Real-Time Misbehavior Detection in IEEE 802.11 Based Wireless Networks: An Analytical Approach

Jin Tang, *Student Member, IEEE,* Yu Cheng, *Senior Member, IEEE,*
and Weihua Zhuang, *Fellow, IEEE*

Abstract—The distributed nature of the CSMA/CA based wireless protocols, e.g., the IEEE 802.11 distributed coordinated function (DCF), allows malicious nodes to deliberately manipulate their backoff parameters and thus unfairly gain a large share of the network throughput. In this paper, we first design a real-time backoff misbehavior detector, termed as the *fair share detector* (FS detector), which exploits the non-parametric cumulative sum (CUSUM) test to quickly find a selfish malicious node without any *a priori* knowledge of the statistics of the selfish misbehavior. While most of the existing schemes for selfish misbehavior detection depend on heuristic parameter configuration and experimental performance evaluation, we develop a Markov chain based analytical model to systematically study the performance of the FS detector in real-time backoff misbehavior detection. Based on the analytical model, we can quantitatively compute the system configuration parameters for guaranteed performance in terms of average false positive rate, average detection delay and missed detection ratio under a detection delay constraint. We present thorough simulation results to confirm the accuracy of our theoretical analysis as well as demonstrate the performance of the developed FS detector.

Index Terms—Selfish misbehavior, real-time detection, IEEE 802.11, CUSUM test, Markov chain model.

1 Introduction

The IEEE 802.11 based wireless local area networks (WLANs) have been widely deployed over recent years due to their high-speed access, easy-to-use features and economical advantages. To resolve the contention issue among the multiple participating nodes, 802.11 employs the carrier sense multiple access/collision avoidance (CSMA/CA) protocol to ensure that each node gets a reasonably fair share of the network. This is particularly the case for the distributed cooperation function (DCF) of 802.11, where every node accesses the network in a cooperative manner and randomly delays transmissions to avoid collisions by following a common backoff rule [1]. However, in such a distributed environment without a centralized controller, a malicious node may deliberately choose a smaller backoff timer and selfishly gain an unfair share of the network throughput at the expenses of other normal nodes' channel access opportunities. Moreover, only to make things worse, the easily available programmable and reconfigurable wireless network devices nowadays [2], [3] make the backoff misbehavior much more feasible.

To efficiently detect the backoff misbehavior, a detection scheme needs to address the two main correlated challenges: 1) *unknown misbehavior strategy*, 2) *real-time detection of the misbehavior*. For the first challenge, since a malicious node can first behave as a normal node and then manipulate its backoff timer to a random small value at any time, we have no way to know the misbehavior strategy *a priori*. For the second, the misbehavior needs to be detected in real time and we can then isolate the malicious node to prevent it from bringing more harm to the network as soon as possible. The existing solutions either can not address both issues at the same time [4], [5], [6], [7], or require modifications to the 802.11 protocols [8], [9].

Addressing the challenges, in this paper we first design a real-time backoff misbehavior detector, termed as the *fair share detector* (FS detector), which exploits the non-parametric cumulative sum (CUSUM) test to quickly find a selfish malicious node without any *a priori* knowledge of the statistics of the selfish misbehavior. The work in [3] develops a robust detector for backoff misbehavior detection based on the Kolmogorov-Smirnov (K-S) test, without *a priori* knowledge of the misbehavior strategy either. The detector resorts to estimating the collision probability of a transmission to establish the distribution of the idle time between two consecutive successful transmissions from a tagged node. The collision estimation is however done with an approximate method for short detection delay; such an approximation in

- *J. Tang and Y. Cheng are with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL 60616. E-mail: {jtang9, cheng}@iit.edu.*
- *W. Zhuang is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. E-mail: wzhuang@uwaterloo.ca.*

This work was supported by NSF grant CNS-1117687. A preliminary version [19] of this paper was presented at IEEE INFOCOM 2011.

fact negatively impacts the performance in both false positive rate and detection delay, to be discussed in detail in Section 7. In our preliminary work [19], we adopt the non-parametric CUSUM test for the backoff misbehavior detection as well. The detector in [19] directly counts the number of successful transmissions of a tagged node within an observation window to obtain a sample. The observation window needs to linearly increase with the number of nodes in the network to fairly count transmissions from each node, which as a result will increase the detection delay. The FS detector newly developed in this paper takes each successful transmission over the network as its observation sample. Such a sampling method is independent of the network size and turns out to result in good performance in both false positive rate and detection delay. Also, the FS detector does not require any modification to the protocols, and can be implemented by any node assuming the role of the detection agent that monitors the network.

Another significant open research issue regarding the selfish misbehavior detection is that most of the existing detection schemes depend on heuristic parameter configuration and experimental performance evaluation [3], [4], [10], [11]. Such a heuristic approach largely limits the flexibility and robustness of the detection scheme; a change of the operation context could trigger the retraining of the configuration parameters by experimenting over a large set of data traces and the performance under those heuristic parameters is not theoretically provable.

To address the issue, in this paper we further develop an analytical model for the FS detector, which can provide quantitative performance analysis and theoretical guidance on system parameter configuration. Specifically, we use a discrete-time Markov chain to model the behavior of the detector, because the detector's next state depends only on its current value and the coming observation sample. This Markov chain based model enables us to conduct rigorous quantitative analysis of the FS detector on three fundamental metrics: *average false positive rate*, *average detection delay*, and *missed detection ratio*, and further compute the system configuration for guaranteed performance. In particular, the Markov chain modeling the FS detector takes different transition probabilities under the normal traffic condition and under the abnormal condition with misbehaving nodes present, respectively. The Markov chain obtained from the normal traffic condition can be used to directly calculate the average false positive rate and also provide the initial states for misbehavior analysis. Based on these initial states, we can then use the Markov chain under the abnormal conditions to analyze the average detection delay and the missed detection ratio. Note that the missed detection ratio is not often considered in the context of CUSUM test due to its "non-stop until detection" property. In this paper, we examine a *missed detection ratio under a detection delay constraint*, which is of importance regarding real-time detection.

In summary, the main contributions of the paper come in four aspects: 1) We develop an effective detector for real-time misbehavior detection in 802.11 based wireless networks. 2) We develop a discrete-time Markov chain based model to characterize the detection system. 3) We utilize the model to conduct rigorous quantitative analysis of the detector and guide the system configuration for guaranteed performance. 4) We provide analytical and simulation results to confirm the accuracy of our theoretical analysis, and demonstrate the robust performance of the developed FS detector under varying network size, against the short-term unfairness, and in the situation when both UDP and TCP traffic exists.

The rest of the paper is organized as follows. Section 2 reviews more related work. Section 3 describes the system model. In Section 4, we present the detector design. Section 5 develops the Markov chain based analytical model, and Section 6 gives the theoretical performance analysis based on the Markov chain model. Section 7 presents the simulation results. Section 8 discusses how to extend our analytical model to address the case of multiple malicious nodes as well as misbehavior beyond backoff timer manipulation. Section 9 concludes the paper.

2 Related Work

The problem of detecting backoff misbehavior over the 802.11-based medium access control (MAC) protocol has been widely studied in the literature. In [8], [9], a modification to the 802.11 protocol is proposed to facilitate the misbehavior detection, where the receiver assigns a backoff timer for the sender. If the number of idle slots between consecutive transmissions from the sender does not comply with the assigned backoff timer, the receiver may label the sender as a selfish node. Modification to the 802.11 protocol and reliance on a trustworthy receiver are the main limitations of the work.

Another approach to deal with the backoff misbehavior is to develop protocols based on the game-theoretic techniques [14], [15], [16]. The goal is to encourage all the nodes to reach a Nash equilibrium. As a result, a malicious node is not able to gain an unfair share compared to well-behaved nodes and thus discouraged from the misbehavior. However, this category of approaches assume that all the nodes are willing to deviate from the protocol when necessary, and the standard protocol needs to be modified. A heuristic sequence of conditions are proposed in [17], [18] to test multiple misbehavior options over the 802.11 MAC based on simple numerical comparisons. This approach, named DOMINO, preserves its advantage of simplicity and easiness of implementation, and still demonstrates its efficiency when dealing with a wide range of 802.11 MAC misbehavior. However, the heuristic nature of the approach limits its applications to specific scenarios.

The sequential probability ratio test (SPRT) method is used in [5], [6], [7] to detect the 802.11 backoff misbehavior. The detection decision is made when a random walk of the likelihood ratio of observations (given two hypotheses) rises to be larger than an upper threshold. The main advantage of SPRT is that it can reach decision very fast, given the complete knowledge of both normal behavior and backoff misbehavior strategy [20]. However, in a realistic setting, the strategy of malicious nodes is hard to know in advance. Further, the existing work normally assumes that the backoff timer of each node is observable, which is again hard to achieve in practice because the transmission attempts involved in a collision are impossible to be distinguished. In our design, we monitor the successful transmission of the tagged node as the observation measurement.

The authors in [3], [4] utilize the Kolmogorov-Smirnov (K-S) significance test for backoff misbehavior detection. This test is able to make the decision by measuring the distribution of the idle time between consecutive successful transmissions from a tagged node and comparing it to the normal backoff behavior. The detection method in [3], [4] requires estimation of the collision probability of a packet transmitted. However, an inaccurate simplification there is to consider that packets from the misbehaving node and those from the normal nodes have the same collision probability. Such inaccuracy impacts both the performance of false positive rate and detection delay, to be demonstrated in Section 7. Furthermore, as a batch test method, the K-S statistic has its own drawback. Fixed-size data samples are needed to perform the test each time, which makes real-time detection difficult.

In our preliminary work [19], we adopt the non-parametric CUSUM test [12] for the backoff misbehavior detection, which has the advantages of both real-time detection and no requirement of *a priori* knowledge of the misbehavior strategy. The detector in [19] directly counts the number of successful transmissions from a tagged node within an observation window[1] to get a sample. Although such a sampling method is easy for implementation, the observation window needs to linearly increase with the number of nodes in the network to fairly count transmissions from each node, which as a result will increase the detection delay. In this paper, we develop the new FS detector, which takes every successful transmission over the network as a sample to trigger its state change. Such a sampling method is independent of the network size and turns out to result in good performance in both false positive rate and detection delay, as to be demonstrated later in this paper.

A common research issue among most of the existing schemes for misbehavior detection is their dependency on heuristic parameter configuration and experimental performance evaluation, which largely limits the flexibility and robustness of the schemes. To address this issue, in [19], we propose a Markov chain based analytical model to theoretically analyze the detection performance and quantitatively configure the system parameters. In this paper, we develop the analytical model according to the newly proposed FS detector. Our analysis demonstrates performance improvement of the FS detector in real-time misbehavior detection over the original detector in [19]. Also, we demonstrate the robustness of the FS detector under varying network size, against the short-term unfairness, and in the situation when both UDP and TCP traffic exists.

3 System Model

3.1 IEEE 802.11 DCF

There are two major functions in the IEEE 802.11 protocols: the point coordination function (PCF) and the distributed coordination function (DCF). The PCF is a centralized function and is an optional feature in 802.11. Here, our focus is on the more widely used DCF protocol. In the DCF, every node contends for access to the wireless medium following the CSMA/CA function [1]. When a node attempts to transmit a packet, it needs to sense the medium idle for a specified time. The time is divided into slots of constant duration, and a node can only transmit at the beginning of a slot time. If the medium is not idle, the node will enter a backoff stage and defer the transmission according to a timer before attempting the next transmission. This backoff timer is a random value uniformly selected from the range $[0, CW_{min} - 1]$, where CW_{min} is called the minimum contention window with a standard value of 32. The timer will decrease if the medium is continuously sensed idle and freeze whenever the medium is sensed busy. After the timer reaches 0, the node will attempt another transmission. Each unsuccessful transmission will double the contention window size until it reaches the maximum value $CW_{max} = 2^m CW_{min}$, where m is called the maximum backoff stage with a standard value of 5. This operation is also referred to as the *binary exponential backoff* scheme. After a successful transmission, the node will reset the contention window to CW_{min} and continue sensing the medium if it has more packets to transmit.

3.2 Backoff Misbehavior in IEEE 802.11 DCF

As a distributed protocol, the DCF assumes that every node in the network operates in accordance with the standard to obtain a fair share of the wireless medium. Since there is no central controlling unit which assigns the backoff timer for each node, a malicious node can

[1] An observation window is defined as a certain number of consecutive successful transmissions over the whole network [19].

continuously choose a small backoff timer and then gain significant advantages in channel access probability over others. Moreover, because the increased transmission probability of the malicious node causes more collisions, normal nodes are forced to further exponentially defer their transmissions as they operate according to the protocol. The backoff misbehavior can drastically decrease the transmission probability of normal nodes and subsequently severely reduce their throughput. In an extreme case where a malicious node sets its own backoff timer to a very small constant value, it will lead to denial of service (DoS) of the whole network. Thus, a detection scheme capable of quickly identifying the misbehaving malicious node is highly desired.

4 Detector Design

We consider a saturated situation that a node always has data to send when the channel is available. Although a network in practice is not always saturated, the saturated scenario is of meaningful concern in the context of selfish misbehaving. If the network is lightly loaded, a misbehaving node will not impact much the throughput of normal ones. When the network is close to full utilization, the data buffer in every node has a very small probability to be empty, where the saturated model is a good approximation.

4.1 The Observation Measure

Consider a tagged node v. In our detection system, the *observation measure* is an indicator of whether a successful transmission over the network belongs to the tagged node v, denoted as I^v. We take the popular modeling technique [1] that each node independently accesses an idle channel for transmission with a probability determined by its contention window size. If we use q_s^v to denote the probability that a successful transmission over the network is from node v, the probability distribution of I^v is given by

$$P\{I^v = k\} = \begin{cases} q_s^v & \text{if} \quad k = 1, \\ 1 - q_s^v & \text{if} \quad k = 0. \end{cases} \tag{1}$$

In a normal situation that every node follows the 802.11 DCF standard, it can be seen that $q_s^v = \frac{1}{N}$ due to fair channel sharing, given N nodes in the network. If node v is a malicious node taking a smaller contention window size, it will achieve a q_s^v larger than $\frac{1}{N}$ and thus a larger portion of the network throughput. In Section 6, we will present how to calculate q_s^v given the contention window size. The distribution of I^v in (1) is the basis to establish our analytical model.

Remark In an 802.11 network, a node that has just accomplished a successful transmission will have advantages in grabbing the channel for next transmission in a short period [13]. This is referred to as *short-term unfairness* and is inherent to the 802.11 backoff mechanism. Such an issue implies correlations among the channel accesses, which may impact the accuracy of (1) to model the successful transmission of the tagged node based on the assumption of independent channel access. In [19], we apply a shuffling mechanism to observation samples to mitigate the impact of short-term unfairness. In Section 7, we provide detailed analysis to show that the FS detector is inherently robust against short-term unfairness, and the detection based on (1) does give accurate decisions. The fairness issue also exists when both user datagram protocol (UDP) and transmission control protocol (TCP) traffic flows exist in the network, where the TCP traffic tends to be overwhelmed by UDP traffic due to its congestion control mechanism. In Section 7, we also discuss how to apply the FS detector for robust performance when both UDP and TCP traffic flows exist in the network.

4.2 Fair Share Detector

Let $\{I_n, n = 0, 1, ...\}$ be the sequence of sample values of I^v, observed each time a successful transmission appears on the channel. Here, we drop the superscript v for easier presentation considering the clear context. There are N nodes and one access point (AP) in the network. Suppose that the initial value of the detector is $X_0 = 0$. If a successful transmission upon the nth observation is from the tagged node, i.e., $I_n = 1$, the detector X_n increases by $N - 1$; otherwise, $I_n = 0$, and X_n decreases by 1 until it reaches 0. The intuition of this design is as follows: In the normal situation, each node roughly takes turn to transmit; the increase of X_n caused by one successful transmission from the tagged node can then be equally offset by the successful transmissions from other $N-1$ non-tagged nodes. Thus, the detector X_n will fluctuate around a low value close to zero in the normal situation. On the other hand, when the tagged node turns to misbehave and obtain more chances to transmit, it is not difficult to see that X_n is going to quickly accumulate to a large positive value.

The behavior of the FS detector can be mathematically described as

$$\begin{aligned} X_{n+1} &= (X_n + (NI_n - 1))^+ \\ X_0 &= 0 \end{aligned} \tag{2}$$

where $(x)^+ = x$ if $x \geq 0$ or 0 otherwise. We can see that (2) is actually in the form of a non-parametric CUSUM detector [12]. Let h be the detection threshold. The decision rule of the detector in step n is

$$\delta_n = \begin{cases} 1 & \text{if} \quad X_n \geq h \\ 0 & \text{if} \quad X_n < h \end{cases} \tag{3}$$

where δ_n is also an indicator function of whether the detection event happens or not. The detector value X_n will be reset back to 0 as soon as it exceeds the threshold and the detection procedure starts over again.

5 Markov Chain Based Model

Consider the sequence $\{X_n\}$ as a discrete random process, which takes values from a finite set $A = \{0, 1, 2, ..., h\}$. The process is said to be in state i at time n if $X_n = i$ with $i \in A$. The state transition happens when a successful transmission over the network is observed. According to (2), the next state X_{n+1} depends only on the current state X_n and is independent of any other previous states, where the transition probability is

$$P_{ij} = P\{X_{n+1} = j | X_n = i\} \quad i, j \in A. \tag{4}$$

Thus the random process $\{X_n\}$ satisfies the Markov property and can be modeled as a discrete-time Markov chain.

Given the decision threshold h, the Markov chain is then described by a $(h + 1) \times (h + 1)$ transition probability matrix as

$$\mathbf{P} = \begin{pmatrix} P_{00} & P_{01} & P_{02} & \dots & P_{0h} \\ P_{10} & P_{11} & P_{12} & \dots & P_{1h} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{h0} & P_{h1} & P_{h2} & \dots & P_{hh} \end{pmatrix}.$$

This transition probability matrix can be divided into three distinct groups based on the operation of the FS detector.

Group 1 consists of P_{ij} for $i = 0$ and $j \in [0, h]$, with values

$$P_{0j} = \begin{cases} P\{I_n = 0\} & \text{if } j = 0, \\ P\{I_n = 1\} & \text{if } j = N-1 \text{ and } N-1 \leq h, \\ P\{I_n = 1\} & \text{if } j = h \text{ and } N-1 > h, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

This group is related to the transitions from state 0 to other states. According to the state transition equation (2), the detector variable X_n jumps out of state 0 only when the observed successful transmission is from the tagged node, that is, $I_n = 1$. Further, X_n makes a transition to either $N - 1$ or h depending on whether $N - 1$ is greater than h or not. Note that the state h in fact incorporates all possible states $X_n \geq h$, as the detector will raise an alarm when the state hits h.

Group 2 consists of P_{ij} for $i \in [1, h-1]$ and $j \in [0, h]$, with values

$$P_{ij} = \begin{cases} P\{I_n = 0\} & \text{if } j = i-1, \\ P\{I_n = 1\} & \text{if } j = i+N-1 \text{ and} \\ & \quad i+N-1 \leq h, \\ P\{I_n = 1\} & \text{if } j = h \text{ and} \\ & \quad i+N-1 > h, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

This group describes the typical behavior of the detector. The state can transit to left (i.e., to a smaller value) when $I_n = 0$ or to right (i.e., to a larger value) when $I_n = 1$, according to the state transition equation (2).

Finally, group 3 consists of P_{ij} for $i = h$ and $j \in [0, h]$, with values

$$P_{hj} = \begin{cases} 1 & \text{if } j = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This group is related to the transitions out of state h. Since the detector value will be reset to 0 as soon as it reaches or exceeds h, $P_{h0} = 1$.

6 Theoretical Performance Analysis

In this section, we conduct theoretical performance analysis of the FS detector based on the Markov chain model in terms of the three fundamental metrics to change detection: average false positive rate, average detection delay, and missed detection ratio under a detection delay bound. Then we show how we can configure the system parameters to achieve guaranteed performance. We also analyze the performance of the detector when the number of nodes is varying, which is a typical scenario in the 802.11 based wireless networks.

6.1 Average False Positive Rate

The average false positive rate P_{fp} is the rate that the detector value X_n hits state h given the fact that there is no node in the network misbehaving. According to the theory on the discrete-time Markov chain, such a rate is equal to the steady-state probability that the Markov chain describing the FS detector stays at h in the normal condition.

In the normal condition with a fair share of the channel access, we have $q_s^v = \frac{1}{N}$ for a tagged node. We can calculate the distribution of I_n according to (1), and further obtain the transition probabilities matrix $\mathbf{P}$ according to (5)−(7).

Let $(\pi_0, ..., \pi_h)$ denote the steady state probabilities of the Markov chain, which can be solved from the equations

$$\pi_j = \sum_{i=0}^{h} \pi_i P_{ij}, \qquad j \in \{0, ..., h\}, \tag{8}$$

$$\sum_{j=0}^{h} \pi_j = 1. \tag{9}$$

Then we can get the average false positive rate

$$P_{fp} = \pi_h. \tag{10}$$

The analytical result (10) allows us to numerically examine the impact of the fundamental parameter h on the average false positive rate P_{fp} of the FS detector. As an example, we compute the results for a network with $N = 10$ nodes, and the results are illustrated in Fig. 1. From the figure, we can observe that a larger h yields a smaller false positive rate, as expected.

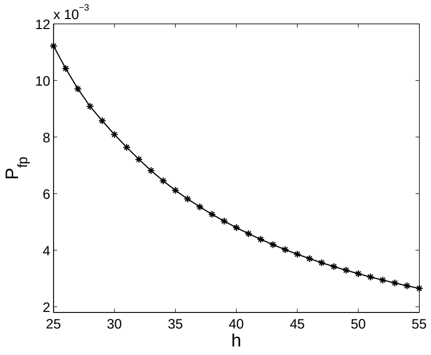

Fig. 1. Average false positive rate.

6.2 Average Detection Delay

In this subsection, we analyze the average detection delay denoted as $E[T_D]$, which is the average number of samples observed from the moment that the tagged node starts to misbehave until the misbehavior is detected. With the Markov chain under the abnormal condition (abnormal Markov chain), $E[T_D]$ can be computed as the expected number of transitions required for the state variable to hit state h, starting from the moment when the misbehavior starts. To carry out the analysis, we need to find the transition probabilities of the abnormal Markov chain and determine the initial state of the FS detector when the misbehavior starts.

6.2.1 Transition Probabilities under the Misbehavior

We consider a network consisting of two classes of nodes. Class 1 includes the one misbehaving node with a small minimum contention window CW_{min} denoted as W^1, and class 0 includes all the normal nodes with the standard minimum contention window denoted as W^0. According to the classic modeling approach for the 802.11 DCF [1], we consider that each node independently accesses an idle channel for transmission. Let p_t^i denote the probability that a class i ($i \in 0, 1$) node transmits at a random time slot and p_c^i denote the collision probability of a class i node. Also recall that N is the number of nodes and m is the maximum backoff stage. According to [1], we have the following equations:

$$\begin{cases} p_t^0 = \dfrac{2(1-2p_c^0)}{(1-2p_c^0)(W^0+1)+p_c^0W^0(1-(2p_c^0)^m)} \\ p_t^1 = \dfrac{2(1-2p_c^1)}{(1-2p_c^1)(W^1+1)+p_c^1W^1(1-(2p_c^1)^m)} \\ p_c^0 = 1-(1-p_t^1)(1-p_t^0)^{N-2} \\ p_c^1 = 1-(1-p_t^0)^{N-1} \end{cases} \quad (11)$$

from which the four parameters p_t^0, p_t^1, p_c^0 and p_c^1 can be solved.

Note that a node can get a successful transmission under the circumstance that there is no collision while the node transmits. Thus from the solutions of (11), we can obtain the probability that a node gets a successful transmission at a random time slot:

$$p_s^0 = p_t^0(1-p_c^0), \quad (12)$$
$$p_s^1 = p_t^1(1-p_c^1). \quad (13)$$

We can then calculate the probability $\hat{q}_s$ that a successful transmission over the network is from the malicious node as (14):

$$\hat{q}_s = \frac{p_s^1}{p_s^1+(N-1)p_s^0}. \quad (14)$$

Using $\hat{q}_s$ in (1), we can obtain the distribution of I_n for the misbehaving node; using this distribution in (5)−(7), we can then compute the transition probability matrix $\hat{\mathbf{P}}$ for the abnormal Markov chain.

It is worth noting that although we only include two classes of nodes in the above analysis, the model of (11) to (14) can be easily extended to cases where multiple classes of misbehaving nodes with different intensities of misbehavior exist. This will enable us to analyze much more complicated misbehaving scenarios. We will discuss this issue in Section 8.

6.2.2 Initial States

A natural thought of the initial state of X_n is 0 when the misbehavior starts. However, this may not be the case; before a malicious node starts to misbehave, it can behave like a normal node and still affect X_n. Thus X_n can be initially at any state following the normal Markov chain except for state h, as we do not consider an already "alarmed" state as an initial state.

We can calculate the steady state probabilities of the normal Markov chain according to (8) and (9). Since we are interested in detection starting from an unalarmed state, under such a constraint the conditional initial state probabilities should be

$$\pi_i' = \frac{\pi_i}{\sum_{i=0}^{h-1}\pi_i} \quad \text{for } i \in \{0,...,h-1\}. \quad (15)$$

6.2.3 Average Detection Delay

As we have various initial states, the average detection delay $E[T_D]$ should be calculated as the weighted average of the expected numbers of transitions from every initial state to state h based on the transition probability matrix $\hat{\mathbf{P}}$ for the abnormal Markov chain.

Let μ_{ih}, $i \in [0, h-1]$, denote the expected number of transitions for state i to state h. According to [21], the values of μ_{ih} can be solved from the equations

$$\mu_{ih} = 1 + \sum_{r \neq h} \hat{P}_{ir}\mu_{rh}, \qquad i \in \{0,...,h-1\} \quad (16)$$

where $\hat{P}_{ir}$ is the transition probability from state i to r of $\hat{\mathbf{P}}$. Based on the solutions of (15) and (16), we can obtain the average detection delay $E[T_D]$ as

$$E[T_D] = \sum_{i=0}^{h-1}\pi_i'\mu_{ih}. \quad (17)$$

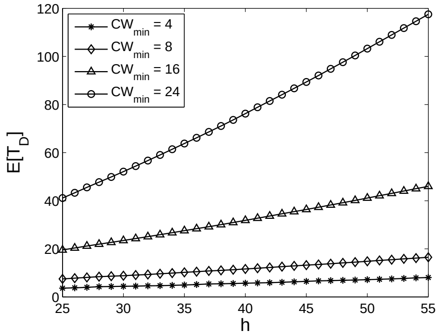

Fig. 2. Average detection delay.

The analytical result (17) allows us to numerically examine the impact of h on the average detection delay $E[T_D]$. As an example, we compute the results for a network with $N = 10$ nodes, and the results are shown in Fig. 2 with four misbehaving intensities $CW_{min} = 4, 8, 16$ and 24, respectively. As we expect, the curves in Fig. 2 show that a more intense misbehavior leads to a shorter detection delay. Also, we observe that a smaller h yields better performance in average detection delay.

6.3 Missed Detection Ratio

In this subsection, we discuss the missed detection ratio, denoted as P_{md}. The FS detector exploits the non-parametric CUSUM test. The missed detection ratio is not often considered in the context of CUSUM test due to its "non-stop until detection" property. We however examine P_{md} under a given detection delay constraint D, which is of importance regarding real-time detection.

The detection event happens only when X_n hits state h. Thus the missed detection ratio P_{md} under the delay constraint D is the summation of the probabilities of X_n staying at a state other than h at time D. With the transition probability matrix $\hat{\mathbf{P}}$, the missed detection ratio can be computed in an iterative manner. Let the row vector $\vec{P}(j) = [P_0(j), \cdots, P_h(j)]$ denote the probabilities of the state variable at step j with $0 \leq j \leq D$. The computation starts from the initial states given in (15), setting

$$P_i(0) = \pi'_i \quad \text{for } i \in \{0, ..., h-1\}, \tag{18}$$

$$P_h(0) = 0. \tag{19}$$

At each transition step $j \in [0, D-1]$, the state probabilities are updated as

$$\vec{P}(j) = \vec{P}(j-1) \cdot \hat{\mathbf{P}}, \tag{20}$$

$$P_h(j) = 0. \tag{21}$$

At each step, $P_h(j)$ is set to 0 for next step computation because we are interested in the missed detection cases. The missed detection ratio under the delay bound constraint D can be obtained as

$$P_{md} = \sum_{i=0}^{h-1} P_i(D). \tag{22}$$

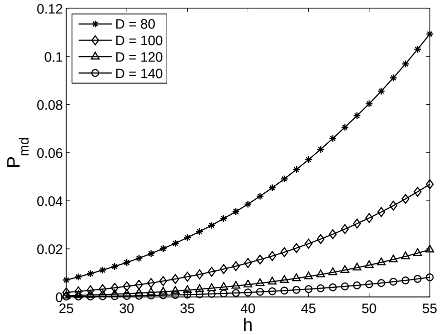

Fig. 3. Missed detection ratio.

Fig. 3 demonstrates the missed detection ratios P_{md} of our analysis under the delay constraints $D = 80, 100, 120$ and 140, respectively, for a misbehaving node with the moderate misbehavior of $CW_{min} = 16$. We observe that the larger the delay constraint is, the lower the missed detection ratio will be. In other words, the probability of detection increases with a cost of longer delay. Also, a smaller detection threshold h yields a lower missed detection ratio.

6.4 Configuration for Guaranteed Performance

The above theoretical analysis provides us a guideline to configure the system parameter h for guaranteed performance in a target scenario. For each performance metric, we can obtain the feasible ranges of h to satisfy the performance constraints. With the intersection of the parameter ranges under all the constraints, a proper configuration of h can be obtained to meet the performance requirements of all the metrics. Moreover, once we determine the configuration parameter, we can explicitly estimate the performance measures given a misbehaving scenario. In practice, as we do not have *a priori* knowledge of the misbehavior, the analytical model allows us to conservatively configure the system so that even the misbehavior with a low intensity can be detected with good performance. For example, if we select $h = 40$ for a network with $N = 10$, our analytical model indicates that, even for the moderate misbehavior with $CW_{min} = 16$, we can target a high level of performance with the average false positive rate of 0.005, the average detection delay of 31.8357 samples, and the missed detection ratio of 0.0141 with the delay constraint $D = 100$. In Section 7, we will use simulation results to demonstrate that our target performance measures are indeed achievable.

6.5 Detection with Network Size Change

In an 802.11 based wireless network, it is typical that nodes are mobile and thus the number of nodes (i.e., the network size) changes from time to time. The proposed FS detector is robust against such a scenario. As we directly include the number of nodes N in the detector design, when N changes, the detector can adjust and respond in real time.

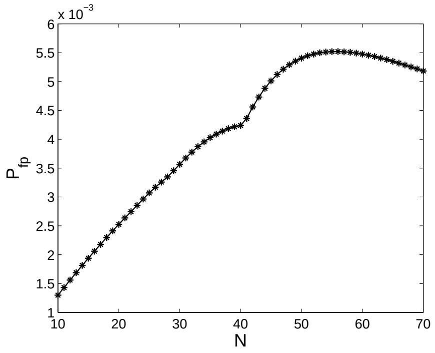

Fig. 4. Impact of network size change on average false positive rates at $h = 80$.

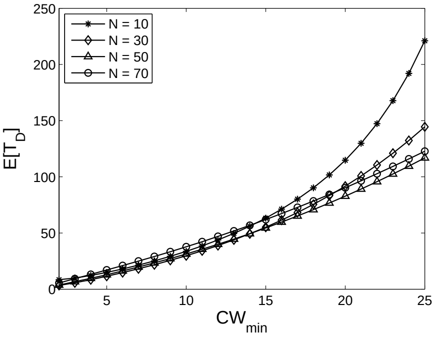

Fig. 5. Impact of network size change on average detection delays at $h = 80$.

Fig. 4 shows the average false positive rates P_{fp} of the detector versus the number of nodes N, at $h = 80$. The threshold h is intentionally set to be greater than the maximum number of nodes to avoid alarm being triggered by just one successful transmission from the tagged node. As shown in Fig. 4, there is a dent on the curve at $N = 40$ and P_{fp} has a sharper increase when N gets greater than 40. This is because, when $N \leq 40$, at least three or more consecutive successful transmissions from the tagged node are needed to drive X_n to h from an initial state of 0, raising a false alarm; however, when $41 \leq N \leq 70$, it will take only two consecutive transmissions to reach h, which largely increases the possibility of false positive. Furthermore, note that P_{fp} does not monotonically increase with N and has an upper bound of $P_{fp} = 0.0055$. The explanation is that, when the number of nodes contending for the channel becomes larger, the transmissions from a tagged node are more likely to be interrupted by transmissions from those non-tagged nodes, and the accumulation of the detector X_n will be more aggressively offset by such non-tagged nodes, thus resulting in a smaller P_{fp}. If the target performance of $P_{fp} \leq 0.0055$ is allowed, we can see that the configuration $h = 80$ satisfies the false positive performance requirement even when N changes dynamically in a wide range. Note that a typical 802.11 based wireless local area network covers up to tens of users.

Fixing $h = 80$, we now investigate the average detection delay $E[T_D]$ of the detector for different misbehavior intensities, indicated by the CW_{min} value of a misbehaving node, with results shown in Fig. 5. The misbehavior intensities with $CW_{min} > 25$ are not included in our discussion, as their effects are minimal. Practically, a misbehaving node needs to choose more intense misbehavior, e.g., $CW_{min} \leq 16$, to gain more benefits from the network throughput. From Fig. 5, we see that for misbehavior in this range, the change of N does not affect $E[T_D]$ much. The reason is that, when a misbehaving node grabs the channel, very likely it will consecutively send a certain number of packets, driving the detector to hit the threshold. For a smaller value of N, it may just take a couple of more samples for the detector to hit the threshold (note that each transmission from the tagged node increases the detector state by $N - 1$), which only slightly increases the detection delay. With less intense misbehavior ($16 < CW_{min} \leq 25$), we do observe obviously larger detection delays for a small N. The reason is that, when the misbehaving intensity is low, the accumulation procedure of X_n is more often to be offset by transmissions from those non-tagged normal nodes; for a small N, it will take even more samples from the misbehaving node to raise the alarm, leading to a longer detection delay.

It is noteworthy that the relationship between N and detection delay in Fig. 5 is not rigorously monotonic. Such phenomenon is due to two contradicting factors: Given the CW_{min}, the misbehaving node will get less transmissions when N gets larger and thus less chances to accumulate X_n, potentially increasing the detection delay; the increase of X_n (by a value of $N - 1$) caused by one transmission from the misbehaving node however becomes larger too, potentially decreasing the detection delay. In summary, the results in Fig. 5 again demonstrate that the FS detector with a fixed threshold (larger than N) has a robust performance for a typical misbehaving scenario, even when the number of nodes in the network changes.

In a situation where a fixed constraint on P_{fp} is imposed, we can dynamically calculate the h value corresponding to a certain N through the analytical model. Further, if we do the calculation beforehand and maintain a table of "h versus N" values under the given P_{fp} constraint, we can quickly adjust h as soon as changes on N are observed. Fig. 6 shows the average detection delays $E[T_D]$ of the detector for different misbehaving intensities, given a false positive constraint as $P_{fp} = 0.005$. Similar to Fig. 5, Fig. 6 shows that the detection delays under different N are similar when the misbehavior is very intense. Under a lower misbehaving intensity (i.e., a larger CW_{min}), the detection delays increase more obviously with the number of nodes, because a larger threshold h is required for a larger N to meet the false

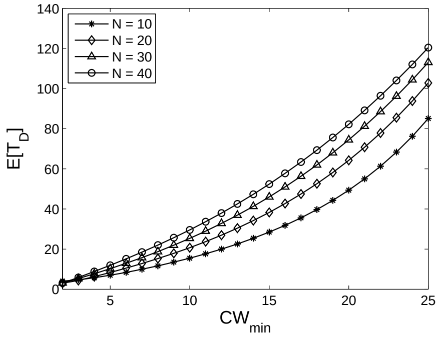

Fig. 6. Impact of network size change on average detection delays at $P_{fp} = 0.005$.

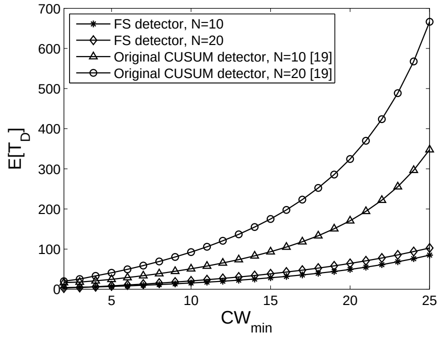

Fig. 7. Comparison with the original CUSUM detector in [19] at $P_{fp} = 0.005$.

positive requirement. However, the delay increase is not dramatic. Even for $CW_{min} = 25$ and $N = 40$, it only takes about 120 successful transmissions over the whole network to detect the misbehavior.

6.6 Comparison with the CUSUM Detector in [19]

In order to show how we have improved in real-time misbehavior detection, we compare the FS detector to the detector developed in our preliminary work [19], referred to as the "original CUSUM detector" for convenience. The observation measure of the original CUSUM detector is the number of successful transmissions of the tagged node in every M successful transmissions over the whole network. It means getting one observation sample for the original CUSUM detector requires M successful transmissions, whereas the FS detector will update state upon every successful transmission over the network. Also, M needs to be at least as large as the number of nodes N and linearly increase with N to fairly count transmissions from each node. Moreover, besides h, there is another parameter u in the original detector design, which is the upper bound of the observation measure's expectation. To determine a proper u, we need to take into account both the sample size M and the number of nodes N, adding the complexity of the detection system. In the FS detector, u is not present, which leads to one less parameter impacting the detection performance and thus makes parameter configuration much simpler.

Fig. 7 shows the average detection delays of the two detectors for different misbehavior intensities under the same false positive constraint of $P_{fp} = 0.005$. Here we consider the cases of $N = 10$ and $N = 20$. For the FS detector, given the P_{fp} and N, the threshold h can be determined from the analytical model. With h, the detection delay for a given misbehaving intensity can then be calculated and plotted in Fig. 7. We intentionally configure the original CUSUM detector for a small detection delay so the advantage of the FS detector can be demonstrated more convincingly. The sample size M for the original CUSUM detector is set to its minimum value N (i.e., 10 and 20 for the two cases considered, respectively) in order to minimize the impact of the observation window size on the detection delay. With such an observation window selection, on average one successful transmission from each node can be expected in each window, i.e., $u = 1$. Given the P_{fp}, N and u, the parameter h can then be determined from the analytical model in [19]. With h and u, the detection delay for a given misbehaving intensity with the original CUSUM detector can be calculated and plotted in Fig. 7.

As shown in Fig. 7, for the same N, the FS detector shows clear advantages over the original CUSUM detector, especially when the misbehavior becomes less intense. Observing the delays of the original CUSUM detector, we can see that the delays with $N = 20$ are roughly two times of the delays with $N = 10$ for almost all the misbehavior intensities. The fact clearly indicates the impact of the observation window size on detection delay in the original CUSUM detector. Another advantage of the FS detector is that its detection delay curves are quite flat against the misbehaving intensity and not much impacted by the network size N, showing very robust performance.

7 Simulation Results

7.1 Simulation Setup

We establish an 802.11 DCF based wireless network consisting of 10 competing nodes ($N = 10$) and an access point through ns-2 [22] simulation. We first consider that the network works under the saturated condition and every node sends packets with UDP towards the AP. Then we include the TCP traffic in our simulation to further analyze the performance of the FS detector in more general scenarios. The AP also acts as the detection agent which monitors the transmissions from every competing node with a separate FS detector. The nodes are located close enough to sense the transmissions from each other and thus avoid the hidden terminal problem. There is 1 misbehaving node among the 10 competing nodes, which accesses the wireless channel using the binary exponential backoff scheme but can manipulate its

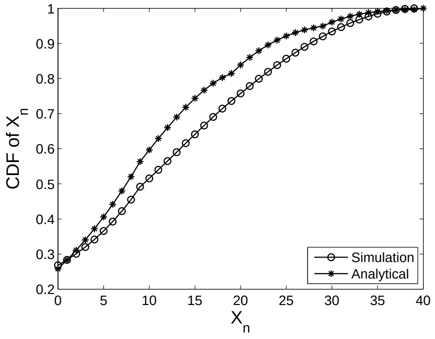

Fig. 8. CDF of X_n.

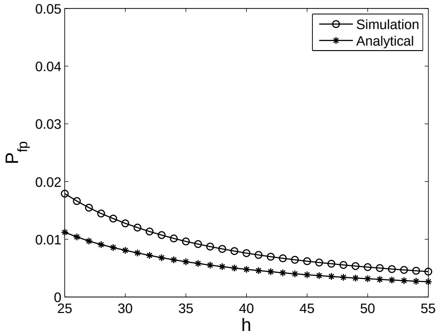

Fig. 9. Average false positive rate.

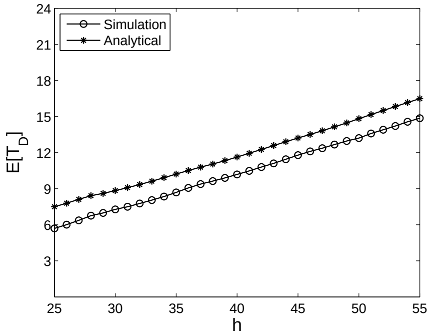

Fig. 10. Average detection delay with $CW_{min} = 8$.

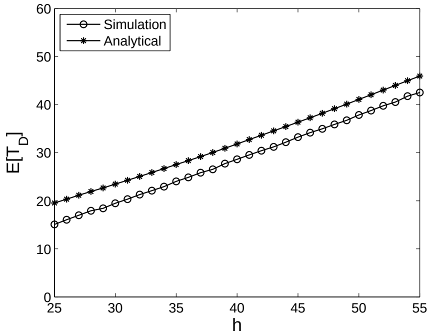

Fig. 11. Average detection delay with $CW_{min} = 16$.

minimum contention window CW_{min} to any value between 1 and 32.

Due to the conflicting nature of the three performance metrics (average false positive rate, average detection delay, and missed detection ratio), it can be difficult to find the system configuration parameter that achieves best performance at all fronts. Using our analytical model, we find that, for $N = 10$, setting the detection threshold $h = 40$ can achieve a good tradeoff among all the metrics (referring to Section 6.4). Therefore in our simulation, if not specified, we set $h = 40$ to further evaluate the performance of our detection.

7.2 Robustness against Short-Term Unfairness

In an 802.11 network, a node that has just accomplished a successful transmission will have advantages in grabbing the channel for next transmission in a short period [13]. This is referred to as *short-term unfairness* and is inherent to the 802.11 backoff mechanism. Such an issue implies correlations among the channel accesses, which impact the accuracy of the transition probability calculation based on the assumption of independent channel access. The system configuration based on an inaccurate model can lead to inaccurate detection results. In this section, we study how the short-term unfairness affects the performance of our detector.

We first examine the impact of short-term unfairness on the distribution of the detector X_n under the normal traffic condition. In Fig. 8, we present the simulation results of the cumulative distribution function (CDF) of X_n, compared with the analytical CDF. Note that even though the analytical results are based on the independent model of (1), the two curves are still close to each other. We then examine the average false positive rate P_{fp} versus h, comparing the analytical results with the simulation results in Fig. 9. Again, despite a bigger gap when h is smaller, the P_{fp} curve obtained from simulations still largely resembles the analytical one. The observations show that our FS detector is robust against the impact of short-term unfairness.

We then obtain the average detection delays $E[T_D]$ under different misbehaving intensities. Figs. 10 and 11 present both the simulation and analytical $E[T_D]$ curves versus h for $CW_{min} = 8$ and $CW_{min} = 16$, respectively. The closeness of the two curves in both cases again confirms the robustness of the FS detector against the short-term unfairness.

Technically, the FS detector by nature can mitigate the impact due to the short-term unfairness. In the normal situation, every node in the network has the same opportunity to experience a short period of advantages in transmissions. At a sampling moment, if the tagged node under observation is accessing the channel more aggressively due to short-term unfairness, it will increase the detector state value more aggressively according to (2), tending to be false positive. However, if at other sampling moments, those non-tagged nodes are accessing the channel more aggressively, it will in turn decrease the detector

TABLE 1
Comparison of Analytical and Simulation Results with $N = 10$, $h = 40$, $D = 100$

	P_{fp}	$E[T_D]$	P_{md}
Analysis	0.005	31.8357	0.0141
Simulation	0.0076	28.5744	0.0255

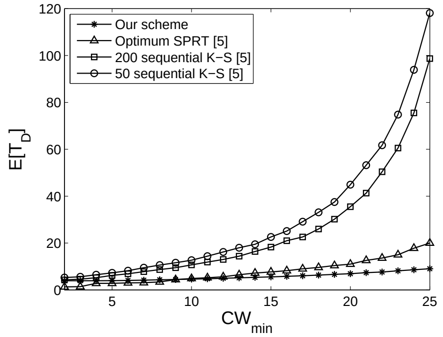

Fig. 12. Comparison with the detection schemes in [3].

state value more aggressively and mitigate the false positive effect. Therefore, as an aggregate effect, the FS detector only degrades slightly in the false positive performance. In the misbehaving situation, extra channel access (in addition to that resulting from the backoff misbehavior) due to the short-term unfairness effect in fact benefits the misbehavior detection, with the detector being driven to hit the threshold h sooner, as shown in Figs. 10 and 11. We did design a shuffling mechanism based on the similar idea as that applied in [19] to address the impact of the short-term unfairness, and found that it would sacrifice a lot in detection delay to achieve just a moderate gain in mitigating false positive rate. Thus according to the theoretical and simulation investigations given above, we decide to apply the FS detector without an extra mechanism for the short-term unfairness issue.

7.3 Performance Guarantee

Given the configuration parameter $h = 40$, we compare the target performance measures with the simulation results under the same setting, shown in Table 1, to examine whether the target performance is guaranteed. We can see that simulation results are very close to the target values in all three performance metrics. The small gap between the values is largely due to the variance in the observation samples; also the effect of the short-term unfairness is not 100% overcome according to Figs. 8 and 9. Considering such a small gap, in practice we can on purpose select configuration parameters to conservatively provision the detection performance.

With the same parameter configuration as above, we compare our FS detector to the sequential K-S test and the optimal SPRT for 802.11 backoff misbehavior detection used in [3] in Fig. 12. The sample used in those solutions is collected every successful transmission of the tagged node, whereas in our scheme, the sample is collected every successful transmission from any node in the network. The average detection delays in terms of the number of successful transmissions from the tagged node for different detection schemes are compared in Fig. 12. For a fair comparison, we map our samples (the total number of successful transmissions over the network) to that used in [3]. For such a mapping, we only need to count the number of successful transmissions from the tagged node within the total successful transmissions. Also note that the desired false positive rate in [3] is fixed at $P_{fp} = 0.05$, which is one order larger than our target 0.005 as given in Table 1. Even with a much more strict constraint on P_{fp}, Fig. 12 shows that our detector has comparative detection delays against high intensities of backoff misbehavior and becomes superior to all other schemes as the misbehavior turns less intense.

It is interesting to discuss why our FS detector has better performance even than the optimal SPRT (when the misbehaving intensity is not high) in [3]. An optimal SPRT has the "optimal" performance only when the normal behavior distribution could be accurately obtained. However, to establish the normal behavior distribution, the detectors in [3] need to first estimate the collision probability over the 802.11 channel. In [3], there are two aspects of inaccuracy in estimating the collision probability, which degrade the performance of false positive rate and detection delay, respectively.

The first aspect of inaccuracy in [3] is that the collision probability is estimated from only tens of samples, over which the variance may lead to overestimating the collision probability. The behavior monitored by the detector is the idle time between consecutive successful transmissions; an overestimated collision probability will lead to an overestimated idle time (longer than its real value). With such an estimation error by the detector, a normal idle time observed will appear smaller than the "thought-to-be" normal behavior and thus misunderstood as misbehaving. That is, the overestimation of the collision probability leads to a higher false positive rate.

The second aspect of inaccuracy is that, according to the IEEE 802.11 model, a conditional collision probability (given that the tagged node is sending a packet) should be used to characterize the backoff procedure and further estimate the distribution of the idle time between consecutive successful transmissions. The study in [3] however uses an unconditional collision probability estimated over all nodes to approximate the conditional one. Regarding the misbehaving node, the unconditional collision probability will be an underestimate of the conditional one. The conditional collision probability associated with the tagged misbehaving node is determined by transmissions from other normal nodes. When estimating with

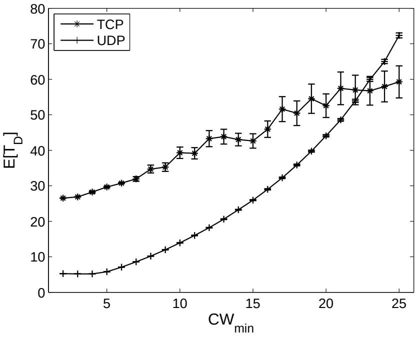

Fig. 13. Average detection delay for TCP traffic.

an unconditional collision probability, transmissions from the misbehaving node are also included in the estimation [3]; note that many transmissions from the misbehaving node will not experience collisions due to the misbehaving node's aggressive access to the channel. Thus, the collision probability will be underestimated, which then makes the detector to underestimate the normal idle time between consecutive successful transmissions. Such underestimation of the normal model makes the misbehavior deviation less obvious and incurs a longer time for detection.

7.4 Performance with UDP and TCP Traffic

We also consider scenarios where TCP traffic exists in the network. Fig. 13 shows the average detection delay of a misbehaving node versus the misbehaving intensity in a network of 10 nodes. The detection threshold $h = 40$. We compare the detection delays in the two scenarios that all the nodes send TCP traffic or saturated UDP traffic to the AP, respectively. As shown in Fig. 13, in most cases, the detection delay in the TCP scenario is larger than that in the UDP one, especially when the misbehavior is more intense. The reason is that TCP multiplicatively decreases the transmission rate upon a packet loss due to its congestion control mechanism; the impact of congestion control is more obvious in wireless networks where collisions are common. The congestion control mechanism by nature mitigates the selfish misbehavior. Aggressive transmissions will lead to more collisions, which in turn decreases the sending rate through the congestion control. Thus, it takes a longer time to detect (compared to the UDP case) the misbehavior due to the mitigating effect of the congestion control. With a low misbehaving intensity ($CW_{min} > 20$), the congestion control effect applies more to the normal nodes, where the detection delay will be shorter than that in the UDP case. In Fig. 13, we also plot the 95% confidence interval, measured from a large number of detections (in the order of 10^6), which demonstrates that TCP congestion control brings a high degree of dynamics to the system.

A more general scenario would be a wireless network consisting of both TCP and UDP nodes.[2] There are three possible cases. 1) All the nodes have a normal behavior. In this case, the FS detector will have a high false positive rate to indicate a certain normal UDP node as a misbehaving node, since the throughput of UDP nodes will overwhelm those TCP ones. 2) A misbehaving UDP node exists. Note that when both UDP and TCP flows exist, it is impossible for a TCP node to aggressively grab more throughput due to the congestion control. As a misbehaving UDP node will easily overwhelm those normal TCP nodes, the detection delay of a misbehaving node will be even shorter than that in an all-UDP case. 3) To avoid being detected, a smart misbehaving node may establish a TCP connection to the AP, but does not implement the congestion control mechanism (i.e., actually transmit according to UDP).

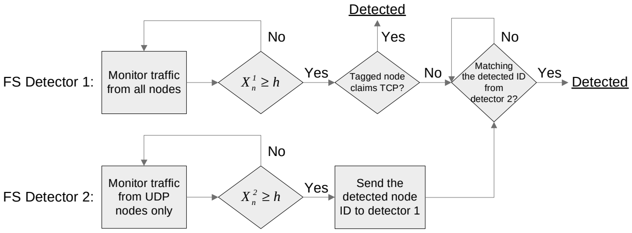
Detected
No
Yes
No
FS Detector 1:
Monitor traffic from all nodes
$X_n^1 \geq h$
Yes
Tagged node claims TCP?
No
Matching the detected ID from detector 2?
Yes
Detected
No
FS Detector 2:
Monitor traffic from UDP nodes only
$X_n^2 \geq h$
Yes
Send the detected node ID to detector 1

Fig. 14. Detection with TCP/UDP hybrid traffic.

For robust detection performance in the complex scenario when both UDP and TCP traffic flows exist, we design a *dual-detector* implementation as shown in Fig. 14. FS detector 1 monitors traffic from all the nodes; if a detection event happens, detector 1 further checks whether the tagged node claims to use TCP or UDP. If it claims to use TCP, detector 1 can then determine that it is a smart misbehaving node actually using UDP (case 3 mentioned above); if it claims to use UDP, detector 1 turns to listen to the decision from FS detector 2 (to avoid false positive in case 1). FS detector 2 starts simultaneously with detector 1 but monitors only the traffic from the UDP nodes. When detector 2 identifies misbehavior from a UDP node, it sends the detected node ID to detector 1. If this detected node ID matches that alarmed by detector 1, the dual-detector system will then determine that the node is misbehaving (case 2). We run simulations to verify the performance of the dual-detector. For example, in a network of 10 nodes where 5 nodes use TCP and the other 5 nodes use UDP, the average false positive rate over a normal UDP node is 0.0047. Also, for the moderate misbehavior of $CW_{min} = 16$, the average detection delay of a misbehaving node is 18.1953 when it lies to be a TCP node. The detection delay increases to 36.4728 (for confirmed detection in both detectors) when the attacker is honest with its

[2] Without loss of generality, we consider the situation that some nodes have a UDP flow and others have a TCP flow. If a node has both UDP flows and TCP flows, in a saturated situation, the aggregate traffic behaves similar to the UDP traffic.

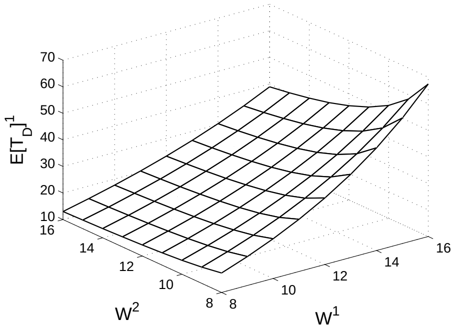

Fig. 15. Average detection delays under two misbehaving nodes.

UDP behavior, which is similar to that in the all UDP case listed in Table 1.

8 Discussion

8.1 Detection of Multiple Misbehaving Nodes

In this section, we discuss how our analytical model can be extended to the cases where multiple classes of malicious nodes with different intensities of misbehavior exist. The key to the analysis is to obtain the abnormal Markov chain, which in fact is determined by the probability that a successful transmission over the network is from the tagged malicious node.

Consider a network of N nodes, k of which are malicious and the rest are normal. Suppose that malicious node i sets its minimum contention window CW_{min} as W^i and all the $N-k$ normal nodes use the standard minimum contention window denoted as W^0. We can expand (11) to have the following equations:

$$\begin{cases} p_t^0 = \dfrac{2(1-2p_c^0)}{(1-2p_c^0)(W^0+1)+p_c^0W^0(1-(2p_c^0)^m)} \\ \vdots \\ p_t^k = \dfrac{2(1-2p_c^k)}{(1-2p_c^k)(W^k+1)+p_c^kW^k(1-(2p_c^k)^m)} \\ p_c^0 = 1-\displaystyle\prod_{i=1}^{k}(1-p_t^i)(1-p_t^0)^{N-k-1} \\ \vdots \\ p_c^k = 1-\displaystyle\prod_{i=1}^{k-1}(1-p_t^i)(1-p_t^0)^{N-k} \end{cases} \tag{23}$$

From the solutions of (23), we can obtain the probability that a node gets a successful transmission at a random time slot:

$$p_s^0 = p_t^0(1-p_c^0), \tag{24}$$

$$\vdots$$

$$p_s^k = p_t^k(1-p_c^k). \tag{25}$$

Then we can calculate the probability q_s^l that a successful transmission over the network is from the tagged malicious node l with $CW_{min}=W^l$ as

$$\hat{q}_s^l = \frac{p_s^l}{\sum_{i=1}^{k} p_s^i + (N-k)p_s^0}. \tag{26}$$

Using q_s^l in (26), we can obtain the transition probability matrix of the abnormal Markov chain $\hat{\mathbf{P}}^l$; using $\hat{\mathbf{P}}^l$ and initial states of the detector when misbehavior starts, which are determined in the same way as in Section 6.2.2, we can analyze average detection delay and missed detection ratio for the tagged malicious node l accordingly.

We consider an example that there are two misbehaving nodes in a network of 10 nodes, one setting its minimum contention window as W^1 and the other as W^2. Fig. 15 plots the average detection delays to identify the misbehaving node 1, denoted as $E[T_D]^1$, under different misbehaving intensity pairs (W^1, W^2). Note that even in this simple case the two malicious nodes are competing with each other. There is a tradeoff between the two nodes. Certainly it takes longer to detect one malicious node if the other chooses more intense misbehavior. It will be an interesting problem to determine how the multiple malicious nodes can find certain misbehaving strategies to collaboratively maximize their collective benefit from the network throughput while avoiding being detected as long as possible. In our future work, we will carry out more in-depth studies of the scenario with multiple misbehaving nodes.

8.2 Misbehaviors beyond CW Manipulation

Beyond just manipulating CW_{min} values, there can be more sorts of strategic misbehavior. As nowadays virtualization technologies are common, a malicious node can create multiple virtual adapters associated with one physical adapter. Combining this with the backoff misbehavior, the malicious node can initiate sybil attacks to gain more benefits from the network and still remain undetected. Further, the malicious node can even spoof the MAC addresses of well-behaved nodes and then start misbehaving. This may lead to false accusation of well-behaved nodes if the FS detector is directly applied. To address these issues, based on the fact that every node needs to contact the AP initially to join the network, one approach is to let the AP impose authentication to every node joining the network to ensure that each MAC address is associated with exactly one physical node. After the authentication, the FS detector can then take actions to monitor the node. We will conduct more in-depth studies in our future work.

9 Conclusion

In this paper we propose a novel fair share (FS) detector for real-time backoff misbehavior detection in IEEE 802.11 based wireless networks. Also, we develop a Markov chain based model to theoretically analyze the detection performance of the scheme. While most existing work for backoff misbehavior detection depends on heuristic parameter configuration and experimental performance evaluation, we are able to use our model for a quantitative study to achieve guaranteed detection performance in terms of average false positive rate, average detection delay and missed detection ratio. Moreover, we present simulation results that confirm the accuracy of our theocratical analysis and demonstrate the robustness of the FS detector. For our future work, we plan to systematically study the generic scenario with multiple misbehaving nodes in a multi-hop wireless network.

REFERENCES

[1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Select. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[2] The MAdWiFi Driver, [Online.] Available: http://www.madwifi.org/.

[3] A. Toledo and X. Wang, "Robust Detection of Selfish Misbehavior in Wireless Networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 6, pp. 1124-1134, Aug. 2007.

[4] A. Toledo and X. Wang, "A Robust Kolmogorov-Smirnov Detector for Misbehavior in IEEE 802.11 DCF," *Proc. IEEE ICC*, 2007, pp. 1564–1569.

[5] S. Radosavac, J. S. Baras and I. Koutsopoulos, "A Framework for MAC Protocol Misbehavior Detection in Wireless Networks," *Proc. ACM Workshop on Wireless Security*, 2005, pp. 33–42.

[6] S. Radosavac, G. Moustakides, J. Baras and I. Koutsopoulos, "An Analytic Framework for Modeling and Detecting Access Layer Misbehavior in Wireless Networks," *ACM Trans. Information and Systems Security*, vol. 11, no. 4, article no. 19, Jul. 2008.

[7] Y. Rong, S. Lee and H. Choi, "Detecting Stations Cheating on Backoff Rules in 802.11 Networks Using Sequential Analysis," *Proc. IEEE INFOCOM*, 2006, pp. 1–13.

[8] P. Kyasanur and N. Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," *Proc. IEEE DSN*, 2003, pp. 173–182.

[9] P. Kyasanur and N. Vaidya, "Selfish MAC Layer Misbehavior in Wireless Networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 502-516, Sept.-Oct. 2005.

[10] P. Serrano, A. Banchs, V. Targon and J. Kukielka, "Detecting selfish configurations in 802.11 WLANs" *IEEE Communication Letter*, vol.14, no.2, pp. 142-144, Feb. 2010.

[11] 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, Jan. 2011.

[12] B. Brodsky and B. Darkhovsky, *Nonparametric Methods in Change-Point Problems*. Kluwer Academic Publisher, 1993.

[13] C. E. Koksal, Hi. Kassab and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," *Proc. ACM SIGMETRICS*, 2000.

[14] M. Cagalj, S. Ganeriwal, I. Aad and J. Hubaux, "On Cheating in CSMA/CA Ad Hoc Networks," Tech. Rep. LCA-REPORT-2004-017, 2004.

[15] J. Konorski, "Protection of Fairness for Multimedia Traffic Streams in a Non-Cooperative Wireless LAN Setting," *Proc. 6th International Conference on Protocols for Multimedis Systems (PROMS)*, 2001.

[16] J. Konorski, "Multiple Access in Ad-Hoc Wireless LANs with Noncooperative Stations," *Proc. 2nd International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications (NETWORKING)*, 2002.

[17] M. Raya, J. Hubaux and I. Aad, "DOMINO: A System to Detect Greedy Behavior in IEEE 802.11 Hotspots," *Proc. ACM MobiSys*, 2004.

[18] M. Raya, I. Aad, J. Hubaux and A. El Fawal, "DOMINO: Detecting MAC Layer Greedy Behavior in IEEE 802.11 Hotspots," *IEEE Trans. Mobile Comput.*, vol. 5, no. 12, pp. 1691-1705, Dec. 2006.

[19] J. Tang, Y. Cheng and W. Zhuang, "An Analytical Approach to Real-Time Misbehavior Detection in IEEE 802.11 Based Wireless Networks," *Proc. IEEE INFOCOM*, 2011.

[20] H. V. Poor and O. Hadjiliadis, *Quickest Detection (1st ed.)*, Cambridge Univ. Press, 2008.

[21] J. R. Morris, *Markov Chains*, Cambridge Univ. Press, 1997.

[22] The Network Simulator - ns-2, [Online.] Available: http://www.isi.edu/nsnam/ns.

Jin Tang received the B.S. degree in Computer Science from Fudan University, Shanghai, China, in 2004 and the Master's degree in Information Technology and Management from Illinois Institute of Technology, USA, in 2007. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Illinois Institute of Technology. His current research interests include wireless network security, intrusion detection and security in VoIP applications.

Yu Cheng received the B.E. and M.E. degrees in Electrical Engineering from Tsinghua University, Beijing, China, in 1995 and 1998, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2003. Since August 2006, he has been with the Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, USA, and now an Associate Professor. His research interests include next-generation Internet architectures and management, wireless network performance analysis, network security, and wireless/wireline interworking. He received a Postdoctoral Fellowship Award from the Natural Sciences and Engineering Research Council of Canada (NSERC) in 2004, and a Best Paper Award from the conferences QShine 2007 and ICC 2011. He received the National Science Foundation (NSF) CAREER award in 2011. He served as a Co-Chair for the Wireless Networking Symposium of IEEE ICC 2009, a Co-Chair for the Communications QoS, Reliability, and Modeling Symposium of IEEE GLOBECOM 2011, and a Technical Program Committee (TPC) Co-Chair for WASA 2011. He is an Associated Editor for IEEE Transactions on Vehicular Technology.

Weihua Zhuang has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, since 1993, where she is a Professor and a Tier I Canada Research Chair in Wireless Communication Networks. Her current research focuses on resource allocation and QoS provisioning in wireless networks. She is a co-recipient of the Best Paper Awards from the IEEE Multimedia Communications Technical Committee in 2011, IEEE Vehicular Technology Conference (VTC) Fall 2010, IEEE Wireless Communications and Networking Conference (WCNC) 2007 and 2010, IEEE International Conference on Communications (ICC) 2007, and the International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine) 2007 and 2008. She received the Outstanding Performance Award 4 times since 2005 from the University of Waterloo, and the Premier's Research Excellence Award in 2001 from the Ontario Government. Dr. Zhuang is the Editor-in-Chief of IEEE Transactions on Vehicular Technology, and the Technical Program Symposia Chair of the IEEE GLOBECOM 2011. She is a Fellow of the IEEE, a Fellow of the Canadian Academy of Engineering (CAE), a Fellow of the Engineering Institute of Canada (EIC), and an elected member in the Board of Governors of the IEEE Vehicular Technology Society. She was an IEEE Communications Society Distinguished Lecturer (2008-2011).