

Chapter 2

Real-Time Misbehavior Detection in IEEE 802.11TM: An Analytical Approach

Abstract In this chapter, we address the selfish misbehavior in the IEEE 802.11TM based wireless network. After a brief description on selfish misbehavior in 802.11TM, we first design a real-time backoff misbehavior detector, termed as the fair share detector (FS detector), which exploits the non-parametric cumulative sum (CUSUM) test to quickly find a selfish malicious node without any a priori knowledge of the statistics of the selfish misbehavior. We then develop a Markov chain based analytical model to systematically study the performance of the FS detector. Based on the analytical model, we can quantitatively compute the system configuration parameters for guaranteed performance in terms of average false positive rate, average detection delay and missed detection ratio under a detection delay constraint. We present simulation results to confirm the accuracy of our theoretical analysis as well as demonstrate the performance of the FS detector.

2.1 Selfish Misbehavior in 802.11TM

2.1.1 IEEE 802.11TM DCF

There are two major functions in the IEEE 802.11TM protocols: the point coordination function (PCF) and the distributed coordination function (DCF). The PCF is a centralized function and is an optional feature in 802.11TM. In this book, 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 $\{0, 1, \dots, CW_{min} - 1\}$, where CW_{min} is called the minimum con-

tention 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.

2.1.2 Backoff Selfish Misbehavior in IEEE 802.11TM DCF

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 802.11TM wireless network.

2.2 Fair Share Detector Design

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

2.2.1 The Observation Measure

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

$$P\{I^v = k\} = \begin{cases} q_s^v & \text{if } k = 1, \\ 1 - q_s^v & \text{if } k = 0. \end{cases} \quad (2.1)$$

In a normal situation that every node uses the same contention window size and follows the 802.11TM 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 Sect. 2.4, we will present how to calculate q_s^v given the contention window size. The distribution of I^v in (2.1) is the basis to establish our analytical model.

Remark 2.1: In an 802.11TM network, a node that has just accomplished a successful transmission will have advantages in grabbing the channel for next transmission in a short period [3]. This is referred to as *short-term unfairness* and is inherent to the 802.11TM backoff mechanism. Such an issue implies correlations among the channel accesses, which may impact the accuracy of (2.1) to model the successful transmission of the tagged node based on the assumption of independent channel access. In our preliminary work [5], we apply a shuffling mechanism to observation samples to mitigate the impact of short-term unfairness. In Sect. 2.5, we will show with detailed analysis that the FS detector is inherently robust against short-term unfairness, and the detection based on (2.1) does give accurate decisions. The fairness issue also exists when both user datagram protocol (UDP) and transmission control protocol (TCP) traffic flows exist in the network, where the TCP traffic tends to be overwhelmed by UDP traffic due to its congestion control mechanism. In Sect. 2.5, we will also discuss how to apply the FS detector with a robust performance when both UDP and TCP traffic flows exist in the network.

2.2.2 Fair Share Detector

Let $\{I_n, n = 0, 1, \dots\}$ be the sequence of sample values of I^v , observed each time a successful transmission appears on the channel. Here, we drop the superscript v for easier presentation considering the clear context. Let N denote the number of nodes existing in the network. Suppose that the initial value of our detector X_n is 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 802.11TM DCF model, each node roughly takes turn to transmit; the increase of X_n caused by one successful transmission from the tagged node can then be equally offset by the successful transmissions from other $N - 1$ non-tagged nodes. Thus 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, it is not difficult to see that X_n is going to quickly accumulate to a large positive value.

The behavior of the FS detector can be mathematically described as

$$\begin{aligned} X_n &= (X_{n-1} + (NI_n - 1))^+ \\ X_1 &= 0 \end{aligned} \quad (2.2)$$

where $(x)^+ = x$ if $x \geq 0$ or 0 otherwise. We can see that (2.2) is actually in the form of a non-parametric CUSUM detector [2].

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

$$\delta_n = \begin{cases} 1 & \text{if } X_n \geq h \\ 0 & \text{if } X_n < h \end{cases} \quad (2.3)$$

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

2.3 Markov Chain Based Analytical Model

Consider the sequence $\{X_n\}$ as a discrete random process, which takes values from a finite set $A = \{0, 1, 2, \dots, h\}$. 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 when a successful transmission over the network is observed. According to (2.2), 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

$$P_{ij} = P\{X_n = j | X_{n-1} = i\}. \quad (2.4)$$

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

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

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

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

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

$$P_{0j} = \begin{cases} P\{I_n = 0\} & \text{if } j = 0, \\ P\{I_n = 1\} & \text{if } j = N - 1 \text{ and } \\ & N - 1 \leq h, \\ P\{I_n = 1\} & \text{if } j = h \text{ and } \\ & N - 1 > h, \\ 0 & \text{otherwise.} \end{cases} \quad (2.5)$$

This group is related to the transitions from state 0 to other states. According to the state transition equation (2.2), the detector variable X_n jumps out of state 0 only when the observed successful transmission is from the tagged node, that is, $I_n = 1$. Further, X_n makes a transition to either $N - 1$ or h depending on whether $N - 1$ is greater than h or not. Note that the state h in fact incorporates all possible states $X_n \geq h$, as the detector will raise an alarm when the state hits h .

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

$$P_{ij} = \begin{cases} P\{I_n = 0\} & \text{if } j = i - 1, \\ P\{I_n = 1\} & \text{if } j = i + N - 1 \text{ and } \\ & i + N - 1 \leq h, \\ P\{I_n = 1\} & \text{if } j = h \text{ and } \\ & i + N - 1 > h, \\ 0 & \text{otherwise.} \end{cases} \quad (2.6)$$

This group describes the typical behavior of the detector. The state can transit to left (i.e., to a smaller value) when $I_n = 0$ or to right (i.e., to a larger value) when $I_n = 1$, according to the state transition equation (2.2).

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

$$P_{hj} = \begin{cases} 1 & \text{if } j = 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2.7)$$

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

2.4 Theoretical Performance Analysis

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

2.4.1 Average False Positive Rate

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

In the normal condition with a fair share of the channel access, we have $q_s^v = \frac{1}{N}$ for a tagged node. We can calculate the distribution of I_n according to (2.1), and further obtain the transition probabilities matrix \mathbf{P} according to (2.5)–(2.7).

Let (π_0, \dots, π_h) denote the steady state probabilities of the Markov chain, which can be solved from the equations

$$\pi_j = \sum_{i=0}^h \pi_i P_{ij}, \quad j \in \{0, \dots, h\}, \quad (2.8)$$

$$\sum_{j=0}^h \pi_j = 1. \quad (2.9)$$

Then we can get the average false positive rate

$$P_{fp} = \pi_h. \quad (2.10)$$

The analytical result (2.10) allows us to numerically examine the impact of the fundamental parameter h on the average false positive rate P_{fp} of the FS detector. As an example, we compute the results for a network with $N = 10$ nodes, and the results are illustrated in Fig. 2.1. From the figure, we can observe that a larger h yields a smaller false positive rate, as expected.

2.4.2 Average Detection Delay

In this subsection, we analyze the average detection delay denoted as $E[T_D]$, which is the average number of samples observed from the moment that the tagged node

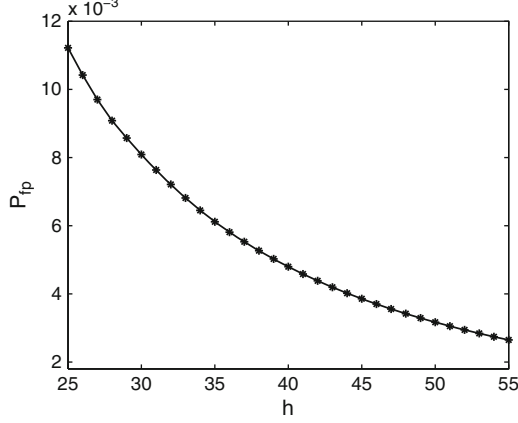


Fig. 2.1 Average false positive rate

starts to misbehave until the misbehavior is detected. With the Markov chain under the abnormal condition (abnormal Markov chain), $E[T_D]$ can be computed as the expected number of transitions required for the state variable to hit state h , starting from the moment when the misbehavior starts. To carry out the analysis, we need to find the transition probabilities of the abnormal Markov chain and determine the initial state of the FS detector when the misbehavior starts.

2.4.2.1 Transition Probabilities Under the Misbehavior

We consider a network consisting of two classes of nodes. Class 1 includes the one misbehaving node with a small minimum contention window CW_{min} denoted as W_1 , and class 0 includes all the normal nodes with the standard minimum contention window denoted as W_0 . According to the classic modeling approach for the 802.11TM DCF [1], we consider that each node independently accesses an idle channel for transmission. Let p_i^j 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:

$$\begin{cases} p_t^0 = \frac{2(1-2p_c^0)}{(1-2p_c^0)(W_0+1) + p_c^0 W_0(1-(2p_c^0)^m)} \\ p_t^1 = \frac{2(1-2p_c^1)}{(1-2p_c^1)(W_1+1) + p_c^1 W_1(1-(2p_c^1)^m)} \\ p_c^0 = 1 - (1-p_t^1)(1-p_t^0)^{N-2} \\ p_c^1 = 1 - (1-p_t^0)^{N-1} \end{cases} \quad (2.11)$$

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

Note that a node can get a successful transmission under the circumstance that there is no collision while the node transmits. Thus from the solutions of (2.11), we can obtain the probability that a node gets a successful transmission at a random time slot:

$$p_s^0 = p_t^0(1 - p_c^0), \quad (2.12)$$

$$p_s^1 = p_t^1(1 - p_c^1). \quad (2.13)$$

We can then calculate the probability \hat{q}_s that a successful transmission over the network is from the malicious node as (2.14):

$$\hat{q}_s = \frac{p_s^1}{p_s^1 + (N-1)p_s^0}. \quad (2.14)$$

Using \hat{q}_s in (2.1), we can obtain the distribution of I_n for the misbehaving node; using such I_n distribution in (2.5)–(2.7), we can then compute the transition probability matrix $\hat{\mathbf{P}}$ for the abnormal Markov chain.

It is worth noting that although we only include two classes of nodes in the above analysis, the model of (2.11)–(2.14) can be easily extended to cases where multiple classes of misbehaving nodes with different intensities of misbehavior exist. This will enable us to analyze much more complicated misbehaving scenarios. We will discuss this issue in Sect. 2.4.4.

2.4.2.2 Initial States

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

We can calculate the steady state probabilities of the normal Markov chain according to (2.8) and (2.9). Since we are interested in detection starting from an unalarmed state, under such a constraint the conditional initial state probabilities should be

$$\pi'_i = \frac{\pi_i}{\sum_{i=0}^{h-1} \pi_i} \quad \text{for } i \in \{0, \dots, h-1\}. \quad (2.15)$$

2.4.2.3 Average Detection Delay

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

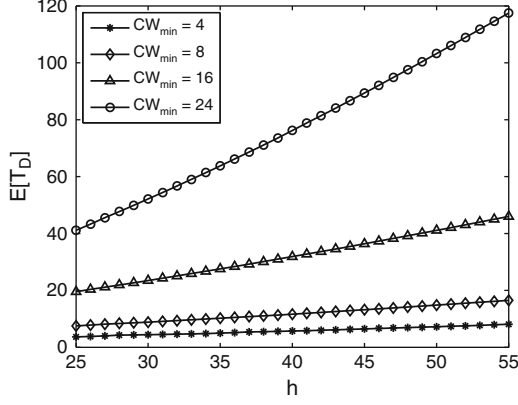


Fig. 2.2 Average detection delay

Let μ_{ih} , $i \in [0, h-1]$, denote the expected number of transitions for state i to state h . According to [4], the values of μ_{ih} can be solved from the equations

$$\mu_{ih} = 1 + \sum_{r \neq h} \hat{P}_{ir} \mu_{rh}, \quad i \in \{0, \dots, h-1\} \quad (2.16)$$

where \hat{P}_{ir} is the transition probability from state i to r of $\hat{\mathbf{P}}$. Based on the solutions of (2.15) and (2.16), we can obtain the average detection delay $E[T_D]$ as

$$E[T_D] = \sum_{i=0}^{h-1} \pi_i' \mu_{ih}. \quad (2.17)$$

The analytical result (2.17) allows us to numerically examine the impact of h on the average detection delay $E[T_D]$ of the FS detector. As an example, we compute the results for a network with $N = 10$ nodes, and the results are illustrated in Fig. 2.2. Specifically, Fig. 2.2 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 yields better performance in average detection delay.

2.4.3 Missed Detection Ratio

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

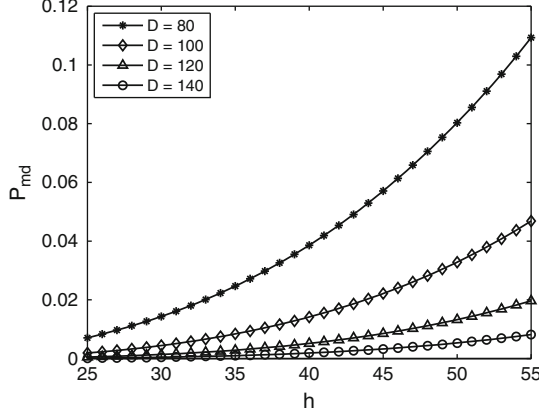


Fig. 2.3 Missed detection ratio

The detection event happens only when X_n hits state h . Thus the missed detection ratio P_{md} under the delay constraint D is the summation of the probabilities of X_n staying at a state other than h at time D . With the transition probability matrix $\hat{\mathbf{P}}$, the missed detection ratio can be computed in an iterative manner. Let the row vector $\mathbf{P}(j) = [P_0(j), \dots, P_h(j)]$ 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 (2.15), setting

$$P_i(0) = \pi'_i \quad \text{for } i \in \{0, \dots, h-1\}, \quad (2.18)$$

$$P_h(0) = 0. \quad (2.19)$$

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

$$\mathbf{P}(j) = \mathbf{P}(j-1) \cdot \hat{\mathbf{P}}, \quad (2.20)$$

$$P_h(j) = 0. \quad (2.21)$$

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

$$P_{md} = \sum_{i=0}^{h-1} P_i(D). \quad (2.22)$$

Figure 2.3 demonstrates the missed detection ratios P_{md} of our analysis under the delay constraints $D = 80$, $D = 100$, $D = 120$ and $D = 140$, for a misbehaving node with the moderate misbehavior of $CW_{min} = 16$. We observe that the larger the delay constraint is, the lower the missed detection ratio will be. In other words, the probability of detection increases with a cost of longer delay. Also, a smaller detection threshold h yields a lower missed detection ratio.

2.4.4 Discussion on Detection of Multiple Misbehaving Nodes

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

Consider a network of N nodes, k of which are malicious and the rest are normal. Suppose that malicious node i sets its minimum contention window CW_{min} as W_i and all the $N - k$ normal nodes use the standard minimum contention window denoted as W_0 . We can expand (2.11) to have the following equations:

$$\begin{cases} p_t^0 = \frac{2(1 - 2p_c^0)}{(1 - 2p_c^0)(W_0 + 1) + p_c^0 W_0(1 - (2p_c^0)^m)} \\ \vdots \\ p_t^k = \frac{2(1 - 2p_c^k)}{(1 - 2p_c^k)(W_k + 1) + p_c^k W_k(1 - (2p_c^k)^m)} \\ p_c^0 = 1 - \prod_{i=1}^k (1 - p_t^i)(1 - p_t^0)^{N-k-1} \\ \vdots \\ p_c^k = 1 - \prod_{i=1}^{k-1} (1 - p_t^i)(1 - p_t^0)^{N-k} \end{cases} \quad (2.23)$$

From the solutions of (2.23), we can obtain the probability that a node gets a successful transmission at a random time slot:

$$p_s^0 = p_t^0(1 - p_c^0), \quad (2.24)$$

$$\vdots$$

$$p_s^k = p_t^k(1 - p_c^k). \quad (2.25)$$

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

$$\hat{q}_s^l = \frac{p_s^l}{\sum_{i=1}^k p_s^i + (N - k)p_s^0}. \quad (2.26)$$

Using q_s^l in (2.26), we can obtain the transition probability matrix of the abnormal Markov chain $\hat{\mathbf{P}}^l$; using $\hat{\mathbf{P}}^l$ and initial states of the detector when misbehavior starts, which are determined in the same way as in Sect. 2.4.2.2, we can analyze average detection delay and missed detection ratio for the tagged malicious node l accordingly.

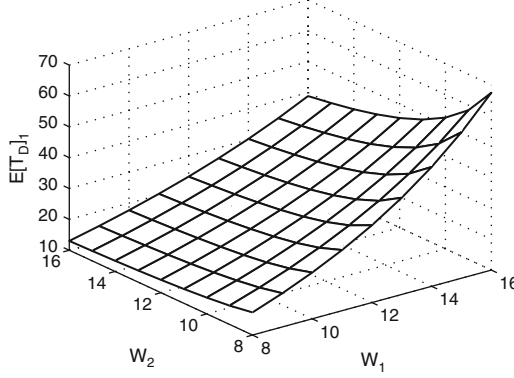


Fig. 2.4 Average detection delay under two misbehaving nodes

We consider an example that there are two misbehaving nodes in a network of 10 nodes, one setting its minimum contention window as W_1 and the other as W_2 . Figure 2.4 plots the average detection delays to identify the misbehaving node 1, denoted as $E[T_D]_1$, under different misbehaving intensity pairs (W_1, W_2) . Note that even in this simple case the two malicious nodes are competing with each other. There is a trade-off between the two nodes. Certainly it takes longer to detect one malicious node if the other chooses more intense misbehavior. It will be an interesting problem to determine how the multiple malicious nodes can find certain misbehaving strategies to collaboratively maximize their collective benefit from the network throughput while avoiding being detected as long as possible. In next chapter, we will also carry out in-depth studies of the scenario with multiple misbehaving nodes when our more advanced adaptive detector is considered.

2.4.5 System Configuration Under Performance Constraints

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

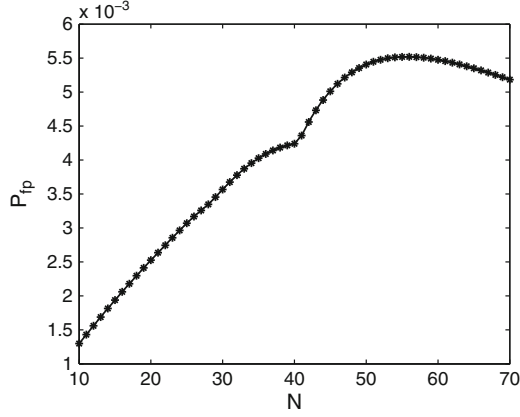


Fig. 2.5 Impact of network size change on average false positive rates at $h = 80$

samples, and the missed detection ratio of 0.0141 with the delay constraint $D = 100$. In Sect. 2.5, we will use simulation results to demonstrate that our target performance measures are indeed achievable.

2.4.6 Detection Performance Against Network Size Change

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

Figure 2.5 shows the average false positive rates P_{fp} of the detector versus the number of nodes N , at $h = 80$. The threshold h is intentionally set to be greater than the maximum number of nodes to avoid alarm being triggered by just one successful transmission from the tagged node. As shown in Fig. 2.5, there is a dent on the curve at $N = 40$ and P_{fp} has a sharper increase when N gets greater than 40. This is because, when $N \leq 40$, at least three or more consecutive successful transmissions from the tagged node are needed to drive X_n to h from an initial state of 0, raising a false alarm; however, when $41 \leq N \leq 70$, it will take only two consecutive transmissions to reach h , which largely increases the possibility of false positive. Furthermore, note that P_{fp} does not monotonically increase with N and has an upper bound of $P_{fp} = 0.0055$. The explanation is that, when the number of nodes contending for the channel becomes larger, the transmissions from a tagged node are more likely to be interrupted by transmissions from those non-tagged nodes, and the accumulation of the detector X_n will be more aggressively offset by such non-tagged nodes, thus resulting in a smaller P_{fp} . If the target performance of $P_{fp} \leq 0.0055$ is allowed,

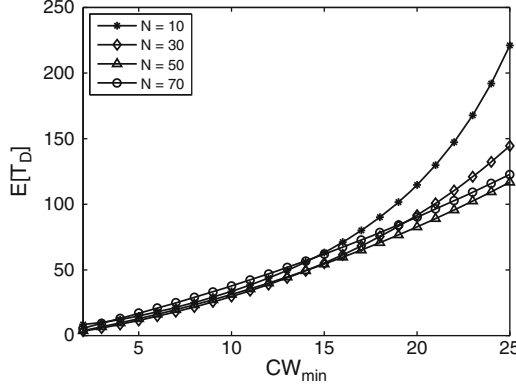


Fig. 2.6 Impact of network size change on average detection delays at $h = 80$

we can see that the configuration $h = 80$ satisfies the false positive performance requirement even when N changes dynamically in a wide range. Note that a typical 802.11TM based wireless local area network covers up to tens of users.

Fixing $h = 80$, we now investigate the average detection delay $E[T_D]$ of the detector for different misbehavior intensities, indicated by the CW_{min} value of a misbehaving node, with results shown in Fig. 2.6. The misbehavior intensities with $CW_{min} > 25$ are not included in our discussion, as their effects are minimal. Practically, a misbehaving node needs to choose more intense misbehavior, e.g., $CW_{min} \leq 16$, to gain more benefits from the network throughput. From Fig. 2.6, we see that for misbehavior in this range, the change of N does not affect $E[T_D]$ much. The reason is that, when a misbehaving node grabs the channel, very likely it will consecutively send a certain number of packets, driving the detector to hit the threshold. For a smaller value of N , it may just take a couple of more samples for the detector to hit the threshold (note that each transmission from the tagged node increases the detector state by $N - 1$), which only slightly increases the detection delay. With less intense misbehavior ($16 < CW_{min} \leq 25$), we do observe obviously larger detection delays for a small N . The reason is that, when the misbehaving intensity is low, the accumulation procedure of X_n is more often to be offset by transmissions from those non-tagged normal nodes; for a small N , it will take even more samples from the misbehaving node to raise the alarm, leading to a longer detection delay.

It is noteworthy that the relationship between N and detection delay in Fig. 2.6 is not rigorously monotonic. Such phenomenon is due to two contradicting factors: Given the CW_{min} , the misbehaving node will get less transmissions when N gets larger and thus less chances to accumulate X_n , potentially increasing the detection delay; the increase of X_n (by a value of $N - 1$) caused by one transmission from the misbehaving node however becomes larger too, potentially decreasing the detection delay. In summary, the results in Fig. 2.6 again demonstrate that the FS detector with a fixed threshold (larger than N) has a robust performance for a typical misbehaving scenario, even when the number of nodes in the network changes.

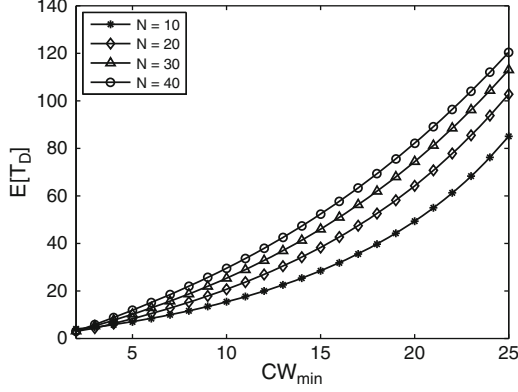


Fig. 2.7 Impact of network size change on average detection delays at $P_{fp} = 0.005$

In a situation where a fixed constraint on P_{fp} is imposed, we can dynamically calculate the h value corresponding to a certain N through the analytical model. Further, if we do the calculation beforehand and maintain a table of “ h versus N ” values under the given P_{fp} constraint, we can quickly adjust h as soon as changes on N are observed. Figure 2.7 shows the average detection delays $E[T_D]$ of the detector for different misbehaving intensities, given a false positive constraint as $P_{fp} = 0.005$. Similar to Figs. 2.6 and 2.7 shows that the detection delays under different N are similar when the misbehavior is very intense. Under a lower misbehaving intensity (i.e., a larger CW_{min}), the detection delays increase more obviously with the number of nodes, because a larger threshold h is required for a larger N to meet the false positive requirement. However, the delay increase is not dramatic. Even for $CW_{min} = 25$ and $N = 40$, it only takes about 120 successful transmissions over the whole network to detect the misbehavior.

2.4.7 Comparison with the Original CUSUM Detector

In order to show how we have improved in real-time misbehavior detection, we compare the FS detector to the detector developed in our preliminary work [5], referred to as the “original CUSUM detector” for convenience. The observation measure of the original CUSUM detector is the number of successful transmissions of the tagged node in every M successful transmissions over the whole network. It means getting one observation sample for the original CUSUM detector requires M successful transmissions, whereas the FS detector will update state upon every successful transmission over the network. Also, M needs to be at least as large as the number of nodes N and linearly increase with N to fairly count transmissions from each node. Moreover, besides h , there is another parameter u in the original detector design, which is the upper bound of the observation measure’s

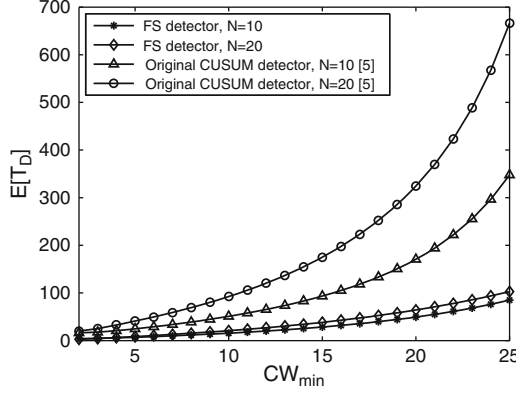


Fig. 2.8 Comparison with the original CUSUM detector in [5] at $P_{fp} = 0.005$

expectation. To determine a proper u , we need to take into account both the sample size M and the number of nodes N , adding the complexity of the detection system. In the FS detector, u is not present, which leads to one less parameter impacting the detection performance and thus makes parameter configuration much simpler.

Figure 2.8 shows the average detection delays of the two detectors for different misbehavior intensities under the same false positive constraint of $P_{fp} = 0.005$. Here we consider the cases of $N = 10$ and $N = 20$. For the FS detector, given the P_{fp} and N , the threshold h can be determined from the analytical model. With h , the detection delay for a given misbehaving intensity can then be calculated and plotted in Fig. 2.8. We intentionally configure the original CUSUM detector for a small detection delay so the advantage of the FS detector can be demonstrated more convincingly. The sample size M for the original CUSUM detector is set to its minimum value N (i.e., 10 and 20 for the two cases considered, respectively) in order to minimize the impact of the observation window size on the detection delay. With such an observation window selection, on average one successful transmission from each node can be expected in each window, i.e., $u = 1$. Given the P_{fp} , N and u , the parameter h can then be determined from the analytical model in [5]. With h and u , the detection delay for a given misbehaving intensity with the original CUSUM detector can be calculated and plotted in Fig. 2.8.

As shown in Fig. 2.8, for the same N , the FS detector shows clear advantages over the original CUSUM detector, especially when the misbehavior becomes less intense. Observing the delays of the original CUSUM detector, we can see that the delays with $N = 20$ are roughly two times of the delays with $N = 10$ for almost all the misbehavior intensities. The fact clearly indicates the impact of the observation window size on detection delay in the original CUSUM detector. Another advantage of the FS detector is that its detection delay curves are quite flat against the misbehaving intensity and not much impacted by the network size N , showing very robust performance.

2.5 Simulation Results

2.5.1 Simulation Setup

We establish an 802.11TM DCF based wireless network consisting of 10 competing nodes ($N = 10$) and an access point (AP) through ns-2 [6] simulation. We first consider that the network works under the saturated condition and every node sends packets with UDP towards the AP. Then we include the TCP traffic in our simulation to further analyze the performance of the FS detector in more general scenarios. The AP also acts as the detection agent which monitors the transmissions from every competing node with a separate FS detector. The nodes are located close enough to sense the transmissions from each other and thus avoid the hidden terminal problem. There is 1 misbehaving node among the 10 competing nodes, which accesses the wireless channel using the binary exponential backoff scheme but can manipulate its minimum contention window CW_{min} to any value between 1 and 32.

Due to the conflicting nature of the three performance metrics (average false positive rate, average detection delay, and missed detection ratio), it can be difficult to find the system configuration parameter that achieves best performance at all fronts. Using our analytical model, we find that, for $N = 10$, setting the detection threshold $h = 40$ can achieve a good tradeoff among all the metrics (referring to Sect. 2.4.5). Therefore in our simulation, if not specified, we set $h = 40$ to further evaluate the performance of our detection.

2.5.2 Robustness Against Short-Term Unfairness

In an 802.11TM network, a node that has just accomplished a successful transmission will have advantages in grabbing the channel for next transmission in a short period [3]. This is referred to as *short-term unfairness* and is inherent to the 802.11TM backoff mechanism. Such an issue implies correlations among the channel accesses, which impact the accuracy of the transition probability calculation based on the assumption of independent channel access. The system configuration based on an inaccurate model can lead to inaccurate detection results. In this section, we study how the short-term unfairness affects the performance of our detector.

We first examine the impact of short-term unfairness on the distribution of the detector X_n under the normal traffic condition. In Fig. 2.9, we present the simulation results of the cumulative distribution function (CDF) of X_n , compared with the analytical CDF. Note that even though the analytical results are based on the independent model of (2.1), the two curves are still close to each other. We then examine the average false positive rate P_{fp} versus h , comparing the analytical results with the simulation results in Fig. 2.10. Again, despite a bigger gap when h is smaller, the P_{fp} curve obtained from simulations still largely resembles the analytical one. The observations show that our FS detector is robust against the impact of short-term unfairness.

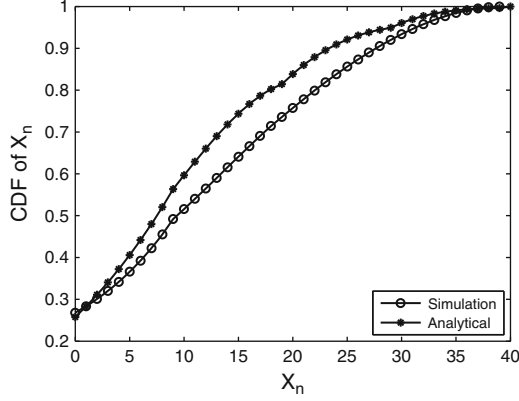


Fig. 2.9 CDF of X_n

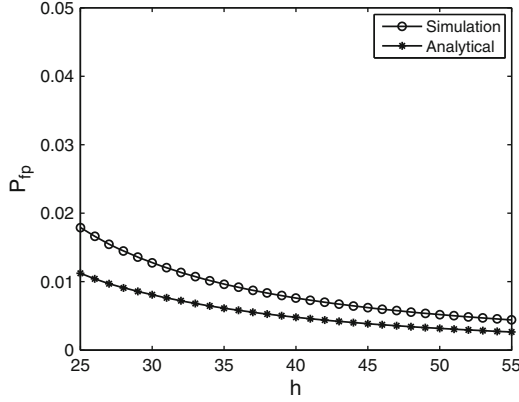


Fig. 2.10 Average false positive rate

We then obtain the average detection delays $E[T_D]$ under different misbehaving intensities. Figures 2.11 and 2.12 present both the simulation and analytical $E[T_D]$ curves versus h for $CW_{min} = 8$ and $CW_{min} = 16$, respectively. The closeness of the two curves in both cases again confirms the robustness of the FS detector against the short-term unfairness.

Technically, the FS detector by nature can mitigate the impact due to the short-term unfairness. In the normal situation, every node in the network has the same opportunity to experience a short period of advantages in transmissions. At a sampling moment, if the tagged node under observation is accessing the channel more aggressively due to short-term unfairness, it will increase the detector state value more aggressively according to (2.2), tending to be false positive. However, if at other sampling moments, those non-tagged nodes are accessing the channel more aggressively, it will in turn decrease the detector state value more aggressively and mitigate the false positive effect. Therefore, as an aggregate effect, the FS detector only degrades slightly in the false positive performance. In the misbehaving situa-

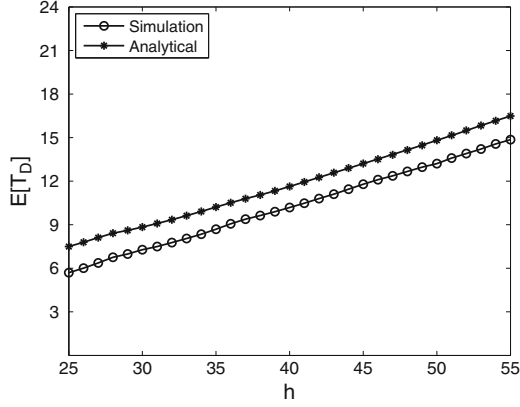


Fig. 2.11 Average detection delay with $CW_{min} = 8$

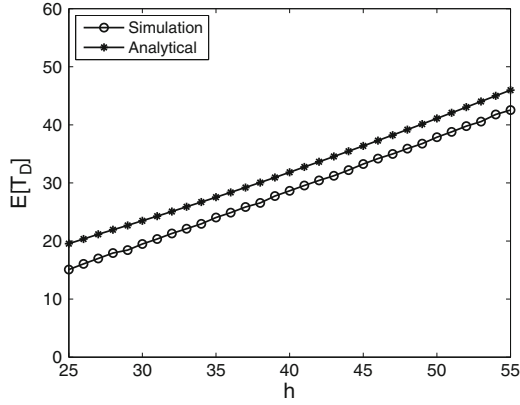


Fig. 2.12 Average detection delay with $CW_{min} = 16$

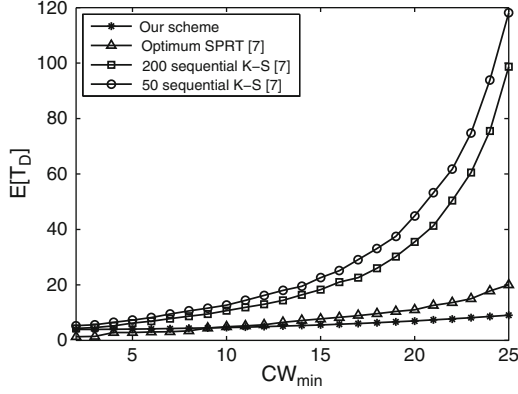
tion, extra channel access (in addition to that resulting from the backoff misbehavior) due to the short-term unfairness effect in fact benefits the misbehavior detection, with the detector being driven to hit the threshold h sooner, as shown in Figs. 2.11 and 2.12. We did design a shuffling mechanism based on the similar idea as that applied in [5] to address the impact of the short-term unfairness, and found that it would sacrifice a lot in detection delay to achieve just a moderate gain in mitigating false positive rate. Thus according to the theoretical and simulation investigations given above, we decide to apply the FS detector without an extra mechanism for the short-term unfairness issue.

2.5.3 Performance Guarantee

Given the configuration parameter $h = 40$, we compare the target performance measures with the simulation results under the same setting, shown in Table 2.1, to

Table 2.1 Comparison of analytical and simulation results with $N = 10$, $h = 40$, $D = 100$

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

**Fig. 2.13** Comparison with the detection schemes in [7]

examine whether the target performance is guaranteed. We can see that simulation results are very close to the target values in all three performance metrics. The small gap between the values is largely due to the variance in the observation samples; also the effect of the short-term unfairness is not 100 % overcome according to Figs. 2.9 and 2.10. Considering such a small gap, in practice we can on purpose select configuration parameters to conservatively provision the detection performance.

With the same parameter configuration as above, we compare our FS detector to the sequential K–S test and the optimal SPRT for 802.11TM backoff misbehavior detection used in [7] in Fig. 2.13. The sample used in those solutions is collected every successful transmission of the tagged node, whereas in our scheme, the sample is collected every successful transmission from any node in the network. The average detection delays in terms of the number of successful transmissions from the tagged node for different detection schemes are compared in Fig. 2.13. For a fair comparison, we map our samples (the total number of successful transmissions over the network) to that used in [7]. For such a mapping, we only need to count the number of successful transmissions from the tagged node within the total successful transmissions. Also note that the desired false positive rate in [7] is fixed at $P_{fp} = 0.05$, which is one order larger than our target 0.005 as given in Table 2.1. Even with a much more strict constraint on P_{fp} , Fig. 2.13 shows that our detector has comparative detection delays against high intensities of backoff misbehavior and becomes superior to all other schemes as the misbehavior turns less intense.

It is interesting to discuss why our FS detector has better performance even than the optimal SPRT (when the misbehaving intensity is not high) in [7]. An optimal SPRT has the “optimal” performance only when the normal behavior distribution

could be accurately obtained. However, to establish the normal behavior distribution, the detectors in [7] need to first estimate the collision probability over the 802.11TM channel. In [7], there are two aspects of inaccuracy in estimating the collision probability, which degrade the performance of false positive rate and detection delay, respectively.

The first aspect of inaccuracy in [7] is that the collision probability is estimated from only tens of samples, over which the variance may lead to overestimating the collision probability. The behavior monitored by the detector is the idle time between consecutive successful transmissions; an overestimated collision probability will lead to an overestimated idle time (longer than its real value). With such an estimation error by the detector, a normal idle time observed will appear smaller than the “thought-to-be” normal behavior and thus misunderstood as misbehaving. That is, the overestimation of the collision probability leads to a higher false positive rate.

The second aspect of inaccuracy is that, according to the IEEE 802.11TM model, a conditional collision probability (given that the tagged node is sending a packet) should be used to characterize the backoff procedure and further estimate the distribution of the idle time between consecutive successful transmissions. The study in [7] however uses an unconditional collision probability estimated over all nodes to approximate the conditional one. Regarding the misbehaving node, the unconditional collision probability will be an underestimate of the conditional one. The conditional collision probability associated with the tagged misbehaving node is determined by transmissions from other normal nodes. When estimating with an unconditional collision probability, transmissions from the misbehaving node are also included in the estimation [7]; note that many transmissions from the misbehaving node will not experience collisions due to the misbehaving node’s aggressive access to the channel. Thus, the collision probability will be obviously underestimated, which then makes the detector to underestimate the normal idle time between consecutive successful transmissions. Such underestimation of the normal model makes the misbehavior deviation less obvious and incurs a longer time for detection.

2.5.4 Performance with UDP and TCP Traffic

We also consider scenarios where TCP traffic exists in the network. Figure 2.14 shows the average detection delay of a misbehaving node versus the misbehaving intensity in a network of 10 nodes. The detection threshold $h = 40$. We compare the detection delays in the two scenarios that all the nodes send TCP traffic or saturated UDP traffic to the AP, respectively. As shown in Fig. 2.14, in most cases, the detection delay in the TCP scenario is larger than that in the UDP one, especially when the misbehavior is more intense. The reason is that TCP multiplicatively decreases the transmission rate upon a packet loss due to its congestion control mechanism; the impact of congestion control is more obvious in wireless networks where collisions are common. The congestion control mechanism by nature mitigates the

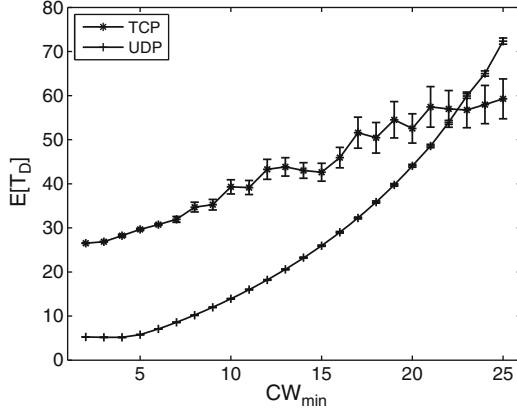


Fig. 2.14 Average detection delay under TCP traffic

selfish misbehavior. Aggressive transmissions will lead to more collisions, which in turn decreases the sending rate through the congestion control. Thus, it takes longer time to detect (compared to the UDP case) the misbehavior due to the mitigating effect of the congestion control. With a low misbehaving intensity ($CW_{min} > 20$), the congestion control effect applies more to the normal nodes, where the detection delay will be shorter than that in the UDP case. In Fig. 2.14, we also plot the 95 % confidence interval, measured from a large number of detections (in the order of 10^6), which can show that TCP congestion control brings a high degree of dynamics to the system.

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

For robust detection performance in the complex scenario when both UDP and TCP traffic flows exist, we design a *dual-detector* implementation as shown in Fig. 2.15. FS detector 1 monitors traffic from all the nodes; if a detection event happens, we check whether the tagged node claims to use TCP or UDP. Then, if

¹ Without loss of generality, we can consider the situation that some nodes have a UDP flow and some have a TCP flow. If a node has both UDP flows and TCP flows, in a saturated situation, the aggregate traffic behaves similar to the UDP traffic.

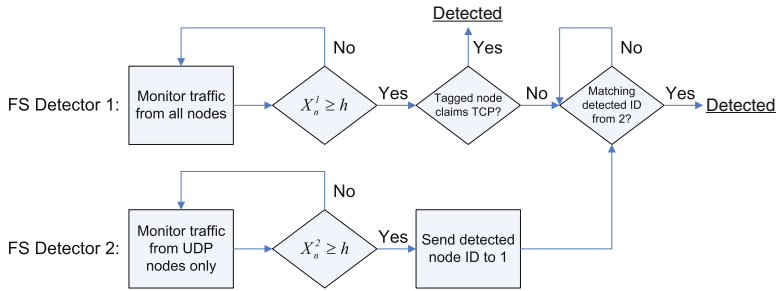


Fig. 2.15 Detection with TCP/UDP hybrid traffic

it claims to use TCP, we decide that it is a smart misbehaving node actually using UDP (case 3 mentioned above); if it claims to use UDP, we turn to listen to the decision from FS detector 2 (to avoid false positive in case 1). FS Detector 2 starts simultaneously with detector 1 but monitors only the traffic from the UDP nodes. When detector 2 identifies misbehavior from a UDP node, it sends the detected node ID to detector 1. If the detected node ID from detector 2 matches that alarmed by detector 1, the dual-detector system will then determine that the node is misbehaving (case 2). We run simulations to verify the performance of the dual-detector. For example, in a network of 10 nodes where 5 nodes use TCP and the other 5 nodes use UDP, the average false positive rate over a normal UDP node is 0.0047. Also, for the moderate misbehavior of $CW_{min} = 16$, the average detection delay of a misbehaving node is 18.1953 when it lies to be a TCP node. The detection delay increases to 36.4728 (for confirmed detection in both detectors) when the misbehaving node is honest with its UDP behavior, which is similar to that in the all UDP case listed in Table 2.1.

2.6 Summary

In this chapter, we develop a novel fair share (FS) detector for real-time backoff misbehavior detection in IEEE 802.11TM based wireless networks. Also, we develop a Markov chain based model to theoretically analyze the performance of the detector. While most existing work for backoff misbehavior detection depends on heuristic parameter configuration and experimental performance evaluation, we are able to use our model for a quantitative study to achieve guaranteed detection performance in terms of average false positive rate, average detection delay and missed detection ratio. Moreover, we present simulation results that confirm the accuracy of our theoretical analysis and demonstrate the robustness of the FS detector.

References

1. G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE Journal on Selected Areas of Communication*, vol. 18, no. 3, pp. 535–547, Mar. 2000.
2. B. Brodsky and B. Darkhovsky, *Nonparametric Methods in Change-point Problems*, Kluwer Academic Publisher, 1993.
3. C. E. Koksal, H. Kassab and H. Balakrishnan, “An Analysis of Short-Term Fairness in Wireless Media Access Protocols,” in *Proc. ACM SIGMETRICS*, 2000.
4. J. R. Morris, *Markov Chains*, Cambridge Univ. Press, 1997.
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.
6. The Network Simulator - ns-2, [Online.] Available: <http://www.isi.edu/nsnam/ns/>.
7. A. Toledo and X. Wang, “Robust Detection of Selfish Misbehavior in Wireless Networks,” in *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1124–1134, Aug. 2007.

Intrusion Detection for IP-Based Multimedia
Communications over Wireless Networks

Tang, J.; Cheng, Y.

2013, X, 86 p. 31 illus., Softcover

ISBN: 978-1-4614-8995-5