Real Time Detection of MAC Layer DoS Attacks in IEEE 802.11 Wireless Networks

Mallesham Dasari
Stony Brook Univeristy
Stony Brook, New York, USA.
Email: mdasari@cs.stonybrook.edu

***Abstract*—In this paper, a real time detection of Medium Access Control (MAC) layer attacks in IEEE 802.11 wireless networks is proposed. There can be different kinds of Denial of Service (DoS) attacks observed at the MAC layer such as misbehaviour and selfish attacks. The malicious nodes manipulate the MAC protocol parameters such as back-off time, network allocation vector value and short inter frame space, or flood the network with huge volume of dummy packets. With this, the attacker nodes capture entire network bandwidth causing legitimate nodes not communicate with other nodes, consequently decreasing the throughput of the nodes significantly. This paper gives an effective real time detection of these attacks with minimal detection delay. We collect the delay and throughput data and apply a change point detection algorithm to observe the change of distribution. To observe the effect of these attacks, two types of attacks: back-off manipulation and RTS flooding are simulated using Network Simulator (NS-3). The simulation results shows the efficiency of the detection algorithm in terms of delay and throughput results.**

***Index Terms*—IEEE 802.11 Wireless Networks, Denial of Service Attacks, MAC Layer Attacks, Change Point Detection.**

I. Introduction

The IEEE 802.11 MAC protocols execute Carrier Sensing Multiple Access with Collision Avoidance (CSMA/CA) [1] for media access. These protocols lies under the assumption that all nodes in the network has same regulations. However, the vulnerability of these protocols is very high when a malicious node manipulates its protocol stack with optimal parameters classified into different Denial of Service (DoS) attacks. Several attacks are being studied in the literature to illustrate the effect on the performance of the system. Legacy network security provides a cryptography based protection for the packets using encryption standards such as Wired Equivalent Privacy (WEP) and WiFi Protected Access (WPA), however, this cannot be applied at the MAC layer to detect the such attacks. These DoS attacks are categorized into Selfish and Misbehavior attacks [2] as shown in Fig .1. The misbehaving nodes target the network to drain the performance of other nodes by monopolizing the channel. This kind of attacks include, Single Adversary Attack (SAA), Colluding Adversary Attack (CAA), Misdirection and RTS flooding attacks. In SAA. a single malicious node sends enormous flows to legitimate nodes, consequently draining the energy and channel capacity in its vicinity. In CAA, two attackers send continuous data to each other causing other nodes to delay their transmission and reducing the throughput. The misdirection is kind of man in the middle attack where the attacker redirects the packet from a legitimate node to another malicious node to gain the knowledge. The RTS flooding is the attackers sends continuous RTS packets without any data packets. The legitimate nodes reply with CTS packets although there would be no data flow and wait for SIFS duration.

The selfish attacks include RTS dropping, manipulating its contention parameters such as DCF Inter Frame Space (DIFS), Short Inter Frame Space (SIFS), Network Allocation Vector (NAV) and back-off contention window. In reducing the DIFS and SIFS, the selfish node can get the channel very frequently in sending RTS, CTS or actual data frames. In NAV attack, the attacker send the RTS packet with longer duration updated in NAV to other nodes.In RTS dropping attack, the selfish node intentionally drops the RTS packets to enhance the probability of accessing the channel. Both kind of attacks pose a potential threat to the network in terms of channel utilization, throughput, Quality of Service (QoS) etc. In this paper, we consider four types of attacks: 1) Back-off attack using shorter contention window and 2) RTS flooding. We model the data distribution such as delay and throughput over time series and apply a change point detection method to detect the abnormal change in the distribution. The rest of the paper is organized as follows: Section II gives the overview of literature work. In Section III, the proposed real time detection method is explicated. Section IV shows the simulation results and the concluding remarks presented in Section V.

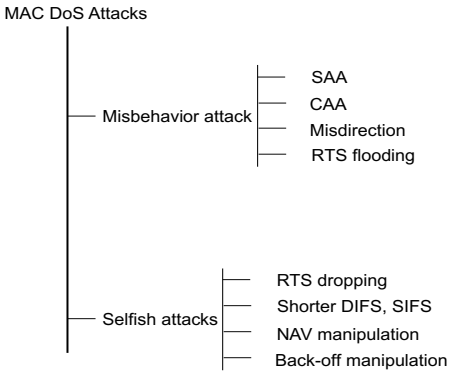

Fig. 1: Types of MAC layer attacks

978-1-5090-6196-9/17/$31.00 ©2017 IEEE

II. Related Work

In [3], the authors propose a MAC layer attack detection technique by Distributed Coordination Function (DCF) which observes the reason for increased number of collisions due to DoS attack. In this method, although the attacks are detected successfully, it is very difficult to deploy the 802.11 stack by modifying the protocol stack. In [4], a cross layer based intrusion detection method is proposed to detect the malicious attacks by analyzing the pattern of trace files. This method also indicates the lower occurrences of false positives by using a watch-dog monitor to observe the mis direction and intended dropping of packets. In [5], a real time DoS attacks detection method is proposed for IEEE 802.11 networks, where the method indicates how the channel status i.e, the number of collisions computed probabilistically. This probabilistic distribution of collision explainablity shows a sensitive indicator such that it gives minute changes the in the network conditions even in the case of changing number of nodes in the network. In [2], a new machine learning approach is proposed to come up with classifiers that enable the administrator to choose among a host of classification algorithms. This method mainly focused on precision (accuracy) and recall (detection rate) using different machine learning approaches. In [6][7], the authors propose a distributed method by calculating the CTS packet distribution with Markov analysis. This method calculates the theoritical bound on the CTS packet statistics using Markov chains and compare with experiencing CTS statistics then identify the attacker using MAC address if there is any unexpected behaviors. In [8], the authors discuss the MAC layer misbehaviors of attackers in two scenarios: a)individual, b)colluding selfish attacks. This method focuses particularly on back-off manipulation attack to ensure the honest behavior of non-colluding participants. A similar analysis is done in [9][10] for 802.11 based Mobile Ad hoc Networks (MANETs), where there are single and multiple adversary attacks with flooding enormous amounts of RTS packets injected into the network. The authors in [10], proposed a packet by packet authentication process by embedding shared key communicated over a secured channel. This however, is again a big deal to change the packet format of the 802.11 protocols.

In [11], a cross layer attack is analyzed to increase the attacker's power and to reduce the risk of being attacked, with trust fusion based multi layer defense mechanism. In [12], a classification of different MAC layer DoS attacks is given. This classification takes different standards' MAC layer attacks considering types of attacks such as unfairness, excessive collisions, sleep attacks and security violation. In [13], the MAC level packet jamming attacks are analyzed to detect the commercially available jamming devices already. This method takes the packet send ratio and packet delivery ratio in estimating the depth of attack. In [14], a centralized solution is proposed that acts as a back end server to observe the DoS attacks by maintaining three tables and a timer. In [15], a robust detection and defense of selfish misbehaviors is proposed for multi-hop ad hoc networks. This method has an observer that calculates the joint probability for a node and compares with normal nodes to determine if it is a selfish node. They come up with a penalty mechanism that brings down the performance of the selfish node. In [16], a real time detection of DoS attacks is proposed based on statistical process control. This method is excerpted from the industrial field in quality management. The method detects the greedy behaviors by observing the abnormal throughput and inter packet interval for each node. In [17], an efficient Markov-RED-FT based technique that calculates the flow trust values to safeguard the legitimate flows. However this method analyses only back-off attacks. In [18], the authors propose an analysis of RTS/CTS attacks where two attackers send continuous RTS and CTS packets each other. A real time detection of 802.11p attacks is proposed in [19], to analyze the jamming of periodic position messages. This gives detection times compared to the limited performance evaluations of previous work.

III. Real Time Change Point Detection

A. System Model

The traffic generated at each node in the network follows a unique distribution in terms of delay and throughput measurements. This distribution changes when an attacker intrudes the network and causing the performance degradation to the legitimate nodes. This leads to abnormal increase in the delay and a decrease in throughput. However, this can also be observed in the case of a congested network with many the legitimate flows. The attack scenario can be easily distinguishable from the network congestion because the change is triggered by all nodes in the later case where as only one node is responsible for change in the former case. We collect the data packets received over time series with end-to-end delay from sender to receiver. The cumulative measurements are taken instead of individual elements to minimize the false alarm rates. That is, the algorithm should not alarm the base station in the case of congested network, as the traffic can be highly unexpected in 802.11 networks due to contention.

The collected data points $X[i] = \{x_0, x_1, ..., x_t, ...\}$ over time series can be considered as stochastic process over a continuous time interval with standard deviation σ_X. A change in the probability distribution of this stochastic process can be considered as a change point. When there is an attack, the data distribution becomes $X[i] + Y[i]$ with standard deviation σ_{X+Y}, where $Y[i] = \{y_{t+1}, y_{t+2}, ...\}$. Initially, the data points follows a distribution $F(.)$ with a probability density function f. At a change point ξ, the distribution changes to $G(.)$ with corresponding density function g, due to attack. More precisely, the legitimate node chooses one of the following two hypotheses.

$$\mathcal{H}_0 : \qquad X[i], \qquad \forall n = 0, ..., t, .. \qquad \sim f$$

$$\mathcal{H}_1 : \exists \xi \in [0,t], \ni \begin{cases} X[i] & n = 0, ..., \xi - 1 \\ X[i] + Y[i] & n = \xi, ..., t.. \end{cases} \sim g$$

The hypothesis $\mathcal{H}_0$ is true as long as there is no attack. If there is an attack, then the hypothesis $\mathcal{H}_1$ is true and the algorithm alarms a change point. The detection method should have accurate detection with less false alarm rates and low detection delay. Let $\xi^{'}$ is the time when an alarm raises. The detection delay l_d or the false alarm case are shown below.

$$l = \xi^{'} - \xi, \ni \begin{cases} \xi^{'} \geq \xi & Accurate \\ \xi^{'} < \xi & False\ alarm \end{cases}$$

B. Detection

The most suitable change point detection methods for attack detection are Cumulative Sum (CUSUM) algorithms [20], [21] when pre and post change distributions are known. These methods use log likelihood ratio give below.

$$L_i = \log \left\{ \frac{g(x+y)}{f(x)} \right\}$$

And, the expected value of these pre and post change distributions are given as

$$\mathbb{E}_f = \int f(x) \log\{\frac{g(x+y)}{f(x)}\} dx$$

$$\mathbb{E}_g = \int g(x+y) \log\{\frac{g(x+y)}{f(x)}\} dx$$

The cumulative sum (s_t) is compared with a predefined threshold h. If s_t is greater than h, then it reports an alarm, as the algorithm detects a consistent positive drift after the k^{th} sample. The s_t is computed as:

$$s_t = \max_{k \leq t} \left\{ \sum_{i=1}^{t} L_i - \sum_{i=1}^{k} L_i \right\} = \max_{k \leq t} \sum_{i=k+1}^{t} L_i$$

$$s_{t+1} = \left\{ s_t + L_i(t+1) \right\}^+, \forall t \geq 0$$

Hence, the s_t can be computed recursively with $s_0 = 0$. The algorithm alarms the change as soon as it detects that s_t exceeds h. Knowing distributions f and g is high unlikely in the case of dynamic traffic in the network. Further, choosing the detection threshold for such unexpected traffic is not easy. Despite its effectiveness, this algorithm is not suitable for the networks with varying load. Hence, we come up with a method that requires no prior knowledge about distribution.

We consider a non parametric method assuming σ_X and σ_{X+Y} are unknown. Let $F_0(.)$ and $F_1(.)$ are the data distributions of pre and post change due attack. The cumulative sum is computed as

$$F_0(x) = \int_0^x \sqrt{\frac{2}{\pi \sigma_X}} e^{-\frac{y^2}{2\sigma_X}} dy$$

$$F_1(x) = \int_0^x \sqrt{\frac{2}{\pi(\sigma_X + \sigma_{X+Y})}} e^{-\frac{y^2}{2(\sigma_X + \sigma_{X+Y})}} dy$$

The rank statistic idea from [22] can be adopted to design change point detection for attack detection in nonparametric category. Let $R(i \smile t)$ is the rank of the i^{th} sample out of first t samples. The random samples of these ranks are $R(i \smile t)$, $R(i \smile t)$,...$R(i \smile t)$,.. very low compared to the ranks after attack, i.e, post change point ($t \geq \xi$). Let $R(i \frown t)$ is the inverse permutation of ranking of these samples such that $R(R(i \smile t) \frown t) = i$. The likelihood ratio of $\xi = k$ in taking a rank R is given as

$$\lambda_k^t(R) = \frac{P_{\xi=k}\{X_{R(1\frown t)} < ... < X_{R(t\frown t)}\}}{P_f\{X_{R(1\frown t)} < ... < X_{R(t\frown t)}\}}$$

$$= \sum_{m=0}^{n} \Upsilon_{k,m}^t(R), \forall 1 \leq k \leq n$$

where, $\Upsilon_{k,m}^t(R)$ can be computed from [22], with p, q, α and β are constant satisfying $p\alpha \geq q\beta$

$$\Upsilon_{k,m}^t(R) = \binom{t}{m} (\frac{1}{2})^t (\frac{p\alpha}{q\beta})^{U_k(t,m)} (2q\beta)^{t+1-k}$$

$$\times \prod_{i=1}^{m} \left(1 + \frac{V_k(i,t)}{i}(\beta - 1)\right)^{-1}$$

$$\times \prod_{i=m+1}^{t} \left(1 + \frac{U_k(i-1,t)}{n+1-i}(\alpha - 1)\right)^{-1}$$

where, $U_k(t,m) = \sum_{j=k}^{t} I(R(j \smile t) > m)$ and $V_k(t,m) = n + 1 - k - U_k(t,m)$. Finally, the Shiryayev-Roberts statistic $R_n = \sum_{k=1}^{t} \lambda_k^t$ and compared with the threshold h for detecting change point. Using this method, many types of DoS attacks not only at the MAC layer but also other layers of network stack without knowing the distribution of traffic in the network.

IV. Simulation Results

A. Simulation Setup

To study the performance of the nodes under these DoS attacks at the MAC layer, we used NS-3 Network Simulator. We hacked 802.11 Wifi MAC layer to create the an attacker node different from normal nodes. Due to the popularity of the IEEE 802.11, detecting MAC layer misbehavior has focused on this protocol. A selfish user in the IEEE 802.11 can use a whole range of methods to maximize its access to medium. The most effective way is to deploy different schemes for manipulating the rules of the MAC layer. For example, the attacker can manipulate the size of the Network Allocation Vector(NAV) and reserve large time periods to its neighbors, and it can also decrease the size of DIFS and SIFS etc. For all the simulation results unless otherwise specified, we use the topology shown in Fig. 2. We consider 9 nodes scattered in a grid of 100x100 terrain area. Out of all nodes, there are 8 normal nodes and one attacker node. Node 5 is sink node and remaining nodes are source nodes. Node 4 is the attacker.

In this paper, we focus on two important misbehavior attacks: back-off manipulation attack and RTS-dropping attack.

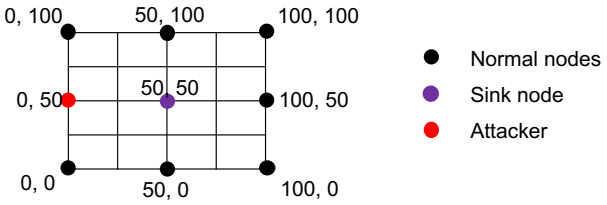

Fig. 2: Simulation Topology

B. Simulation Results

1) Back-off Manipulation Attack: In this simulation, we consider the vulnerability of the 802.11 MAC towards back-off attack. We consider 3 parameters of Distributed Coordinated Function (DCF) to establish the attack: DIFS, SIFS, Slot time. The malicious node manipulates its back-off time to a shorter value so that it monopolizes the channel by capturing the channel for all the time. A UDP echo client application is installed on all the source nodes. The sink node is in the center of source nodes, and source nodes are around it. The simulation time is 100s. All the 7 normal source nodes send packets to sink node from the start of the simulation with a packet interval ratio of 0.1 seconds. To make the network more congested with the traffic, the attacker node send packets to sink node with a very high packet interval of 0.001 seconds. The attacker starts transmitting its packets from simulation time 20 seconds.

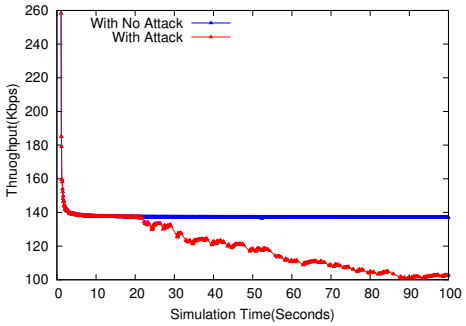

Fig. 3: Throughput versus Simulation Time: Back-off Attack

We investigated 3 performance metrics in this simulation: Throughput, Delay and Jitter. In Fig. 3, when there is an attack, the throughput of the a normal node is started decreasing when the attack started at 20 seconds simulation time. The curve does not have data points of throughput at some simulation time because of packet loss due to high packet interval ratio from the attacker. When there is no attack, the throughput curve is above the curve with attack. The delay statistics are taken for different values of back-off time slots (τ). In Fig. 4, as expected, the delay for most of the packets is very high in the case of attack compared to

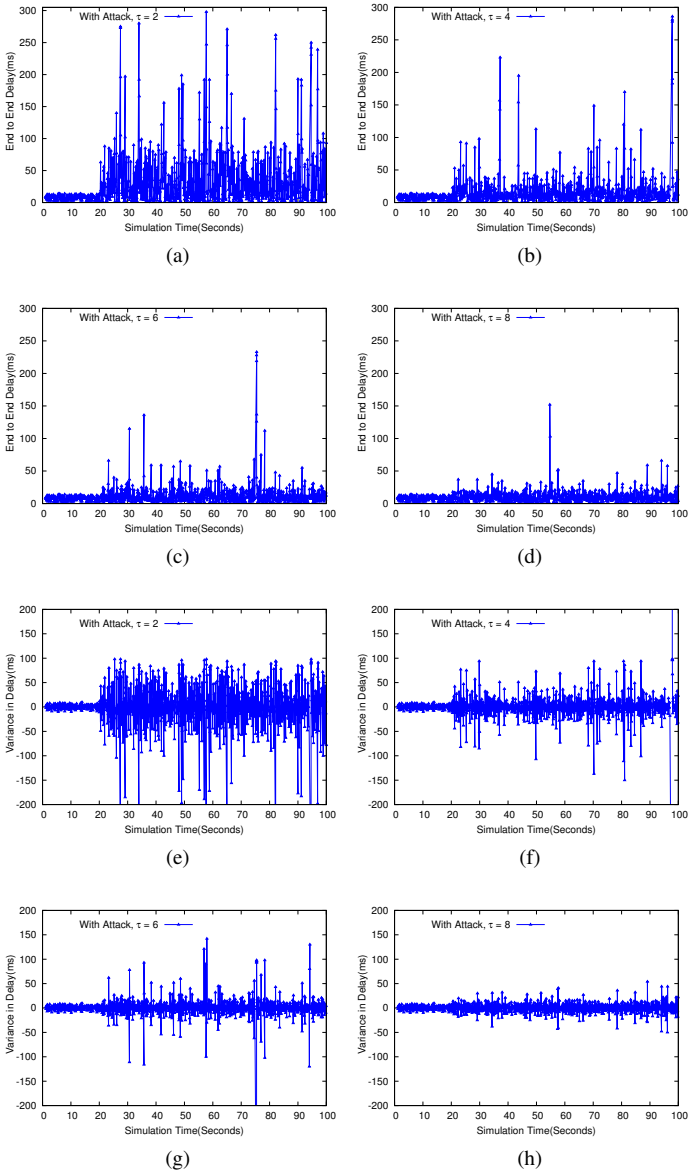

Fig. 4: Delay and Jitter with Simulation Time: Back-off Attack
(a)-(e):These have the highest effective attack with $\tau = 2$; (b,c)-(f,g): These plots have $\tau = 4$ and 6 respectively; (d)-(h): These plots have least effective attack with $\tau = 8$ with almost no difference with no attack. In all scenarios, the distribution of delay and jitter changes abruptly at 20 seconds simulation time and varies based on the value of τ

when there is no attack. The delay is very proportional to τ. In Fig. 4(a) with τ value = 2, the delay of the packets very high with a mean delay above 100ms and highest delay of 250ms, with a mean delay less than 50ms when there is no attack as shown in Fig. 4(e). The least effect of this attack is observed when the $\tau = 8$ with the mean delay almost equal to when there is no attack, in Fig. 4(d). To illustrates the effect the τ, we simulated the attack with τ value 4 and 6 in Fig. 4(b) and (c) respectively.

To observe the variance in delay distribution, we calculated the jitter induced for each packet delivery using different sce-

narios, as shown in Fig. 4. In Fig. 4(e), when there is an attack with $\tau = 2$, it shows highest effect of the attack compared to all other attack scenarios. It shows that, a maximum of 150ms faster than an earlier packet at 36 and 64 simulation time and a maximum of 100ms slower than an earlier packet at many simulation time points, with an average jitter of 50ms. However, when there is no attack, it shows a very minimum of average jitter bounded by 10ms approximately. Although, it has some peak jitter values at few points, it is induced because of higher data rates from other normal nodes. Similar effect is observed on the least effect side with $\tau = 8$ attack. In Fig. 4(h), both the curves experiencing almost similar jitter, due to the mean τ selected by all the normal nodes is around 8 in such scenario. To explicate more details of the effect, two more experiments with $\tau = 4$ and $\tau = 6$ attacks is shown in Fig. 4(f) and (g) respectively. As expected, the jitter distribution is getting saturated to same as in the case of no attack with increasing in τ.

2) RTS flooding Attack: In this simulation, we consider RTS flooding attack as explained in Section I. The simulation setup is same as IV-B1 except the data rates of attacker node. The attacker disseminates continuous RTS packets to legitimate nodes and capture the channel for all the time. Here, although the attacker sends too many packets, it still has DCF parameters such as DIFS, SIFS and back-off time same as normal nodes. This varies based on the frequency (θ) of RTS packets sent by attacker. The throughput and channel utilization of the legitimate nodes is inversely proportional to θ. We measure delay and throughput performance by varying the θ distribution. Fig. 5 shows the end to end delay for a normal flow for four different scenarios of θ. In Fig. 5(a), when there is an attack with $\theta = 10000$, it shows the delay for most of the packets is very high compared to other cases. The normal node experience a maximum of 70ms which is very low when compared to back-off attack. This is due to the nodes execute the back-off procedure with same parameters. Hence, we observe less effect even though attacker has higher θ. In Fig. 5(b) and (c), the delay distribution is further decreased to an average of 20ms and 15ms due to θ values are 5000 and 1000 respectively. A very minimal effect is observed when $\theta = 500$ in Fig. 5(d), as the channel contention is minimized. In Fig. 6, as expected, the throughput of the curve in the case of attack ($\theta = 1000$) falls below the curve without attack. At simulation time 20s, the attack started and hence change point is started at same time. When there is no attack as shown in Fig. 5(e), the throughput is saturated to a value of 821kpbs approximately and continued the same until the end of simulation. However, in the case of attack, the throughput further decreased after the attack started. The false alarms can reduced if we use mean of the delay distribution instead of delay for each packet. Fig. 5(e-h) shows the mean delay with respect to simulation time. As expected, the change point is observed at time 20 seconds.

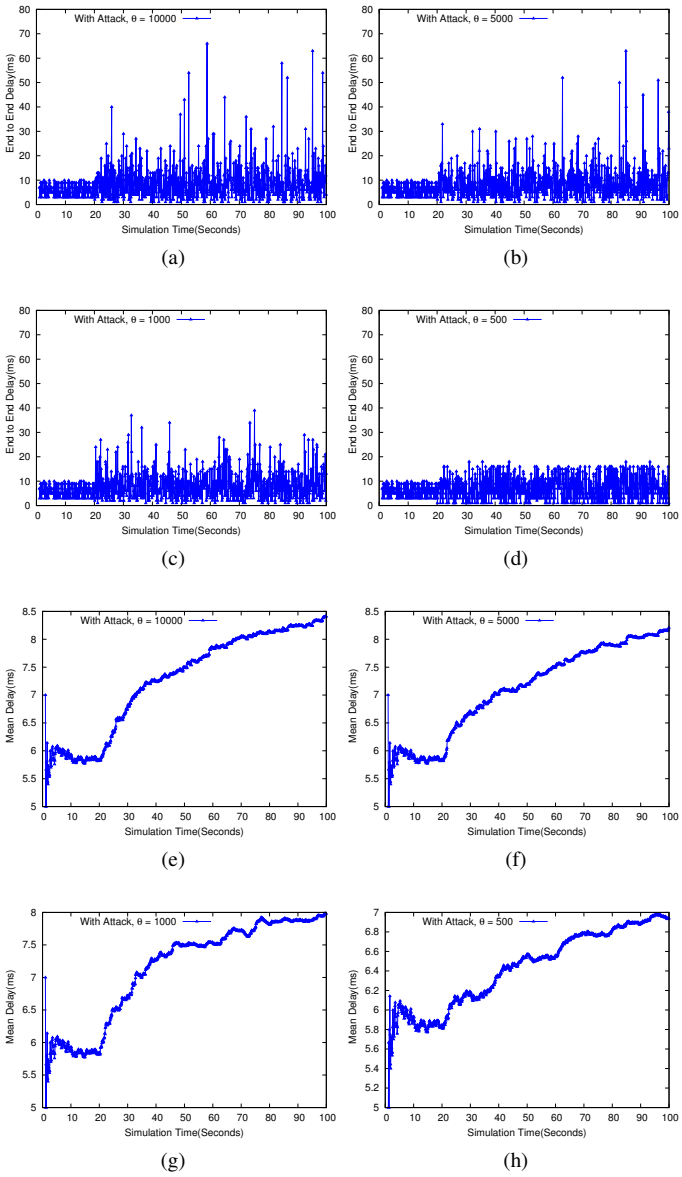

Fig. 5: Delay with respect to Simulation time: RTS Flooding
(a-e):These plots have highest effective attack with $\theta = 10000$; (b,c)-(f,g): These plots have $\theta = 5000$ and 1000 respectively; (d-h):These plots have least effective attack with $\theta = 500$ with almost no difference with no attack. Similar to Back-off attack, the change point in the delay distribution can be observed at 20 seconds simulation time that varies with respect to θ

C. Detection Accuracy

The attack detection results are given in Table I. The metrics are considered for 2 different flows f_1 and f_2, where f_1 has a sudden change in the distribution and f_2 has monotonous change. In CUSUM, the threshold need to selected manually which is not the case with our approach. This has greater impact in detecting the attach. Whereas, in our approach, we need to select the window size, which is not critical in the detection accuracy. In Table 1, for flow f_2, as expected, the threshold is not suitable for identifying the attack, hence, CUSUM is not able detect the attack.

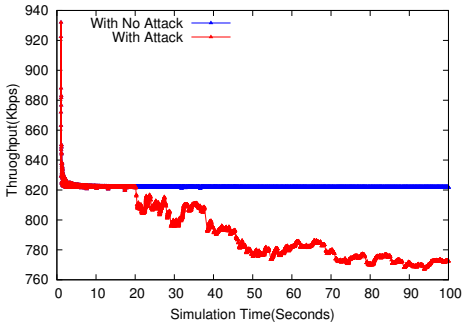

Fig. 6: Throughput versus Simulation Time: RTS Flooding

V. Conclusion

In this work, a real time detection of MAC layer attacks in IEEE 802.11 wireless networks is studied. We used two types of attacks: back-off manipulation and RTS flooding. We have come up with new change point detection algorithm which does not require fixed thresholds to compared the change of distribution unlike the cumulative sum algorithms. To observe the change point, we used delay and throughput results over time series and applied the change point detection algorithm which analyises the data distribution in a retrospective manner. From the simulation results, it can be concluded that the back-off attack has greater effect in attacking compared to RTS flooding, as malicious node still has wait for back-off time in later case. The attack can be made more effective by applying all kinds attacks together. In general, the wireless networks can be protected from these attacks in two phases: 1) Attack detection and 2) Applying a defensive mechanism. Here, we used a fast attack detection method using data analytics in real time. In future work, an appropriate defensive mechanism can be applied to detect the malicious node using localization techniques and stopping the node from capturing the network. Further, this method can be extended to MAC protocols in mobile adhoc networks such as in [23] [24].

References

[1] "Ieee draft standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE P802.11-REVmc/D3.0, June 2014 (Revision of IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, IEEE Std 802.11ad-2012, IEEE Std 802.11ac-2013, and IEEE Std 802.11af-2013)*, pp. 1–3701, Feb 2015.

[2] M. Agarwal, S. Biswas, and S. Nandi, "Detection of de-authentication dos attacks in wi-fi networks: A machine learning approach," in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 246–251.

TABLE I: Comparison of Detection Performance

Algorithms	Flow ID	Detected	False Alarms	Latency(s)
Our Approach	f1	Yes	4	0.24
Our Approach	f2	Yes	0	1.21
CUSUM	f1	Yes	>30	-
CUSUM	f2	NO	-	-

[3] R. Negi and A. Rajeswaran, "Dos analysis of reservation based mac protocols," in *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, vol. 5. IEEE, 2005, pp. 3632–3636.

[4] S. Bose and A. Kannan, "Detecting denial of service attacks using cross layer based intrusion detection system in wireless ad hoc networks," in *Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on*. IEEE, 2008, pp. 182–188.

[5] A. L. Toledo and X. Wang, "Robust detection of mac layer denial-of-service attacks in csma/ca wireless networks," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 347–358, 2008.

[6] J. Soryal and T. Saadawi, "Ieee 802.11 denial of service attack detection in manet," in *Wireless Telecommunications Symposium (WTS), 2012*. IEEE, 2012, pp. 1–8.

[7] ——, "Byzantine attack isolation in ieee 802.11 wireless ad-hoc networks," in *Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on*. IEEE, 2012, pp. 1–5.

[8] S. Radosavac, A. A. Cárdenas, J. S. Baras, and G. V. Moustakides, "Detecting ieee 802.11 mac layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers," *Journal of Computer Security*, vol. 15, no. 1, pp. 103–128, 2007.

[9] C. Alocious, H. Xiao, and B. Christianson, "Analysis of dos attacks at mac layer in mobile adhoc networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*. IEEE, 2015, pp. 811–816.

[10] Y. Zhou, D. Wu, and S. M. Nettles, "Analyzing and preventing mac-layer denial of service attacks for stock 802.11 systems," in *Workshop on Broadband Wireless Services and Applications (BROADNETS)*, 2004.

[11] W. Wang, Y. Sun, H. Li, and Z. Han, "Cross-layer attack and defense in cognitive radio networks," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[12] Y. M. Amin and A. T. Abdel-Hamid, "Classification and analysis of ieee 802.15. 4 mac layer attacks," in *Innovations in Information Technology (IIT), 2015 11th International Conference on*. IEEE, 2015, pp. 74–79.

[13] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 2, pp. 245–257, 2011.

[14] P. Ding, "Central manager: A solution to avoid denial of service attacks for wireless lans." *IJ Network Security*, vol. 4, no. 1, pp. 35–44, 2007.

[15] M. Li, S. Salinas, P. Li, J. Sun, and X. Huang, "Mac-layer selfish misbehavior in ieee 802.11 ad hoc networks: Detection and defense," *Mobile Computing, IEEE Transactions on*, vol. 14, no. 6, pp. 1203–1217, 2015.

[16] A. Aaroud, M.-A. El Houssaini, A. El Hore, and J. Ben-Othman, "Real-time detection of mac layer misbehavior in mobile ad hoc networks," *Applied Computing and Informatics*, 2015.

[17] S. K. Thambi and N. Sakthivel, "Real-time mac-layer selfish misbehavior detection and prevention technique for wireless networks," *Indian Journal of Science and Technology*, vol. 8, no. 21, 2015.

[18] P. Nagarjun, V. A. Kumar, C. A. Kumar, and A. Ravi, "Simulation and analysis of rts/cts dos attack variants in 802.11 networks," in *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on*. IEEE, 2013, pp. 258–263.

[19] N. Lyamin, A. V. Vinel, M. Jonsson, and J. Loo, "Real-time detection of denial-of-service attacks in ieee 802.11 p vehicular networks." *IEEE Communications letters*, vol. 18, no. 1, pp. 110–113, 2014.

[20] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *Ann. Statist.*, vol. 14, no. 4, pp. 1379–1387, 12 1986. [Online]. Available: http://dx.doi.org/10.1214/aos/1176350164

[21] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954. [Online]. Available: http://www.jstor.org/stable/2333009

[22] L. Gordon and M. Pollak, "A robust surveillance scheme for stochastically ordered alternatives," *Ann. Statist.*, vol. 23, no. 4, pp. 1350–1375, 08 1995. [Online]. Available: http://dx.doi.org/10.1214/aos/1176324712

[23] H. Sindhwal, M. Dasari, and N. Vattikuti, "Slot conflict resolution in tdma based mobile ad hoc networks," in *2015 Annual IEEE India Conference (INDICON)*, Dec 2015, pp. 1–6.

[24] N. Vattikuti, H. Sindhwal, M. Dasari, and B. R. Tamma, "Delay sensitive tdma slot assignment in ad hoc wireless networks," in *Communications (NCC), 2015 Twenty First National Conference on*, Feb 2015, pp. 1–5.