Paper review : Minimum Contention Window Estimation and Collision Identification in Wi-Fi Systems

Hari Prasad Sankar
dept. of ECE
UC San Diego
PID : A59003560

I. Introduction

Wireless networks have become ubiquitous in this age of information. Every person has atleast 2 devices that is transmitting data to the cloud server at any point of time leading to generation of petabytes of data. However,when such wireless networks scale up in size, they become more susceptible to security and privacy issues. Identifying such malicious users even in a small wireless network environment is a very big challenge as there are multiple parameters that needs to be estimated/tracked which again is a highly data intensive process and cannot be done very easily in real-time. In this context, wi-fi networks are susceptible to various kinds of attacks which allows a malicious user to gain access to the whole network resources and prevent other genuine users from transmitting and receiving data. Such attacks in general are called as Denial of Service (DoS) attacks. These DoS attacks on wireless networks can be orchestrated by different ways taking use of the wireless network Physical (PHY) layer and Medium access control (MAC) layer parameters. The paper that is reviewed in this project focuses on identifying malicious users who take control of wireless network resources by modifying the MAC layer parameter called minimum contention window (CW_{min}). It is done by monitoring the contention window length (CW) values of all stations connected to that particular access point and create the PMF of various CW length and comparing with the PMF of Wi-fi standard CW length PMF. Depending on how close or far away a particular station's CW PMF is, we decide if that device is a malicious user or not. We try to compare this technique with other standard methods to find and prevent DoS attacks in a wi-fi network on a theoritical level. Also, in the end, we also suggest some improvements and cons that are present in the method. The paper reviewed in this project is here

II. Problem Setup

The section talks about the basics of wi-fi transmission very briefly followed by the ways to orchestrate a DoS attack and then on (CW_{min}) based DoS attack in more detail.

Identify applicable funding agency here. If none, delete this.

A. Primer on wi-fi transmission

An Wi-fi network consists of an Access point (AP) and multiple stations (STA) connected to it. The AP serves the STA with the wi-fi service. In case of a single AP and a single STA1, the wireless channel is exclusively available for the STA1 to use. However, when multiple users are attached to the same AP, then the same wireless channel resource has to be shared among many users. This is done based on a protocol called Distributed Co-ordination Function (DCF). The DCF uses Carrier Sensing and Multiple Access/ Collision Avoidance (CSMA/CA) mechanism to prevent the packet loss due to collision. Here collision is defined when two STAs transmit at the same time resulting in packet collision and packet loss. To prevent collisions, the STA senses the channel for a duration called DCF inter frame space (DIFS) and transmits only if the channel is idle for that duration. If the channel is not idle, the STA postpones its transmission and waits for a randomly chosen backoff time. The backoff time chosen is from the set of [0,1,....,CW-1].This is the essence of CSMA/CA. The flowchart is given in figure 1.

B. DoS Attacks on wi-fi networks

Keeping in mind how the wi-fi transmission takes place using CSMA/CA, there are ways to exploit the parameters and make the other users not being able to access the channel. Some of the common ways to orchestrate such an attack are :

- Send high powered signal such that the other signal from other users aren't received
- Modify the Protocol parameter in favor of the malicious user gaining access to the channel. Example parameters include, DIFS time, increase the transmission time even when the STA doesn't have data to transmit, modify the length of CW_{min}
- Transmit dummy packets so that the malicious user doesn't allow other users to transmit or keep causing collision.

C. DoS attack by reducing CW_{min}

Consider the scenario where there are 3 users STA1,STA2 and STA3 in which STA3 has a abnormally low minimum contention window length CW'_{min} than the other two stations who have standard CW_{min}. When the user has a abnormally

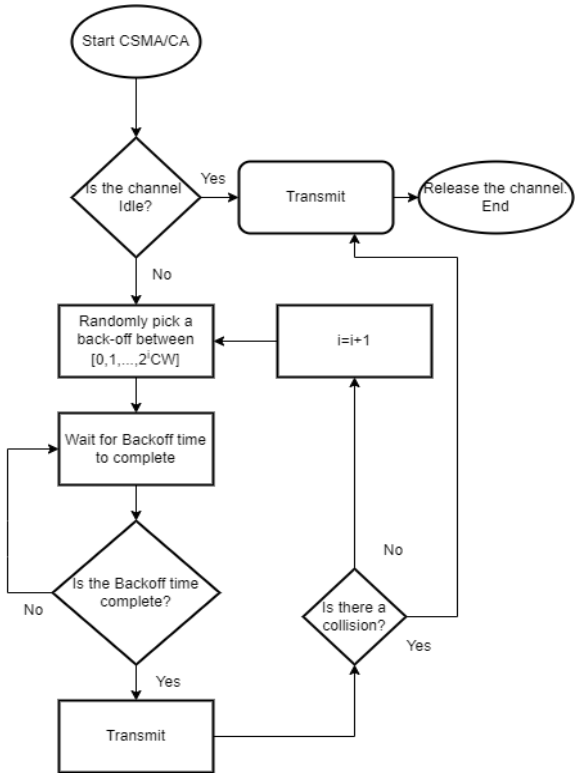

Fig. 1. Flowchart of CSMA/CA

low CW'_{min}, the backoff duration that is picked will be automatically be in the range [0,1,...CW'_{min}]. This means there are high chances of the user to wait less than the other 2 users which may result in throughput loss to the STA1 and STA2 while STA3 has more access to the channel.

The paper discussed this and provided a simulation result that shows the throughput dip for STA1 and STA2 while STA3 has a good throughput. This is shown in fig. 2.

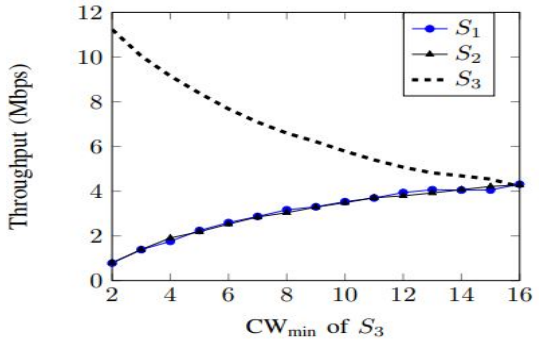

Fig. 2. Throughput comparison between genuine and malicious user in DoS attack. [1]

III. DoS attack Prevention by CW_{min} Estimation

In this section, we will discuss the method proposed by the paper that we are reviewing and outline the steps in the process. Let us consider a scenario where there are N stations where 1 is an aggressor and other N-1 are good stations. In order to identify the CW_{min} of each station and the filter out the outliers, the paper takes a 2 phase approach. These two phases are outlined as follows:

A. Monitor Phase

In the monitor phase, the AP monitors all the STA in the network and collects the data of back-off values of each station that each station randomly selected to transmit. The AP while monitoring does this by listening to channel and find out which devices are idle and note the back-off values. But the AP has to make sure that the STA is idle due to waiting for its turn to transmit and not because it does not have anything to transmit in first place. The AP can make sure of this by tracking only STA which have a non-empty $queueSize$ field from the previous transmission.
This way the AP can collect the back-off value of all the STA's that are waiting for transmission turn.

B. Need for Collision Identification Technique

Since we do not intend to make any changes to the standard protocol of wi-Fi, we need to identify the collision occurrences so that we can suitably change the backoff value collection. This is because when we have a collision, the backoff value of the STAs involved in the collision is changed and those STAs pick a backoff value in the range [0,1, ... $2^k * CW_{min}$], where 'k' is the number of occurrences a particular STA has collided with other N-1 stations. Also, the wi-fi standard has a maximum allowable re-transmission range M after which the packet is lost and the STA goes idle, setting the backoff value to the original CW_{min}. The AP has to note this so that it can suitably find if the STA is trying to transmit or if it has dropped and moved to idle to correctly identify its backoff value. As a part of the review, we have decided to focus less on this aspect as it has less probabilistic and statistical aspects involved.

C. Identifying the Malicious station

Once the AP has monitored all the STAs for a required monitor time, it constructs a PMF of backoff value chosen by each STA. Let us assume that the PMF of each station is H_j where j takes value from $[1, 2.....N]$ indicating N stations. By using Markov analysis, the PMF of backoff values of a station with $CW_{min} = l$ in a wi-fi network with N-1 good station and 1 malicious station can be computed. Let this PMF be denoted as $P^N = [P_2^N, P_3^N, ...P_{Ws}^N]$, where W_s is the standard CW_{min} and the malicious STA takes values lesser than that.

To compare the divergence of the nominal PMF with the PMF of the STA, the jensen shannon divergence measure is used. This a statistical measure that indicates the similarity between two PMFs. The Divergence measure is given by :

$$JSD(p,q) = \frac{1}{2}\left(KL(p,m) + KL(q,m)\right) \qquad (1)$$

where $KL(p,q)$ is the Kullback-Leibler divergence between p and q, and m is the average of the two distributions, defined as:

$$m(x) = \frac{p(x) + q(x)}{2}$$

The overall flow of estimation of contention window is given in fig.3

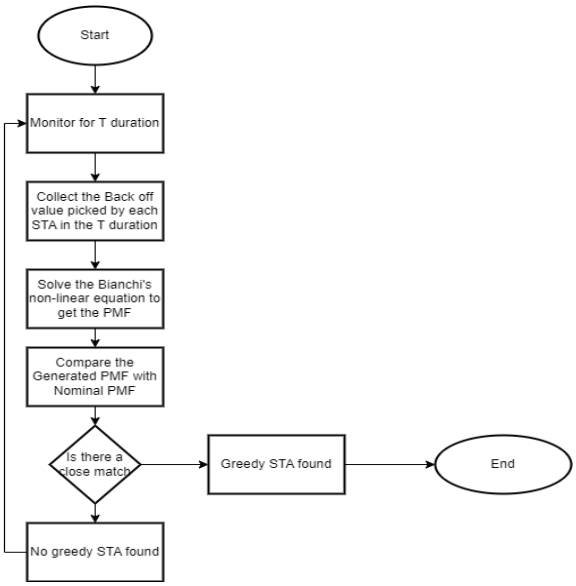

Fig. 3. Flowchart for CW min Estimation. [1]

D. Mathematical analysis on Identifying H_j and P^N

1) Construction of H_j: Let us assume that the AP has collected the backoff values selected by an STA in the buffer,

$$K_j = [k_j(1), k_j(2), ..., k_j(L_j)] \tag{2}$$

where L_j denotes the different numbers of random backoff values selected by that particular AP.

From this, the PMF can be constructed by using the indicator method,

$$H_j(n) = \frac{\sum_{i=1}^{L_j} I\left(K_j(i) == n\right)}{|K_j|}, \quad n \in \left\{0, 1, \ldots, 2^M W_s - 1\right\} \tag{3}$$

where M is the maximum allowable re-transmission. Therefore, we can identify that the PMF of CW_{min} actually depends on the probability of collision.

2) P^N vector computation: If a STA does not suffer from collision, it will randomly choose a backoff value from the Uniform distribution

$$d_0 = U_{[0, CW_{min}-1]} \tag{4}$$

But if the same STA suffers a collision, it will randomly choose a backoff value from the uniform distribution,

$$d_m = U_{[0, 2^m * CW_{min}-1]} \tag{5}$$

Therefore, the PMF for a particular contention window 'W', P_W^N is nothing but the composition of all such 'm' Uniform distribution.

$$P_W^N = d_0 * d_1 * d_2 * ....d_m \tag{6}$$

In other words, P_W^N is the weighted superposition of the collision probability.Therefore, consider X, an random variable that denotes the collision, then

$$P_W^{(N)} = \sum_{i=0}^{M} \Pr[X = i] \times d_i \tag{7}$$

3) Calculating the Collision Probability : Markov Analysis: The Markov state diagram for CSMA/CA is as shown below:

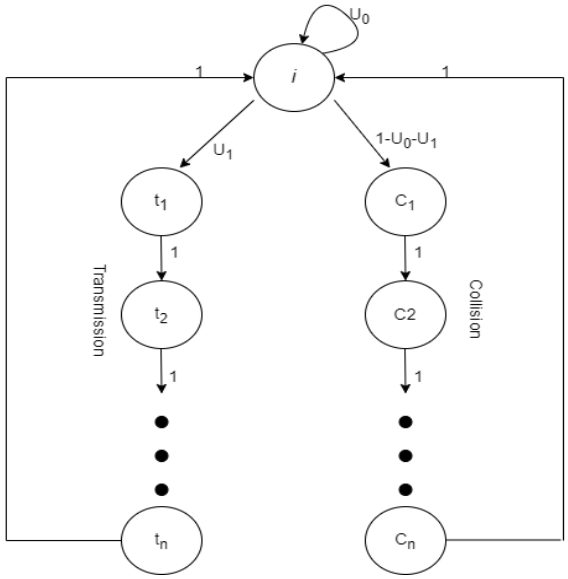

Fig. 4. General Markov chain for CSMA/CA. U_0 and U_1 represent the probability that an STA stays idle and transmits.

The collision probability can be derived using the following formula:

$$P(\text{collision}) = 1 - P(\text{success}) \tag{8}$$

where $P(\text{success})$ is the probability that a transmission will be successful.

In general, the probability of success is a function of the number of devices in the network, the length of the backoff time, and the transmission rate.

For example, suppose there are N devices in the network, each with a backoff time of T seconds. The probability of success is given by:

$$P(\text{success}) = 1 - (1 - \frac{T}{N})^N \tag{9}$$

Thus, the collision probability is given by:

$$P(\text{collision}) = 1 - (1 - \frac{T}{N})^N \tag{10}$$

To refine this further, as mentioned in the paper, we can use Bianchi's 2-D markov model for successful transmission for a

particular contention window length. The below figure shows the Markov model.

The probability of transmission in any random slot given by Bianchi's model [4] is,

$$\tau = \frac{2(1-2p)(1-p)}{(1-2p)\left(W_{\min}+1\right)+pW_{\min}\left(1-(2p)^{m}\right)} \tag{11}$$

But the above probability works in an all good scenario only. It does not suite when one has an aggressor who has contention window range less then others. In such a case, the probability of successful transmission in any slot is [5] :

$$\begin{cases} \tau^0 = \dfrac{2\left(1-2p^0\right)}{\left(1-2p^0\right)\left(W+1\right)+p^0W\left(1-\left(2p^0\right)^{m}\right)} \\ \tau^1 = \dfrac{2\left(1-2p^1\right)}{\left(1-2p^1\right)\left(g^1W+1\right)+p^1g^1W\left(1-\left(2p^1\right)^{m}\right)} \\ \cdots \\ \tau^l = \dfrac{2\left(1-2p^l\right)}{\left(1-2p^l\right)\left(g^lW+1\right)+p^lg^lW\left(1-\left(2p^l\right)^{m}\right)} \\ p^0 = 1-\left(1-\tau^0\right)^{n-l-1}\prod\limits_{1\le i\le l}\left(1-\tau^i\right) \\ p^1 = 1-\left(1-\tau^0\right)^{n-l}\prod\limits_{2\le i\le l}\left(1-\tau^i\right) \\ \cdots \\ p^l = 1-\left(1-\tau^0\right)^{n-l}\prod\limits_{1\le i\le l-1}\left(1-\tau^i\right) \end{cases} \tag{12}$$

Where τ^i denotes the probability of successful transmission in any slot by the i^{th} greedy user and p^i probability of collision.τ^0 and p^0 represent normal stations. This is a generic rule for 'l' stations not following standard probability. In the case we are considering, $l = 1$. From the value of p, the $Pr(X = i)$ can be calculated as it is nothing but the STA being in backoff stage i and p which is the probability of collision is direct translation of the STA being in backoff stage i.

IV. Comparison with other solutions to the problem

In this section, we compare the paper in discussion [1] with 2 other papers [2] [3] that try to solve the same problem but with a different approach and identify why the current paper in discussion is better than the other solutions.

A. Intelligent CW : AI-based Framework

The paper Intelligent CW tries to solve the same problem but not by following a probabilistic approach but an AI approach. The main difference between the paper in consideration and intelligent CW paper is the following :

- The first main difference is the fact that intelligent CW tries to solve the problem from the STA point of view and not from AP point of view. This raises some concern. This is because, usually the STA's are normal IoT devices or mobile phones and these devices Run lean on power. As a result, running big ML models in them seems a little infeasible.
- Moreover, The Gini loss function used in the model,

$$G = 1 - \sum_{i=1}^{N} p_i^2$$

where N is the number of classes and p_i is the probability of the i^{th} class. However, the loss function is not sensitive to outliers. The problem that we are trying to solve is exactly the same, which has an outlier.
- Most importantly, the intelligent CW paper assumes that all nodes operating in an network knows all its neighbors. This is the foundation of the whole solution which is not the case in majority of the real life scenarios. Most of the wi-fi networks suffer from Hidden terminal problems or STA deafness.
  The hidden terminal situation is when a STA is only visible to the AP to which it is connected to and not to the other STAs to which it is connected. The following figure explains represents this :

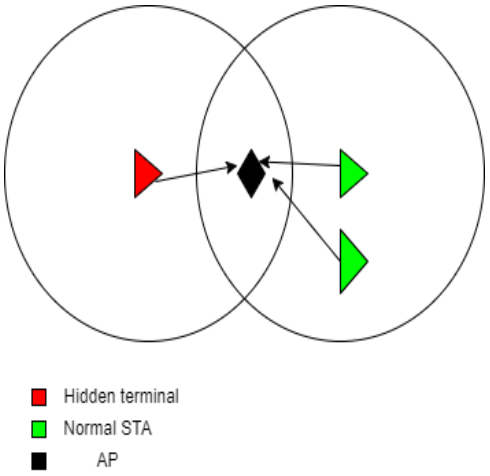

Fig. 5. Throughput comparison between genuine and malicious user in DoS attack

When compared to the markovian model that we are reviewing, the intelligent CW fails to address this aspect. The pros of this model is that it can it acts on dynamically changing the network fair by nature unlike the method under consideration which requires to a lot of compute time.

B. DOMINO : Detecting MAC Layer Greedy Behavior

The Domino mechanism introduced in the paper [**?**] tries to combat all kind of MAC level greedy behavior on test basis. The test for identifying the greedy device is:

$$\text{condition} \ : B_{ac}\left[M_i\right] < \alpha_{ac} \times B_{\text{acnom}} \tag{13}$$

The value $B_{ac}[M_i]$ is the average backoff value observed by Ap of STA M_j, the value α_{ac} is a tuneable parameter to tune TPR vs FPR and B_{acnom} is the expected value of Back-off. This is calculated by observing the network. If the AP doesn't

have enough samples, then it can use the $E[B_{ac}]$ which can be calculated by,

$$E\left[B_{ac}\right] = \sum_{i=0}^{W-1} ip(i) \tag{14}$$

where W is the maximum contention window length.

Now, $p(i)$ is the probability that i is the actual backoff length. This is the ratio of probability of all backoff values and the probability that i is the backoff value.

$$p(i) = \frac{p_{ac}(i)}{\sum_{i=0}^{W-1} p_{ac}(i)} \tag{15}$$

Using this,

$$E\left[B_{ac}\right] = \frac{(W-1)q_{ac}^{W+1} - Wq_{ac}^{W} + q_{ac}}{\left(1-q_{ac}^{W}\right)\left(1-q_{ac}\right)}. \tag{16}$$

where,

$$p_{ac}(i) = q_{ac}^{i}(1-\tau)^{n-1} \tag{17}$$

and τ is used from equation (12)

However, while calculating these backoffs the paper just takes into account scenarios when there are no collisions. As a result, it doesn't take into consideration the exponential back off. Instead it only solves when the contention window length is between $[0, 1, ...CW_{min} - 1]$This is because it is really diffcult to identify the STAs involved in collision. An important feature of the paper in review is the fact that it has solved that issue also through the CIT algorithm. Thus, it is able to handle both the hidden terminal case and the case of collision.

V. Drawbacks in the Approach

Even though the paper [1], there are several drawbacks in it. They are listed as below :

- The markovian approach that is taken is too rigid. As a result, a similar approach cannot be used to solve other MAC level greedy behavior. The Domino method introduced in [3] took a broad stroke approach to solve the issue while [1] took a really narrow approach.
- The model can become too computationally complex even for AP when the WLAN system scales. As then the AP has to keep collecting huge data set of back off values and solve multiple non-linear equations. For example, for l greedy devices, $l+1$ non-linear equations need to be solved.
- WLAN networks are generally very dynamic and mercurial in nature. The PMF that is computed at a time $'T'$, if when used at $'T+1'$,can be an unfit measure as the participating STAs nature could have changed.

VI. Suggested Improvements

While [1] has solved almost majority of the problems that exist in CW_{min} estimation, we have found some scope of improvements also. They are as follows :

A. Divergence Measure

[1] uses the Jensen-shannon Divergence (JSD) measure as a metric to choose if the two PMF's are close to one another. But while Jensen-Shannon divergence measure is a reasonably good metric, it can indicate that two distributions are identical/similar even if they are not truly and vice vera.

We found that 2 other similarity/divergence measure can be more useful for the application in hand.

- Bhattacharya Divergence : The metric is given by :

$$D_B(P,Q) = \sum_i \sqrt{p(i)q(i)}, \tag{18}$$

  $p(i)$ and $q(i)$ are the probabilities of event i in the distributions P and Q, respectively
- Kolmogorov Smirnov test

$$D_n = \max_{1\le i\le n} |F_n(x_i) - F(x_i) \tag{19}$$

  Here, $F_n(x_i)$ is the empirical CDF of the sample, and $F(x_i)$ is the CDF of the reference distribution.

The limitation in JSD is that it is based on a non symmetric KL measure. Therefore, it is not a truly symmetric metric, whereas Bhattacharya divergence is perfectly symmetric metric. Also, Since JSD is computed based on how dissimilar two distributions look, it is very sensitive to minor perturbations in PMF of contention window. But Bhattacharya divergence is based on how similar two PMFs look. To our application, since we are concerned about how similar our nominal PMFs are compared to the calculated PMFs. Thus we believe using Bhattacharya divergence can be more suitable.

B. Extension to Identify Packet Sniffers

The same idea can ideally be extended to identify Rf sniffers. RF sniffers are devices who dont really transmit very huge data or contend for the channel like other STAs do but they listen to the communication links. This is a serious privacy issue and the same method with a small tweak can be made to identify them. Instead of looking to identify the most matching PMFs, we need to look at the PMF of STAs which have least contested to the channel. Once that is done, we can nearly identify RF sniffers. It can be represented as,

$$\arg\min \left(\sum_{i=0}^{CW-1} C_j(i) \right) \tag{20}$$

where $C_j(i)$ is the number of times backoff value i has been chosen by the j^{th} STA.

VII. Conclusion

In this paper review project, we took a problem that is most pressing in the WLAN network i.e the greedy behavior of certain STAs and the associated unfairness with it. We decided to analyze the root cause of the problem and understand the mathematics by taking one of the latest algorithms developed for it. We also analyzed and understood the probabilistic and statistical concepts that are involved in the algorithm. We compared the results and scenarios where the model performs

good with other standard algorithms. We also analyzed the scenarios where the algorithm in review performed better than the standard and other updated solutions. The CW_{min} algorithm is able to estimate the minimum contention window for about 98% accuracy with +/-2% tolerance level. We also suggested some improvements in the statistical metrics used along with an idea of extending the same model to other scenarios.

VIII. Acknowledgement

I wish to thank the authors of [1], [2], [3]. I have used them analyze the problem in hand and section III and IV are written with [1], [2], [3] as primary references.

IX. Code and Simulation Environment

The code emulating CSMA/CA and the estimation can be found here : ECE 225A Project Code The files are in Matlab .m format and one needs to run the cwmin_main.m

References

[1] Abyaneh, Amirhossein Yazdani and Marwan Krunz. "CWmin Estimation and Collision Identification in Wi-Fi Systems." 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS) (2021): 490-498.

[2] A. H. Y. Abyaneh, M. Hirzallah and M. Krunz, "Intelligent-CW: AI-based Framework for Controlling Contention Window in WLANs," 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), 2019, pp. 1-10, doi: 10.1109/DySPAN.2019.8935851.

[3] M. Raya, I. Aad, J. . -P. Hubaux and A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots," in IEEE Transactions on Mobile Computing, vol. 5, no. 12, pp. 1691-1705, Dec. 2006, doi: 10.1109/TMC.2006.183.

[4] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," in IEEE Journal on Selected Areas in Communications, vol. 18, no. 3, pp. 535-547, March 2000, doi: 10.1109/49.840210.

[5] Y. Rong, S. . -K. Lee and H. . -A. Choi, "Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis," Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, 2006, pp. 1-13, doi: 10.1109/INFOCOM.2006.305.