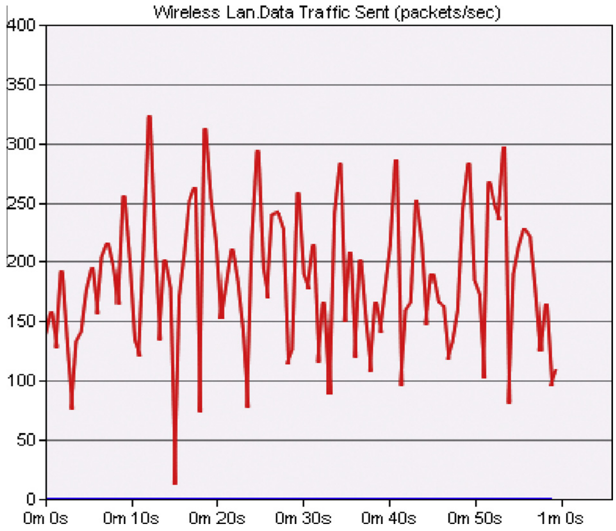


Fig. 2 Data traffic sent comparison using DSSS technology.

Firstly, we obtain the Transmission Probability for each area to calculate the throughput for this specific area and finally obtain the individual throughput for each located in this specific area.

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

n_a, n_b, and n_c are the number of nodes in areas A, B, and C respectively.

$$\frac{1}{b_{0,0}} = \frac{1-p^{L+1}}{2(1-p)} + \frac{w_0}{2}\left[1 + \frac{2p-(2p)^{m+1}}{(1-2p)} + \frac{2^m(p^{m+1}-p^{L+1})}{1-p}\right] \tag{1}$$

$$\tau = \sum_{j=0}^{L} b_{j,0} = \frac{1-p^{L+1}}{1-p} \times b_{0,0}$$

$$\tau_x = \frac{1-p_x^{L+1}}{(1-p_x)\left\{\frac{1-p_x^{L+1}}{2(1-p_x)} + \frac{w_0}{2}\left[1 + \frac{2p_x-(2p_x)^{m+1}}{(1-2p_x)} + \frac{2^m(p_x^{m+1}-p_x^{L+1})}{1-p_x}\right]\right\}} \tag{2}$$

τ in the different areas

$$\tau_a = \frac{1-p_a^{L+1}}{(1-p_a)\left\{\frac{1-p_a^{L+1}}{2(1-p_a)} + \frac{w_0}{2}\left[1 + \frac{2p_a-(2p_a)^{m+1}}{(1-2p_a)} + \frac{2^m\left(p_a^{m+1}-p_a^{L+1}\right)}{1-p_a}\right]\right\}}$$

$$\tau_b = \frac{1-p_b^{L+1}}{(1-p_b)\left\{\frac{1-p_b^{L+1}}{2(1-p_b)} + \frac{w_0}{2}\left[1 + \frac{2p_b-(2p_b)^{m+1}}{(1-2p_b)} + \frac{2^m\left(p_b^{m+1}-p_b^{L+1}\right)}{1-p_b}\right]\right\}}$$

$$\tau_c = \frac{1-p_d^{L+1}}{(1-p_d)\left\{\frac{1-p_d^{L+1}}{2(1-p_d)} + \frac{w_0}{2}\left[1 + \frac{2p_d-(2p_d)^{m+1}}{(1-2p_d)} + \frac{2^m\left(p_d^{m+1}-p_d^{L+1}\right)}{1-p_d}\right]\right\}}$$

According to the given topology, p in the different area

$$p_a = 1-(1-\tau_d)^{n_d}(1-\tau_a)^{n_a-1}$$

$$p_b = 1-(1-\tau_d)^{n_d}(1-\tau_e)^{n_e}(1-\tau_b)^{n_b-1}$$

$$p_c = 1-(1-\tau_d)^{n_d-1}(1-\tau_a)^{n_a}(1-\tau_b)^{n_b}$$

Throughput in the different area:

$P_{i,tr}$ is defined as the probability that least one transmission occurs within node i's coverage area in a random time slot.

$$P_{i,tr} = 1-(1-\tau_i)\prod_{u=all\ i's\ neighbours}(1-\tau_u) \tag{3}$$

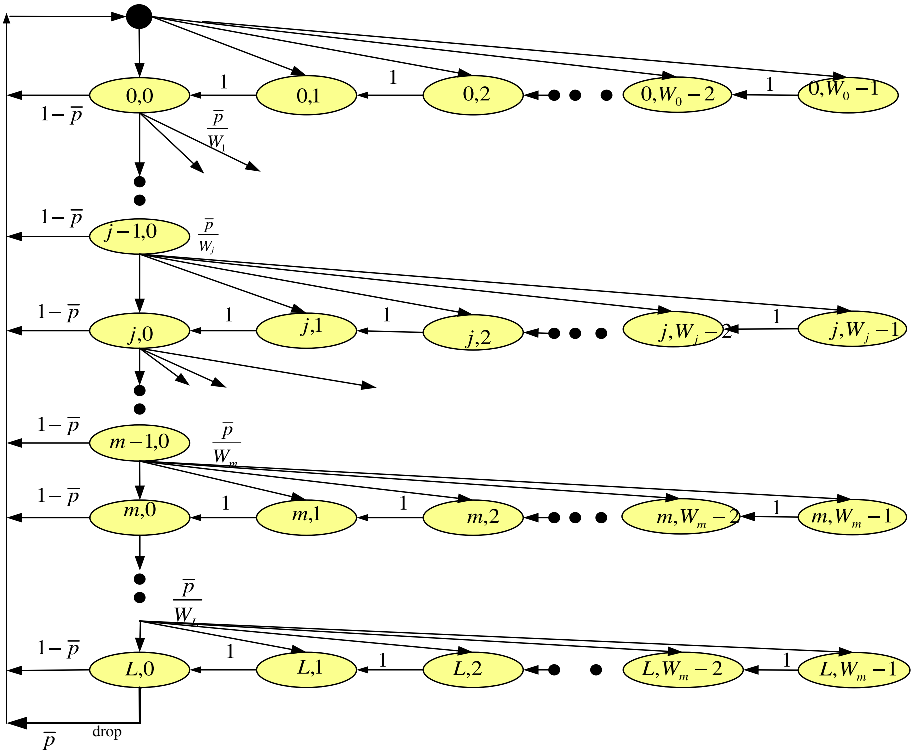


Fig. 3 Two-dimensional Markov Chain model for a given IEEE 802.11 wireless network.

$P_{i,success}$ is the probability that node i successfully transmits its packet to another node, and this equals the probability that exactly only one node transmits on the channel covered by node i in a given time slot, and no hidden node transmits either. Hence the formulas for $P_{i,tr}$ and $P_{i,success}$ are given by

$$P_{i,success} = \tau_i \prod_{u=all\ i's\ neighbours} (1-\tau_u) \prod_{v=i's\ hidden\ station} (1-\tau_v) \qquad (4)$$

Let $throughput_i$ be the normalized capacity of node i,

$$throughput_i = \frac{P_{i,success}E[P]}{(1-P_{i,tr})\sigma + P_{i,success}T_S + [P_{i,tr} - P_{i,success}]T_C} \qquad (5)$$

$E[length]$ is the average length of a slotted time and $E[payload]$ is the average packet payload size. $P_{i,success}E[payload]$ *is* the average amount of payload information successfully sent out in a time slot. $E[length]$ will be $(1-P_{i,tr})\sigma + P_{i,success}T_S + [P_{i,tr} - P_{i,success}]T_C$. σ is the duration of a time slot. Here the term $(1-P_{i,tr})$ accounts for an idle time slot with probability $1-P_{i,tr}$. $P_{i,success}T_S$ is the successful transmissions of node i with successful probability of $P_{i,success}$. The term $[P_{i,tr} - P_{i,success}]$ T_C represents the collision duration. T_S is the average time needed for a successful transmission, and T_C is the average duration for the collision. T_C and T_S are then derived for the RTS/CTS mechanism. Obtaining the throughputs for RTS/CTS accesses the mechanism:

Then we obtain τ_x and p_x

$$\begin{aligned} T_{S,rts} &= [t_{phy} + \text{RTS}] + \text{SIFS} + \delta + [t_{phy} + \text{CTS}] + \text{SIFS} + \delta \\ &\quad + [t_{phy} + t_{\text{MAC}} + E[packet]] + \text{SIFS} + \delta + [t_{phy} + \text{ACK}] \\ &\quad + \text{DIFS} + \delta T_{C,rts} \\ &= [t_{phy} + \text{RTS}] + \text{DIFS} + \delta \end{aligned}$$

$$throughput_a = \frac{P_{a,success}E[P]}{(1-P_{a,tr})\sigma + P_{a,success}T_S + [P_{a,tr} - P_{a,success}]T_C} \qquad (6)$$

$$throughput_b = \frac{P_{b,success}E[P]}{(1-P_{b,tr})\sigma + P_{b,success}T_S + [P_{b,tr} - P_{b,success}]T_C} \qquad (7)$$

$$throughput_c = \frac{P_{c,success}E[P]}{(1-P_{c,tr})\sigma + P_{c,success}T_S + [P_{c,tr} - P_{c,success}]T_C} \qquad (8)$$

To validate the theoretical results described above, we compared the numerical results produced by solving the Markov Chain using parameters listed in Table 1 with the results generated by OPNET [4] simulator under the saturation condition. Matlab [15] was used to solve the Markov Chain and obtain the numerical results.

Table 2 shows the values obtained from Markov Chain modelling and from OPNET simulation to show the average achievable throughput (packets/s) for each area for both FHSS and DSSS under saturation conditions. Since all nodes have the same condition, then every node has the same probability 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 theoretical calculations and empirical results and shows the significance of the detection thresholds accuracy. It is noted that the results for areas A and B are slightly different in the simulation results because of the imperfection of wireless nature. It is also noted in Table 2 that the theoretical results 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 environments. One benefit of using the theoretical results as opposed to empirical results that the theoretical results generate higher values of thresholds which help eliminating false positives. As shown in the previous section that the number of the CTS packets received is equal to number of data packets transmitted.

Detection process

According to the IEEE 802.11 implementations, the number of successful data packets transmitted by any given node is equal to the CTS packets received by this specific node. The CTS packets are designed to be heard by every single node within its coverage area. All the nodes besides the one that the CTS packet is destined to, will have to update their NAV so other nodes halt transmitting any packets during the NAV period to eliminate the chances of collisions. We modified the OPNET [4] 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 received CTS packets is equal to the number of data packets sent. Simulation results show that the number of CTS received by node_1 is the same number of packets sent by this specific node to other nodes in the network. Based on that concept, the detection algorithm depends on modifying the IEEE 802.11 DCF firmware to equip each node to monitor the network with very low cost (in terms of processing and memory consumption) solution without 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 discarding it. Upon network communication initialization, which includes the initial exchange of Hello packets, every node maps out which nodes it can sense in its range and compile a list of MAC addresses that it can communicate with. This list is broadcasted by all the nodes. Then each node compares 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 marks that domain for node count (area A or B in Fig. 1).

If the two lists do not match then this node identifies itself as an overlapping node that shares two domains (area C in Fig. 1). The lack of cooperation from the attacker does not impact the results because the detection threshold has enough tolerance to account for a missing count from a node. The algorithm has two phases that run in series. The first phase

Table 2 Comparison between average throughputs (packets/s) for each area.

PHY technology	Area A	Area B	Area C
FHSS (simulation)	100	110	100
FHSS (theoretical)	105	110	105
DSSS (simulation)	360	360	270
DSSS (theoretical)	510	520	510

is the network mapping where all the nodes determine their coverage area to decide which Markov Chain Throughput equation should be used, either an exclusive domain (“A” or “B”) or an overlapping area (“C”). Accordingly each node chooses the appropriate Markov Chain equation to generate the throughput. The lists created during the network mapping phase are appended to the Hello packets and is only exchanged once among the nodes after the initialization of the network. Each node further processes each received list to derive the number of the nodes in each area.

Example to explain the network mapping technique – using Fig. 1:

Area “A” has 2 nodes: a1, a2.
Area “B” has 3 nodes: b1, b2,b3.
Area “C” has 2 nodes: c1, c2.

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:

a1: (a2, c1, c2) a2: (a1, c1, c2).
b1: (b2, b3, c1, c2) b2: ((b1, b3, c1, c2) b3: ((b1, b2, c1, c2).
c1: (a1, a2, b1, b2, b3, c2) c2: (a1, a2, b1, b2, b3, c1).

Now, for instance node a1 compares its own list with the others and it finds that the list from a2 is identical to its own list except for the node itself, then it decides that a1 and a2 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 happens with all other nodes. When it is c1’s turn to compare the lists, it finds that c2 has the same number of nodes which leads node c1 to conclude that c1 and c2 belong to the same region. In addition, c1 finds its list (a1, a2, b1, b2, b3, c2), is longer than the others heard then node c1 realizes that its location is in the overlapping area in Fig. 1 and will use these numbers for the calculation of the throughput.

Phase I is triggered after the exchange of the first round of Hello packets and the lists are 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 is triggered to detect the attackers based on the network topology discovered in phase I.

The algorithm

The Algorithm that resides at each node is as follows:

```
  Phase I: Network Mapping:
  Each node maps 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 should 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, $k = A,B$, or C */
Broadcast List_x
Receive List_1 through List_$n_k$ /* (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 neighbours 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 > Number of 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 knows how many nodes in its immediate area and other areas – also, the nodes in overlapping area know themselves) */
  Phase II: Detection:
  Each node implements the detection algorithm
Count $n_k$ /* “Number of Nodes in the immediate area and other areas” */
Create $n_k$ Counters
Calculate Average Throughput for each node /* based on Markov Chain modelling above for each area */
When CTS Received
If (Destination Address = My Address)
Do Nothing
Else
{
Update Counter (Destination Address)
Calculate Rate
  /* rate of received CTS packets/second for each Destination Address */   }
If
CTS_node_x rate < Average Individual Throughput
Do Nothing
Else
Announce “node_x is implementing DoS attack” /* it is shown as print command in our OPNET simulation and used it as output */
End
```

For the simulation, we use Matlab [15] to solve the Markov Chain mathematical model and feed the results to OPNET simulator for the detection threshold. The numerical results are considered the average number of packets any node can send in the presence of other number of nodes (as calculated in Markov Chain modelling), 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 to attack the network.

Results and discussion

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 shows that the thresholds shown in Table 2 are exceeded whenever an attacker is present in the network which enables the innocent nodes to detect the attacker using the theoretical baselines generated by 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

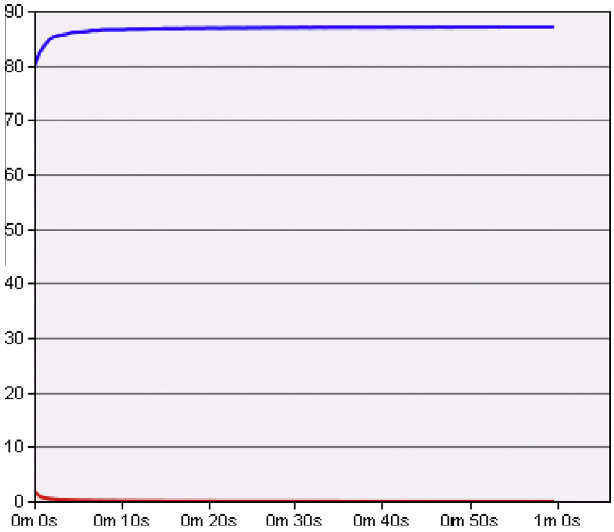


Fig. 4 FHSS – Node c1 – Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker (a1 represented by the blue line).

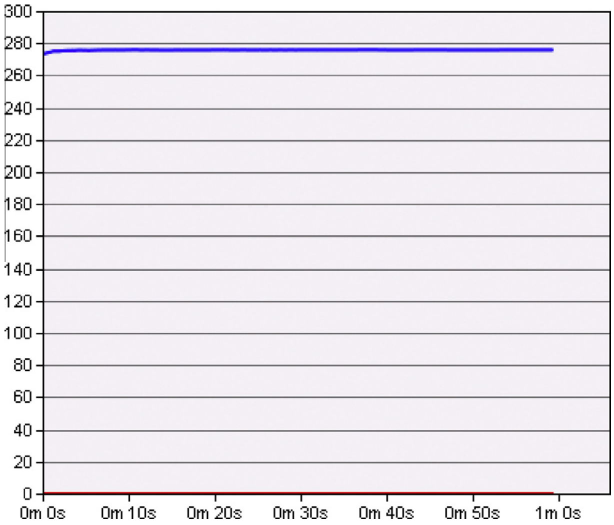


Fig. 5 DSSS – Node c2 – Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker (a1 represented by the blue line).

as an attacker, the algorithm does not react to instantaneous spike but rather looks for a moving average over time to ensure that any spike by an innocent node is not mistaken for an attacker. The simulation setting examined the presence of the attacker node in two regions (A and C). So one round of simulation runs assumed that the attacker is in area A and the second run assumed that is in area C.

In Fig. 4 for the FHSS case and Fig. 5 for the DSSS case, an innocent node in Area C is listening to the CTS packets sent in the medium and finds that one node in Area A 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 line 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 C, the channel capacity is 105 Packets/s (57 Packets/s per node) for FHSS and for DSSS is 510 Packets/s (250 Packets/s 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.

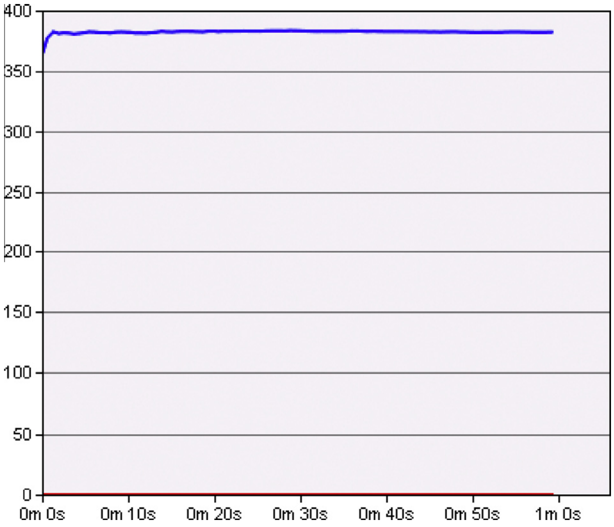


Fig. 6 DSSS – Node a2 – Number of CTS packets heard by innocent node for two other nodes – one of them is an attacker (c1 represented by the blue line).

In Fig. 6, an innocent node in area A was listening to the CTS packets sent in the medium and found that one node in area C is exceeding the threshold calculated for the channel in this area divided by the number of nodes in this area. According to the thresholds calculated in area C, for DSSS is 510 Packets/s (250 Packets/s per node) with the existence of two nodes, the attacker achieved number of transmitted packets well over the threshold and is detected by this innocent node and marked as an attacker.

Conclusions

A novel approach 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 utilizing the numerical results obtained by solving the Markov Chain model. Combining the numerical results with the specifications of the IEEE 802.11 DCF RTS/CTS protocol, a developed code was embedded into IEEE 802.11 code to enable individual nodes to monitor the network and detect the attacker. The simulation results proved our concept with very high accuracy without any false positives recorded and this in part caused by taking advantage of the higher values of the theoretical results generated by solving Markov Chain model. This solution is scalable and applicable for distributed environment where there is no centralized authority overseeing the communication process and transaction among the nodes. In the future, a method to combat the attack based on a game theoretic approach will be developed and will be appended to the presented algorithm.

Conflict of interest

The authors have declared no conflict of interest.

References

[1] IEEE Standard 802.11 – Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications; 1999.

[2] Bianchi G. Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J Sel Areas Commun 2000;18(3):535–47.

[3] Liu X, Saadawi T. Throughput analysis of IEEE 802.11 multihop ad hoc wireless networks under saturation condition. In: Proceedings of ISCC; 2010. p. 245–8.

[4] www.OPNET.com.

[5] Lolla VN, Law LK, Krishnamurthy SV, Raishankar C, Manjunath D. Detecting MAC layer back-off timer violations in mobile ad hoc networks. In: ICDCS '06 Proceedings of the 26th IEEE international conference on distributed computing systems; 2006. 63p.

[6] Bellardo J, Savage S. 802.11 Denial-of-service attacks: real vulnerabilities and practical solutions. In: Proceedings of the USENIX security symposium, Washington, DC; 2003.

[7] Raya M, Hubaux J, Aad I. DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots. In: Proceedings of MOBISYS; 2004.

[8] Radosavac S, Cárdenas AA, Baras JS, Moustakides GV. Detecting IEEE 802.11 MAC layer misbehavior in ad hoc networks: robust strategies against individual and colluding attackers. J Comput Secur – Special Issue Secur Ad-hoc Sensor Netw 2007;15(1):103–28.

[9] Kyasanur P, Vaidya NH. Detection and handling of MAC layer misbehavior in wireless networks. In: Proceedings of 2003 international conference of dependable systems and networks; 2003. p. 173–82.

[10] Bora RP, Harihar D, Sehrawat S. Detection, penalization and handling of misbehavior in ad hoc wireless networks. IAENG Int J Comput Sci 2007;33:1, IJCS_33_1_3.

[11] Alsahag AM, Othman M. Enhancing wireless medium access control layer misbehavior detection system in IEEE 802.11 network. J Comput Sci 2008;4(11):951–8.

[12] Rong Yanxia. Detecting MAC layer misbehavior and rate adaptation in IEEE 802.11 networks: modeling and SPRT algorithms. PhD dissertation, The George Washington University; 2008. doi:3320934.

[13] Soryal J, Saadawi T. IEEE 802.11 DoS attack detection and mitigation utilizing cross layer design. Ad Hoc Networks 2014;14:71–83.

[14] Cárdenas AA, Radosavac S, Baras JS. Detection and prevention of MAC layer misbehavior in adhoc networks. In: SASN 2004 Proceedings of the 2nd ACM workshop on security of ad hoc and sensor networks; 2004.

[15] www.mathworks.com.