Selfish Misbehavior Detection in 802.11 Based Wireless Networks: An Adaptive Approach Based on Markov Decision Process

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

***Abstract*—The open and distributed nature of the IEEE 802.11 based wireless networks provides selfish users the opportunity to to gain an unfair share of the network throughput by manipulating the protocol parameters, say, using a smaller contention window. In this paper, we propose an adaptive approach for real-time detection of such selfish misbehavior. An adaptive detector is necessary in practice, as it needs to deal with different misbehaving scenarios where the number of selfish users and the contention windows exploited by each selfish user are different. In this paper, we first design a basic misbehavior detector based on the non-parametric cumulative sum (CUSUM) test. While the basic detector can be modeled with a Markov chain, we further resort to the Markov decision process (MDP) technique to enhance the basic detector to an adaptive design. In particular, we develop a novel reward function based on which the optimal policy of the MDP can be determined. The optimal policy indicates how the adaptive detector should operate at each state. Another important feature of our detector is that it enables an effective iterative method to detect multiple misbehaving nodes. We present thorough simulation results to confirm the accuracy of our analysis, and demonstrate the efficiency of the adaptive detector compared to a static solution.**

I. Introduction

The IEEE 802.11 based wireless networks are widely deployed to provide wireless Internet access. The IEEE 802.11 protocol relies on the carrier sense multiple access/collision avoidance (CSMA/CA) based distributed cooperation function (DCF) for a distributed medium access control (MAC). According to the DCF MAC, all the users will fairly share the network throughput if everyone follows the standard in a cooperative manner [1], [2]. However, due to the lack of a central controller, a selfish user can simply manipulate protocol parameters, e.g., choosing a smaller minimum contention window size, to gain unfair access to the wireless channel. Such misbehavior not only gives advantages to the selfish user, but also results in a much less share of the network throughput or even denial of service to other normal users who play by the rule. In this paper, we propose an adaptive approach for real-time detection of such selfish misbehavior in 802.11 based wireless networks.

We adopt the non-parametric cumulative sum (CUSUM) test [3] to design a basic misbehavior detector. Our detector takes observation measures each time a successful transmission is observed over the channel, and updates its value based on a rule of fairly sharing the channel among the contending nodes. When monitoring a normal node, the detector value remains around zero; when monitoring a selfish misbehaving node, the detector value will quickly accumulate into a high positive value and further raise an alarm upon hitting a threshold. An adaptive detector is necessary in practice, as it needs to deal with different misbehaving scenarios where the number of selfish users and the contention windows exploited by each selfish user are different. In particular, upon an observation sample, the detector needs to determine an action of whether more aggressively increase or decrease the value of the detector. At a state, if the observed node could be a misbehaving one with a high probability, aggressively increasing the detector value can lead to a smaller detection delay while does not impact the false alarm rate much. On the other hand, if the observed node is inferred to be a normal one with a high probability, aggressively decreasing the detector value can mitigate the false alarm rate without impacting the detection delay much.

Regarding the adaptive detector, a proper decision on the action at each state is crucial for improving the detection performance, as bad decisions may degrade the system performance. We will indicate that our CUSUM based detector can be modeled as a discrete time Markov chain. Thus, we can further resort to the *Markov decision process* (MDP) (Ch. 4 of [4]) technique to guide the adaptive design. In a MDP formulation, the optimal action for each detection state will be obtained by solving an optimization problem to maximize a reward function. This paper develops a novel reward function which will generate a positive reward for a correct decision and a negative penalty for a wrong decision (for example, aggressively increasing the detector value when monitoring a normal node). The Markovian model also enables us to theoretically analyze the detection performance in terms of false positive (or false alarm) rate, average detection delay, and missed detection ratio. Efficiency of the MDP based adaptive design is demonstrated with comparison to the performance of the basic static detector. To the best of our knowledge, this paper is the first work applying the MDP to improve the performance of the CUSUM-based detector.

We would like to emphasize that a particular advantage of our detector is its capability to detect multiple misbehaving

This work was supported by NSF grant CNS-1117687.

978-1-4673-5946-7/13/$31.00 ©2013 IEEE

nodes in a network. In such a scenario of multiple misbehaving nodes, it is difficult to detect all misbehaving nodes by just analyzing the fairness in accessing the channel. Both the normal nodes and the selfish nodes with lower misbehaving intensity will be overwhelmed by those selfish nodes with higher misbehaving intensity. Our detector design is equipped with the capability to readily shield the detector operation from the impact of the traffic associated with any specified node. Such a capability enables an iterative method to effectively detect multiple misbehaving nodes. After detecting the node with the most intense misbehavior, the detector can discard the packets from that node and continue the detection among the leftover nodes. Thus, the misbehaving nodes will be detected one by one.

The contributions of this paper can be summarized in five aspects. 1) We present a CUSUM based misbehavior detector, and further enhance it with an adaptive design. 2) We develop a MDP based model to guide the design of the adaptive detector. 3) We develop a novel reward function for formulating the MDP problem. 4) An iterative algorithm is presented for detecting multiple misbehaving nodes. 4) Both theoretical analysis and simulation results are provided to demonstrate the performance of our detector.

The rest of the paper is organized as follow. Section II reviews more related work. Section III describes the detector design. Section IV develops the MDP based modeling and analysis. Section V presents the theoretical performance analysis based on the Markovian model, and Section VI presents the simulation results. Section VII concludes the paper.

II. Related Work

The problem of detecting selfish misbehavior over 802.11 networks has been studied in various scenarios and under several mathematical frameworks in the literature. The approaches in [9]–[11] focus on developing protocols based on the game-theoretic techniques. The goal is to encourage all the nodes to reach a Nash equilibrium. As a result, a misbehaving node is not able to gain an unfair share over well-behaved nodes and thus discouraged from misbehaving. 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 required.

The studies [12], [13] present a modification to the 802.11 protocol for 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 timer, the receiver will consider that the sender potentially deviates from the protocol. Continuous deviations will let the receiver label the sender as a selfish node. Modifications to the 802.11 protocol and reliance on the receiver are the main limitations of the work. A heuristic sequence of conditions are proposed in [14], [15] 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. However, the heuristic nature of the approach limits its applications to specific protocols.

The authors in [16], [17] utilize the Kolmogorov-Smirnov (K-S) test for misbehavior detection. This test is able to make decisions 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 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. Also, as a batch test method, the K-S test needs fixed-size data samples to perform detection, which makes real-time detection difficult.

In [5], we adopt the non-parametric CUSUM test for misbehavior detection. The detector there counts the number of successful transmissions from a tagged node within an observation window to get a sample. Although such a sampling method is easy to implement, 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, the detector 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 achieve good performance in both false positive rate and detection delay.

III. Detector Design

A. Basic Detector

In this paper, we consider a single-hop IEEE 802.11 based wireless local area network (WLAN). The access point (AP) will run a separate misbehavior detector to monitor each node. We consider the saturated model that every node always has data for transmission. Note that in practice a node may be unsaturate, but when one or more selfish misbehaving nodes exist, their aggressive transmissions will drive the network to the saturated point.

In our detection system, the *observation measure* is an indicator of whether a successful transmission over the network belongs to a tagged node, denoted as I. Let $\{I_n, n = 0, 1, ...\}$ be the sequence of sample values of I, observed each time a successful transmission appears on the channel. Let N denote the number of nodes existing in the network, which is readily known to the AP. For our basic detector, suppose its initial value X_n to be 0. When the current successful transmission over the network is from the tagged node, i.e., $I_n = 1$, we increase X_n by $N - 1$; otherwise, when the transmission is from any non-tagged node, i.e., $I_n = 0$, we decrease X_n by 1 until it reaches 0. The intuition of this design is as follows: In the normal situation where every node follows the 802.11 DCF model, each node roughly takes turns 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 in the normal situation, the detector X_n will fluctuate around a low value close to zero. On the other hand, when the tagged node turns to misbehave and obtain more chances to transmit, we can see that X_n is going to quickly accumulate to a large positive value.

The behavior of the basic detector can be mathematically described as

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

where $(x)^+ = x$ if $x \geq 0$ or 0 otherwise. And it can be seen that (1) follows the form of a non-parametric CUSUM detector.

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

$$\delta_n = \begin{cases} 1 & \text{if} \quad X_n \geq h \\ 0 & \text{if} \quad X_n < h \end{cases} \tag{2}$$

where δ_n is also an indicator function of whether the detection event happens or not. The detector X_n will be reset to 0 once it exceeds h and the detection procedure will start over again.

Note that the performance of the basic detector is fully analyzed in our work [18] under varying network size, against the short-term unfairness, and in the situation when both UDP and TCP traffic exists. In this paper, we consider the new issue of enhancing the basic detector with an adaptive design.

B. Adaptive Detector

The basic detector (1) can be conveniently enhanced to incorporate adaptive operations. The basic idea is that at a certain state, that if the tagged node is inferred to be misbehaving, the detector can increase its value more aggressively for a shorter detection delay; otherwise, the detector can decrease more aggressively to mitigate false positives. Let X_n denote the state of the detector at time n (i.e., the state after processing n observation samples). We use u_n to denote the adaptive actions associated with the state X_n. In this paper, u_n can take the value of a positive integer, a negative integer, or 0. How to properly determine the action for each state will be discussed in the following section. With the adaptive actions, the behavior of the adaptive detector can be mathematically described as

$$\begin{aligned} X_{n+1} &= (X_n + (NI_n + u_n - 1))^+ \\ X_0 &= 0. \end{aligned} \tag{3}$$

Also, the adaptive detector has the same decision rule as (2). And similarly, the detector value will be reset to 0 once a detection event happens.

IV. Markov Decision Process Based Modeling

Consider the sequence of the adaptive detector value $\{X_n\}$ as a discrete random process, which takes values from a finite set $\mathcal{A} = \{0, 1, 2, ..., h\}$. The process is said to be in state i at time n if $X_n = i$. We can see that both the basic detector (1) and the adaptive detector (3) have the Markov property, that is, given the current state X_n, the next state X_{n+1} is independent of previous states. We have applied Markov chain analysis to study the basic detector in [18]. We here apply the Markov decision process to study the adaptive detector.

Given an action $u_n = u$ associated with the state X_n, the transition probability can be expressed as

$$P_{ij}(u) = P\{X_{n+1} = j | X_n = i, u_n = u\}. \tag{4}$$

A MDP is a 4-tuple: $(\mathcal{A}, \mathcal{U}, P_{ij}(u), R_{ij}(u))$.

- $\mathcal{A}$ is a finite set of states. In this paper, the detector value X_n is defined as the state and takes values from the set $\mathcal{A} = \{0, 1, 2, ..., h\}$.
- $\mathcal{U}$ is a finite set of actions. The action is the value of the adjustment u_n chosen at the state X_n. Given positive integer u_{max}, we consider the action set consisting of the integers in the range $[-u_{max}, u_{max}]$. How to select the action u_n according to the state X_n is called a *policy*.
- $P_{ij}(u)$ is the transition probability given that action u is taken at state X_n.
- $R_{ij}(u)$ is the reward received with the transition from state i to state j under the action u.

In the following, we will discuss how to determine the transition probability $P_{ij}(u)$, design the reward function $R_{ij}(u)$, as well as decide the optimal policy.

A. Transition Probability

The transition probability is the critical element for analyzing a Markov model. In our detection system, a state transition happens when a successful transmission over the network is observed, which can either be from the tagged node, i.e., our observation measure $I_n = 1$, or from any other node, i.e., $I_n = 0$. Then if we use q_s to denote the probability that a successful transmission over the network is from the tagged node, the probability distribution of I_n is given by

$$P\{I_n = k\} = \begin{cases} q_s & \text{if} \quad k = 1, \\ 1 - q_s & \text{if} \quad k = 0. \end{cases} \tag{5}$$

In a normal situation that every node uses the same contention window size and follows the 802.11 DCF, it can be seen that $q_s = \frac{1}{N}$ under the independent channel access assumption and fair channel sharing, given N nodes in the network. If the tagged node is a selfish node taking a smaller contention window, it will achieve a q_s larger than $\frac{1}{N}$ and thus a larger portion of the network throughput. In Section IV-B, we will present how to calculate q_s given the contention window size.

Using the distribution of I_n, and also taking into account the value of the detection threshold h and the action u, we can then calculate the transition probability $P_{ij}(u)$. Based on the operation of the adaptive detector, the calculation of $P_{ij}(u)$ is divided into four distinct cases.

Case 1 consists of $P_{ij}(u)$ for $i \in [0, h-1]$ and $j \in \{0, h\}$, with values

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

This case is related to transitions from any state other than state h to state 0 or h. According to the state transition equation (3), the detector variable X_n can only jump from the current state i to two other states, i.e., a larger value $i + N + u - 1$ when $I_n = 1$ and a smaller value $i + u - 1$ when $I_n = 0$. Thus, if the larger value $i + N + u - 1 \leq 0$, X_n will for sure jump to 0; if the smaller value $i + u - 1 \geq h$, X_n will for sure jump to h. 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. Further, if $i + u - 1 \leq 0$ and $i + N + u - 1 > 0$, X_n can only jump from i to 0 when $I_n = 0$. Also, if $i + N + u - 1 \geq h$ and $i + u - 1 < h$, X_n can only jump from i to h when $I_n = 1$.

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

$$P_{ij}(u) = \begin{cases} P\{I_n = 0\} & \text{if } j = i + u - 1 \text{ and} \\ & \ i + u - 1 > 0, \\ P\{I_n = 1\} & \text{if } j = i + N + u - 1 \text{ and} \\ & \ i + N + u - 1 < h, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

This case is related to the typical behavior of the detector, describing the transitions from any state other than state h to any state other than 0 and h. The state can transit to state $i + u - 1$ when $I_n = 0$ or to state $i + N + u - 1$ when $I_n = 1$, according to the state transition equation (3).

Case 3 consists of $P_{ij}(u)$ for $i = h$ and $j \in \{0, h\}$, with values

$$P_{hj}(u) = \begin{cases} 1 & \text{if } j = 0 \text{ and } N + u - 1 \leq 0, \\ P\{I_n = 0\} & \text{if } j = 0,\ u - 1 \leq 0 \text{ and} \\ & \ N + u - 1 > 0, \\ P\{I_n = 1\} & \text{if } j = h,\ u - 1 < h \text{ and} \\ & \ N + u - 1 \geq h, \\ 1 & \text{if } j = h \text{ and } u - 1 \geq h, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Case 4 consists of $P_{ij}(u)$ for $i = h$ and $j \in [1, h-1]$, with values

$$P_{hj}(u) = \begin{cases} P\{I_n = 0\} & \text{if } j = u - 1 \text{ and} \\ & \ u - 1 > 0, \\ P\{I_n = 1\} & \text{if } j = N + u - 1 \text{ and} \\ & \ N + u - 1 < h, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Cases 3 and 4 are related to transitions from state h. Such transitions are singled out as state h is special. According to the detector operation, we reset the detector value X_n to 0 right after it reaches h. This transition from h to 0 is not triggered by any successful transmission over the network and the two values happen in the same sampling period. This state has the value of h when it is entered and the value of 0 when it is left. Therefore, the transition probabilities as shown in Cases 3 and 4 are in fact equivalent to those associated with state 0 contained in Cases 1 and 2, respectively. Note that the the policy at state h will also be the same as that at state 0. The reason we still maintain the state h is that it is the state to trigger the alarm of a misbehavior detected.

B. Reward Function

With a MDP formulation, the optimal policy at each state will be solved from an optimization problem that maximizes a reward function. Thus, the reward function needs to be properly designed. It should have the property that a positive reward will be collected for a right action that can improve the performance, while a negative penalty will be given to an improper action that degrades the performance. Regarding our problem of misbehavior detection, the evaluation of an action will depend on the analysis of the node behavior. We use $P\{M|X_n = i\}$ to denote the probability that the tagged node is misbehaving, given the current detector state $X_n = i$. We propose to define a reward function for an action u in state i as

$$R\{i, u\} = -(1 - P\{M|X_n = i\})u + P\{M|X_n = i\}u. \tag{10}$$

We can see that the reward function will encourages choosing a positive value for the action u when the tagged node is misbehaving, represented by the positive reward $P\{M|X_n = i\}u$, where the positive u represents an aggressive increase towards fast detection. On the other hand, the reward function stimulates the selection of a negative value for u when the tagged node is a normal one, represented by the positive reward $-(1 - P\{M|X_n = i\})u$, where the reward is for an aggressive decrease to mitigate the false positives. The reward function is represented as a probabilistic average, considering the randomness in practice. It is not a trivial issue to calculate the probability $P\{M|X_n = i\}$ though.

We consider a general multiple attacker scenario for the analysis. Specifically, we assume that each node in a network of N nodes could be a misbehaving one independently with a probability of Q, i.e., $P\{M\} = Q$. With good majority, normally $Q < 50\%$. We further assume that when a node is misbehaving, it randomly chooses the value of its minimum contention window CW_{min} from a set $\mathcal{W}$, with a uniform probability of $\frac{1}{|\mathcal{W}|}$, where $|\mathcal{W}|$ is the cardinality of the set $\mathcal{W}$. The CW_{min} defined in the IEEE DCF standard is 32, and all the elements $\mathcal{W}$ are less than 32. According to the contention windows selected, the nodes in the network can be divided into S classes. All the nodes in the same class use the same CW_{min}. Specifically, we denote that a class i node uses $CW_{min} = W_i$ with $W_i \in \{\mathcal{W}, 32\}$. If each class has N_i (≥ 0) nodes, we have $\sum_{i=1}^{S} N_i = N$.

With our model, we are to calculate the probability $P\{X_n = i\}$, which is the steady state probability that the detector X_n stays in state i when monitoring a node, averaging all possible misbehaving cases in the network. We will also calculate the probability $P\{X_n = i|M\}$, which is the steady state probability that the detector X_n stays in state i, given that a misbehaving node is monitored. Note that the two probabilities will be calculated with $u = 0$, which represents the performance if no extra adaptive action is taken and serves as the standpoint for the optimal policy design. The probabilities will allow us to determine the reward function through calculating the probability

$$P\{M|X_n = i\} = \frac{P\{X_n = i|M\}P\{M\}}{P\{X_n = i\}}. \tag{11}$$

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 node transmits at a random time slot and p_c^i denote the collision probability of a given transmission from a class i node. We further let m denote the maximum backoff stage. Given the number of nodes in each class $N_1, \cdots, N_S$ and the CW_{min} taken by each class $W_1, \cdots, W_S$, according to [1], we have the following equations:

$$\begin{cases} p_t^1 = \dfrac{2(1-2p_c^1)}{(1-2p_c^1)(W_1+1)+p_c^1W_1(1-(2p_c^1)^m)} \\ \vdots \\ p_t^S = \dfrac{2(1-2p_c^S)}{(1-2p_c^S)(W_S+1)+p_c^SW_S(1-(2p_c^S)^m)} \\ p_c^1 = 1-(1-p_t^1)^{N_1-1}\displaystyle\prod_{i=2}^{S}(1-p_t^i)^{N_i} \\ \vdots \\ p_c^S = 1-(1-p_t^S)^{N_S-1}\displaystyle\prod_{i=1}^{S-1}(1-p_t^i)^{N_i} \end{cases} \tag{12}$$

from which the parameters p_t^i, p_c^i of each class i node 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 (12), we can obtain the probability that a node gets a successful transmission at a random time slot:

$$p_s^1 = p_t^1(1-p_c^1), \tag{13}$$

$$\vdots$$

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

Suppose that the tagged node belongs to class v. We can then calculate the probability q_s^v that a successful transmission over the network is from the tagged node with $CW_{min} = W_v$ as

$$q_s^v = \frac{p_s^v}{\sum_{i=1}^{S} N_i p_s^i}. \tag{15}$$

Note that q_s^v is a conditional probability given a specific configuration of CW_{min} and number of nodes in each class, i.e.,

$$q_s^v(\cdot) = f(W_1, \cdots, W_S, N_1, \cdots, N_S). \tag{16}$$

We can further analyze other conditional and unconditional successful transmission probability resorting to our misbehaving model. Recall that in our model every node misbehaves with a probability Q and the node who misbehaves randomly chooses its CW_{min} from the set $\mathcal{W}$. Let $P(W_i)$ denote the probability that a node sets its CW_{min} as W_i $(\in \mathcal{W})$. We then have

$$P(W_i) = \begin{cases} \dfrac{Q}{|\mathcal{W}|} & \text{if} \quad W_i \in \mathcal{W} \\ 1-Q & \text{if} \quad W_i = 32 \end{cases}. \tag{17}$$

Let $\bar{q}_s(W_v)$ denote the conditional probability that a successful transmission is from the tagged node, given that the tagged node has a CW_{min} size of W_v. Let $\bar{q}_s$ denote the unconditional probability that a successful transmission is from the tagged node, averaging over all the possible network scenarios. We can see that

$$\bar{q}_s(W_v) = \sum_{\substack{W_i \in \mathcal{W}: \\ i \neq v \text{ and} \\ W_i \neq W_j \text{ for } i \neq j}} \sum_{\substack{(N_1,\cdots,N_S): \\ N_v \geq 1 \text{ and} \\ \sum_i N_i = N}} q_s^v(\cdot)P(W_v)^{(N_v-1)} \prod_{i=1, i\neq v}^{S} P(W_i)^{N_i} \tag{18}$$

and further

$$\bar{q}_s = \frac{Q}{|\mathcal{W}|}\sum_{W_v \in \mathcal{W}} \bar{q}_s(W_v) + (1-Q)\bar{q}_s(32). \tag{19}$$

In (18) and (19), we consider the different combinations of the CW_{min} taken by each class and the number of nodes in each class.

Applying $\bar{q}_s$ to (5), we can obtain the probability distribution of our observation measure I_n for a tagged node. We can then calculate $P\{X_n = i\}$ with $u = 0$ as mentioned above. By using the I_n distribution in (6)−(9), we can compute the transition probabilities $P_{ij}(0)$. Let $(\pi_0, ..., \pi_h)$ denote the steady state probabilities when $u = 0$, which can be solved from the equations

$$\pi_j = \sum_{i=0}^{h} \pi_i P_{ij}(0), \qquad j \in \{0, ..., h\}, \tag{20}$$

$$\sum_{j=0}^{h} \pi_j = 1. \tag{21}$$

We can then get $P\{X_n = i\} = \pi_i$ from (20) and (21).

Similarly, if we apply $\bar{q}_s(W_v)$ obtained in (18) to (5) and further compute the transition probabilities and then solve the equations (20) and (21) based on such transition probabilities, we will obtain the steady state probability $P\{X_n = i|CW_{min} = W_v\}$, which is the distribution of the detector state given that the tagged node takes a CW_{min} of W_v. Then as the tagged node may chooses its CW_{min} uniformly from $\mathcal{W}$ when it is misbehaving, we can calculate $P\{X_n = i|M\}$ as

$$P\{X_n = i|M\} = \frac{P\{X_n = i, M\}}{P\{M\}} = \frac{1}{|\mathcal{W}|} \sum_{W_v \in \mathcal{W}} P\{X_n = i|CW_{min} = W_v\}. \qquad (22)$$

With the probabilities $P\{X_n = i|M\}$ and $P\{X_n = i\}$, we can calculate the probability $P\{M|X_n = i\}$ according to (11) and then obtain the reward function by (10).

C. Optimization Problem Formulation

Our goal is to determine the optimal policy, i.e., how to choose the action at a certain state, to achieve the maximum benefit based on reward function developed above. In particular, we will find the steady-state probability π_{iu} of being in state i and choosing action u when the optimal policy is used. Hence, given that the state space is determined by a certain detection threshold h, i.e., $i, j \in [0, h]$, the problem can be formulated as

$$\begin{aligned} &\max \sum_i \sum_u \pi_{iu} R(i,u) \\ &\text{subject to} \sum_i \sum_u \pi_{iu} = 1, \\ &\qquad \sum_u \pi_{ju} = \sum_i \sum_u \pi_{iu} P_{ij}(u) \quad \text{for all } j, \\ &\qquad \pi_{ju} \geq 0 \quad \text{for all } i,\ u. \end{aligned} \qquad (23)$$

This is in fact a linear programming problem. Solving (23) using the simplex method, we can obtain the set of π^*_{iu} maximizing the overall reward, which also indicates the optimal policy of how to choose an action u at state i. We also find that for each i, π^*_{iu} is zero for all but one value of u, which is due the property of the linear programming problem (Section 4.10 of [4]). Thus there is only one action, i.e., u^*, to be taken for a state in the optimal policy, which indicates how our adaptive detector will operate under a certain detection threshold h. Then we compare the total reward of various detection thresholds and select an h with the largest reward to be our detection threshold. As an example, for a network with $N = 8$ nodes, each of which tends to misbehave with a probability of $Q = 0.25$, we determine a detection threshold of $h = 20$ which achieves the largest reward, and also obtain the optimal configuration u^* associated with the h.

V. Theoretical Performance Analysis

In this section, we conduct theoretical performance analysis of the adaptive detector whose operation is characterized by the optimal policy obtained from last section, in terms of three fundamental metrics: *average false positive rate*, *average detection delay*, and *missed detection ratio* under a detection delay bound. In fact, with the optimal policy for each state is obtained, the MDP will reduce to a Markov chain. When we analyze this Markov chain in a specific scenario, we can numerically evaluate the performance of the adaptive detector in that case. In this section, we will also compare the performance of the adaptive detector to that of the basic detector.

A. Average False Positive Rate

The average false positive rate P_{fp} is the rate that the detector X_n hits state h given the fact that no node in the network is misbehaving. Such a rate is equal to the steady-state probability, denoted by π^*_h, that the adaptive detector stays at h in the normal condition.

In the normal condition with a fair share of the channel access, we have $q_s = \frac{1}{N}$ for a tagged node. We can calculate the distribution of I_n according to (5). And further, using the optimal u^* for each state obtained from (23), we calculate the transition probabilities $P_{ij}(u^*)$ according to (6)−(9). Then using (20) and (21), we can get π^*_h and subsequently the average false positive rate as

$$P_{fp} = \pi^*_h. \qquad (24)$$

With a detection threshold $h = 20$, we obtain P_{fp} as 0.0013.

B. Average Detection Delay

The average detection delay $E[T_D]$ is the average number of samples observed from the moment that the tagged node starts to misbehave until the misbehavior is detected. With the transition probabilities under the abnormal condition, i.e., $\hat{P}_{ij}(u^*)$, $E[T_D]$ can be computed as the expected number of transitions required for the detector X_n to hit state h, starting from the moment when the misbehavior starts. To carry out the analysis, we need to find $\hat{P}_{ij}(u^*)$ and determine the initial state of X_n when the misbehavior starts.

1) Transition Probabilities under Misbehavior: Given the number of nodes and the minimum contention window size CW_{min} of each misbehaving node, we can calculate the probability that a successful transmission is from a tagged misbehaving node through (12)−(15). Given the optimal policy u^* for each state, we can further obtain the transition probabilities $\hat{P}_{ij}(u^*)$ according to (6)−(9).

2) Initial States: Before a selfish 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 transition probabilities $P_{ij}(u^*)$ except for state h, as we do not consider an already "alarmed" state as an initial state.

We can calculate the steady state probabilities under the normal condition through (20) and (21). Since we are interested in detection starting from an unalarmed state, under such a constraint the conditional initial state probabilities should be

$$\pi'^*_i = \frac{\pi^*_i}{\sum_{j=0}^{h-1} \pi^*_j} \quad \text{for } i \in \{0, ..., h-1\}. \qquad (25)$$

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 probabilities $\hat{P}_{ij}(u^*)$.

Let μ_{ih}, $i \in [0, h-1]$, denote the expected number of transitions for state i to state h. According to [6], the values of μ_{ih} can be solved from the equations

$$\mu_{ih} = 1 + \sum_{r \neq h} \hat{P}_{ir}(u^*)\mu_{rh}, \qquad i \in \{0, ..., h-1\} \tag{26}$$

Based on the solutions of (25) and (26), we can obtain the average detection delay $E[T_D]$ as

$$E[T_D] = \sum_{i=0}^{h-1} \pi_i'^* \mu_{ih}. \tag{27}$$

TABLE I

Average detection delay of the adaptive detector with $P_{fp} = 0.0013, h = 20$

	$W_2 = 2$	$W_2 = 4$	$W_2 = 8$	$W_2 = 16$
$W_1 = 2$	12.8151	4.321	4.096	4.0168
$W_1 = 4$	$1.98*10^4$	16.2949	7.3324	6.1042
$W_1 = 8$	$7.11*10^5$	270.0457	31.1042	17.4963
$W_1 = 16$	$1.59*10^7$	$5.61*10^3$	307.9787	111.3373

As an example, we consider a network with $N = 8$ nodes; among them there are two nodes, denoted as node 1 and node 2, misbehaving. We use the same optimal parameter configuration as in the example in Section V-A. We treat node 1 as the tagged node and compute $E[T_D]$ for detecting the node. Varying the CW_{min} of both of the misbehaving nodes, the delays for detecting node 1 are shown in Table I. Here W_1 denotes the CW_{min} of node 1 and W_2 denotes the CW_{min} of node 2. From the table, as expected, we can see that more intense misbehavior of node 1, i.e., a smaller W_1, leads to a shorter detection delay for node 1; however, more intense misbehavior of node 2 leads to a longer detection delay for node 1. Note that when $W_1 = W_2$, the two misbehaving nodes are "fairly" competing with each other to transmit. But once W_2 becomes larger than W_1, the detection delay for node 1 incurs a drastic decline, as node 1 starts to gain advantages over node 2 and gets more opportunities to transmit. Moreover, we see that for the cases where $W_1 > W_2$, the detection delays for node 1 become very large since node 2 is gaining advantages over node 1. Such long detection delays do not make much practical sense. We will discuss our method for dealing with the multiple misbehaving nodes scenarios in Section V-D. And later in our simulation results, we will demonstrate the effectiveness of the method.

We then compare the detection performance of the adaptive detector to the basic detector with $u = 0$ in each state, to examine whether the adaptive detector has indeed achieved better performance. Here we still consider the same network setting where there are 2 misbehaving nodes in a network with $N = 8$ nodes.

TABLE II

Average detection delay of the basic detector with $P_{fp} = 0.0013, h = 70$

	$W_2 = 2$	$W_2 = 4$	$W_2 = 8$	$W_2 = 16$
$W_1 = 2$	22.4527	9.9944	9.5412	9.3783
$W_1 = 4$	$2.90 * 10^7$	26.1878	15.2037	13.2251
$W_1 = 8$	$5.67*10^{12}$	174.044	38.721	27.3762
$W_1 = 16$		$3.18*10^5$	198.6957	85.3163

Table II shows the detection delays of the basic detector for detecting node 1, giving the same false positive constraint of $P_{fp} = 0.0013$. To achieve this false positive rate, the basic detector needs to set its detection threshold as $h = 70$. Comparing the results in Table II to Table I, we can see that for most cases of $W_1 \leq W_2$, the adaptive detector can achieve much quicker detection delay than the basic detector. The reason for this better performance is that when the detector is in smaller states, the optimal policy mostly instructs it to choose a negative u; whereas in larger states, the optimal policy mostly instructs the detector to choose a positive u. As a result, it is more difficult for the detector to increase its value in the normal condition since initially the detector value X_n is small and there are relatively few transmissions from the tagged node to increase X_n. However, when the tagged node starts to misbehave, consecutive transmissions from the node trigger X_n to become large, and the optimal policy at this time make the increase even greater, resulting in quicker detection delay. Overall, the adaptive detector is able to achieve better performance in both false positive and detection delay. Note that we do not show the result for the case of $W_1 = 16$ and $W_2 = 2$, as the number is too large to be considered as practically detectable.

C. Missed Detection Ratio

According to (3) and (2), our detector only stops when the misbehavior is detected. Thus we need to examine the missed detection ratio P_{md} under a given detection delay constraint D, which is an important performance measure regarding the 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. Let $\hat{\mathbf{P}}^*$ denote the transition probability matrix of $\hat{P}_{ij}(u^*)$. With $\hat{\mathbf{P}}^*$, P_{md} 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 (25), setting

$$P_i(0) = \pi_i'^* \quad \text{for } i \in \{0, ..., h-1\}, \tag{28}$$

$$P_h(0) = 0. \tag{29}$$

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

$$\vec{P}(j) = \vec{P}(j-1) \cdot \hat{\mathbf{P}}^*, \tag{30}$$

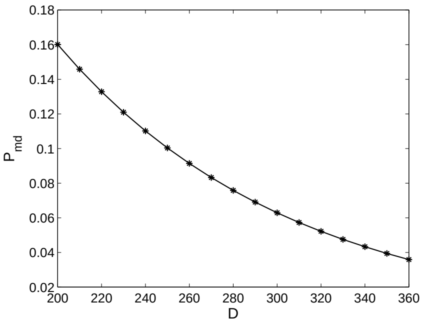


Fig. 1. Missed detection ratio for 1 node with $CW_{min} = 16$.

$$P_h(j) = 0. \tag{31}$$

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 then be obtained as

$$P_{md} = \sum_{i=0}^{h-1} P_i(D). \tag{32}$$

Fig. 1 demonstrates the missed detection ratio P_{md} of detecting 1 of 2 misbehaving nodes both with $CW_{min} = 16$, in a network of 8 nodes. As shown in the figure, to detect such moderate misbehavior, we can set $D = 330$ to obtain a P_{md} less than 0.05.

D. Multiple Misbehaving Nodes Scenario

In cases where multiple misbehaving nodes exist in a wireless network, our detector can detect the misbehaving nodes one by one. We monitor the transmissions from every node using a separate detector. As the node with the most intense misbehavior has the most opportunities to transmit, we will be able to quickly detect it with our detector. After that, we will discard packets from that node and continue the detection among the leftover nodes. Clearly, the node with the second most intense misbehavior will be detected at this time. We will continue such detection iteratively to identify every misbehaving node. The effectiveness of this approach will be demonstrated in Section VI-B.

What needs to be noted here is that we are not assuming eliminating the wireless channel access of those nodes with more intense misbehavior after they are detected, as practically it is not easy to achieve such elimination, and the detected misbehaving nodes can continue transmitting packets to keep on impacting contentions in the network. The specifics of how to eliminate misbehaving nodes, however, are out of the scope of this paper. In our scheme, whether to eliminate the nodes from the network or not is not a problem for continuing detecting other nodes with less intense misbehavior. The detector can just drop the packets from those nodes already detected so that the detector is shielded from the impact of those nodes. This is an special advantage of our detector over existing misbehavior detectors [16], [17] in detecting multiple misbehaving nodes. The observation measures of those detectors have to include information from the whole network, and thus transmissions from any node in the network will always have impact on the detection of a tagged misbehaving node. The "shielding" option is not available to those detection methods.

VI. Simulation Results

A. Comparison with Analytical Results

We first establish an 802.11 DCF based wireless network consisting of 8 competing nodes ($N = 8$) and an AP through ns-2 simulation [7]. The network works under the saturated condition and every node transmits packets over the User Datagram Protocol (UDP) towards the AP. The AP also acts as the detection agent which monitors the transmissions from every competing node with a separate detector. The nodes are located close enough to sense the transmissions from each other and thus avoid the hidden terminal problem. There are 2 misbehaving nodes, marked as node 1 and node 2, among the 8 competing nodes, the same as in the theoretical analysis. The 2 misbehaving nodes can manipulate their minimum contention window to a value from $\{2, 4, 8, 16, 32\}$.

TABLE III
Average detection delay of the adaptive detector for $W_1 \leq W_2$ with $h = 20$

	$W_2 = 2$	$W_2 = 4$	$W_2 = 8$	$W_2 = 16$
$W_1 = 2$	8.3378	5.518	5.5385	5.6583
$W_1 = 4$		9.9695	5.6956	5.5825
$W_1 = 8$			20.2828	12.655
$W_1 = 16$				79.2084

Through simulation, we obtain average detection delays of the adaptive detector, and the results for detecting node 1 are shown in Table III. Note that we do not include the cases of $W_1 > W_2$ in this table, as most of them are not practically detectable in the first place. We will discuss this issue in next subsection.

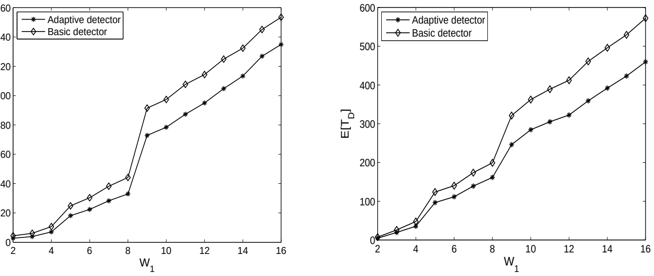
Given the same detection threshold of $h = 20$, we compare the simulation results in Table III to the analytical results in Table I. Even though the two sets of results are close, generally the simulation results are smaller than the analytical results. The reason is that our analysis is based on the assumption of independent channel access; however, in a practical 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 [8], which implies correlations among the channel accesses. As a result, a selfish misbehaving node can obtain even more channel accesses in addition to those resulting from the misbehavior. Therefore, the average detection delays obtained from simulations are shorter. However, the analytical results still provide a conservative estimation for the detector's performance, which is meaningful for us to guide the detection system configuration.

B. Multiple Misbehaving Nodes Scenario

In the cases of $W_1 > W_2$, the misbehaving node 1 is practically not detectable simultaneously with node 2 as its

TABLE IV
AVERAGE DETECTION DELAY OF THE ADAPTIVE DETECTOR FOR $W_1 > W_2$ WITH $h = 20$

	$W_2 = 2$	$W_2 = 4$	$W_2 = 8$	$W_2 = 16$
$W_1 = 2$				
$W_1 = 4$	38.3886			
$W_1 = 8$	70.3503	66.9261		
$W_1 = 16$	243.7508	197.8860	131.2686	



(a) $W_2 = 4$, $W_3 = 8$, $W_4 = 16$. (b) $W_2 = 2$, $W_3 = 4$, $W_4 = 8$.

Fig. 2. Average detection delay in multiple misbehaving nodes scenarios.

channel accesses are much less than node 2. However, as node 2 can be detected quickly in these cases, after the detection of node 2, we can simply discard the transmissions from node 2 and only continuing our detection among the remaining 7 nodes to detect node 1. As we monitor the transmissions from every competing node with a separate detector, this approach is very easy to implement. Table IV shows the results obtained from the approach. The detection delays are the original delays to detect node 2 plus the additional delays to detect node 1 after transmissions from node 2 are discarded. As we can see, the results are significantly smaller than the analytical results from Table I, which makes the detection of multiple misbehaving nodes much more feasible.

Then to be more general, we consider a network of 20 nodes with 4 nodes misbehaving, one of which is the tagged node. The tagged node varies its minimum contention window size W_1 from 2 to 16, while the other 3 misbehaving nodes set their windows as W_2, W_3 and W_4 respectively to represent a wide range of misbehavior. Fig. 2(a) shows the average delays for detecting the tagged node when $W_2 = 4$, $W_3 = 8$ and $W_4 = 16$, representing a more moderate misbehaving environment; Fig. 2(b) shows the results when $W_2 = 2$, $W_3 = 4$ and $W_4 = 8$, representing a more intense misbehaving environment. The sudden increases of delays in the figures, e.g., when $W_1 = 5$ and 9 in Fig. 2(a), are due to the reason that starting from these window values, there will be one more misbehaving node detected before the tagged node, which obviously increases the delay. Overall, we can see that the adaptive detector is faster than the basic detector in all the misbehaving cases. Also as expected, more intense misbehavior from other nodes makes it longer to detect the tagged node.

VII. CONCLUSION

In this paper, we propose an adaptive approach to address real-time selfish misbehavior detection in IEEE 802.11 based wireless networks. By enhancing a basic detector based on the non-parametric CUSUM test, we design an adaptive detector where actions are added to control how the detector value is updated. Further, we model the adaptive detector using the Markov decision process, which enables us to find an optimal policy to determine the proper action to be taken in each state of the adaptive detector. The optimal policy characterizes the operation of the adaptive detector, which also enables us to theoretically analyze the detection performance. The simulation results confirm the accuracy of our analysis, and also demonstrate the detector's ability to address the cases of multiple misbehaving nodes. In our future work, we plan to extend the work to the detection of selfish misbehavior in multi-hop wireless networks.

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

[3] B. Brodsky and B. Darkhovsky, *Nonparametric methods in change-point problems*. Kluwer Academic Publisher, 1993.

[4] S. M. Ross, *Introduction to Probability Models*. Elsevier Academic Press, 9th edtion, 2007.

[5] J. Tang, Y. Cheng, and W. Zhuang, "An analytical approach to real-time misbehavior detection in IEEE 802.11 based wireless networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1638–1646.

[6] J. R. Morris, *Markov chains*, Cambridge Univ. Press, 1997.

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

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

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

[10] J. Konorski, "Protection of fairness for multimedia traffic streams in a non-cooperative wireless LAN setting," in *Proc. PROMS*, 2001.

[11] J. Konorski, "Multiple access in Ad-hoc wireless LANs with noncooperative stations," in *Proc. NETWORKING*, 2002.

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

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

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

[15] 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.

[16] 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.

[17] 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.

[18] J. Tang, Y. Cheng, and W. Zhuang, "Real-time misbehavior detection in IEEE 802.11 based wireless networks: an analytical approach," *IEEE Trans. Mobile Comput.*, accepted, preprint version available in IEEEXplore.