CW$_{\text{min}}$ Estimation and Collision Identification in Wi-Fi Systems

Amir-Hossein Yazdani-Abyaneh and Marwan Krunz
Department of Electrical and Computer Engineering, University of Arizona, AZ, USA
Email: {yazdaniabyaneh, krunz}@email.arizona.edu

***Abstract*—Wi-Fi networks are susceptible to aggressive behavior caused by selfish or malicious devices that reduce their minimum contention window size (CW$_{\text{min}}$) to below the standard CW$_{\text{min}}$. In this paper, we propose a scheme called *Minimum Contention Window Estimation* (CWE) to detect aggressive stations with low CW$_{\text{min}}$'s, where the AP estimates the CW$_{\text{min}}$ value of all stations transmitting uplink by monitoring their backoff values over a period of time and keeping track of the idle time each station spends during backoff. To correctly estimate each backoff value, we present a cross-correlation based technique that uses the frequency offset between the AP and each station to identify stations involved in uplink collisions. The AP constructs empirical distributions for the monitored backoff values and compares them with a set of nominal PMF's, created via Markov analysis of the DCF protocol to estimate CW$_{\text{min}}$ of various stations. After detecting the aggressive stations, the AP can choose to stop serving those stations. Simulation results show that the accuracy of our collision detection technique is 96%, 94%, and 88% when there are 3, 6, and 9 stations in the WLAN, respectively. For the former WLAN settings, the estimation accuracy of CWE scheme is 100%, 98.81%, and 96.3%, respectively.**

I. Introduction

Wi-Fi end-users, also known as stations, demand fair allocation of the channel airtime. The 802.11 MAC protocol [1], known as the Distributed Coordination Function (DCF), uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) with exponential backoff to provide fair channel access in a distributed manner, provided that all stations comply with the DCF protocol. Under DCF, a station that wants to transmit must first sense the channel for a fixed duration, called the DCF initial Inter-Frame Space (DIFS). If the channel is sensed to be idle during the DIFS period, the station starts its transmission; otherwise, the station defers its transmission and waits for a random backoff period. The backoff period consists of k idle slots, where k is randomly chosen from $\{0, 1, ..., \text{CW} - 1\}$. Initially, CW is set to a default value CW$_{\text{min}}$. After a collision, CW$_{\text{min}}$ is doubled until it reaches the maximum allowable contention window (CW$_{\text{max}}$). Generally, a station that has consecutively collided for j times chooses its k randomly from $\{0, 1, ..., \min(2^j \text{CW}_{\text{min}}, \text{CW}_{\text{max}}) - 1\}$. The exponential increase in CW helps stations avoid collisions. Following a successful transmission, a station resets its CW to CW$_{\text{min}}$.

In a Wi-Fi system, aggressive behavior for channel access can be attributed to malicious reasons to degrade the network's performance [2], [3], [4], [5] or it can be caused by selfish stations that try to gain more access to channel airtime [6], [7], [8], [9]. An example of malicious behavior is channel jamming attacks [2], [10], which can be considered as a particular type of Denial-of-service (DoS) attack [3], [11], [12]. In addition to transmitting a high-power signal to disrupt other users' transmissions, a malicious station can also transmit fake packets to prevent normal users from communicating [6]. In contrast, a selfish station alters its protocol parameters to get an unfair share of the channel airtime at the expense of other well-behaving stations. For example, this station may reduce the value of its SIFS or DIFS below the standard values. It may choose a larger value of the remaining transmission duration field in the MAC header to force other stations to back off for longer periods. It may also lower its CW$_{\text{min}}$ so that it captures the channel more often than other stations. Although DCF does an excellent job in ensuring fairness among devices and reducing collisions, it is still vulnerable to aggressive stations that do not abide by the standard protocol, hence harming the performance of compliant stations.

To cast more light on this issue, consider a Wi-Fi network with three backlogged stations, all following the DCF protocol with access category A_3 [1]. All stations are in each other's sensing range. Stations S_1 and S_2 select a standard CW$_{\text{min}}$ of 16. In Figure 1, we show the per-station throughput for different values of S_3's CW$_{\text{min}}$. Whenever CW$_{\text{min}}$ of S_3 is less than 16, S_1 and S_2 have lower throughputs than S_3. Our goal is to enable the AP to detect aggressive stations by estimating their CW$_{\text{min}}$ and comparing them with a standard defined value. The problem of detecting stations with low CW$_{\text{min}}$ values has been studied in the literature, as described in Section V. However, prior works propose protocol modifications [8], [13], [14], [15], assume backlogged stations [6], [16], [17], [18], [19], or they do not consider the hidden terminal problem [6], [18], [19]. In this paper, we consider the problem of *detecting* aggressive stations with low CW$_{\text{min}}$ setting, but without imposing any computational overhead on any station. Our approach is only implemented at the AP, but without altering the DCF protocol. We make no assumptions about the traffic type of any station. Further, we take into consideration the hidden terminal problem by introducing a new correlation-based technique for collision detection. Our *Minimum Contention Window Estimation* (CWE) has two phases, a monitoring phase and an estimation phase. In the monitoring phase, the AP monitors transmission activity of each station and notes down the idle durations in which stations decrease their backoff counters. The AP translates the monitored idle durations to

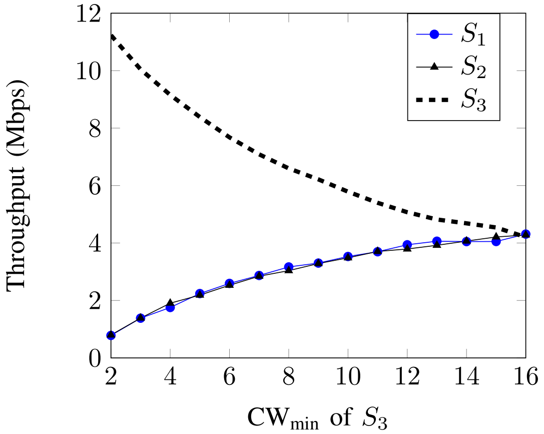


Fig. 1: Per-station throughput for a network of three stations vs. CW$_{\text{min}}$ of S_3 (CW$_{\text{min}}$ $= 16$ for S_1 and S_2).

their representing backoff values and constructs an empirical distribution (PMF) of these backoff values. The AP makes sure that the periods of time each station is idle is due to its backoff process and not caused by an empty transmission buffer. We assume that all transmitted frames include the `Queue Size` subfield of the `QoS Control` field of the MAC header [1][1]. Thus, the AP derives backoff values only for packets that their prior transmission indicated a non-empty transmission buffer, i.e. non-zero `Queue Size` value. Another important aspect of correctly estimating the backoff value is for the AP to detect collisions in the uplink and identify stations that are involved in a collision. Most prior works related to misbehavior detection do not consider the possibility of hidden terminals, and if they do, they discard any observation related to a collision. To determine the identities of colliding stations, we present a correlation-based technique that uses the frequency offset (FO) between each station and the AP. For each station-AP pair, the technique calculates the cross-correlation of a collided signal with the 802.11 preamble that is modified to include the FO effects of the station-AP pair, and looks for peaks in the cross-correlation that are higher than a given threshold to identify colliding stations.

In the second phase of CWE, the constructed PMF's are compared with a set of nominal PMF's, which are derived based on Markov Chain (MC) analysis of the CSMA/CA protocol [20] whereby all stations but one are compliant and the CW$_{\text{min}}$ of the non-compliant station is changed within a range to construct different nominal PMF's (one per CW$_{\text{min}}$ value of the non-compliant station). The observed and nominal PMF's are compared using Jensen-Shannon divergence measure. The CW$_{\text{min}}$ with a nominal PMF of least divergence measure with the observed PMF is taken as the estimated CW$_{\text{min}}$ for the station under observation. Stations with estimated CW$_{\text{min}}$ values lower than the standard value are considered as aggressors.

Simulation results with three, six, and nine stations show that our collision detection technique achieves an accuracy of 96%, 94%, and 88%, respectively. The corresponding accuracy of the CW$_{\text{min}}$-estimation algorithm is 100%, 98.81%, and 96.3%, respectively. The paper is organized as follows. In Section II, we introduce CWE along with the backoff value estimation algorithm. Our collision identification technique and evaluation results are presented in Sections III and IV, respectively. Finally, we survey related works and conclude the paper in Sections V and VII, respectively.

II. Minimum Contention Window Estimation (CWE)

Our system model includes a WLAN with N stations and an AP. We denote the jth station by S_j and the standard CW$_{\text{min}}$ by W_s. To estimate CW$_{\text{min}}$ of an arbitrary station, say S_j, the AP tracks the backoff values selected by S_j over an observation period T. The set of backoff values selected by S_j are denoted by $K_j = [K_j(1), K_j(2), ..., K_j(L_j)]$, where $K_j(i)$ is the ith backoff value selected by S_j during T and L_j is the total number of selected backoff values by S_j. In Section II-A, we explain the process of obtaining the vector K_j. For now, we assume that $K_j(i)$'s have been estimated by the AP. The AP constructs an empirical probability mass function (PMF) from the vector K_j, as:

$$H_j(n) = \frac{\sum_{i=1}^{L_j} \mathbb{1}(K_j(i) == n)}{|K_j|}, \quad n \in \{0, 1, ..., 2^M W_s - 1\} \tag{1}$$

where M is the maximum number of allowed retransmissions. Consider an arbitrary station $S \in \{S_1, ..., S_N\}$ with CW$_{\text{min}}$$=$ W. If S is compliant, then W is the standard value. Different CW$_{\text{min}}$ settings for S result in different backoff values, thus different PMF's. After each successful transmission, S samples its backoff values from a uniform distribution $U_{[0,W-1]}$. Because stations double their contention window after a collision, S will select its backoff values from $U_{[0,2W-1]}$ after any collision that follows a successful transmission. Considering the possibility of collisions, the overall PMF of the backoff values selected by S, denoted by H, depends on W and the collision probability in the WLAN.

The AP maintains a set of nominal PMF's, denoted by $\mathcal{P}^{(N)} = \{P_2^{(N)}, P_3^{(N)}, ..., P_{W_s}^{(N)}\}$, where $P_l^{(N)}$ is the PMF of selected backoff values of a station with a CW$_{\text{min}}$$=$ l in a WLAN of N stations, where all other $N-1$ stations have a CW$_{\text{min}}$$=$ W_s. To obtain W, the AP compares H with each $P_l^{(N)}$ for $l \in \{2, 3, ..., W_s\}$, the l with the nominal PMF of $P_l^{(N)}$ that has the least difference from H is considered as the estimated CW$_{\text{min}}$ for station S. If S does not collide during its transmissions, then it will always randomly select a backoff value k from the uniform distribution $d_0 = U_{[0,W-1]}$. On the other hand, if S is involved in j consecutive collisions, it will randomly select k from $d_j = U_{[0,2^jW-1]}$. Therefore, the overall PMF of backoff value selections of a station with CW$_{\text{min}}$$=$ W, i.e. $P_W^{(N)}$, is a composition of $d_0, d_1, ..., d_M$, where M is the maximum number of allowed retransmissions. To derive $P_W^{(N)}$, we define a random variable

[1] IEEE 802.11 standard requires all QoS data packets to include the `Queue Size` subfield in their MAC headers.

X that represents the backoff stage of S. X takes values from $\{0, 1, ..., M\}$. For instance, $X = j$ means that S has collided j consecutive times and will randomly select its backoff value from $\{0, 1, ..., 2^j W - 1\}$. The overall distribution of k is the weighted superposition of $d_0, d_1, ..., d_M$:

$$P_W^{(N)} = \sum_{i=0}^{M} \Pr[X = i] \times d_i. \quad (2)$$

To find $\Pr[X]$'s, we need to find the collision probabilities for when S selects a CW$_{\min}$= W. In [16], authors obtain collision and packet transmission probabilities for different CW$_{\min}$ settings each station in the WLAN. Their analysis is based on Bianchi's MC modeling of the 802.11's DCF protocol [20]. For this work we only need to obtain the former probabilities for when all stations except one are compliant (CW$_{\min}$= W_s). This simplification does not degrade the performance of CWE; instead, it further reduces the computational complexity of CWE from $O(N^{W_s - 1})$ to $O(W_s - 1)$.

Bianchi developed a bidimensional MC for a WLAN with N stations, where all stations have a CW$_{\min}$= W_s, and assumed that the collision probability is constant, denoted by p. Let $\{s(t), b(t)\}$ represent the state of the MC, where $s(t) \in \{0, 1, ..., M\}$ is a stochastic process that represents the backoff stage of a station, and $b(t)$ is a stochastic process that represents the backoff counter for the station. At a stage $s(t) = i$, $i \in \{0, 1, ..., M\}$, $b(t)$ can take values from the set $\{0, 1, ..., W_i - 1\}$, where $W_i = 2^i W_s$. The one step transition probabilities are represented as:

$$\begin{cases} \Pr[i, k|i, k+1] = 1 & k \in \{0, ..., W_i - 2\} \quad i \in \{0, ..., M\} \\ \Pr[0, k|i, 0] = \frac{1-p}{W_s} & k \in \{0, ..., W_s - 1\} \quad i \in \{0, ..., M\} \\ \Pr[i, k|i-1, 0] = \frac{p}{W_i} & k \in \{0, ..., W_s - 1\} \quad i \in \{0, ..., M\} \\ \Pr[M, k|M, 0] = \frac{p}{W_M} & k \in \{0, ..., W_M - 1\}, \end{cases} \quad (3)$$

where $\Pr[i, k|j, l] = \Pr[s(t+1) = i, b(t+1) = k|s(t) = j, b(t) = l]$. Deriving the steady state probabilities, The probability of a transmission in a randomly chosen time slot, denoted by τ is:

$$\tau = \frac{2(1-2p)}{(1-2p)(W_s+1) + pW_s(1-(2p)^M)}. \quad (4)$$

And the probability of a collision can be represented as:

$$p = 1 - (1-\tau)^{N-1}. \quad (5)$$

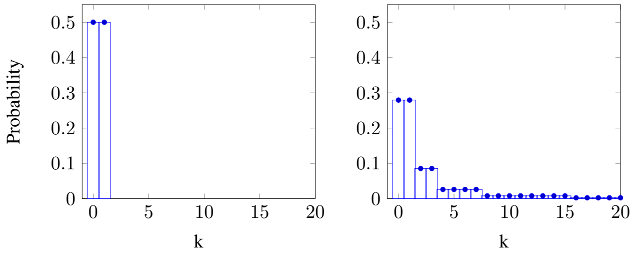
Equations 4 and 5 form two nonlinear equations with two unknowns which can be solved numerically to obtain p and τ. It is important to note that Equations 4 and 5 are only valid when all the N stations have CW$_{\min}$ = W_s, which is not the case in our system model, since an aggressor has a lower CW$_{\min}$ value than the standard value. To calculate the collision probability for S with CW$_{\min}$= W, which is needed to obtain the proper $\Pr[X]$'s in Equation 2, we assume that all $N-1$ other stations have a CW$_{\min}$= W_s. We denote the transmission and collision probabilities for S by τ and p, respectively. The transmission and collision probabilities for the compliant stations (CW$_{\min}$= W_s) are denoted by $\tau^{'}$ and $p^{'}$, respectively. Following the analysis presented in [16] the probabilities can be obtained by solving the following four nonlinear equations:

$$\begin{cases} \tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^M)} \\ p = 1 - (1-\tau^{'})^{N-1} \\ \tau^{'} = \frac{2(1-2p^{'})}{(1-2p^{'})(W_s+1)+p^{'}W_s(1-(2p^{'})^M)} \\ p^{'} = 1 - (1-\tau)(1-\tau^{'})^{N-2}. \end{cases} \quad (6)$$

In Equation 2, $\Pr[X]$'s are needed to be calculated to construct nominal PMF's ($P_W^{(N)}$'s). $\Pr[X = i]$ can be interpreted as the steady state probability of being in a backoff stage i ($\Pr[s(t) = i]$), and it can be calculated as:

$$\Pr[X = i] = \begin{cases} 1-p & i = 0 \\ (1-p)p^i & i = 1, ..., M-1 \\ p^M & i = M \end{cases} \quad (7)$$

In Figures 2(a) and 2(b), we show the PMF of backoff value selections of S when $W = 2$, $N = 10$, and $M = 7$, where S does not and does double its contention window in case of collisions, respectively. To obtain Figure 2(b), we solve Equation 6 and calculate $p = 0.612$. To compare the constructed empirical PMF (i.e., H) with each $P_l^{(N)} \in \mathcal{P}^{(N)}$ for $l \in \{2, 3, ..., W_s\}$, we use Jensen-Shannon divergence [21], which is based on Shannon's concept of uncertainty (entropy), to measure the similarity between two probability distributions. The Jensen-Shannon divergence measure between two PMF's H and P is denoted by $J(H, P)$, and it is calculated as:



(a)

(b)

Fig. 2: Backoff value distribution of S with $W = 2$, $M = 7$, and $N = 10$ where S (a) does not and (b) does double its contention window in the case of collisions.

$$J(H, P) = \frac{1}{2}\Bigg[\sum_{i=1}^{|H|} P(i) ln\left(\frac{2P(i)}{P(i)+H(i)}\right) + \sum_{i=1}^{|H|} H(i) ln\left(\frac{2H(i)}{P(i)+H(i)}\right)\Bigg], \quad (8)$$

where $P(i)$ and $H(i)$ are the ith elements of P and H, respectively, and $|.|$ is the cardinality operator. The estimated CW$_{\min}$ value for S_j, i.e. W_j, can be estimated as:

$$W_j = \underset{l \in \{2,3,\ldots,W_s\}}{\operatorname{argmin}} \; J(H_j, P_l^{(N)}). \tag{9}$$

A. Backoff Counter Estimation

To estimate backoff values selected by S for each channel access attempt, the AP monitors S's transmission activity and notes down the idle durations in which S decreased its backoff counter. Afterwards, the monitored idle durations are translated to backoff values which caused those specific idle periods. During the monitoring period, the AP needs to be accurate in sensing S's transmission, hence it needs to detect any uplink collisions and identify stations involved in them. We tackle the former by introducing *Collision Identification Technique* (CIT) that helps identify all stations involved in an uplink collision. Using CIT, the AP will be able to monitor the channel and associate each channel busy time to a subset of stations. We explain CIT in Section III. Also, the AP needs to make sure that the duration S spent in an idle states was due to decreasing its backoff counter and not caused by an empty transmission buffer. The former is always true when the WLAN is in a saturated traffic scenario, where stations are backlogged with packets to transmit. To tackle the stated challenge, the AP will use the information that QoS packets must include in their QoS Data Field of their MAC headers, namely the `Queue Size` subfield [1]. The `Queue Size` subfield indicates the number of bytes that are present in the queue of the transmitter at the time of transmission. Therefore, a none-zero `Queue Size` value will suggest that the transmitter entered backoff stage immediately after that packet transmission, hence all sensed idle durations were due to decreasing the backoff counter. Also, we know that a packet that experiences collision will be set for retransmission for at most an M number of retransmissions, this means that the `Queue Size` of packets that are inside a collision will be considered as nonzero for backoff value estimation, too. Nonetheless, there is a low possibility that a packet might fail to successfully transmit for m times and get dropped. In this case, if there are no packets left to transmit at the buffer of S, on average, our algorithm will mistakenly measure the backoff value to be more than $\frac{2^{m+1}-1}{2}W_s$. For this case, the AP will disregard that backoff value estimation.

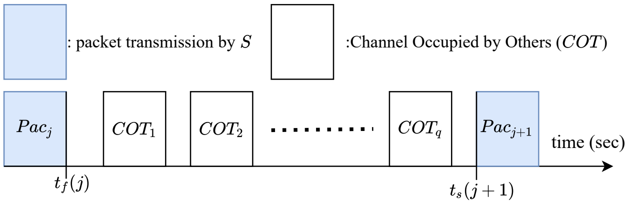


Fig. 3: An example of gathering observations for estimating the $(j+1)$th backoff value selected by S.

We explain how each backoff value is estimated during two successive packet transmissions of S. We denote the jth packet transmission by S during T by Pac_j. Also, we define $COT_{j+1}(i)$ to be the ith duration of time that the channel becomes occupied by stations other than S during the contention period for transmitting the $(j+1)$th packet of S. Figure 3 shows an example of observations seen during two packet transmissions, i.e. Pac_j and Pac_{j+1}. To find the value of $K(j+1)$ (i.e., the $(j+1)$th backoff value selected by S during T), the AP has to mark the time instant of the end of Pac_j's transmission, denoted by $t_f(j)$, and the time instant Pac_{j+1} started getting transmitted, denoted by $t_s(j+1)$. We assume that Pac_j has a non-zero value for its `Queue Size` subfield; otherwise, the AP would have disregarded the estimation of $K(j+1)$. It is important to note that either Pac_j or Pac_{j+1} may be involved in collisions; however, using CIT the AP will be able to determine the start of packet transmissions by different stations in a collision, and by assuming fixed packet sizes, the AP can estimate the end of a packet transmission in a collision, too. Therefore, our backoff estimation example holds for the case of collisions, too. The value of $K(j+1)$ can be calculated as:

$$K(j+1) = \frac{t_s(j+1) - t_f(j) - \sum_{i=1}^{q} COT_i - q \times T_{DIFS}}{T_{MAC}}, \tag{10}$$

where T_{MAC} and T_{DIFS} are the MAC time slot and the $DIFS$ period values, respectively, and q is the total number of times the channel was occupied by other stations during T. Algorithm 1 we explains how to estimate backoff values for S. The algorithm can be configured to obtain backoff value selections for all stations in the WLAN. The output of the algorithm, i.e. vector K, will be used by CWE to estimate the CW$_{\min}$ of S.

III. Collision Detection & Identification Technique (CIT)

In [22], authors propose an algorithm to decode collided packets. For their algorithm to work, the number of distinct collisions (different overlapping combinations) that are needed to be gathered is the same as the number of colliding stations. Since, our backoff estimation algorithm only needs the ID's of colliding stations, we develop *Collision Identification Technique* (CIT) in which a single collision is sufficient to identify all colliding stations without needing to decode packets. CIT uses wireless channel and hardware characteristics of each station-AP pair to identify stations involved in any collisions by following a correlation-based technique. Before going through details of CIT, we briefly present some background on digital communication and some physical phenomenons that affect signals transmitted over the wireless channel, namely, frequency offset, channel attenuation and channel phase shift.

A. Digital Communications

Packets are consisted of bits, for these bits to get transmitted over the wireless channel, they have to be modulated into complex stream of numbers. For example, the BPSK modulation scheme converts a bit of value 0 and 1 into complex symbols

Algorithm 1 Backoff Estimation for S

```
Input $T$ : Monitoring period ;
Variables: $q \leftarrow 0, idle \leftarrow 0, Q \leftarrow 1$;
Output: $K$;
while current time $< T$ do
    Monitor the channel until it gets occupied
    $idle \leftarrow idle +$ channel idle duration;
    if correct packet reception then
        if transmitter's MAC address match S's MAC address then
            if $Q$ has a nonezero value then
                add $\frac{idle - q \times T_{DIFS}}{T_{MAC}}$ to $K$;
                $idle, q \leftarrow 0$;
            end if
            $Q \leftarrow$ Queue Size subfield value;
        else
            $q \leftarrow q + 1$;
        end if
    else
        perform CIT to identify colliders
        if $S$ is a collider then
            add $\frac{idle - q \times T_{DIFS}}{T_{MAC}}$ to $K$;
            $idle, q \leftarrow 0$;
        else
            $q \leftarrow q + 1$;
        end if
    end if
end while
```

$e^{j\pi} = -1$ and $e^{j0} = 1$, respectively. The transmitter generates symbols each T_s seconds. We denote the nth symbol generated by the transmitter by $x(n)$. Considering there is only one transmitter, we denote the nth symbol received by a receiver with a sampling rate of $\frac{1}{T_s}$ by $y(n)$, which has the following relation with $x(n)$:

$$y(n) = Hx(n) + \mathcal{N}(n), \tag{11}$$

where $H = Me^{j\phi}$ is a complex number with a magnitude of M and an angle of ϕ, modeling the channel attenuation and phase shift effects, respectively. Also, $\mathcal{N}$ models an AWGN channel. We consider the AP to be the receiver in our system model (uplink transmissions), which serves N number of stations. Therefore, Equation 11 can be generalized as follows:

$$y(n) = \sum_{i=1}^{N} H_i x_i(n - \eta_i) u(n - \eta_i) + \mathcal{N}(n), \tag{12}$$

where H_i represents the channel between S_i and the AP, and $x_i(n)$ represents the nth symbol transmitted by S_i, $u(n)$ is the unit step function, and η_i is the index of the received symbol at the AP where S_i starts transmitting.

For the AP to correctly receive transmitted symbols of S_i, it has to compensate for frequency offset (FO), sampling offset, inter-symbol interference, and channel equalization. However, for the purpose of collision detection we only need to explain the effects of FO, channel attenuation, and channel phase shift (PS) on transmitted symbols.

FO is the absolute difference of transmitter and receiver oscillators' frequencies that are supposed to be centered as the exact same frequency. The FO between a pair of transmitter-receiver results in a linear phase shift in received symbols that increases over time. The receiver usually estimates FO and compensates for it. As for PS, i.e. $e^{j\phi_i}$ in Equation 12, the phase of all symbols transmitted by S_i is shifted by a value of ϕ_i. The AP should compensate for the channel phase shift effect to correctly detect collisions. Typically, receivers estimate the channel response and compensate for the channel effects as they do for FO effects by using the 802.11 preamble [1]. Equation 12 can be further generalized to account for frequency offsets between AP and its stations as follows:

$$y(n) = \sum_{i=1}^{N} H_i x_i(n - \eta_i) e^{j2\pi(n-\eta_i)\delta_f(i)T_s} u(n - \eta_i) + \mathcal{N}(n), \tag{13}$$

where $\delta_f(i)$ is the frequency offset between AP and S_i.

CIT relies on studying the architecture of the 802.11 legacy preamble. Standardized preambles are designed to satisfy certain properties, including high FO estimation range, good frame detection accuracy, low dynamic range and low *peak-to-average power ratio* (PAPR) [23]. Every PHY-layer frame starts with a preamble, which begins with two essential fields, short training field (STF) and long training field (LTF). Figure 4 shows the legacy preamble where the sampling frequency is 20 Msps. The STF contains ten identical short training sequences (STS's), which represent ten replicas of a particular periodic signal with period $\lambda_{STF} = 0.8\mu sec$. The STF is used for coerced FO estimation and frame detection [24]. The LTF consists of two long training sequences (LTS's), which represent two cycles of another known periodic signal with period $\Delta_{LTF} = 4\Delta_{STF}$, plus a 1.6 μsec cyclic prefix. The LTF is used for channel estimation and further FO estimation. The legacy preamble is included in all the 802.11 enhancements. This is for the backward compatibility of the newer amendments with the legacy versions. For CIT to be applicable for all 802.11 versions, we will consider the legacy preamble to develop our algorithm, and refer to the legacy preamble as the "preamble", throughout the paper.

B. Collision Identification Technique (CIT)

If the AP receives a signal y that it fails to correctly decode, it will initiate CIT. The heart of our collision detection technique is to leverage the fact that the 802.11 packets start with a known set of samples (i.e., 802.11 preamble). CIT uses this fact and calculates the cross-correlation of the known preamble, which is modified to incorporate the FO and channel PS effects, with the collided signal, and looks for peaks in this cross-correlation that exceed a detection threshold. Equation 13 shows that FO and PS effects change the phase of the transmitted symbols, hence the cross-correlation might not peak where the known preamble overlaps with the start of a Wi-Fi transmission. We overcome the former

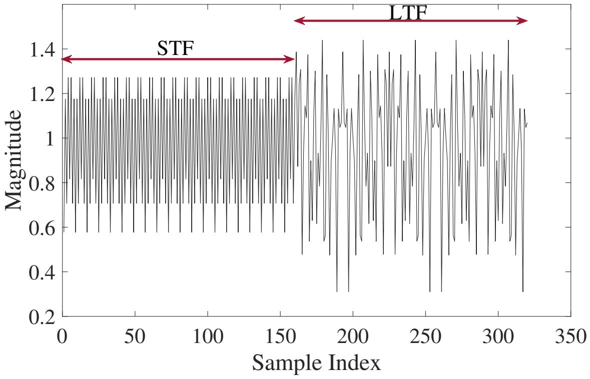


Fig. 4: Legacy preamble of an 802.11 packet.

challenge by requiring the AP to keep the latest FO and PS estimations of each successful transmission in the vectors $\delta_f = [\delta_f(1), \delta_f(2), ..., \delta_f(N)]$ and $\phi = [\phi(1), ..., \phi(N)]$, respectively, where $\delta_f(i)$ and $\phi(i)$ are the latest estimated FO and PS between station S_i and the AP, respectively. Upon receiving a collided signal y, the AP first, modifies the known preamble by incorporating the effects of FO and PS on the known preamble, then it computes the cross-correlation with y. The modified preamble is denoted by P_i and obtained as:

$$P_i(n) = e^{j\phi(i)} e^{j2\pi n \delta_f(i) T_s} P(n) \quad n = 1, .., L, \tag{14}$$

where P is the original preamble and L is its length. The cross-correlation between P_i and y is denoted by Γ_i and can be calculated as:

(i) $0 \leq m \leq |y|$:

$$\Gamma_i(m) = \frac{\left|\sum_{n=1}^{L} P_i^*(n) y(n+m)\right|}{\sqrt{\sum_{n=1}^{L} P_i^*(n) P_i(n)} \sqrt{\sum_{n=1}^{L} y^*(n+m) y(n+m)}} \tag{15}$$

(ii) $m < 0 \quad$ or $\quad m > |y|$:

$$\Gamma_i(m) = 0, \tag{16}$$

where "$*$" is the complex conjugate operator. Also we zero-pad y to gather cross-correlation results for $|y| - L < m < |y|$. In Equation 15, the two factors in the denominator are normalizing the value of Γ_i for it to be in the range $[0, 1]$. To keep track of the highest cross-correlation values over all Γ_i's for each received symbol, $m \in \{0, 1, ..., |y|\}$), the AP builds a composite cross-correlation vector Γ as:

$$\Gamma(m) = max(\Gamma_1(m), ..., \Gamma_N(m)), m \in \{0, 1, ..., |y|\}. \tag{17}$$

Then, each m that has a $\Gamma(m) > th_c$ will be assigned to a station to be identified as a collider. In our analysis of 802.11 packets we have seen that if a station, say S_i, is a collider then Γ_i will have comparable cross-correlation values for values of m that are surrounding η_i, i.e., the index of the first received symbol transmitted by S_i. This is due to preamble's periodic nature. In Figure 5, we show the auto-correlation of the 802.11 preamble while having a sampling rate of 20 Msps. We can see that several high-value peaks reside in the neighborhood of the start of the preamble. And from Equation 17, we can see that only the highest values of cross-correlation will be considered for each index of Γ. So, to prevent the false detection of stations as colliders and falsely not detecting colliding stations, we need to eliminate the surrounding cross-correlation peak values around highest peak values. We realize the former by defining a filtering window of size ζ that assigns a value of zero for cross-correlation values of m's that are within the ζ range of the highest cross-correlation values. CIT's filtering approach proceeds dynamically by first zeroing out the cross-correlation values of indices surrounding the index of highest cross-correlation value, then updating Γ_j's, and doing the former for the second highest peak, and proceeding similarly for all the remaining cross-correlation values higher than th_c. We are assuming that in a collision, packets sent by different stations are apart from one another at least by a MAC time slot T_{MAC}. Therefore, $\zeta = \frac{T_{MAC}}{T_s}$ is a reasonable choice. After filtering, the AP constructs the composite cross-correlation vector Γ by the updated Γ_i's using Equation 17, once more. Then each value in Γ that is higher than a detection threshold th_c is assigned to the station with that specific cross-corelation value at that specific received symbol index. Each station with an assigned peak value will be considered as a participant transmitter for the received signal y. After assigning all the peak values, CIT will construct a vector, ID, that will consist of the IDs of all the colliding stations during y.

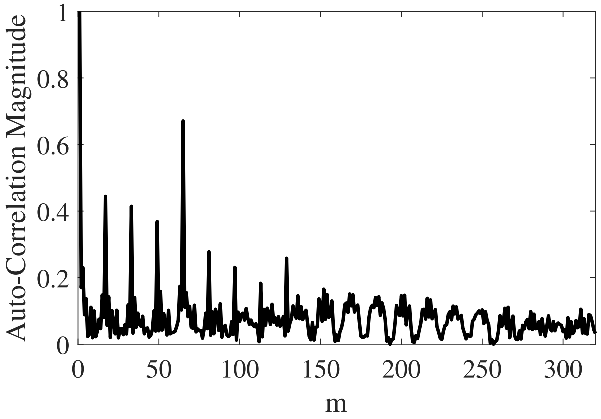


Fig. 5: Auto-correlation magnitude value of the 802.11 preamble.

To better understand the process of CIT, we provide a simulation example using MATLAB's Wireless Waveform Generator [25]. Consider a WLAN with six stations and one AP, where S_1 and S_2 are hidden terminals. S_1 starts its transmission and the AP senses the channel to busy and start receiving the transmitting signal, during S_1's transmission, all stations except S_2 freeze their backoff counter. After the AP receives about 2000 samples, with a sampling rate of

20 Msps, S_2 starts its transmission, while the AP continues receiving samples but it will not be able to successfully decode any packets. So it initiates CIT which starts by calculating the cross-correlation values for each station and adopting a filtering of $\zeta = \frac{T_{MAC}}{T_s} = \frac{9\mu s}{\frac{1}{20Msps}} = 180$. In Figure 6, we present the magnitude of the cross-correlation values of the modified preambles with y for 6 stations, where each station is 5 meters away from the AP. We use itu-r m.2135-1 channel path loss model with $SNR = 10$ dB. The center frequency and bandwidth for both transmission and reception are 2.4 GHz and 20 MHz, respectively. We randomly select the elements of the vectors δ_f and ϕ from the intervals $[-125, 125]$ KHz, and $[0, 2\pi]$, respectively. We set $\zeta = 180$ and $th_c = 0.6$. It can be seen that all Γ_i's have peak values at m indices that correspond to the start of a Wi-Fi transmission. However, the highest cross-correlation value is for the Γ_i that is correctly modifying the transmitted preamble (incorporating the right values for FS and PS in Equation 14). Looking at Figure 6, Γ_1 has the largest cross-correlation value at $m = 0$ ($\Gamma_1(0) = 0.9483$), hence $\Gamma(0) = \Gamma_1(0)$. Following the same procedure and constructing Γ for the remaining m values, it can be seen that Γ will have only one other value larger than th_c, which is at $m = 2000$ ($\Gamma(2000) = 0.6695$). $\Gamma(2000)$ is associated to S_2, since $\Gamma_2(2000) = \Gamma(2000) > th_c$. Since there are no other $\Gamma(m)$'s larger than th_c, CIT will terminate with $ID = \{S_1, S_2\}$, with S_1 and S_2 having transmission start indices of $m = 0$ and $m = 2000$ during y, respectively. .

IV. Evaluations

A. Colision Identification technique

To obtain the accuracy of CIT, we conduct simulations, using MATLAB's WLAN toolbox, for three settings of a WLAN with $N =$ 3, 6, and 9. In all the settings, stations S_1 and S_2 are hidden terminals. We set the center frequency and bandwidth for transmission of all stations to 2.4 GHz and 20 MHz, respectively. Our algorithm's performance is dependent on th_c and ζ. Therefore, we vary th_c from 0 to 1 and set $\zeta = 180$. For each th_c value, we run 100 different simulations, with different random seeds, each including 1000 different collision combinations of S_1 and S_2. We derive the accuracy as:

$$\text{Accuracy} = \frac{\text{Number of correct detections}}{\text{Total number of detections}} \times 100\%, \quad (18)$$

where a correct detection translates into correct identification of all colliders in y. We randomly select the PS values of all stations to be in the range $[0, 2\pi]$. Also the FO values of all stations is randomly selected from $[-125, 125]$ kHz for each simulation run, which is the acceptable FO for 2.4 GHz center frequency [1]. It is important to note that as the FO values of stations get closer to each other the possibility of a miss-detection increases. To fully illustrate the effect of the former, we include a new parameter Δ into our evaluations, which effects the random selection of FO values. The value of Δ indicates that for any element of δ_f, e.g. $\delta_f(i)$, the only element of δ_f residing in the frequency range $[\delta_f(i) - \frac{\Delta}{100}\delta_f(i), \delta_f(i) + \frac{\Delta}{100}\delta_f(i)]$ is $\delta_f(i)$.

In Figures 7(a), 7(b), and 7(c), we show CIT's accuracy vs. th_c for $N =$ 3, 6, and 9, respectively. It can be seen that for $th_c = 0.5$, CIT can achieve 96%, 94%, 88% collision identification accuracy for when we have $N =$ 3, 6, and 9, respectively.

B. CW$_{min}$ Estimation

Our simulation evaluations are based on a C++-based discrete-event simulator called CSIM [26]. CSIM includes functions and classes for generating and synchronizing process-oriented events. We implement the DCF as detailed in 802.11 ac standard, including all timing requirements. An indoor scenario is considered, where a number of Wi-Fi devices are uniformly distributed in a square area of length 80 meters. In this section we present CW$_{\text{min}}$ estimation (CWE) accuracy results for WLAN's where $N =$ 3, 6, and 9. In all our simulation settings, the CW$_{\text{min}}$ value of all stations are randomly selected from $\{2, 3, ..., 16\}$. We conduct 93, 70, and 51 simulation setups for $N =$ 3, 6, and 9, respectively, which results into 279, 420, and 459 total estimations in total. In Figures 8(a) and 8(b) , we show the accuracy performance of CWE vs. the monitoring period (i.e., T) for when we have a collision detection accuracy of 100% and for when we use CIT, respectively. It can be seen that by using CIT with $T = 60$ sec we can achieve 100%, 98.81%, and 96.3% CWE accuracy, for when we have $N =$ 3, 6, and 9, respectively.

V. Related Works

In [16], Rong et al.'s misbehavior detection scheme is based on the sequential hypothesis testing. Instead of monitoring the backoff values selected by stations they first developed analytical models for packet inter-arrival time distribution from each station in the network, where multiple cheating stations coexist. Using the characteristics of this probability distribution, they developed an algorithm to detect cheating stations based on the throughput degradations observed at normal stations. However, they only considered saturated traffic and they assumed that all stations are implementing RTS/CTS exchange. To detect misbehavior in 802.11 WLAN's, Tang et al. [17] assumed that the number of aggressors in the WLAN is known and they derived Markov chains for different settings of aggressors and well-behaved stations, then they analyzed the successful transmission rate of the tagged station to see whether it will reach beyond the rate of a standard station's to be considered as an aggressor. The authors assumed that stations are saturated with traffic. Also, they only assumed one aggressor and considered stations to be in each others sensing ranges, hence eliminating the chance of collisions and hidden terminals. The authors in [6], proposed mechanisms to detect and penalize aggressors that choose a low CW$_{\text{min}}$. The detection is applied on multiple observations of backoff values and then compared to a supposed average backoff value to determine whether it is less than the supposed value, if so, then the station is considered as an aggressor. Each station

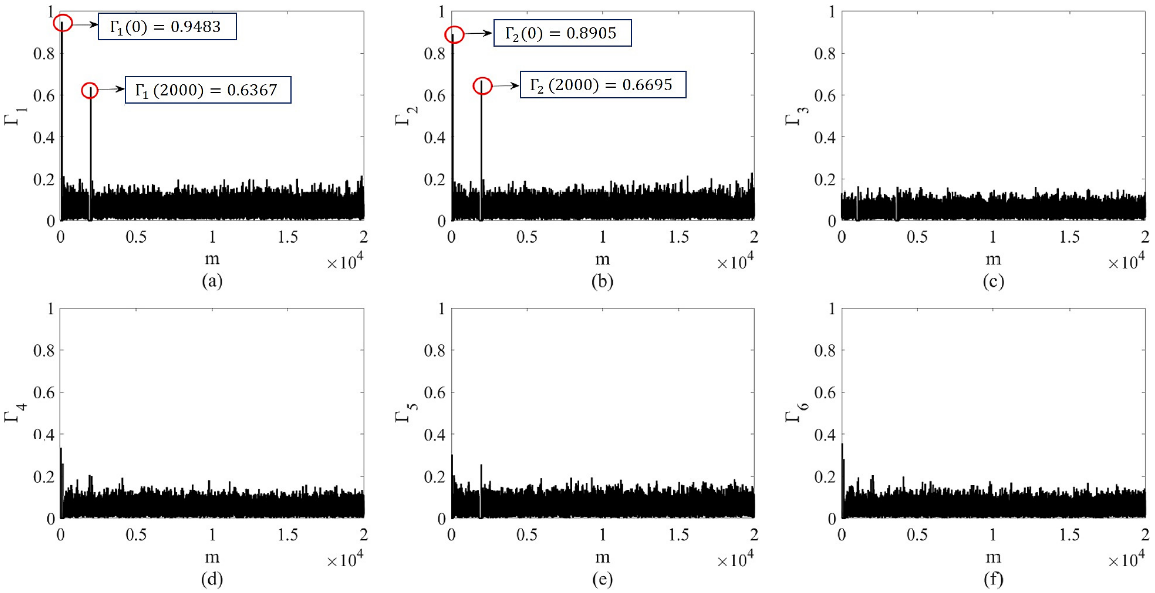


Fig. 6: The cross-correlation of the modified preambles with the collided signal, y for (a)S_1, (b)S_2, (c) S_3, (d) S_4, (e) S_5, and (f) S_6.

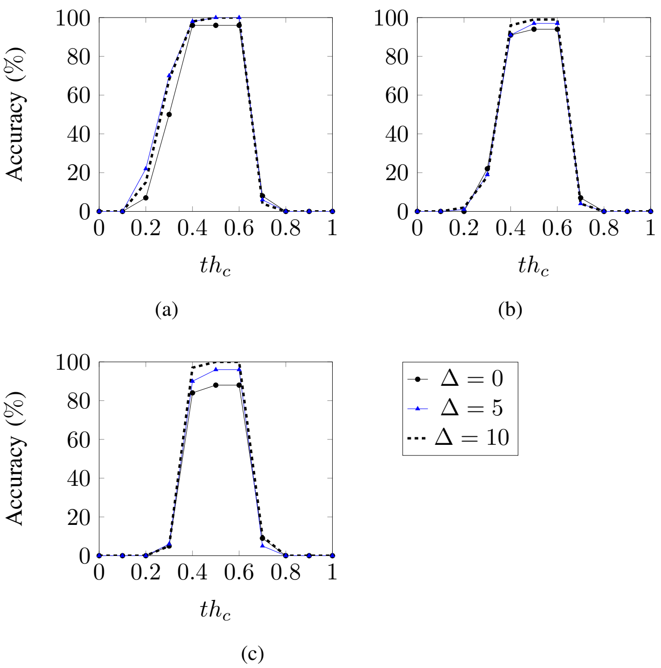


Fig. 7: Accuracy of CIT vs. the collision detection threshold (i.e., th_c) for (a) $N = 3$, (b) $N = 6$, and (c) $N = 9$.

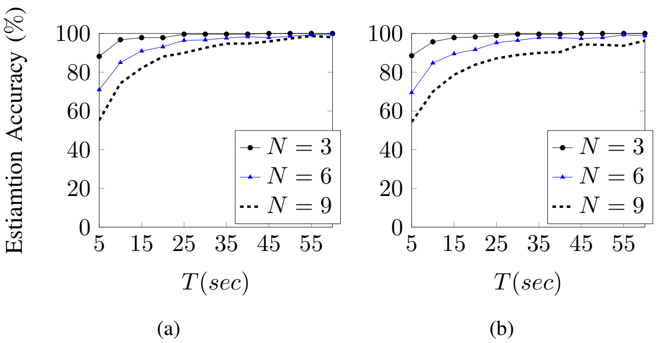


Fig. 8: CWE accuracy vs. the monitoring period, T, for $N = 3$, 6, and 9, when (a) the collision detection accuracy is 100% and (b) we use CIT for collision detection.

is observed by all its one hop neighbors and all stations are considered to have backlogged traffic. The authors claimed that the hidden terminal problem is solved by taking the majority vote for deciding whether a station is aggressive. For the AP to detect stations with low CW_{min} values, Raya et al. [18] proposed that the AP should first gather backoff value traces from each station and then compare the average value of selected backoff values with a nominal average backoff value which is used by a standard compliant station, if the measured average value of the backoff value is less than the nominal average value, then the stations will be considered as aggressors. The authors assumed backlogged traffic for all stations and did not take collisions into account. Machine learning has also found its way in various wireless communication applications [19], [27], [28]. In [19], the authors tackle the aggressive behavior of stations in the WLAN by equipping the standard stations with a machine learning module, specifically random forests, to adapt their CW_{min} to get their fair share of channel airtime. Their framework only enhances the performance of stations that utilize their module, thus the performance of standard-compliant stations that do not use their adaptation algorithm might decrease. Also, for collision detection, the work closest to ours is [22], which we introduced in section III. Zhao et al [29] have also developed an algorithm to resolve RTS collisions, by analyzing the payload of the RTS as a vector and obtaining its distribution, and reformulating the RTS resolution as a sparse-recovery problem.

VI. Conclusions

In this work, we showed the unfairness that will be created when aggressive stations with low $\mathrm{CW}_{\mathrm{min}}$ exist in the WLAN. We proposed a novel solution for the AP to detect aggressors in the WLAN by estimating their $\mathrm{CW}_{\mathrm{min}}$'s, i.e., CWE. Using CWE, the AP needed to monitor the backoff values of its stations for $\mathrm{CW}_{\mathrm{min}}$ estimation, which required the AP to keep track of the idle time each station spent backing off. The former also needed the AP to be able identify colliding stations in an 802.11 uplink collision, which we tried to resolve by introducing our collision detection and identification technique (CIT). Overall, our collision detection algorithm obtains accuracies of 96%, 94%, and 88%, our $\mathrm{CW}_{\mathrm{min}}$-estimation algorithm has estimation accuracies of 100%, 98.81%, and 96.3%, when we have 3, 6, and 9 stations in the WLAN, respectively.

VII. Acknowledgments

This research was supported in part by NSF (grants CNS-1563655, CNS-1731164, and IIP-1822071), and in part by the U.S. Army Small Business Innovation Research Program Office and Army Research Office under Contract No. W911NF-21-C-0016. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF or Army.

References

[1] "IEEE standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534.

[2] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *In Proc. of the ACM MobiHoc*, 2005, p. 46–57.

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

[4] M. Manzo, T. Roosta, and S. Sastry, "Time synchronization attacks in sensor networks," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2005, p. 107–116.

[5] D. R. Raymond, R. C. Marchany, M. I. Brownfield, and S. F. Midkiff, "Effects of denial-of-sleep attacks on wireless sensor network mac protocols," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 367–380, 2009.

[6] 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," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1203–1217, 2015.

[7] S. Radosavac, J. S. Baras, and I. Koutsopoulos, "A framework for mac protocol misbehavior detection in wireless networks," in *Proceedings of the 4th ACM Workshop on Wireless Security*, 2005, p. 33–42.

[8] P. Kyasanur and N. H. Vaidya, "Selfish mac layer misbehavior in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 5, pp. 502–516, 2005.

[9] Z. Lu, W. Wang, and C. Wang, "On order gain of backoff misbehaving nodes in csma/ca-based wireless networks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.

[10] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols," *ACM Trans. Sen. Netw.*, 2009.

[11] K. El-Khatib, "Impact of feature reduction on the efficiency of wireless intrusion detection systems," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1143–1149, 2010.

[12] V. Gupta, S. Krishnamurthy, and M. Faloutsos, "Denial of service attacks at the mac layer in wireless ad hoc networks," in *In Proc. of the IEEE MILCOM*, 2002, pp. 1118–1123.

[13] J. Konorski, "Protection of fairness for multimedia traffic streams in a non-cooperative wireless lan setting," in *In Proc. of the PROMS*, 2001, pp. 116–129.

[14] ——, "Multiple access in ad-hoc wireless lans with noncooperative stations," in *In Proc. of the NETWORKING*, 2002, pp. 1141–1146.

[15] M. Cagalj, S. Ganeriwal, I. Aad, and J.-P. Hubaux, "On cheating in csma/ca ad hoc networks," Tech. Rep., 2004.

[16] Y. Rong, S. . Lee, and H. . Choi, "Detecting stations cheating on backoff rules in 802.11 networks using sequential analysis," in *Proc. of the IEEE INFOCOM*, 2006, pp. 1–13.

[17] J. Tang, Y. Cheng, and W. Zhuang, "Real-time misbehavior detection in ieee 802.11-based wireless networks: An analytical approach," *IEEE Transactions on Mobile Computing*, vol. 13, no. 1, pp. 146–158, 2014.

[18] M. Raya, I. Aad, J. . Hubaux, and A. El Fawal, "Domino: Detecting mac layer greedy behavior in ieee 802.11 hotspots," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1691–1705, 2006.

[19] A. H. Y. Abyaneh, M. Hirzallah, and M. Krunz, "Intelligent-CW: AI-based Framework for Controlling Contention Window in WLANs," in *Proc. of the IEEE DySPAN*, 2019, pp. 1–10.

[20] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, 2000.

[21] S.-H. Cha, "Taxonomy of nominal type histogram distance measures," in *In Proc. of the American Conference on Applied Mathematics*, 2008, p. 325–330.

[22] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *Proc. of the ACM SIGCOMM 2008 Conference on Data Communication*, 2008, p. 159–170.

[23] H. Rahbari and M. Krunz, "Rolling preambles: Mitigating stealthy fo estimation attacks in ofdm-based 802.11 systems," in *In Proc. of the IEEE CNS*, 2016, pp. 118–126.

[24] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE Transactions on Communications*, pp. 1613–1621, 1997.

[25] "WLAN toolbox version 3.0," R2020a, the MathWorks, Natick, MA, USA.

[26] "Csim20," [http://www.mesquite.com], accessed: 2019-08-30.

[27] A. Yazdani Abyaneh, V. Pourahmadi, and A. Hosein Gharari Foumani, "CSI-based authentication: Extracting stable features using deep neural networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, 2020.

[28] W. Zhang, M. Feng, M. Krunz, and A. Hossein Yazdani Abyaneh, "Signal detection and classification in shared spectrum: A deep learning approach," in *Proc. of the IEEE INFOCOM*, 2021, pp. 1–10.

[29] S. Zhao, Z. Qu, Z. Luo, Z. Lu, and Y. Liu, "Comb decoding towards collision-free WiFi," in *Proc. of the NSDI*, 2020, pp. 933–951.