Real-Time Misbehavior Detection in IEEE 802.11e Based WLANs

Xianghui Cao*, Lu Liu*, Wenlong Shen*, Jin Tang† and Yu Cheng*
*Department of Electrical and Computer Engineering, Illinois Institute of Technology, USA
Email: xcao10@iit.edu; {lliu41,wshen7}@hawk.iit.edu; cheng@iit.edu
†AT&T Labs, USA. Email: jin.tang@att.com

***Abstract*—The Enhanced Distributed Channel Access (EDCA) specification in the IEEE 802.11e standard supports heterogeneous backoff parameters and arbitration inter-frame space (AIFS), which makes a selfish node easy to manipulate these parameters and misbehave. In this case, the network-wide fairness cannot be achieved any longer. Many existing misbehavior detectors, primarily designed for legacy IEEE 802.11 networks, become inapplicable in such a heterogeneous network configuration. In this paper, we propose a novel real-time hybrid-share (HS) misbehavior detector for IEEE 802.11e based wireless local area networks (WLANs). The detector keeps updating its state based on every successful transmission and makes detection decisions by comparing its state with a threshold. We develop mathematical analysis of the detector performance in terms of both false positive rate and average detection rate. Numerical results show that the proposed detector can effectively detect both contention window based and AIFS based misbehavior with only a short detection window.**

***Index Terms*—IEEE 802.11e; contention window; AIFS; misbehavior detection; real-time; false positive rate; detection rate**

I. Introduction

To support rapid growing applications (especially multimedia ones) of wireless local area networks (WLANs), the IEEE 802.11e standard adopts the Enhanced Distributed Channel Access (EDCA) mechanism to provide media access control (MAC) level differentiation in quality of service (QoS) [1], [2]. With EDCA, network traffic is prioritized and classified into several access categories (ACs). Service differentiation is realized by assigning different parameters for each AC, including the minimum and maximum contention window sizes (CWmin and CWmax, respectively), the arbitration inter-frame space (AIFS) number and transmission opportunity (TXOP) limit [3].

In IEEE 802.11e based WLANs, a selfish/misbehaving node can deliberately manipulate those parameters to gain advantage over others. For example, it can use a smaller AIFS to wait for shorter time than others in the same AC before accessing the medium. As a result, it can access the medium more frequently, and hence gain a higher priority for data transmission. It is even possible for an intensively misbehaving node to block the transmissions from other nodes and cause the so-called denial of service attack. Therefore, real-time misbehavior detection is demanded in order to isolate such a node and alleviate its impact to the network.

Due to the random access which is based on the carrier sense and multiple access with collision avoidance (CSMA/CA) MAC protocol, we usually need to monitor each node for a period of time to judge whether it is misbehaving or not. Since it is difficult to extract necessary information from collided transmissions, information conveyed in successful transmissions is perhaps the only measurement can be utilized for detection. Toledo *et al.* proposed to detect backoff misbehavior by checking whether the idle time between consecutive successful transmissions from a target node obeys the normal distribution [4]. Exploiting the fairness property across the network, Tang *et al.* designed a light-weight fair-share detector, which does not rely on the idle time distribution [5]. However, with multiple ACs in an IEEE 802.11e WLAN, the network-wide fairness as achieved in legacy IEEE 802.11 based networks dose not hold any longer [6], making the above detectors generally inapplicable.

To detect backoff misbehavior in IEEE 802.11e networks, Szott *et al.* proposed a χ^2 detector by comparing the measured and expected backoff values [7]. However, the exact values of backoff periods followed by unsuccessful transmissions may be hard to measure. The detector in [8], however, takes advantage of the fact that the interval between two consecutive successful transmissions is uniformly distributed in [0, CWmin) providing that the packet in the second transmission was not retransmitted before. Nevertheless, the detector delay could be very high. While there are works well addressed the TXOP misbehavior [9], efficient and real-time detection of contention window and AIFS misbehavior still remains open.

In this paper, we propose a new detector to deal with misbehavior in IEEE 802.11e networks. We focus on both contention window and AIFS misbehavior. The major contributions in this paper can be summarized as follows. We analyze the misbehavior strategy in IEEE 802.11e networks and show that a selfish node can gain significant advantage over other nodes by manipulating its contention window or AIFS. We also demonstrate that the existing fair-share based detector for legacy IEEE 802.11 networks is unable to detect certain misbehavior in the IEEE 802.11e cases with multiple priority classes. Then, we propose a mathematical model of the percentage of resource sharing for a node in each priority class. Based on this, we design a novel hybrid-share detector and develop analytical results of the detector performance in terms of false positive rate and average detection rate. We

978-1-4799-3512-3/14/$31.00 ©2014 IEEE

also present numerical results to demonstrate the performance in various aspects including different threshold, misbehaving intensity and detection window. The remainder of this paper is organized as follows. Section II overviews the problem. Following the mathematical MAC model in Section III, our detector is designed and evaluated in Section IV. Numerical results are presented in Section V and the paper is concluded in Section VI.

II. Problem Overview

A. IEEE 802.11e EDCA

In IEEE 802.11 based wireless local area networks (WLANs), the channel access among nodes is coordinated by the CSMA/CA mechanism. Time is divided into equal slots. Before transmission, a node should sense the medium idle until a backoff timer expires. Each node takes the binary exponential backoff strategy to access the channel with the backoff timer at each backoff stage initialized at a value randomly chosen from $[0, CW-1]$. The contention window size CW is initialized at CWmin and doubles (until CWmax) once a transmission is unsuccessful (a packet will be retransmitted at most for a certain number of times). Once the medium is busy, the backoff timer will be suspended until it becomes idle again. The CSMA/CA mechanism also uses an inter-frame space time to defer a transmission or backoff period in order to give way to high priority messages. Unlike the distributed coordination function (DCF) mechanism in legacy IEEE 802.11 standard, the Enhanced Distributed Channel Access (EDCA) specification in IEEE 802.11e supports hybrid backoff parameters and arbitration inter-frame space (AIFS). In default, there are four priority classes (access categories) defined in IEEE 802.11e EDCA [3], as shown in Table II-A.

TABLE I
EDCA default settings.

Access category	CWmin	CWmax	AIFSN
Background AC_{BK}	aCWmin	aCWmax	7
Best Effort AC_{BE}	aCWmin	aCWmax	3
Video AC_{VI}	(aCWmin+1)/2-1	aCWmin	2
Voice AC_{VO}	(aCWmin+1)/4-1	(aCWmin+1)/2-1	2

In this paper, we consider the general cases that there are c priority classes, each of which is assigned contention window sizes CWmin$_i$ and CWmax$_i$, and inter-frame space AIFS$_i$=AIFSN$_i$*aSlotTime+aSIFSTime, where AIFSN is the number of slots after a short inter-frame space duration a node should defer before either invoking a backoff or starting a transmission. The parameters are assigned by the AP.

B. Misbehavior Analysis

A misbehaving node may use different parameters other than those assigned by the AP, to gain a higher sharing of the resource. Fig. 1 illustrates the impact of a misbehaving node and shows the percentages of resource sharing of the misbehaving node and a normal node. Here, the percentage of resource sharing is defined as the portion of throughput contribution from a particular node over the total network throughput. In this figure, we consider a network consisting of 10 normal nodes and one misbehaving node. Each node always has packets in its buffer for sending out. Each normal node takes MAC parameters as $\text{CWmin} = 15, \text{CWmax} = 1023$ and $\text{AIFSN} = 2$, while the misbehaving node takes $\text{CWmin} = 1 \sim 32, \text{CWmax} = 1023$ and $\text{AIFSN} = 0 \sim 2$.

The figure clearly demonstrates that the misbehaving node can gain significant advantage over the other nodes by manipulating its MAC parameters. Moreover, the impacts of CWmin and AIFSN are different. For example, in order to achieve 10% more throughput, the misbehaving node needs to reduce its CWmin to a much smaller value (e.g., from 15 to less than 7); while, this can also be achieved by simply reducing its AIFSN from 2 to 1. In other words, the misbehaving impact on the network is more sensitive to AIFSN than CWmin.

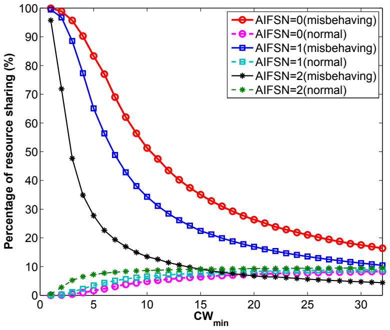


Fig. 1. Impact of MAC misbehavior.

In this paper, we consider both CWmin and AIFSN misbehavior. While the proposed detector is also effective for CWmax misbehavior, a malicious node may prefer to manipulate CWmin over CWmax since the former strategy has greater impact. We focus on saturated traffic case, i.e., each node always has packets for transmitting to the AP. Otherwise, for a light-loaded network, a misbehaving node may not have much impact on the other normal nodes. Our problem is to detect such misbehavior at the AP in real-time.

C. Fair-Share Detector and Challenges

Consider a WLAN consisting of one AP and n nodes which locate inside each other's communication range. The nodes compete for accessing a channel and sending packets to the AP. In the legacy IEEE 802.11 standard, the DCF mechanism guarantees that each node will share the same portion of the channel resource and maintains fairness across the network. For an arbitrary node v, let a binary variable I_v be the indicator of whether a packet received by the AP is from node v or not. In normal cases, due to the network-wide fairness guarantee, we have probability $\mathbb{P}[I_v = 1] = \frac{1}{n}$.

A misbehaving node can gain unfair share of the resource by manipulating its backoff parameters, e.g, using a smaller CWmin. If the AP records all the received packets, it can notice that more packets are from the selfish one, i.e., $\mathbb{P}[I_v = 1] > \frac{1}{n}$ if v is misbehaving. In [5], we take advantage of this important feature to design a nonparametric cumulative sum (CUSUM) based fair-share misbehavior detector (called

FS detector) to detect such misbehavior in real-time, which is described as follows.

For a target node v, let X_k be the state of the detector for v. X_k initializes at 0, i.e., $X_0 = 0$. For the kth packet received by the AP, the state of the detector is updated as follows.

$$X_{k+1} = \left[X_k + (nI_k - 1)\right]^+, \tag{1}$$

where $x^+ = x$ if $x > 0$ and 0 otherwise. In other words, if the packet is from v, we have $I_k = 1$ and $X_{k+1} = X_k + n - 1$; otherwise, $I_k = 0$ and $X_{k+1} = X_k - 1$. The idea behind is that, due to fair sharing, the nodes roughly take turns to transmit packets. Therefore, the detector state X_k is likely to be bounded. In presence of misbehaving nodes, since $\mathbb{P}[I_v = 1] > \frac{1}{n}$, the unfair portion of channel sharing will accumulate such that the state of the FS detector associated with each misbehaving node finally becomes unbounded. Thus, we can employ a detection threshold h to decide whether v is misbehaving (i.e., $\delta_k = 1$) or not (i.e., $\delta_k = 0$) as follows.

$$\delta_k = \begin{cases} 1, & \text{if } X_k \geq h, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Because every received packet is counted by the AP in making the detection decisions, with satisfactory accuracy, the FS detector can identify the misbehaving node much faster than many existing detectors. Moreover, the FS detector is nonparametric and lightweight in terms of computation complexity, it is thereby able to provide real-time misbehavior detection services [5].

Because the underlying assumption of network-wide fairness no longer holds in the EDCA situation, the FS detector cannot be applied directly in networks with hybrid priority classes. However, since sub-network fairness can be still achieved among the nodes in the same class, a direct extension of the FS detector is to design a distinct one for each class. Specifically, for a node in class i, the associated detector should use the number of nodes in this class other than a common n across the whole network as in (1). Nevertheless, such extended FS detector encounters two challenges:

- If there is only one node in a class, its misbehavior may not be detected. To see this, substituting $n = 1$ into (1) we can obtain that $X_{k+1} = [X_k - (1 - I_k)]^+ \equiv 0$ if $X_0 = 0$. As a result, $\delta_k \equiv 0$.
- If all the nodes in one class misbehave, some or all of them may not be detected. Specifically, if they use the same manipulated MAC parameters, the detector sees that none of them is misbehaving; otherwise, at least the one with the lowest throughput among them will be considered as a normal node.

Therefore, to overcome the shortcomings of the FS detector, only considering a single priority class is not enough. In the following, we propose a novel hybrid-share detector based on the following analytical MAC model.

III. MAC Analytical Model

For an arbitrary node v in class i, denote s_i as its percentage of resource sharing. In this section, we propose a model for calculating s_i. We assume there are c priority classes; each contains n_i nodes which compete for channel access using parameters $W_i = \text{CWmin}_i$, $\text{CWmax}_i = 2^{m_i}(W_i + 1) - 1$ and AIFS_i, where $1 \leq i \leq c$ and $m_i = \log_2 \frac{\text{CWmax}_i + 1}{\text{CWmin}_i + 1}$ is the maximum backoff stage. Here, for simplicity, we assume the maximum retransmission limit for node v is the same as m_i[1]. Thus, the contention window size of this node in its jth backoff stage is

$$W_{i,j} = 2^j (W_i + 1) - 1. \tag{3}$$

Let p_i be the frame blocking probability, i.e., the probability that node v senses a busy channel (and thereby suspends its backoff timer countdown) in a generic slot. According to [10], the transmitting probability of node v in a generic slot can be calculated as

$$\begin{aligned} \tau_i &= \frac{1 - p_i^{m_i+1}}{(1 - p_i) \sum_{j=0}^{m_i} p_i^j \left[1 + \frac{1}{1-p_i} \sum_{k=1}^{W_{i,j}} \frac{W_{i,j} - k}{W_{i,j}}\right]} \\ &= \frac{1 - p_i^{m_i+1}}{\sum_{j=0}^{m_i} p_i^j \left(1 - p_i + \frac{W_i}{2}\right)} \\ &= \frac{2(1 - p_i)(1 - 2p_i)}{(1 - 2p_i)^2 + (W_i + 1)(1 - p_i) \frac{1 - (2p_i)^{m_i+1}}{1 - p_i^{m_i+1}}}. \end{aligned} \tag{4}$$

Let $\Delta A_i = \text{AIFSN}_i - \text{AIFSN}_{\text{min}}$ where $\text{AIFSN}_{\text{min}} = \min\{\text{AIFSN}_j | j = 1, \ldots, c\}$. Due to differentiation in AIFS, a node of low priority must wait a longer idle time than a high-priority node after a busy channel period before resuming its backoff timer countdown. Therefore, according to [11], the frame blocking probability of node v can be calculated as follows.

$$\begin{aligned} p_i &= 1 - \left[(1 - \tau_i)^{n_i - 1} \prod_{k=1, k \neq i}^{c} (1 - \tau_k)^{n_k}\right]^{\Delta A_i + 1} \\ &= 1 - \left[\frac{1}{1 - \tau_i} \prod_{k=1}^{c} (1 - \tau_k)^{n_k}\right]^{\Delta A_i + 1} \\ &= 1 - \left(\frac{1 - p_b}{1 - \tau_i}\right)^{\Delta A_i + 1}, \end{aligned} \tag{5}$$

where

$$p_b = 1 - \prod_{k=1}^{c} (1 - \tau_k)^{n_k} \tag{6}$$

is the probability that the channel is busy in a random slot. p_b can be easily measured by the AP. Through the above two equations, the AP can solve the probabilities τ_i and p_i numerically.

In a generic time slot, the probability that node v successfully transmits a packet to the AP is

$$\begin{aligned} p_{s,i} &= \tau_i (1 - \tau_i)^{n_i - 1} \prod_{k=1, k \neq i}^{c} (1 - \tau_k)^{n_k} \\ &= \frac{\tau_i}{1 - \tau_i} (1 - p_b). \end{aligned} \tag{7}$$

[1] In general cases, the hybrid share model will be slightly more complicated, but our modeling methodology and the designed detector are still valid.

Therefore, the percentage of resource sharing of node v (which is also the probability that a successful transmission to the AP is from this node) is given by

$$s_i = \frac{p_{s,i}}{\sum_{j=1}^{c} n_j p_{s,j}} = \frac{\frac{\tau_i}{1-\tau_i}}{\sum_{j=1}^{c} \frac{n_j \tau_j}{1-\tau_j}}. \tag{8}$$

The average number of packets (from any of the nodes) received by the AP in one slot is

$$\begin{aligned}\eta &= \frac{\text{Probability of a successful transmission}}{\text{Average length of a slot time}} \\ &= \frac{p_s}{1 - p_b + p_s T_s + (p_b - p_s) T_c},\end{aligned} \tag{9}$$

where $p_s = \sum_{i=1}^{c} p_{s,i}$ is the probability of a successful transmission, while $p_b - p_s$ is the probability of a collided transmission. $1 - p_b$ is the channel idle probability (i.e., the probability that none of the nodes transmits). T_s and T_c are the numbers of empty slots (i.e., aSlotTime as specified in the standard) of a successful transmission and a collision, respectively. In the case of basic access (without RTS/CTS handshaking), we have [11]

$$T_s = \text{AIFSN}_{\min} + L + 2SIFS + ACK + 2\delta, \tag{10}$$

$$T_c = \text{AIFSN}_{\min} + L + SIFS + ACK + \delta, \tag{11}$$

where L is the length of a packet including the MAC and PHY headers[2]. $SIFS$ and ACK are durations of a short inter-frame space and an ACK transmission period, respectively. δ represents the propagation delay. The units of both T_s and T_c are numbers of empty slots. For the cases with RTS/CTS access mechanism, refer to [11] for the derivation of the corresponding T_s and T_c. Then, the average number of empty slots between two successive transmissions can be given by

$$T = \frac{1}{\eta}, \tag{12}$$

which also describes the frequency of packet arrivals at the AP. Note that, if each misbehaving node is treated as a distinct priority class, the above is able to accommodate both normal and misbehaving nodes.

IV. HYRID-SHARE DETECTOR DESIGN

For a target node belonging to priority class i, the hybrid-share detector (called HS detector) is designed as follows. Due to the high nonlinearity of (4) and (5), the numerical solution of s_i may introduce some error, say ϵ_i. Let $\bar{s}_i$ be the numerical solution of (13), then

$$s_i = \bar{s}_i + \epsilon_i. \tag{13}$$

In the sequel, we shall omit the subscript i since the context is clear. The detector maintains a state X_k with initial state $X_0 = 0$. Once a packet arrives at the AP, the detector state is updated according to

$$X_{k+1} = \left[X_k + (I_k - \bar{s})\right]^+, \tag{14}$$

where I_k is defined below Eq. (1) and $\mathbb{P}[I_k = 1] = s$. Therefore, in normal cases, X_k is expected to remain in $[0, 1]$. We introduce a new detection threshold h and make the decision that whether the target node is misbehaving or not by computing

$$\delta_k = \begin{cases} 1, & \text{if } X_k \geq h, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

Similar as above, $\delta_k = 1$ indicates misbehaving.

Note that, in normal cases, X_k may be able to hit 1 if the AP receives a packet from the target node. Therefore, for the sake of correct detection, h can be set to larger than 1. For detecting real-time misbehavior of the target node, X_k is reset to 0 once it hits the threshold h. If there is only one access class, $s = \frac{1}{n}$, and the HS detector reduces to an FS detector.

We call X_k as the state of the HS detector in step k. Note that the step size may vary from time to time because the packet arrivals at the AP are generally random. However, from (12) we can obtain the average step size as T.

When applying the proposed detector, the AP only needs to compute the MAC model and calculate the percentage of resource sharing once, as long as the network configuration and MAC parameters assigned to each node do not change. As shown in (14), the computation complexity of the proposed detector itself is very low. Therefore, it is worth noting that the proposed detector is light-weight. Moreover, since all received packets are utilized by the detector, misbehavior can be detected in a real-time manner.

Definition *1:* To evaluate the performance of the HS detector, we define the following metrics.

- The *false positive rate* p_f of the HS detector is the conditional probability that the target node is indicated misbehaving (i.e., X_k is no less than the threshold h) when in fact none of the nodes is misbehaving.
- The *detection rate* $p_d(D)$ of the HS detector is the probability that a misbehaving node will be detected in D time slots (empty slots defined in IEEE 802.11e).

p_f can be viewed as the rate of false alarms, while $p_d(D)$ reflects the effectiveness and real-time performance of the HS detector. Below we analytically analyze the detector performance by modeling p_f and $p_d(D)$.

A. False positive rate

Without loss of generality, suppose there exists $\sigma > 0$ such that both $\frac{\bar{s}}{\sigma}$ and $\frac{1-\bar{s}}{\sigma}$ are integers (say L_0 and L_1, respectively). For example, we can use the precision of $\bar{s}$ to determine the above two integers. For any step k between two adjacent detector state resettings, suppose there are k_1 times that $I_\kappa = 1$ and k_0 times that $I_\kappa = 0$, where κ is between the last resetting step and k. Thus, based on (14), $X_k \in \{0, X_{k-1} - \bar{s}, X_{k-1} + 1 - \bar{s}\}$. Furthermore, $X_k \leq k_1(1 - \bar{s})$ which yields that

$$X_k \in \{0, \sigma, 2\sigma, \ldots, k_1 L_1 \sigma\}. \tag{16}$$

Since X_k is multiples of σ, its largest possible value is $\bar{m}\sigma$ where $\bar{m} = \lceil \frac{h}{\sigma} \rceil$ (otherwise X_k is reset). Therefore, the

[2] We assume all the packets are of the same length. Please refer to [12] for the case with diverse packet lengths.

support of X_k can be denoted as

$$\begin{aligned}\mathcal{M} &= \Big\{0, m_1\sigma, m_2\sigma, \ldots, \bar{m}\sigma \Big| m_j \in \mathbb{N}^+, m_j < \bar{m}\Big\} \\ &\subseteq \{0, \sigma, 2\sigma, \ldots, \bar{m}\sigma\}. \end{aligned} \tag{17}$$

Clearly, $\mathcal{M}$ is a finite set.

According to (14), X_{k+1} depends only on X_k and thus the sequence $\{X_k\}$ forms a homogeneous Markov chain. Since the support of X_k may vary from step to step, to calculate the probabilities of the chain's states at any step k, we can consider the bigger set $\{0, \sigma, 2\sigma, \ldots, \bar{m}\sigma\}$ without loss of generality. Define

$$\boldsymbol{x}_k = [\mathbb{P}[X_k = 0], \mathbb{P}[X_k = \sigma], \ldots, \mathbb{P}[X_k = \bar{m}\sigma]]\,. \tag{18}$$

By definition, we have $\boldsymbol{x}'_k * \mathbf{1} = 1$, where $\mathbf{1}$ is a vector with all elements equal to 1. Due to the homogeneity of the chain, we can have $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k\mathbf{P}$, where $\mathbf{P}$ is the step-independent probability transition matrix. $\mathbf{P}$ depends only on s and can be also represented as $\mathbf{P}(s)$. Let $P_{i,j}$ be the (i,j)th entry of $\mathbf{P}$. To describe $\mathbf{P}$, let us consider the steady-state probabilities of the chain $\{X_k\}$: $\boldsymbol{\pi} = \lim_{k\to\infty} \boldsymbol{x}_k = [\pi_0, \pi_1, \ldots, \pi_{\bar{m}}]$. Apparently, $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$. π_m can be calculated by considering the following scenarios:

- If $m = 0$, we have $X_k = 0$ which happens either if $X_{k-1} \le \bar{s}$ and $I_{k-1} = 0$ (i.e., the received packet is not from the target node) or $X_k = \bar{m}\sigma$ and the state is reset subsequently. Therefore,

$$\pi_0 = \sum_{i=0}^{L_0} \pi_i(1-s) + \pi_{\bar{m}}, \tag{19}$$

  which indicates that $P_{i,0} = 1-s, \forall i \le L_0$ and $P_{\bar{m},0} = 1$.
- If $0 < m \le L_1$, we have $X_{k-1} = (m + L_0)\sigma$ if $I_k = 0$. Hence

$$\pi_m = \pi_{m+L_0}(1-s). \tag{20}$$

  i.e., $P_{m+L_0,m} = 1-s$.
- If $0 < m < \bar{m} - L_0$, we have $X_{k-1} = (m+L_0)\sigma$ if $I_k = 0$ and $X_{k-1} = (m - L_1)\sigma\}$ otherwise. Hence

$$\begin{aligned}\pi_m &= \pi_{m+L_0}(1-s) + \pi_{m-L_1}s \\ &= \pi_{m+L_0}P_{m+L_0,m} + \pi_{m-L_1}P_{m-L_1,m}.\end{aligned} \tag{21}$$

- Otherwise if $\bar{m} - L_0 \le m < \bar{m}$, we do not have the case $I_k = 0$. Hence,

$$\pi_m = \pi_{m-L_1}s = \pi_{m-L_1}P_{m-L_1,m}. \tag{22}$$

- Finally, when $m = \bar{m}$,

$$\pi_{\bar{m}} = \sum_{i=1}^{L_1} \pi_{\bar{m}-i}s = \sum_{i=1}^{L_1} \pi_{\bar{m}-i}P_{\bar{m}-i,\bar{m}}. \tag{23}$$

Solving these equations, we can get a unique $\boldsymbol{\pi}$. Based on Definition 1, the false positive rate is given by

$$p_f = \pi_{\bar{m}}. \tag{24}$$

B. Average Detection Rate

Suppose the target node starts to misbehave from step 0 on and assume that the associated Markov chain $\{X_k\}$ for the normal case before step 0 has reached its steady state $\boldsymbol{\pi}$. Note that, with the target node misbehaving (i.e., using different CWmin and/or AIFSN), the MAC model changes. Hence, we add superscript * to the variables defined in previous sections to distinguish the case that the target node is misbehaving from the normal case. Since whether the target node misbehaves or not is not pre-known to the detector, it shall assume that the target node is well-behaving and still use $\bar{s}$ to update its state. Thus, the support of X_k remains the same as above. The only difference lies in the probability of $I_k = 1$, which in turn changes the probability transition matrix from $\mathbf{P}(s)$ to $\mathbf{P}^* = \mathbf{P}(s^*)$.

Then, starting at $\boldsymbol{x}_0 = \boldsymbol{\pi}$, the Markov chain associated with the $\{X_k\}$ evolves with $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k\mathbf{P}^*$. By definition, the average detection rate in time D can be calculated as

$$\begin{aligned}p_d(D) &= 1 - \prod_{k=1}^{\lfloor \frac{D}{T^*} \rfloor} (1 - \mathbb{P}[X_k = \bar{m}\sigma]) \\ &= 1 - \prod_{k=1}^{\lfloor \frac{D}{T^*} \rfloor} (1 - x_{\bar{m},k})\,,\end{aligned} \tag{25}$$

where $\lfloor \frac{D}{T^*} \rfloor$ is the average number of steps in time D and $x_{\bar{m},k}$ is the last element of $\boldsymbol{x}_k$.

V. Performance Evaluation

Consider a WLAN with one AP and 15 nodes locating close to each other so that they can hear each other's transmissions. The nodes are divided into three priority classes. In class 1, there are $n_1 = 6$ nodes each of which uses MAC parameters $W_1 = 15$, $\text{CWmax}_1 = 1023$ and $\text{AIFSN}_1 = 7$. For the other two classes, we set $n_2 = 6$, $W_2 = 15$, $\text{AIFSN}_2 = 3$, $n_3 = 3$, $W_3 = 7$, and $\text{AIFSN}_3 = 2$. CWmax is fixed at 1023 for all the nodes. There is one node (the target node) in class 2 misbehaves.

To evaluate the false positive rate p_f of the HS detector, we consider the case that the target node well-behaves. As shown in Fig. 2, p_f decreases as the detection threshold h increases. This is simply because the higher h is, the less opportunity that the detector state X_k will hit its maximal value (i.e., $\bar{m}\sigma$ as in (18)). The figure also shows that the numerical solution error of the analytical model, as indicated by ϵ in (13), has an impact on the rate p_f: a smaller error can result in lower false positive rate. However, since the second and third curves are very close, we can see that a precision of $\frac{1}{60}$ is enough to deliver satisfactory results.

We then evaluate the detection rate of the HS detector by considering the misbehaving node with various misbehaving strategies. Fig. 3(a) shows the average detection rate $p_d(D)$ under different misbehaving intensities, where we fix $D = 100$ and $h = 2.5$. As the misbehavior is intensified (i.e., the target node uses a smaller AIFSN and/or CWmin), more received packets are from the target node. Hence, the detector state

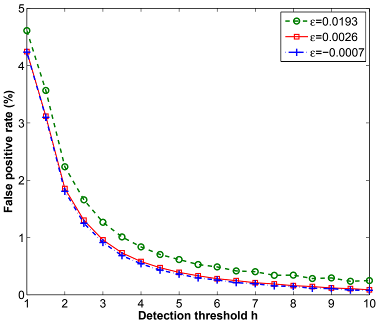


Fig. 2. False positive rate p_f under various detection thresholds. In the figure, the error $\epsilon = 0.0139, 0.0026$ and -0.0007 correspond to that $\sigma = \frac{1}{10}, \frac{1}{60}$ and $\frac{1}{100}$, respectively.

increases more frequently and is more likely to hit its maximal value. As a result, the average detection rate increases, which is clearly depicted in this figure.

Fig. 3(b) shows the performance of the HS detector associated with the target node under different D and h, where the misbehaving strategy is CWmin $= 4$ and AIFSN $= 0$. We can see that, in all cases, the detector becomes more reliable with a higher detection rate as the detection window gets longer. The misbehavior will be captured almost surely when D is larger than 80. However, a larger D indicates a longer detection delay. In this sense, we should keep D small in order to detect real-time misbehavior. For the similar reason as we discussed above about Fig. 2, the higher the detection threshold is, the lower the average detection rate will be achieved. However, since p_f and $p_d(D)$ are two conflict objectives, this figure suggests to carefully choose h and D to balance them.

VI. Conclusion

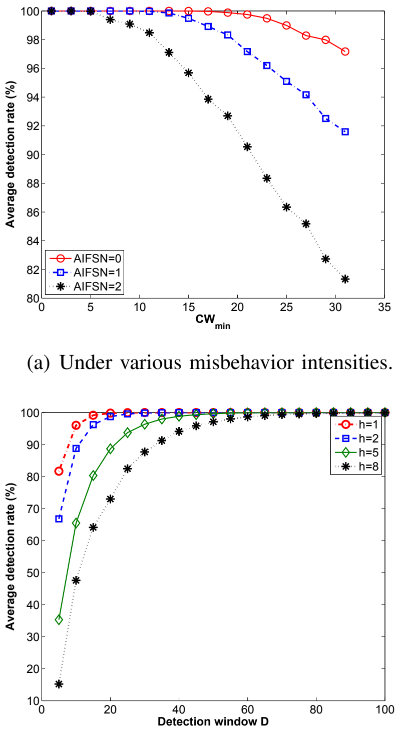
We have investigated the problem of misbehavior detection in IEEE 802.11e based networks where the nodes are able to choose different priority levels and different MAC parameters. We presented a mathematical model of the percentage of resource sharing of each node, based on which we proposed a hybrid-share detector. Theoretical performance of the detector in terms of false positive rate and average detection rate had been analyzed. Through numerical results, we demonstrated that the false positive rate is sensitive to the detection threshold but tolerable to the error involved in computing the MAC model. The results also indicate that our analysis can help choose proper detection threshold and window to meet real-time requirement while balancing false positive rate and average detection rate.

Acknowledgement

This work was supported in part by the NSF under grants CNS-1117687 and CNS-1320736.

References

[1] Q. Zhao, D. H. Tsang, and T. Sakurai, "A scalable and accurate nonsaturated IEEE 802.11e EDCA model for an arbitrary buffer size," *IEEE Trans. Mobile Computing*, vol. 12, no. 12, pp. 2455–2469, 2013.



(a) Under various misbehavior intensities.

(b) With different detection window D.

Fig. 3. Average detection rate $p_d(D)$.

[2] N. Chendeb Taher, Y. Ghamri Doudane, B. El Hassan, and N. Agoulmine, "Towards voice/video application support in 802.11e WLANs: A model-based admission control algorithm," *Computer Communications*, vol. 39, pp. 41–53, 2014.

[3] IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service enhancements," 2005.

[4] A. Lopez Toledo and X. Wang, "A robust Kolmogorov-Smirnov detector for misbehavior in IEEE 802.11 DCF," in *Proc. IEEE ICC*, 2007, pp. 1564–1569.

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

[6] G. Bianchi, I. Tinnirello, and L. Scalia, "Understanding 802.11e contention-based prioritization mechanisms and their coexistence with legacy 802.11 stations," *IEEE Network*, vol. 19, no. 4, pp. 28–34, 2005.

[7] S. Szott, M. Natkaniec, and R. Canonico, "Detecting backoff misbehaviour in IEEE 802.11 EDCA," *European Transactions on Telecommunications*, vol. 22, no. 1, pp. 31–34, 2011.

[8] P. Serrano, A. Banchs, V. Targon, and J. Kukielka, "Detecting selfish configurations in 802.11 WLANs," *IEEE Communications Letters*, vol. 14, no. 2, pp. 142–144, 2010.

[9] Y. W. Ahn, J. Baek, A. M. K. Cheng, P. S. Fisher, and M. Jo, "A fair transmission opportunity by detecting and punishing the malicious wireless stations in IEEE 802.11e EDCA network," *IEEE Systems Journal*, vol. 5, no. 4, pp. 486–494, 2011.

[10] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11 e wireless LANs," *IEEE Trans. Wireless Communications*, vol. 4, no. 4, pp. 1506–1515, 2005.

[11] K. Kosek-Szott, M. Natkaniec, and A. R. Pach, "A simple but accurate throughput model for IEEE 802.11 EDCA in saturation and non-saturation conditions," *Computer Networks*, vol. 55, no. 3, pp. 622–635, 2011.

[12] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.