This paper was presented as part of the main technical program at IEEE INFOCOM 2011

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

Jin Tang, Yu Cheng
Electrical and Computer Engineering
Illinois Institute of Technology
Email: {jtang9, cheng}@iit.edu

Weihua Zhuang
Electrical and Computer Engineering
University of Waterloo
Email: wzhuang@uwaterloo.ca

***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. The non-parametric cumulative sum (CUSUM) test is a promising method for real-time misbehavior detection due to its ability to quickly find abrupt changes in a process without any *a priori* knowledge of the statistics of the change occurrences. While most of the existing schemes for selfish behavior detection depend on heuristic parameter configuration and experimental performance evaluation, we develop a Markov chain based analytical model to systematically study the CUSUM based scheme for real-time detection of the backoff misbehavior. 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. Moreover, we find that the short-term fairness issue of the 802.11 DCF impacts the transition probabilities of the Markov model and thus the detection accuracy. We develop a shuffle scheme to mitigate the short-term fairness impact on the sample series, and investigate the proper shuffle period (in terms of observation windows) that can maintain the randomness in each node's backoff behavior while resolving the short-term fairness issue. We present simulation results to confirm the accuracy of our theoretical analysis as well as demonstrate the performance of the developed real-time detection scheme.**

I. Introduction

The IEEE 802.11 based wireless networks 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]–[3]. However, in such a distributed environment without a centralized controller, a malicious node may deliberately choose a smaller backoff timer and 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 [4], [5] make the backoff misbehavior much more feasible. In this paper, we propose an analytical approach for real-time detection of the backoff misbehavior.

This work was supported in part by NSF grant CNS-0832093.

An efficient 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 [5]–[9], or require modifications to the 802.11 protocols [10], [11]. Considering the challenges, in our very recent work [12], we develop a detection scheme based on the *non-parametric cumulative sum* (CUSUM) test [13] which can quickly find abrupt changes in a process without any prior knowledge of the statistical model of the change occurrences. In [12], we also propose a new observation method to get samples for testing, which directly counts the number of successful transmissions from a tagged node to facilitate the real-time detection. Also, our detection scheme does not require any modification to the protocols. Thus it can be implemented by any node assuming the role of the detection agent that monitors the network.

A significant open research issue regarding the selfish behavior detection is that most of the existing detection schemes depend on heuristic parameter configuration and experimental performance evaluation. 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.

In this paper, we develop an analytical model for the CUSUM based detection scheme, 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 CUSUM detector, because the detector's value in an observation window only depends on its value in the previous window and the observation samples in the current window. To determine the transition

probabilities of the Markov chain, 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. This Markov chain based model enables us to conduct rigorous quantitative analysis of the detection scheme 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 CUSUM 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.

To ensure the accuracy of our analytical model, we must take the well-known *short-term fairness issue* regarding the 802.11 DCF [14] into account. The short-term fairness issue is inherent to the 802.11 backoff mechanism, by which a successful node will have advantages in grabbing the channel for next few transmissions in a short period. 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 will further lead to inaccurate detection results. In order to address the short-term fairness issue, we enhance the detection scheme by applying a shuffle strategy to the observation samples to mitigate the correlations among neighboring observations. Specifically, a sample value of the number of successful transmissions of the tagged node is calculated as the mean value over a few observation windows. Moreover, we also investigate how to find a proper value for the *shuffle period* (in terms of observation windows). A too short shuffle period could not effectively remove the short-term fairness, while a too long period will average out the randomness in the 802.11 backoff behavior. We would like to emphasize that the shuffle scheme will not impact the detection delay, which only requires a few normal observation samples to fill the observation windows in the shuffle period when the system turns up and then works in a sliding window manner for detection.

In summary, the main contributions of the paper come in four aspects: 1) We develop a discrete-time Markov chain model to characterize the CUSUM based detection system for the backoff misbehavior. 2) We utilize the model to guide the detection system configuration based on theoretical analysis for guaranteed performance. 3) We apply a shuffle strategy to resolve the impact of the 802.11 short-term fairness issue on the analysis and detection accuracy. 4) We provide simulation results to confirm the accuracy of our theoretical analysis and demonstrate the performance of the developed detection scheme.

The rest of the paper is organized as follows. Section II reviews more related work. Section III describes the system model. In Section IV, we present the detection scheme design with the enhancement of the shuffle strategy. Section V develops the Markov chain based analytical model, and Section VI gives the theoretical performance analysis based on the Markov chain model. Section VII presents the simulation results, and Section VIII concludes the paper.

II. Related Work

The problem of detecting backoff misbehavior over 802.11 based wireless networks has been studied in various scenarios and under several mathematical frameworks in the literature. In [10], [11], the authors presented a modification to the 802.11 protocol to facilitate the misbehavior detection. In their scheme, 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 will consider that the sender potentially deviates from the protocol and penalize the sender with a smaller backoff timer. Continuous deviations will result in that the receiver labels the sender as a selfish node. However, the above scheme assumes a trustworthy receiver which performs the detection, which may not be the case in a dynamic network environment. Modification to the 802.11 protocol and reliance on the 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, by imposing adequate costs on network operations [15]–[17]. 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 are 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 performance of the network may converge to a suboptimal operation point. Moreover, modifications to the standard protocols are also required.

A heuristic sequence of conditions are proposed in [18], [19] to test multiple misbehavior options in the 802.11 protocol based on simple numerical comparisons. The detection algorithm estimates the average values of the option parameters and raises alarms when the cumulative effect of the misbehavior exceeds a threshold. 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 specific conditions limit its applications to certain protocols.

The sequential probability ratio test (SPRT) method is used in [7]–[9] 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 greater 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; such an issue imposes the major inherent limitation of the SPRT method. 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. Thus in our detection we monitor the successful transmission of the tagged node as the observation measurement.

The authors in [5], [6] utilize the Kolmogorov-Smirnov (K-S) significance test to address the detection problem assuming an *a priori* known misbehavior strategy model. This test is able to make the decision by comparing the distribution of sampled traffic data with the normal-behavior distribution estimated online. Nonetheless, 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 actually makes real-time detection impossible. Even in the modified sequential truncated K-S test, a stage number N [5] still needs to be predetermined before test starts in order to get a proper significant level for each test step. In our very recent work [12], we propose to adopt the non-parametric CUSUM change point test [13] 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. However, a common 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.

III. System Model

In this section we provide a brief description of the IEEE 802.11 DCF and how a malicious node can utilize the vulnerability of the DCF to gain advantages over other normal nodes through backoff misbehavior.

A. 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. In this paper, our concentration is the more widely used DCF which operates in a distributed manner. 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, 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 a set $\{0, 1, ..., CW_{min} - 1\}$, 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 due to reasons such as collisions or lost of ACK messages from the reception node will result in a doubled contention window size until it reaches the maximum contention window $CW_{max} = 2^m CW_{min}$, 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 the minimum value CW_{min} and continue sensing the medium if it has more packets to transmit.

B. Backoff Misbehavior in IEEE 802.11 DCF

A malicious node does not need to misbehave if no one else is contending for the network access with it and the effect of the misbehavior may be ignorable. Therefore we assume that the 802.11 wireless network works in the saturated condition, where every participating node always has packets to transmit.

As a distributed contention based protocol, the DCF assumes that every node in the network operates in accordance with the protocol rules as described above to obtain a fair share of the wireless medium. However, a node which has the smallest backoff timer will obviously be favored by the protocol as it can always obtain more chances to transmit while other nodes are still in the backoff stage. Since there is no central controlling unit which assigns the backoff timer for each node, a malicious node can 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, which results in the malicious node gaining more advantages. The backoff misbehavior can drastically decrease the transmission probability of normal nodes and subsequently severely downgrade their throughput. In some 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 except for the malicious node itself. Thus, a detection scheme capable of quickly and accurately identifying the misbehaving malicious node is highly desired for the normal operation of an IEEE 802.11 wireless network.

IV. Detection Scheme Design

In this section, we describe the CUSUM based backoff misbehavior detection scheme design, enhanced with a shuffle mechanism to mitigate the short-term fairness impact.

A. The Observation Measure

Consider a tagged node v. In our detection system, the *observation measure* is the number of successful transmissions of the tagged node v, denoted as M^v, in every M successful transmissions over the whole network. Under the 802.11 DCF, M_n is a random variable. 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, we could have the binomial distribution of M^v as

$$P\{M^v = k\} = \binom{M}{k}(q_s^v)^k(1-q_s^v)^{M-k}. \quad (1)$$

In a normal situation that every node uses the same contention window size and follows the 802.11 DCF model, it can be seen that $q_s^v = \frac{1}{N}$ under the independent channel access assumption and 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 V, we will present how to calculate q_s^v given the contention window size. The distribution of M_v in (1) is the basis to establish our analytical model.

B. Shuffle Strategy

The short-term fairness issue is inherent to the 802.11 backoff mechanism by which a node that has just accomplished a successful transmission will have advantages in grabbing the channel for next transmission in a short period [14]. Such an issue implies correlations among the channel accesses, which impact the accuracy of (1) to model the successful transmissions of the tagged node based on the assumption of independent channel access, and will in turn impact the accuracy of the analytical model later. Intuitively, if we always assume a fair channel sharing for the normal condition, the short-term fairness issue poses significant difficulties in distinguishing between the backoff misbehavior of a malicious node and the short-term "monopolization" of a normal node, which may result in significant false positives for a detection scheme.

Directly computing the distribution of M^v under the short-term fairness impact is difficult. A possible approach to adapt to the short-term fairness issue is to measure the distribution from the training data. However, this approach requires an excessive number of experiments to have comprehensive training data sets for different traffic conditions and thus lacks the flexibility. We however choose to apply a shuffle strategy to the observation samples to mitigate the short-term fairness issue. Rather than directly using the observation measure in the current observation window, we average the observation measures over I windows, termed as *shuffling with a period of* I. Let $\widetilde{M}_n^v$ denote the raw observation without shuffling at window n, and M_n^v the shuffled observation. We apply a sliding window mechanism for shuffling as

$$M_n^v = \frac{1}{I}\sum_{i=0}^{I-1}\widetilde{M}_{n-i}^v. \quad (2)$$

We expect that different nodes may take short-term advantages in different observation windows; such a shuffling average could dilute the correlations in each window. With the short-term fairness impact mitigated by the shuffle mechanism, the binomial distribution in (1) can still be applied.

The value of the shuffle period I needs to be properly selected. A too short I could not effectively remove the correlations among observation measures, whereas a too long I will average out the random operation of the 802.11 backoff mechanism. We will resort to simulations to show that an optimal I does exist resulting in accurate analytical results based on (1).

C. CUSUM Based Detection

Considering the challenges on real-time detection of the unpredictable backoff misbehavior, we design our detection scheme based on the non-parametric CUSUM test [13] due to its robust ability to quickly find abrupt changes in a process without any prior knowledge of the statistical model of the change occurrences.

For convenience, let $\{M_n, n = 0, 1, ...\}$ be the sequence of the number of successful transmissions of the tagged node in each observation window and u be the upper bound of the expectation of M_n. Here, we drop the superscript of v for easier presentation considering the clear context. We then have the CUSUM test statistic

$$\begin{aligned} X_n &= (X_{n-1} + (M_n - u))^+ \\ X_1 &= 0 \end{aligned} \quad (3)$$

where $(x)^+ = x$ if $x \geq 0$ or 0 otherwise. Note that if the tagged node acts normal, X_n will stay around 0 no matter how long the normal behavior has been observed. However, when the node turns to misbehave, X_n will quickly accumulate to a large positive.

Let h be the detection threshold, then the decision rule of the test in step n is

$$\delta_n = \begin{cases} 1 & \text{if} \quad X_n \geq h \\ 0 & \text{if} \quad X_n < h \end{cases} \quad (4)$$

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.

V. 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 j at time n if $X_n = j$ and in state i at time $n-1$ if $X_{n-1} = i$, where $i, j \in A$. The transition between the states happens at the end of every observation window M. According to (3), the current state X_n depends only on the state X_{n-1} and is independent of any other previous states, where the transition probability is

$$P_{ij} = P\{X_n = j | X_{n-1} = i\}. \quad (5)$$

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 four distinct groups based on the operation of the CUSUM detector.

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

$$P_{i0} = \begin{cases} P\{M_n \leq (u-i)\} & \text{if } u - i \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

This group is related to the transitions to state 0. According to (3), the state X_n is always reset to 0 from i when $i + M_n - u \leq 0$. Thus the transition probability is $P\{M_n \leq (u-i)\}$ as shown in (6). Note that it is impossible to reach state 0 when $u - i < 0$ due to nonnegative value of M_n.

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

$$P_{ij} = \begin{cases} P\{M_n = (j-i+u)\} & \text{if } j - i + u \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

This group is about the transitions to all the states other than 0 and h. Again the transitions are according to the CUSUM detector's behavior (3) and the fact that M_n is nonnegative.

Group 3 consists of P_{ij} for $i \in [0, h-1]$ and $j = h$, with values

$$P_{ih} = 1 - \sum_{j=0}^{h-1} P_{ij}. \quad (8)$$

This group is about the transitions to state h. Note that state h in fact incorporates all possible values $X_n \geq h$.

Finally, group 4 consists of P_{ij} for $i = h$ and $j \in [0, h]$, with values

$$P_{hj} = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

This group is related to the transition from h to any other states, according to the behavior that the detector value will be reset to 0 as soon as it exceeds h.

VI. Theoretical Performance Analysis

In this section, we conduct rigorous theoretical performance analysis of the detection scheme 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 consider a network with N nodes, and each node always has data to transmit (i.e., in the saturated condition). In all the analysis, we consider that the shuffle mechanism is applied and thus the analysis could be based on (1). The observation window size is $M = 30$.

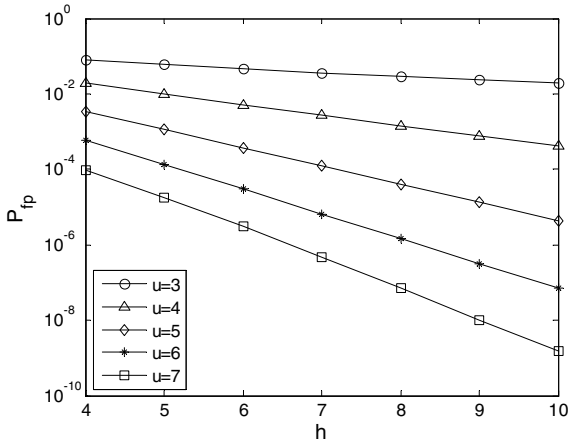


Fig. 1. Average false positive rate.

A. 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 CUSUM 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 M_n according to (1), and further obtain the transition probabilities matrix $\mathbf{P}$ according to (6)−(9).

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

$$\pi_j = \sum_{i=0}^{h} \pi_i P_{ij}, \quad \text{for } j \in \{0, ..., h\} \quad (10)$$

$$\sum_{j=0}^{h} \pi_j = 1. \quad (11)$$

Then we can get the average false positive rate

$$P_{fp} = \pi_h. \quad (12)$$

The analytical result (12) allows us to numerically examine the impact of the two fundamental parameters h and u on the false positive rate P_{fp} of the CUSUM detector. As an example, we compute the results for a network with $N = 10$ nodes, and the results are illustrated in Fig. 1. The logarithmic scale is used in the figure for the vertical axis. The minimum value of u is set to 3 as it represents the upper bound of the expectation of M_n, which is M/N in the normal condition. From the figure, we can observe that a larger h or a larger u yields a smaller false positive rate, as expected.

B. 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 CUSUM detector when the misbehavior starts.

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:

$$\begin{cases} p_t^0 = \dfrac{2(1-2p_c^0)}{(1-2p_c^0)(W^0+1)+p_c^0W^0(1-(2p_c^0)^m)} \\ p_t^1 = \dfrac{2(1-2p_c^1)}{(1-2p_c^1)(W^1+1)+p_c^1W^1(1-(2p_c^1)^m)} \\ p_c^0 = 1-(1-p_t^1)(1-p_t^0)^{N-2} \\ p_c^1 = 1-(1-p_t^0)^{N-1} \end{cases} \tag{13}$$

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 (13), we can obtain the probability that a node gets a successful transmission at a random time slot:

$$p_s^0 = p_t^0(1-p_c^0) \tag{14}$$

$$p_s^1 = p_t^1(1-p_c^1). \tag{15}$$

We can then calculate the probability $\hat{q}_s$ that a successful transmission over the network is from the malicious node as (15):

$$\hat{q}_s = \frac{p_s^1}{p_s^1+(N-1)p_s^0}. \tag{16}$$

Using $\hat{q}_s$ in (1), we can obtain the probability distribution of M_n for the misbehaving node; using such M_n distribution in (6)−(9), 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 (13) to (16) 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.

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 (10) and (11). Since we are interested in detection starting from an unalarmed state, under such a constraint the conditional initial state probabilities should be

$$\pi_i' = \frac{\pi_i}{\sum_{i=0}^{h-1}\pi_i} \quad \text{for } i \in \{0, ..., h-1\}. \tag{17}$$

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 $\hat{\mathbf{P}}$.

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

$$\mu_{ih} = 1 + \sum_{r \neq h} \hat{P}_{ir}\mu_{rh} \quad \text{for } i \in \{0, ..., h-1\}. \tag{18}$$

where $\hat{P}_{ir}$ is the transition probability from state i to r of $\hat{\mathbf{P}}$. Based on the solutions of (17) and (18), we can obtain the average detection delay $E[T_D]$ as

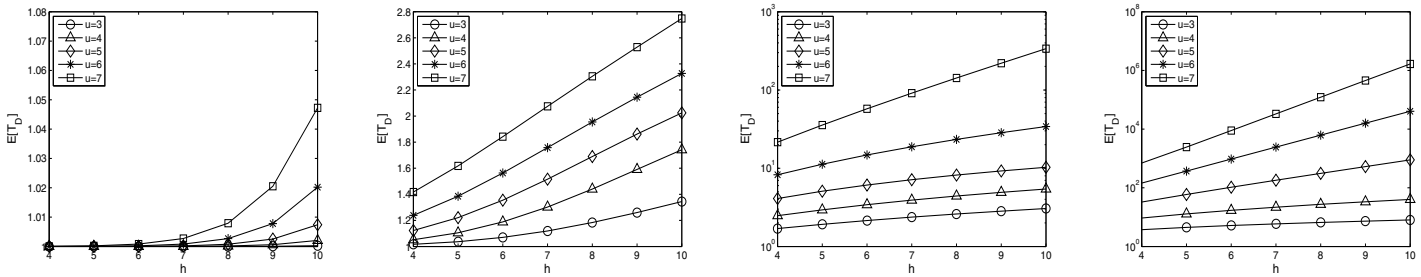
$$E[T_D] = \sum_{i=0}^{h-1} \pi_i'\mu_{ih}. \tag{19}$$

The analytical result (19) allows us to numerically examine the impact of h and u on the average detection delay $E[T_D]$ of the CUSUM detector. As an example, we compute the results for a network with $N = 10$ nodes, and the results are illustrated in Fig. 2. Specifically, Fig. 2 shows the analysis results under four misbehaving intensities $CW_{min} = 4$, $CW_{min} = 8$, $CW_{min} = 16$ and $CW_{min} = 24$. As we expect, the figure illustrates that more intense misbehavior leads to a shorter detection delay. Also, we observe that a smaller h and a smaller u yield better performance in average detection delay.

C. Missed Detection Ratio

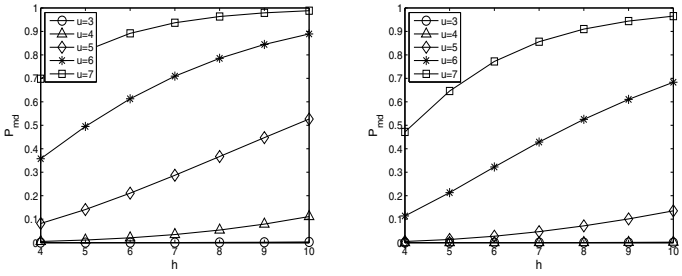
In this subsection we discuss the missed detection ratio, denoted as P_{md}. The missed detection ratio is not often considered in the context of CUSUM test due to its "non-stop until detection" property. We however examine the P_{md} under a given detection delay constraint D, which is of importance 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 could be computed in an iterative



(a) (b) (c) (d)

Fig. 2. Average detection delay. (a) $CW_{min} = 4$. (b) $CW_{min} = 8$. (c) $CW_{min} = 16$. (d) $CW_{min} = 24$.



(a) (b)

Fig. 3. Missed detection ratio. (a) $D = 8$. (b) $D = 16$.

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 (17), setting

$$P_i(0) = \pi'_i, \quad i \in [0, h-1] \tag{20}$$

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

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

$$\vec{P}(j) = \vec{P}(j-1) \cdot \hat{\mathbf{P}} \tag{22}$$

$$P_h(j) = 0. \tag{23}$$

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

$$P_{md} = \sum_{i=0}^{h-1} P_i(D). \tag{24}$$

Fig. 3 demonstrates the missed detection ratios P_{md} of our analysis under the delay constraints $D = 8$ and $D = 16$, for a misbehaving node with the moderate misbehavior of $CW_{min} = 16$. We observe that the longer the constraint is, the lower the missed detection ratio will be. Or in other words, the probability of detection increases with a cost of longer delay. Also, a low h and a low u yield better missed detection ratio.

D. System Configuration under Performance Constraints

The above theoretical analysis provides us a guideline to configure the system parameters h and u for guaranteed performance. For each performance metric, we can obtain the feasible ranges of h and u to satisfy the performance constraint. Then with the intersection of the parameter ranges under all the constraints, we expect to obtain the proper configuration of h and u that meet the performance requirements of all the metrics. In our analytical results presented above, we notice that the metrics are more sensitive to the change of u than h. Thus a rule of thumb for configuration is to first concentrate on finding a proper u and then refine it by determining an appropriate h. Moreover, once we determine a set of configuration parameters, we could explicitly know the target performance measures. 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 could be detected with good performance. For example, if we select $h = 7$ and $u = 5$ for system configuration, our analytical model can tell that even for the moderate misbehavior with $CW_{min} = 16$, we target high level of performance with the false positive rate of 0.00012, the average detection delay of 7.1375 observation windows, and the missed detection ratio of 0.047 with the delay constraint $D = 16$. In next section, we will use simulation results to demonstrate that our target performance measures are indeed achieved.

VII. Simulation Results

A. Simulation Setup

We establish an 802.11 DCF based wireless network consisting of 10 competing nodes and an access point (AP) through ns-2 [22] simulation. The network works under the saturated condition and every node sends constant traffic via UDP towards the AP. The AP also acts as the detection agent which monitors the transmissions from all the competing nodes and runs the CUSUM based detection scheme based on the collected samples. The nodes are located close enough to each other and can then sense the transmissions from others to 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

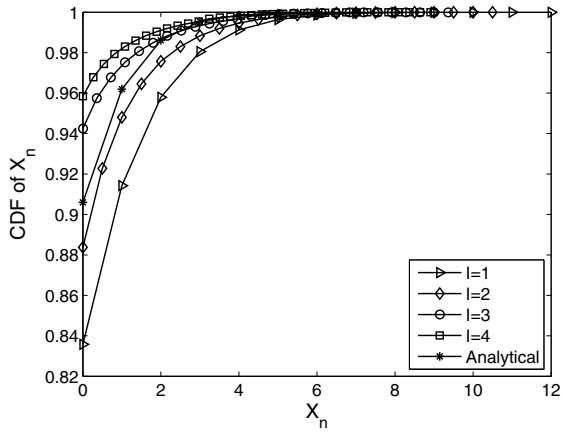


Fig. 4. CDF of X_n under different observation window sizes.

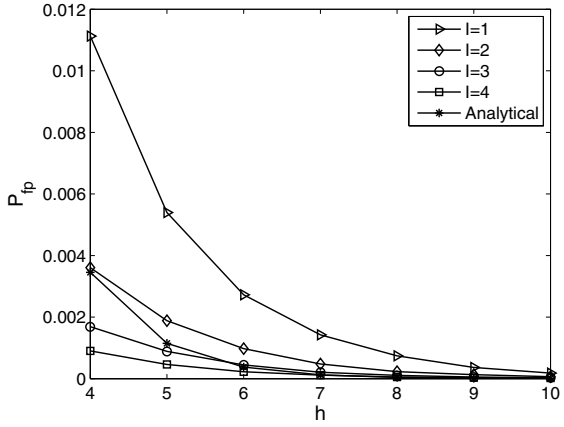


Fig. 5. Average false positive rate.

but can manipulate its minimum contention window CW_{min} to any value between 1 and 32.

Due to the conflicting nature of the three performance metrics, it is impossible to find the system configuration parameters that achieve best performance at all fronts. However, as discussed in the previous section, we find that the performance metrics are more sensitive to u and for $u = 5$ a good tradeoff can be achieved among the metrics. Therefore in our simulation, we adopt 5 as the value of u to further evaluate the performance of our detection.

B. Shuffle Period Selection

As described in the pervious section, the short-term fairness issue makes the binomial modeling of M_n not accurate due to the correlations among successful transmissions, which further impacts the transition probabilities of the Markov chain model and thus the accuracy in system configuration and detection performance. Here we examine the efficiency of the shuffle mechanism in mitigating such an issue. Since the short-term fairness issue is inherent to the 802.11 DCF independent of whether the misbehavior presents or not, we study the selection of the shuffle period in the normal traffic condition and then examine the efficiency of the determined shuffle period in misbehavior detection.

The purpose of the shuffle mechanism is to mitigate the correlations in channel access, thus its efficiency could be demonstrated if the distribution of the CUSUM statistic X_n obtained in simulations with the shuffling applied matches the analytical results based on the independent model of (1). In Fig. 4, we present the simulation results of the cumulative distribution function (CDF) of X_n under different shuffle period, compared with the analytical CDF. The configuration parameters are $h = 7$ and $u = 5$. We note that the curve closest to the analytical one is $I = 2$. We then examine the false positive rate through comparing the analytical results with the simulation results with $u = 5$ and h taking various values in Fig. 5. Again, the false positive rate curve obtained from $I = 2$ most resembles the analytical model. Based on the experimental results, we can heuristically determine that $I = 2$ is the optimal shuffle period.

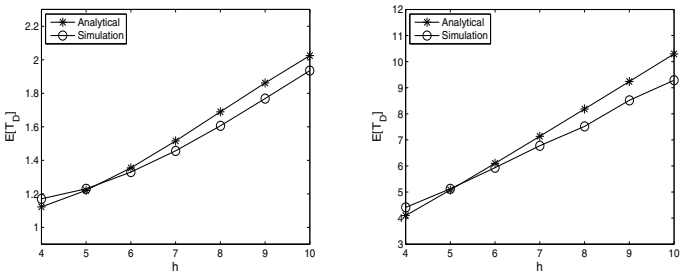


Fig. 6. Average detection delay. (a) $CW_{min} = 8$. (b) $CW_{min} = 16$.

According to our conjecture that the short-term fairness issues is inherent to the 802.11 DCF, an effective shuffle period in the normal condition should also apply in the condition with misbehavior. We then simulate the average detection delays under different misbehaving intensities and apply the shuffle period of 2. The simulation results and analytical results of the average detection delay are presented in Fig. 6. The closeness of the two curves confirm the efficiency of the shuffle mechanism in ensuring an accurate detection system.

C. Performance Guarantee

TABLE I
COMPARISON OF ANALYTICAL AND SIMULATION RESULTS WITH $h = 7, u = 5, D = 16$

	P_{fp}	$E[T_D]$	P_{md}
Analysis	0.00012	7.1375	0.047
Simulation	0.00047	6.4475	0.058

Given the configuration parameters $h = 7$ and $u = 5$, we compare the target performance measures (referring to VI-D) with the simulation results under the same setting in Table I to examine whether the target performance is guaranteed.

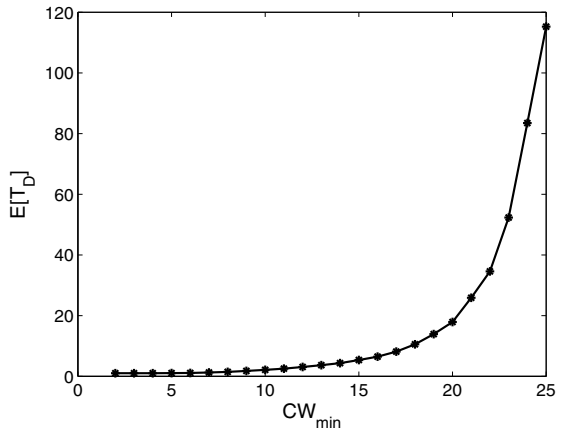


Fig. 7. Average detection delays under different misbehavior intensities.

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 shuffle mechanism is not 100% accurate according to Fig. 4 and 5. Considering such a small gap, in practice we could on purpose select configuration parameters to conservatively provision the detection performance.

Fig. 7 shows the average detection delays of our scheme under different misbehavior intensities, indicated by the size of the minimum contention window CW_{min} of the malicious node. It is observed that our scheme generally performs very well when detecting more intense misbehavior. For example, it takes less than 2 samples for $CW_{min} \leq 10$ and less than 10 samples for $CW_{min} \leq 18$. The detection performance degrades against less intense misbehavior as the malicious node's chance of successful transmissions gets lower and the small advantages gained by the malicious node is easily smoothed out in a shuffle period. However, as practically a malicious node has to choose more intense misbehavior to gain more benefits, our scheme can show its advantages in such cases.

VIII. Conclusion

In this paper we propose an analytical approach to real-time backoff misbehavior detection in IEEE 802.11 based wireless networks. We first develop a Markov chain based model to characterize the behavior of the non-parametric CUSUM detector used in our detection. While most existing work for backoff misbehavior detection depends on heuristic parameter configuration and experimental performance evaluation, we are able to use our model to quantitatively study the system configuration parameters to achieve guaranteed detection performance in terms of the three fundamental metrics. Moreover, realizing that the short-term fairness issue of 802.11 affects the transition probabilities of the Markov chain model and thus the detection accuracy, we apply a shuffle strategy to mitigate the impacts and also investigate the proper value of the shuffle period. Finally, we present simulation results that confirm the accuracy of our theocratical analysis and demonstrate the performance of the CUSUM based detection scheme. In our future work, we will study how to systematically compute optimal system configuration parameters based on our analytical model. Also, we will expand our work to distributed misbehavior detection, where cooperation among several detection agents and joint distribution analysis are to be employed.

References

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[2] H. Zhai, X. Chen and Y. Fang, "How well can the IEEE 802.11 wireless LAN support quality of service?" *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 3084-3094, Nov. 2005.

[3] Y. Cheng, X. Ling, W. Song, L. Cai, W. Zhuang, and X. Shen, "A cross-layer approach for WLAN voice capacity planning," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 678-688, May 2007.

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

[5] A. Toledo and X. Wang, "Robust detection of selfish misbehavior in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1124-1134, Aug. 2007.

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

[7] S. Radosavac, J. S. Baras and I. Koutsopoulos, "A framework for MAC protocol misbehavior detection in wireless networks," in *Proc. ACM Workshop on Wireless Security*, 2005, pp. 33–42.

[8] 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.

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

[10] P. Kyasanur and N. Vaidya, "Detection and handling of MAC layer misbehavior in wireless networks," in *Proc. IEEE DSN*, 2003, pp. 173–182.

[11] P. Kyasanur and N. Vaidya, "Selfish MAC layer misbehavior in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 502-516, 2005.

[12] J. Tang, Y. Cheng, Y. Hao and C. Zhou, "Real-time detection of selfish behavior in IEEE 802.11 wireless networks," in *Proc. IEEE VTC-Fall*, 2010.

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

[14] C. E. Koksal, Hi. Kassab and H. Balakrishnan, "An analysis of short-term fairness in wireless media access protocols," in *Proc. ACM SIGMETRICS*, 2000.

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

[16] J. Konorski, "Protection of fairness for multimedia traffic streams in a non-cooperative wireless LAN setting," in *Proc. 6th International Conference on Protocols for Multimedis Systems (PROMS)*, 2001.

[17] J. Konorski, "Multiple access in Ad-Hoc wireless LANs with non-cooperative stations," in *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.

[18] M. Raya, J. Hubaux and I. Aad, "DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots," in *Proc. ACM MobiSys*, 2004.

[19] 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.

[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.