Journal of Computer Security 15 (2007) 103–128
IOS Press

Detecting IEEE 802.11 MAC layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers

S. Radosavac [a,*], Alvaro A. Cárdenas [a], John S. Baras [a]
and George V. Moustakides [b]
[a] *University of Maryland, College Park, USA*
[b] *University of Thessaly, Greece*

Selfish behavior at the Medium Access (MAC) Layer can have devastating side effects on the performance of wireless networks, with effects similar to those of Denial of Service (DoS) attacks. In this paper we consider the problem of detection and prevention of node misbehavior at the MAC layer, focusing on the back-off manipulation by selfish nodes. We first propose an algorithm that ensures honest behavior of non-colluding participants. Furthermore, we analyze the problem of colluding selfish nodes, casting the problem within a minimax robust detection framework and providing an optimal detection rule for the worst-case attack scenarios. Finally, we evaluate the performance of single and colluding attackers in terms of detection delay. Although our approach is general and can be used with any probabilistic distributed MAC protocol, we focus our analysis on the IEEE 802.11 MAC.

Keywords: Ad hoc networks, MAC layer, intrusion detection, security, IEEE 802.11, back-off manipulation

1. Introduction

With the rise and flexibility of ubiquitous computing, new and unforeseeable ways of user interactions are expected, such as establishing collaborative networks with minimum or almost no central control. One such example can be the use of ad hoc networks for providing fast and efficient network deployment in a wide variety of scenarios with no fixed networking infrastructure and where each node is its own authority. However, in order for this interactions to reach their full potential, these networks should support minimum security and performance guarantees defined by the end users. For example, some current P2P file-sharing networks suffer from the abundance of corrupted files introduced by attackers and from selfish participants who only download files but never share them with other users. These factors limit the utility of P2P file-sharing networks as an efficient way to recover files.

[*] Corresponding author. Address: The Institute for Systems Research, A.V. Williams Building, University of Maryland, College Park, 20742, USA. Tel.: +1-301-405-2942; Fax: +1-301-314-8586; E-mail: svetlana@umd.edu.

The communication protocols in different layers of an ad hoc network can also be subject to manipulation by selfish users. For example, the MAC protocol, the routing protocol and the transport protocol were designed under the assumption that all participating nodes obey the given specifications. However, when these protocols are implemented in an environment where each node has its own authority, nodes can deviate from the protocol specification in order to obtain a given goal, at the expense of honest participants. A selfish user for example, can disobey the rules to access the wireless channel in order to obtain a higher throughput than the other nodes. A selfish user can also change the congestion avoidance parameters of TCP in order to obtain unfair advantage over the rest of the nodes in the network [2]. In devices with limited power resources, certain nodes might refuse to forward packets on behalf of other sources in order to save battery power [3]. In all these cases, the misbehaving nodes will degrade the performance of the network from the point of view of the honest participants.

To fully address these problems, a layered reputation mechanism should be deployed in order to either reward cooperation (e.g., payments) or penalize misbehaving nodes (e.g., revocation). In this paper we focus on the detection of individual and colluding selfish users at the MAC layer in ad hoc networks.

1.1. Summary of our approach

In our approach we point out that a key element that facilitates misbehavior in contention based MAC layer protocols is the fact that they are *probabilistic* distributed protocols. The random nature of these protocols and the nature of the wireless medium makes the detection of misbehaving nodes very difficult, since it is not easy for the detector to distinguish between a peer misbehavior, an occasional protocol malfunction due to a wireless link impairment or a greedy back-off strategy. In order to facilitate the detection of a single attacker, we propose the use of Blum's coin flipping protocol [9] that facilitates the exchange of a truly random parameter that can be used as a seed for a pseudorandom number generator. This allows anyone who monitors the execution of the protocol to determine the exact source of randomness used by the participating nodes and detect any deviations. We believe this idea facilitates the monitoring procedure of misbehavior in any distributed probabilistic MAC layer protocol such as ALOHA [1], SEEDEX [26], MACA [20], MACAW [8] and IEEE 802.11 [19].

Since we assume an ad hoc network where each node is its own authority, the usual assumption of a trusted receiver (e.g., a base station) might no longer hold, and therefore we need to worry about colluding nodes. However, the Blum's scheme cannot be used in the detection of colluding nodes due to the large overhead required for the randomness agreement among more than two nodes. Instead, we base our approach on sequential detection procedures, placing the emphasis on the class of attacks that incur larger gain for the attackers. This approach should also cope with the uncertain environment of a wireless network. Hence, we adopt the minimax robust detection approach where the goal is to optimize performance for the worst-case instance of uncertainty. More specifically, the goal is to identify the least favorable operating point of a system in the presence of uncertainty and subsequently find the strategy the optimizes system performance when operating in that point. In our case, the least favorable operating point corresponds to the worst-case instance of an attack and the optimal strategy amounts to the optimal detection rule.

Throughout this work we assume existence of intelligent adaptive attackers that are aware of the environment and its changes over a given period of time. We assume that, in order to minimize the probability of detection, the attackers choose legitimate over selfish behavior when the level of congestion in the network is low. That is, if neighboring honest nodes have nothing to transmit, then there is no incentive for the selfish node to misbehave, since it will always get access to the channel. However, the attackers will choose adaptive selfish strategies in a congested network in order to obtain better access to the channel. Due to these reasons, we assume a benchmark scenario where all the participants are backlogged, i.e., have packets to send at any given time in both theoretical and experimental evaluations. We assume that the attackers will employ the worst-case misbehavior strategy in this setting, and consequently the detection system can estimate the maximum detection delay. It is important to mention that this setting represents the worst-case scenario with regard to the number of false alarms per unit of time due to the fact that the detection system is forced to make maximum number of decisions per time unit.

Our work contributes to the current literature by: (i) proposing a solution for preventing misbehavior of a single intelligent node, (ii) formulating the problem of optimal detection against misbehavior of intelligent colluding attackers (iii) quantifying performance losses incurred by an attack and defining an uncertainty class such that the focus is only on attacks that incur “large enough” performance losses, (iv) obtaining analytical expressions for the worst-case attack and the optimal detection rule (and its performance), (v) establishing an upper bound on the number of required samples for detection of any of the attacks of interest. Therefore our work constitutes a first step towards understanding the complex issue of collaboration among colluding nodes in wireless networks, obtaining bounds on achievable performance and characterizing the impact of different system parameters on it.

The paper is organized as follows. Section 2 summarizes related work dealing with MAC layer misbehavior. Section 3 deals with misbehavior in IEEE 802.11 DCF protocol. In Section 4 we present an algorithm that prevents the manipulation of backoff values for a single selfish node. In Section 5 we analyze the detection problem in the presence of colluding nodes. Following that, we present the minimax robust detection model and basic assumptions and demonstrate our approach comparing the results with the scenario that includes a single attacker. Finally, Section 7 concludes our study.

2. Background work

2.1. MAC layer misbehavior

The MAC layer in a communication network manages a multiaccess link (e.g., a wireless link) so that frames can be sent by each node without constant interference from other nodes. MAC layer misbehavior is possible in network access cards that run the MAC protocol in software rather than hardware or firmware, allowing a selfish user or attacker to easily change MAC layer parameters. Even network interface cards implementing most MAC layer functions in hardware and firmware usually provide an expanded set of functionalities which can be exploited to circumvent the limitations imposed by the firmware [5]. In the worst case scenario, an untrusted vendor might manufacture NIC cards violating the MAC protocol to create an improved performance of its products.

In this work we assume that a selfish node in the MAC layer attempts to maximize its own throughput and therefore keeps the channel busy. As a side effect of this behavior, regular nodes cannot use the channel for transmission, which leads to a denial of service (DoS) attack [17].

Selfish misbehavior at the MAC layer has been addressed mostly from a game theoretic perspective considering that all nodes are selfish. The goal in a game theoretic setting is to design distributed protocols that guarantee the existence, uniqueness and convergence to a Nash equilibrium with an acceptable throughput for each node. However, if users try to maximize their throughput, every node will attempt to transmit continuously in such way that users deny access to any other node until the network collapses. This collapse is in fact, a (very impractical) Nash equilibrium of the game. In order to obtain a more efficient Nash equilibrium, each node needs to be assigned a cost for each time it accesses the channel. For example [16,22] consider the case of selfish users in Aloha that attempt to maximize their throughput and minimize the cost for accessing the channel (e.g., energy consumption). Another game theoretic scheme for CSMA/CA schemes is presented in [13]. Using a dynamic game model, the authors derive the strategy that each node should follow in terms of controlling channel access probability by adjustment of contention window, so that the network reaches its equilibrium. They also provide conditions under which the Nash equilibrium of the network with several misbehaving nodes is Pareto optimal for each node as well. The underlying assumption is that all nodes are within wireless range of each other so as to avoid the hidden terminal problem, therefore this scheme is mostly intended for wireless LANs, as opposed to ad hoc networks.

Since game theoretic protocols assume all nodes are selfish (the worst case scenario), the throughput achieved in these protocols is substantially less than in protocols where the honest majority cooperates. Under the assumption of an honest majority, detection of misbehaving nodes becomes the primary goal in dealing with misbehavior.

2.1.1. Detecting MAC layer misbehavior

Due to the popularity of the IEEE 802.11, most of the work in detecting MAC layer misbehavior has focused in this protocol. A selfish user in the IEEE 802.11 can implement a whole range of strategies to maximize its access to the medium. The most effective strategy that a selfish user can employ is to use different schemes for manipulating the rules of the MAC layer. For example, the attacker can manipulate the size of the Network Allocation Vector (NAV) and assign large idle time periods to its neighbors, it can decrease the size of Interframe Spaces (both SIFS and DIFS), it can select small back-off values, it can deauthenticate neighboring nodes etc. A successful detection scheme should take into account all possible cheating options in the MAC layer and detect both: users that employ only one scheme and users that employ a combination of several schemes (e.g., first choosing small back-off values, then assigning large NAV values to its neighbors etc.).

However, the most challenging detection task is that of detecting back-off manipulation [5,25]. Due to the randomness introduced in the choice of the back-off, it is difficult to decide if a node has chosen small back-off values by chance or if the small back-off values are part of a misbehavior strategy. The back-off detection scheme provided in [25] works well for adversaries that are unaware of the detection scheme, however an intelligent adversary would try to maximize his own gain (e.g., throughput) while minimizing the chances of being detected. [24] addresses this concern by providing a theoretical foundation for the design of optimal detection schemes against intelligent adversaries. These algorithms however have only focused on individual misbehaving nodes, and do not consider collusion.

Another approach for the detection of single misbehaving nodes was proposed in [21]. In this work, the authors propose a modification to the IEEE 802.11 for facilitating the detection of misbehaving nodes. In their scheme, the receiver (a trusted host-, e.g., a base station-) assigns the back-off value to be used by the sender. The receiver can therefore detect any misbehavior of the sender and penalize it by increasing the back-off values for the next transmission. The protocol consists of Detection, Penalty and Diagnosis Schemes. The sender is considered to be deviating from the protocol if the observed number of idle slots, the actual back-off B_{act}, is smaller than a specified fraction α of the assigned (expected) back-off B_{exp}. For a detected node, a penalty for the next assigned back-off is selected given a measure of the deviation $D = \max(\alpha B_{exp} - B_{act}, 0)$. If the sender deviates repeatedly, i.e., if the sum of misbehavior in a sliding windo+w is bigger than some threshold, then the sender is labeled as misbehaving and the receiver takes drastic measures, for example, by dropping all packets by the sender. However, as we have pointed out in the introduction, the problem of applying this protocol in autonomous ad hoc networks is the fact that the receiver might not be trusted.

2.2. Additional assumptions for detection in a distributed setting

The scenario presented in this work differs from the one presented in [21] due to the fact that we attempt to solve the problem in the environment with no central authority. Consequently, the penalization of misbehaving nodes by the central authority cannot be performed in our setting. Therefore a comprehensive strategy against greedy behavior requires at least three steps: local detection of misbehaving nodes, information propagation to other honest nodes in the network and response.

Upon local detection of misbehavior, the other major issue is propagation of the obtained information throughout the network. Although a misbehaving node can be detected by our system, the detection mechanism opens a new opportunity for attacks since honest nodes can be falsely incriminated by an adversary, imposing the new problem of obtaining secure information from a distributed reputation management system, while maintaining accurate identification of the misbehaving identities and minimizing the probability of false incrimination.

Finally, the system needs to react to the information gathered from the reputation system by other nodes. The response can be either a reward for cooperation (e.g., payments) or the penalization of misbehaving nodes (e.g., revocation). We note that response algorithms can be done more efficiently at different layers (as opposed to doing response just at the MAC layer). For example, a possible response against selfish MAC users is employment of a rate-limiting algorithm at the routing layer that limits the amount of traffic selfish nodes can receive or send. The idea of reacting to MAC layer misbehavior at different layers (routing in this case) coincides with the current interest of cross-layer design for wireless networks [27].

Overall, the issue of designing such a distributed reputation management system is a large and complex subject by itself that has received much attention recently, [10–12,23,28,29]. In this paper however, we focus on the fundamental problem of local detection. This detection is accomplished by the involvement of the neighboring nodes that monitor the behavior of both the sender and the receiver.

We also note that all the schemes presented in the previous sections as well as the ones we propose, require the proper use of MAC layer authentication schemes, providing uniquely verifiable identities in order to prevent impersonation and Sybil attacks [15].

3. IEEE 802.11 DCF

The most frequently used MAC protocol for wireless networks is the IEEE 802.11 MAC protocol, which uses a distributed contention resolution mechanism for sharing the wireless channel. Its design attempts to ensure a relatively fair access to the medium for all participants of the protocol. In order to avoid collisions, the nodes follow a binary exponential back-off scheme that favors the last winner amongst the contending nodes.

In the distributed coordinating function (DCF) of the IEEE 802.11 MAC protocol, coordination of channel access for contending nodes is achieved with carrier sense multiple access with collision avoidance (CSMA/CA). A node with a packet to transmit selects a random back-off value b uniformly from the set $\{0, 1, \ldots, W-1\}$, where W is the (fixed) size of the contention window. The back-off counter decreases by one at each time slot that is sensed to be idle and the node transmits after b idle slots. In case the channel is perceived to be busy in one slot, the back-off counter stops momentarily. After the back-off counter is decreased to zero, the transmitter can reserve the channel for the duration of data transfer. First, it sends a request-to-send (RTS) packet to the receiver, which responds with a clear-to-send (CTS) packet. Thus, the channel is reserved for the transmission. Both RTS and CTS messages contain the intended duration of data transmission in the duration field. Other hosts overhearing either the RTS or the CTS are required to adjust their network allocation vector (NAV) that indicates the duration for which they will defer transmission. This duration includes the SIFS intervals, data packets and acknowledgment frame following the transmitted data frame. An unsuccessful transmission instance due to collision or interference is denoted by lack of CTS or ACK for the data sent and causes the value of contention window to double. If the transmission is successful, the host resets its contention window to the minimum value W.

Figure 1 illustrates the scenario of contending nodes using the protocol.

Typical parameter values for the MAC protocol depend on the physical layer that IEEE 802.11 uses. For example, Table 1 shows the parameters used when the physical layer is using direct sequence spread spectrum (DSSS).

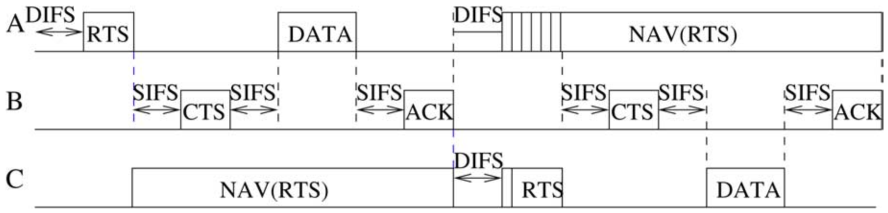

Fig. 1. Nodes A and C contend for accessing node B. The first time A reserves the channel, and in the second time C accesses the channel.

Table 1

Parameters for DSSS

DIFS	50 μs
SIFS	10 μs
SlotTime	20 μs
ACK	112 bits + PHY_header = 203 μs
RTS	160 bits + PHY_header = 207 μs
CTS	112 bits + PHY_header = 203 μs
DATA	MAC_header (30b) + DATA(0-2312b) + FCS(4b)
Timeouts	300–350 μs
CW_{min}	32 time slots
CW_{max}	1024 time slots

IEEE 802.11 DCF favors the node that selects the smallest back-off value among a set of contending nodes. Therefore, a malicious or selfish node may choose not to comply to protocol rules by selecting small back-off intervals, thereby gaining significant advantage in channel sharing over regularly behaving, honest nodes. Moreover, due to the exponential increase of the contention window after each unsuccessful transmission, non-malicious nodes are forced to select their future back-offs from larger intervals after every access failure. Therefore the chance of their accessing the channel becomes even smaller. Apart from intentional selection of small back-off values, a node can deviate from the MAC protocol in other ways as well. He can choose a smaller size of contention window or he may wait for a shorter interval than DIFS, or reserve the channel for a larger interval than the maximum allowed network allocation vector (NAV) duration. In this work, we will adhere to protocol deviations that occur due to manipulation of the back-off value, since the other types of misbehavior have been properly addressed in [5,25].

The nodes that are instructed by the protocol to defer transmission are able to overhear transmissions from nodes whose transmission range they reside in. Therefore, silenced nodes can observe the behavior of transmitting nodes. Due to the fact that the protocol participants are energy-constrained devices, we cannot assume participation of all nodes in the process of detection. Instead, we utilize the fact that each node that needs to access the channel and is forced to defer its transmission due to an ongoing communication will be able to overhear the transmissions of either the transmitter or the receiver (or both). Consequently, each node that attempts to access the channel and has to defer its transmission can serve as a monitoring node and does not need to use any additional power apart from the one used for attempting to access the channel. The question that arises is whether there exists a way to take advantage of this observation capability and use it to identify potential misbehavior instances. If observations indicate a misbehavior event, the observer nodes should notify the rest of the network about this situation or could launch a response action in order to isolate the misbehaving nodes. Detecting misbehavior is not straightforward even in the simplest case, namely that of unobstructed observations. The difficulty stems primarily from the non-deterministic nature of the access protocol that does not lead to a straightforward way of distinguishing between a legitimate sender, that happens to select small back-offs, and a misbehaving node that maliciously selects small back-offs. The open wireless medium and the different perceived channel conditions at different locations add to the difficulty of the problem. Additional challenges arise from the presence of interference due to ongoing concurrent transmissions.

4. Preventing misbehavior of a single node

As it has been mentioned, [21] requires the receiver to be trusted. This assumption is well suited for infrastructure-based wireless networks, where the base station

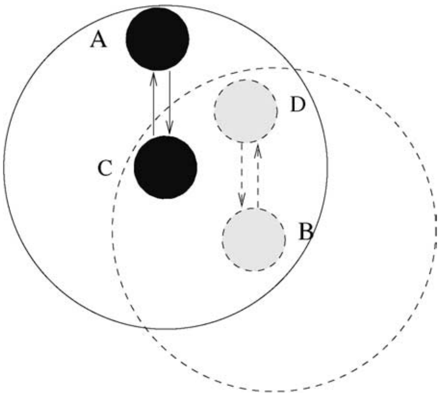

Fig. 2. Node C transmits to A and node B wants to transmit to D. After hearing the back-off assigned by A to C, node D assigns a back-off to node B such that it collides with C.

can be trusted. However, we consider ad hoc networks where the receiver can misbehave by selectively assigning the back-off values to different senders. Depending on the concrete situation, a receiver may benefit by assigning small back-off values to a particular sender (when data from that particular sender need to be received) or by assigning large back-off values to different neighbors (when it wants to degrade overall performance of neighbors and improve its own throughput). Furthermore, existence of multiple sender-receiver pairs in the interference range of each other creates additional security issues. More specifically, a malicious receiver D in Fig. 2 can overhear the back-off value assigned to node A by node C and unilaterally select a back-off for node B in order to create a collision with C.

In this section we propose an extension to the IEEE 802.11 CSMA/CA protocol that ensures a uniformly distributed random back-off, when at least one of the parties is honest. The basic idea follows the protocol for flipping coins over the telephone by Blum [9]. The adopted approach is that the sender and the receiver agree through a public discussion on a random value. The main property of the protocol is that an honest party will always be sure that the agreed value is truly random. For an honest sender this means that he can expect a fair treatment in order to access the channel. On the other hand, an honest receiver can monitor the behavior of the sender (as in [21]) and report a misbehaving node to the reputation management system.

It has been mentioned in the introduction that Blum's protocol can be used to select the seed for a pseudorandom number generator. However, the four way handshake in the IEEE 802.11, that is used every time a new reservation of the channel takes place, is particularly well suited for implementing Blum's protocol as a way of selecting the next back-off value for a node. Selecting the next back-off value in each reservation round, as opposed to selecting a seed for a pseudorandom number generator, has the advantage that there need not be any synchronization between nodes keeping states of random number generators for the other participating nodes (each node would need to keep the seed and the current state of the random generator for other nodes). Furthermore, selecting the next back-off value in each channel reservation, allows any node in the neighborhood to monitor the behavior of the parties accessing the channel, a feature that will be of importance in the next section.

The protocol can be described as follows (the extra messages are appended -denoted by a double bar ||- to the normal message exchange of 802.11):

$$\begin{array}{lcl}
\underline{S} & & \underline{R} \\
n \leftarrow \{0,1\}^{k_{nonce}} & \xrightarrow{RTS\|n} & r \leftarrow \{0,\ldots,W-1\} \\
 & \xleftarrow{CTS\|\sigma} & \sigma = \mathsf{Commit(r||n)} \\
r' \leftarrow \{0,\ldots,W-1\} & \xrightarrow{DATA\|r'} & \sigma' = \mathsf{Open} \\
 & \xleftarrow{ACK\|\sigma'} & \\
\mathsf{Commit(r||n)} \overset{?}{=} \sigma & & \\
b_i = r_i \oplus r'_i & & b_i = r_i \oplus r'_i \\
\text{ for } i \in \{1,\ldots,m\} & & \text{ for } i \in \{1,\ldots,m\}
\end{array}$$

We now explain the protocol step by step.

1. In the first step the sender S selects a nonce: a number n selected uniformly at random from the set $\{0, 1, \ldots, 2^{k_{nonce}}\}$, (denoted as $n \leftarrow \{0,1\}^{k_{nonce}}$). k_{nonce} is a security parameter indicating the level of difficulty of guessing n. For example k_{nonce} can be 64. This step is done in order to prevent an off-line attack on the commitment scheme.
2. In the second step the receiver R selects a random back-off r from the set $\{0, 1, \ldots, W-1\}$ and commits to it. In binary notation r is a random bit string of length m ($r = r_1 r_2 \cdots r_m$), where $m = \log_2 W$ (note that the contention window size W is always a power of two). The commitment scheme Commit is such that the following two properties are satisfied (at least before the timeout for channel reservation: 300 μs–350 μs):

   Binding: After sending Commit(r||n), the receiver cannot open the commitment to a different value $\tilde{r} \neq r$ (except with negligible probability). This protects against a dishonest R that might try to change the committed value depending on the r' received by S.

   Hiding: Given Commit(r||n), S cannot extract any information about r that will enable it to distinguish r from any other bit string of length m (except with negligible probability). This protects against a dishonest S that will try to tailor r' based on its guess of r.
3. After receiving the commitment σ, S selects a random value $r' = r'_1 r'_2 \cdots r'_m$ from $\{0, 1, \ldots, W-1\}$.

4. Finally R opens its commitment to S. Opening a commitment is an operation that reveals the committed value r and some additional information to S. This enables the other party to verify that the revealed and committed values are the same. If the value opened by the R is correct, both sender and receiver compute the back-off $b = b_1 b_2 \cdots b_m$ as the XOR of the bits: $b_i = r_i \oplus r'_i$. Otherwise, the sender can report misbehavior of the node to the reputation management system.

Several commitment schemes are known under very different computational assumptions. Very efficient commitment schemes in terms of computation and communication, can be implemented under the random oracle model [6]. In this setting it is a standard practice to assume that hash functions H, such as SHA-1, are random oracles. Under this assumption it is easy to confirm that the following commitment scheme satisfies the binding (by assuming H is collision resistant) and hiding properties (by assuming H is a random oracle):

```
Commit(r||n)
    $i \leftarrow \{0,1\}^k$
    Output = H(i||r||n)

Open
    Output = (i, r)
```

where k is a security parameter (e.g., $k = 64$). To open the commitment, R has to send both r and i so that S can check validity of the commitment.

We now consider 802.11 with Direct Sequence Spread Spectrum (DSSS) physical layer. In DSSS mode the minimum contention window size is 32 time slots, therefore $m = \log_2 W = 5$, that is, r' and r are only 5 bits long which is an insignificant quantity to be appended to a $DATA$ frame. The acknowledgement frame is appended $k + m = 69$ bits.

If we use SHA-1 to implement the hash function of the commitment then we obtain a message digest of 160 bits. The CTS frame is doubled in size if the full message digest is used. If doubling the size of a CTS frame is a concern, the output of SHA-1 can always be truncated (for example to 80 bits). The security reduction of the message digest has to be evaluated under the birthday paradox: if the message digest has h bits, then it would take only about $2^{h/2}$ messages (out of $2^{k+m+k_{nonce}}$), chosen at random, before one would find two (inputs) with the same value (message digest). Considering the normal timeout between frames to be 300 μs, we can safely assume 2^{40} computations cannot be done in this time. Finally the nonce parameter should discourage off-line attacks, with for example $k_{nonce} = 64$.

In this section we have thus introduced an efficient mechanism to guarantee honest back-off assignments in distributed environments. The computational and communication complexities of our proposed algorithm are kept to the minimum by the use of efficient primitives such as hash functions, and by adding only a small payoff to each message exchange. Once the sender and the receiver have agreed on a given back-off value, each of them can report misbehavior by using the same detection algorithm as the one proposed in [21].

5. Optimal detection of misbehaving colluding nodes

The problem treatment above assumed the existence of a single attacker and did not include the scenario of colluding nodes. To illustrate the difference between detection of a single attacker and colluding attackers we analyze the communication scenario in Fig. 3. We assume that node C is in the wireless range of M and D and that it is capable of monitoring access times of its neighboring nodes. When M reserves the channel following the protocol described in the previous section, any neighboring node can compute M's exact back-off values by listening to the exchanged values n, σ, r', σ' (between M and the receiver) and then computing the back-off as $b_i = r_i \oplus r'_i$. However, nodes D and M may collude and deny network access to nodes B and C. This effect can be easily achieved when back-off values of both sender and receiver are selected a priori (i.e., when both nodes select the back-off values using a pre-specified p.d.f.). Obviously, the previously outlined monitoring procedure does not work in this case due to the fact that both the sender and the receiver follow the specific sequence of back-off values that have been assigned a priori. For example, they can collude by selecting back-off values equal to zero as follows:

$$\begin{array}{lcl}
\underline{M} & & \underline{D} \\
n \leftarrow \{0,1\}^{k_{nonce}} & \xrightarrow{RTS||n} & \\
& \xleftarrow{CTS||\sigma} & \sigma = \mathsf{Commit}(00000||\mathsf{n}) \\
& \xrightarrow{DATA||00000} & \sigma' = \mathsf{Open}(\sigma) \\
& \xleftarrow{ACK||\sigma'} & \\
b = 00000 & & b = 00000
\end{array}$$

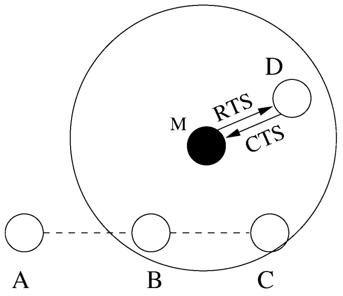

Fig. 3. Nodes M and D collude and interfere in the communication path of nodes B and C.

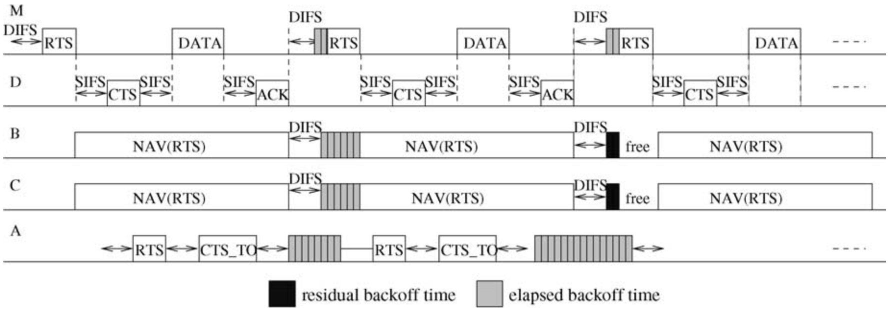

Fig. 4. Nodes M and D collude and select a very small back-off, thereby denying access to node A by causing CTS timeouts.

In this scenario the sender chooses the back-off period equal to zero and sends immediately upon the expiration of its DIFS period. In Fig. 4 we show how the sequence of small backoffs $0, 1, 2, \ldots$ from node M causes the timer for the CTS frame of node A to time out. Node A will therefore back-off repeatedly, making it less likely to access the network.

Obviously, node C cannot detect misbehavior by observing whether nodes D and M deviate from agreed back-off values and other detection procedures need to be applied.

5.1. Detection and attack assumptions

We now consider detection strategies in the presence of an intelligent misbehaving node: a node that is aware of the existence of monitoring neighboring nodes and adapts its access policy in order to avoid detection. In general, we adopt the following assumptions about the colluding nodes:

1. They are *knowledgeable*, i.e., they know everything a monitoring node knows about the detection scheme.
2. They are *intelligent*, i.e., they can make inferences about the situation in the same way as the monitoring nodes can.

Therefore we assume that the goal of the misbehaving hosts is to choose an optimal attack strategy that minimizes the probability of detection P_D (or equivalently a strategy that maximizes the probability of avoiding detection P_M), while maximizing their gain (access to the channel).

However, it is difficult to come up with a universal access policy for misbehaving nodes due to the random nature of the wireless channel and the nature of the access protocol itself. A naive detection system may assume that the misbehaving nodes always select small back-off values. This strategy can be modeled with a scheme that selects backoffs uniformly from the set $\{0, 1, \ldots, W/4\}$. Given this model, the detector raises an alarm when any of the monitored nodes back-off in the interval

$[0, W/4]$ for k consecutive times (where k is chosen given an acceptable false alarm rate P_{FA}). However, an intelligent misbehaving node can easily defeat this detection mechanism by selecting $k-1$ backoffs equal to zero and selecting a value above $W/4$ as the k-th back-off.

Therefore, our desired detection procedure has to fulfill two basic conditions:

- decisions about the occurence of misbehavior should be *robust*, i.e., they need to perform well for a wide range of attack strategies
- decisions should be performed on-line as the observations are revealed to facilitate the quickest attack detection given the desired performance in terms of the false alarm rate P_{FA} and the probability of missing the detection of misbehavior P_M.

The first condition gives rise to the application of a minimax formulation that identifies the rule that optimizes worst-case performance over the class of allowed uncertainty conditions. A minimax formulation translates to finding the detection rule with the minimum required number of observations to reach a decision for the worst instance of misbehavior. Clearly, such scheme guarantees a minimum level of performance which is the best minimum level possible over all classes of attacks.

The second condition implies that sequential detection procedures need to be used. A sequential decision rule consists of a stopping time which indicates when to stop observing and a final decision rule that indicates which hypothesis (i.e., occurrence or not of misbehavior) should be selected. A sequential decision rule is efficient if it can provide reliable decision as fast as possible. It has been shown by Wald [30] that the decision rule that minimizes the expected number of required observations to reach a decision over all sequential and non-sequential decision rules is the sequential probability ratio test (SPRT).

5.1.1. The SPRT

The SPRT collects observations until significant evidence in favor of one of the two hypotheses is accumulated. After each observation at the k-th stage, we choose between the following options: accept one or the other hypothesis and stop collecting observations, or defer decision for the moment and obtain observation $k+1$. The SPRT has two thresholds a and b that aid the decision. The figure of merit at each step is the logarithm of the likelihood ratio of the accumulated sample vector until that stage. For the case of testing between hypotheses $\mathbf{H}_0$ (normal behavior) and $\mathbf{H}_1$ (misbehaving node) that involve probability density functions f_0 and f_1, the logarithm of the likelihood ratio at stage k with accumulated samples $x_1, \ldots, x_k$, where x_i represents the backoff value collected at the k-th stage, is

$$S_k = \ln \frac{f_1(x_1, \ldots, x_k)}{f_0(x_1, \ldots, x_k)}, \tag{1}$$

where $f_i(x_1, \ldots, x_k)$ is the joint probability density function of data $(x_1, \ldots, x_k)$ based on hypothesis $\mathbf{H}_i$, $i = 0, 1$. If the observation samples are statistically independent

$$S_k = \sum_{j=1}^{k} \Lambda_j = \sum_{j=1}^{k} \ln \frac{f_1(x_j)}{f_0(x_j)}, \tag{2}$$

with $f_i(\cdot)$ the probability density function of hypothesis $\mathbf{H}_i$, $i = 0, 1$. The decision is taken based on the criteria:

$$\begin{aligned} S_k &\geqslant a \Rightarrow \text{accept } \mathbf{H}_1, \\ S_k &< b \Rightarrow \text{accept } \mathbf{H}_0, \\ b \leqslant S_k &< a \Rightarrow \text{take another observation.} \end{aligned} \tag{3}$$

Thresholds a and b depend on the specified values of P_{FA} and P_M. From Wald's identity [30]

$$\mathbb{E}[S_N] = \mathbb{E}[N] \times \mathbb{E}[\Lambda], \tag{4}$$

where $\mathbb{E}[\Lambda]$ is the expected value of the logarithm of the likelihood ratio. By using a similar derivation as the one in [18, pp. 339–340], we can derive the following inequalities

$$1 - P_M \geqslant e^a P_{FA} \quad \text{and} \quad P_M \leqslant e^b(1 - P_{FA}), \tag{5}$$

where a and b are the thresholds of SPRT. When the average number of required observations is very large, the increments Λ_j in the logarithm of the likelihood ratio are also small. Therefore, when the test terminates with selection of hypothesis $\mathbf{H}_1$, S_N will be slightly larger than a, while when it terminates with selection of $\mathbf{H}_0$, S_N will be very close to b. Therefore, the above inequalities hold to a good approximation as equalities. Under this assumption, the decision levels a and b that are required for attaining performance (P_{FA}, P_M) are given by,

$$a = \ln \frac{1 - P_M}{P_{FA}} \quad \text{and} \quad b = \ln \frac{P_M}{1 - P_{FA}}. \tag{6}$$

Following the derivations of [18,30],

$$\mathbb{E}[S_N] = aP_D + b(1 - P_D), \tag{7}$$

where $P_D = 1 - P_M$ is the probability of detection of SPRT.

5.2. Minimax robust detection approach

Previously, we stressed the sequential nature of our approach and the implicit need to consider most significant attacks that result in higher chances of channel access for the attacker. An attack in that class would have most devastating effects for the network, in the sense that it would deny channel access to the other nodes and would lead to unfair sharing of the channel. Besides, if we assume that the detection of an attack is followed by communication of the attack event further in the network so as to launch a network response, it would be rather inefficient for the algorithm to consider less significant (and potentially more frequent) attacks and initiate responses for them. Instead, it is meaningful for the detection system to focus on encountering the most significant attacks and at the same time not to consume resources of any kind (processor power, energy, time or bandwidth) for dealing with attacks whose effect on performance is rather marginal.

The approach should also cope with the encountered uncertain operational environment of a wireless network, namely the random nature of protocols and the unpredictable misbehavior or attack instances. Hence, it is desirable to rely on robust detection rules that would perform well regardless of uncertain conditions. In this work, we adopt the minimax robust detection approach where the goal is to optimize the performance for the worst-case instance of uncertainty. More specifically, the goal is to identify the least favorable operating point of a system in the presence of uncertainty and subsequently find the strategy that optimizes system performance when operating at that point. In our case, the least favorable operating point corresponds to the worst-case instance of an attack and the optimal strategy amounts to the optimal detection rule. System performance is measured in terms of number of missed attacks, the number of false alarms and number of required observation samples to derive a decision.

A basic notion in minimax approaches is that of a saddle point. A strategy (detection rule) d^* and an operating point (attack) f^* in the uncertainty class form a saddle point if:

1. For the attack f^*, any detection rule d other than d^* has worse performance. Namely d^* is the optimal detection rule for attack f^* in terms of number of minimum number of required observations.
2. For the detection rule d^*, any attack f other than f^* gives better performance. Namely, detection rule d^* has its worst performance for attack f^*.

We now describe formally our approach. Let hypothesis $\mathbf{H}_0$ denote legitimate operation and thus the corresponding pdf f_0 is the uniform one. Let also Hypothesis $\mathbf{H}_1$ correspond to misbehavior with unknown pdf $f(\cdot)$.

Given the maximum allowed false alarm rate (P_{FA}) and missed detection rate (P_M), the objective of a sequential detection rule is to minimize the number of the required observation samples N so as to derive a decision regarding the existence or not of misbehavior. The performance is therefore quantified by the average number of samples $\mathbb{E}[N]$ needed until a decision is reached, where the average is taken with respect to the distribution of the observations. This number is a function of the adopted decision rule d and the attack p.d.f f, that is

$$\mathbb{E}[N] = \phi(d, f). \tag{8}$$

Let $\mathcal{D}$ denote the class of all (sequential and non-sequential) statistical hypothesis tests d for which the false alarm and missed detection probabilities do not exceed some specified levels P_{FA} and P_M respectively. Generally, a hypothesis test consists of a decision function $g(\cdot)$ that acts on a set of k observations (taking values in $\mathbf{\Omega}$) and takes values in the set of hypotheses, i.e., $g : \mathbf{\Omega}^k \to \{\mathbf{H}_0, \mathbf{H}_1\}$. Let $\mathcal{G}$ be the space of all decision functions. A sequential test is a pair $(g_T(\cdot), T)$ where T is the stopping time and $g_T(\cdot)$ is the decision function that acts on observation samples collected up to time T. Thus, $\mathcal{D} = \mathcal{G} \bigcup (\mathcal{G} \times [0, \infty])$. In the context of the minimax robust detection framework, the problem is to optimize performance in the presence of worst-case attack, that is to find d and f such that

$$\mathbb{E}[N]^* = \min_{d \in \mathcal{D}} \max_{f \in \mathcal{F}_\eta} \phi(d, f), \tag{9}$$

assuming that finite number of samples are needed (otherwise the “min-max” notation should change to “inf-sup”). We proceed to a formal definition of a saddle point.

Definition 1. A pair (d^*, f^*) is called a saddle point of the function ϕ if

$$\phi(d^*, f) \leqslant \phi(d^*, f^*) \leqslant \phi(d, f^*) \quad \forall d \in \mathcal{D}, \quad \forall f \in \mathcal{F}_\eta. \tag{10}$$

A saddle point (d^*, f^*) of ϕ consists of a detection test d^* and an attack distribution f^*. In order to find the solution of problem (9), we find the saddle point of ϕ.

However, as we now show, finding the detection strategy satisfying the saddle point is easy (if we have f^*). First, recall that the optimal detection test in the sense of minimizing expected number of samples needed for detection is the SPRT. This means that the SPRT is the test $d^* \in \mathcal{D}$, such that for a fixed (but unknown) attack f we have $\phi(d^*, f) \leqslant \phi(d, f)$ for all other tests $d \in \mathcal{D}$. The inequality above also holds for $f = f^*$, and hence the second inequality in (10) has been established. Therefore in the remainder of this paper we focus on how to obtain the worst attack distribution f^* satisfying the first inequality of Eq. (10).

5.2.1. Definition of the uncertainty class

Implicit in the minimax approach is the assumption that the attacker has full knowledge of the employed detection rule. Thus, it can create a misbehavior strategy that maximizes the number of required samples for misbehavior detection delaying the detection as much as possible. Therefore, our approach refers to the case of an intelligent attacker that can adapt its misbehavior policy so as to avoid detection. One issue that needs to be clarified is the structure of this attack strategy. Subsequently, by deriving the detection rule and the performance for that case, we can obtain an upper bound on performance over all possible attacks.

In order to quantify the performance of the detection scheme and the attacker, we introduce the parameter η, which defines the class of attacks of interest and specifies the incurred relative gain of the attacker in terms of the probability of channel access. In that sense, η can be interpreted as a sensitivity parameter of the detection scheme with respect to attacks, which is determined according to the IDS requirements.

According to the IEEE 802.11 MAC standard, the back-off for each legitimate node is selected from a set of values in a contention window interval based on a uniform distribution. The length of contention window is $2^i W$ for the i-th retransmission attempt, where W is the minimum contention window. In general, some back-off values will be selected uniformly from $[0, W]$ and others will be selected uniformly from intervals $[0, 2^i W]$, for $i = 1, \dots, I_{\max}$ where $I_{\max}$ is the maximum number of re-transmission attempts. Without loss of generality, we can scale down a back-off value that is selected uniformly in $[0, 2^i W]$ by a factor of 2^i, so that all back-offs can be considered to be uniformly selected from $[0, W]$. This scaling property emerges from the linear cumulative distribution function of the uniform distribution. An attack strategy is mapped to a probability density function based on which the attacker selects the back-off value. Although the possible back-off values are discrete, without loss of generality we use continuous distributions to represent attacks in order to facilitate mathematical treatment and to demonstrate better the problem intuition. We consider continuously back-logged nodes that always have packets to send. Thus, the gain of the attacker is signified by the percentage of time in which it obtains access to the medium. This in turn depends directly on the relative values of back-offs used by the attacker and by the legitimate nodes. In particular, the attacker competes with the node that has selected the smallest back-off value out of all nodes.

Assume that a misbehaving and legitimate node intend to access the channel. In order to have a fair basis for comparison, assume that they start their back-off timers at the same time and that none of the counters freezes due to a perceived busy channel. Let the random variable X_0 stand for the back-off value of a legitimate user, hence it is uniformly distributed in $[0, W]$. Also, let the random variables X_1 and X_2 stand for the misbehaving nodes (attackers), with unknown pdf $f_{12}(x_1, x_2)$ with support $[0, W]$. The relative advantage of the attacker is quantified as the probability of accessing the channel, or equivalently the probability that its back-off is smaller than that of the legitimate node, $\Pr(X_0 < \min(X_1, X_2))$.

Suppose that all nodes were legitimate. If p is the access probability of each node, then the probability of successful channel access achieves fairness for $p^* = 1/3$ for each node. Now, if two nodes collude, they receive gain from their attack if

$\Pr(X_0 < \min(X_1, X_2)) \leqslant \eta/3$. In order to quantify this, let $\eta \in [0, 1]$ and define the class of attacks

$$\mathcal{F}_\eta = \left\{ f_{12}(x_1, x_2) : \int_0^W \int_0^W \frac{\min(x_1, x_2)}{W} f_{12}(x_1, x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \leqslant \frac{\eta}{3} \right\}. \tag{11}$$

This class includes attacks for which the incurred relative gain compared to legitimate operation exceeds a certain amount. The class $\mathcal{F}_\eta$ is the uncertainty class of the robust approach and the parameter η is a tunable parameter. By defining the class $\mathcal{F}_\eta$, we imply that the detection scheme should focus on attacks with larger impact to system performance and not on small-scale or short-term attacks.

5.2.2. Derivation of the worst-case attack

Assuming that the SPRT is used, we seek an attack distribution f^* such that $\phi(d^*, f^*) \geqslant \phi(d^*, f)$ for all other attacks $f \in \mathcal{F}_\eta$.

From Eq. (4) the average number of samples is

$$\mathbb{E}[N] = \frac{\mathbb{E}[S_N]}{\mathbb{E}[\Lambda]} = \frac{C}{\mathbb{E}_{12}\left[\ln \dfrac{f_{12}(X_1, X_2)}{f_0(X_1)f_0(X_2)}\right]}, \tag{12}$$

where $f_0(x_i) = 1/W$ (denotes the uniform distribution of normal operation), $C = aP_D + b(1 - P_D)$, and the expectation in the denominator is with respect to the unknown attack distribution f. Since C is a constant, the problem of finding the attack that maximizes the required number of observations reduces to the problem:

$$\min_{f_{12}} \int_0^W \int_0^W f_{12}(x_1 x_2) \ln f_{12}(x_1 x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \tag{13}$$

subject to the constraints,

$$\int_0^W \int_0^W f_{12}(x_1 x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 = 1, \tag{14}$$

$$\int_0^W \int_0^W \frac{\min(x_1 x_2)}{W} f_{12}(x_1 x_2)\,\mathrm{d}x_1\,\mathrm{d}x_2 \leqslant \frac{\eta}{3}. \tag{15}$$

The first constraint enforces the fact that f is a pdf and the second one holds due to the fact that $f \in \mathcal{F}_\eta$. By applying the Karush-Kuhn-Tucker (KKT) conditions, we find that the function $f^*_{12}(x_1, x_2)$ has the following form:

$$f^*_{12}(x_1, x_2) = e^{-1-\lambda} e^{-\mu \min(x_1, x_2)/W}, \tag{16}$$

where λ and μ are the Lagrange multipliers that correspond to the constraints and are functions of W and η only. These can be obtained by the system of equations:

$$\frac{2W^2(e^{-\mu} + \mu - 1)}{\mu^2} = e^{1+\lambda} \tag{17}$$

$$\frac{2W^2}{\mu^3}(2e^{-\mu} + \mu e^{-\mu} - 2 + \mu) = \frac{\eta}{3}e^{1+\lambda}.$$

Interestingly, Eq. (16) shows that the worst-case attack distribution f_{12}^* is an exponential distribution.

Since $\phi(d^*, f^*) \geqslant \phi(d^*, f)$ for all $f \in \mathcal{F}_\eta$, we proved the left inequality in (10). We have now shown that the pair (d^*, f^*), where d^* is SPRT and $f^*(x)$ is the exponential density constitute a saddle point of ϕ. This means that the so-called minimax equality holds and we can interchange the order of min and sup in the optimization problem above [7]. Then, the problem

$$\max_{f \in \mathcal{F}_\eta} \min_{d \in \mathcal{D}} \phi(d, f) \tag{18}$$

has the same solution with (9).

As was mentioned above, the minimax robust detection approach captures the case of an intelligent adaptive attacker. The SPRT algorithm is part of the intrusion detection system module that resides at an observer node. In other words, the observer (and hence the system) attempts to minimize the number of required samples so as to improve its payoff in terms of improved chances for channel access. On the other hand, an intelligent attacker that knows the detection algorithm attempts to delay this decision as much as possible so as to increase his own benefit in terms of chances for channel access. The attacker aims at a strategy that causes performance degradation for other nodes by remaining undetected.

Naturally, if the attacker is intelligent and is aware of the optimal detection strategy of the given system, he can choose to misbehave until the estimated detection point and after that he can either obey the protocol rules for certain time or choose to relocate. The quickest detection framework employed in our analysis forces the adversary to follow the protocol rules or relocate as often as possible, thereby increasing the cost of launching an attack.

6. Experimental results

We now proceed to experimental evaluation of the analyzed scenario. In order to correctly capture the behavior of colluding attackers and evaluate the advantage over the non-colluding strategies, we compare the performance of a *single optimal attacker* from [24] with the performance of colluding attackers who generate the optimal backoff sequence according to the pdf f^*_{12}. The detection schemes employed in [24,25] use different metrics to evaluate the performance of attackers and the detection algorithms. We believe that the performance of the detection algorithms is better captured by employing the expected time before detection $\mathbb{E}[T_D]$ and the average time between false alarms $\mathbb{E}[T_{FA}]$ instead of detection delay $\mathbb{E}[N]$, used in [24], or throughput, used in [25], as the evaluation parameters.

It is important to note that the chosen values of the parameter a in all the experiments are small and vary from 10^{-2} to 10^{-10}. We claim that this represents an accurate estimate of the false alarm rates that need to be satisfied in actual anomaly detection systems [4,14], a fact that was not taken into account in the evaluation of previously proposed systems.

The backoff distribution of both optimal single attacker from [24] and optimal colluding attackers from Eq. (16) was implemented in the network simulator Opnet and tests were performed for various levels of false alarms and various values of the parameter η. The sequence of optimal backoff values was then exported to Matlab and the quickest detection tests were performed on the given sets of data.

We first analyze the effectiveness of the quickest detection scheme against colluding attackers with different levels of aggressiveness (different values of η). We chose 3 different values of η: 0.3, 0.6 and 0.9, where $\eta = 1$ represents the scenario where all nodes follow the rules of the protocol. The results of the above strategies are presented in Fig. 5. As expected, the detection delay increases with η and is almost identical for higher values of η. This re-confirms the effectiveness of the optimal SPRT-based detection scheme for detection of nodes that significantly deviate from the protocol rules. However, it is important to quantify the advantage of the colluding

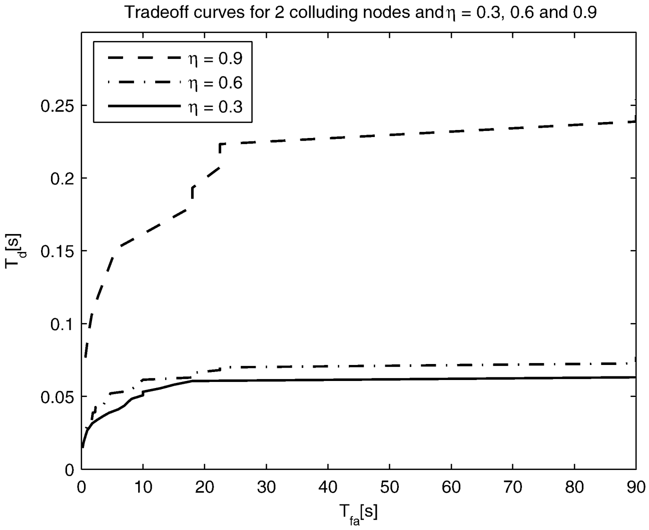

Fig. 5. Tradeoff curves for 2 colluding nodes and $\eta = 0.3$, 0.6 and 0.9.

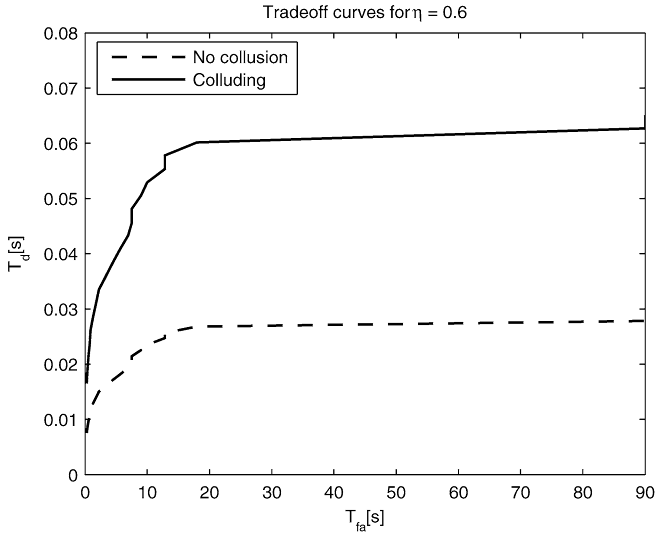

Fig. 6. Tradeoff curves for $\eta = 0.6$: detection times for colluding nodes are up to 2 times longer than for a single node with identical strategy.

scheme over a single attacker in order to justify employment of an additional attacker. It is to be expected that the colluding nodes will experience larger detection delays, depending on the η they choose for their access strategy. Figure 6 compares the performance of colluding and single attackers for $\eta = 0.6$. It is important to mention that the crucial advantage of colluding nodes is that the detection system is not aware of collaboration among the attackers and performs detection on a *single* malicious node. As expected, the detection delay for colluding nodes is approximately 2 times higher than for a single attacker. In order to illustrate the effect of η on the detection delay, we now perform the same test with $\eta = 0.9$. As it can be seen from Fig. 7, the detection delay for colluding nodes increases even further as the aggressiveness of the attackers decreases. Finally, we fix $\eta = 0.9$ for the case of a single attacker and attempt to find the corresponding value of η for the case of colluding nodes that will have the same detection delay. As it can be seen from Fig. 8, the corresponding value of η is approximately 0.4, which represents a significant gain (recall that $\eta = 0$ represents the DoS attack) and enables colluding attackers to significantly deviate from the protocol rules with the detection delay equivalent to the one when there is almost no misbehavior.

Finally, it is important to address the issue of overhead of the proposed detection algorithm. The SPRT is highly efficient since no observation vectors need to be stored. The only storage complexity is the one needed for the pdfs f_1 and f_0, the thresholds “a” and “b” and the current statistic S_n. In addition to that, the SPRT algorithm is also time-efficient, since in order to compute the log-likelihood we only need to compute the ratio of two functions (f_0 and f_1, which are very simple to

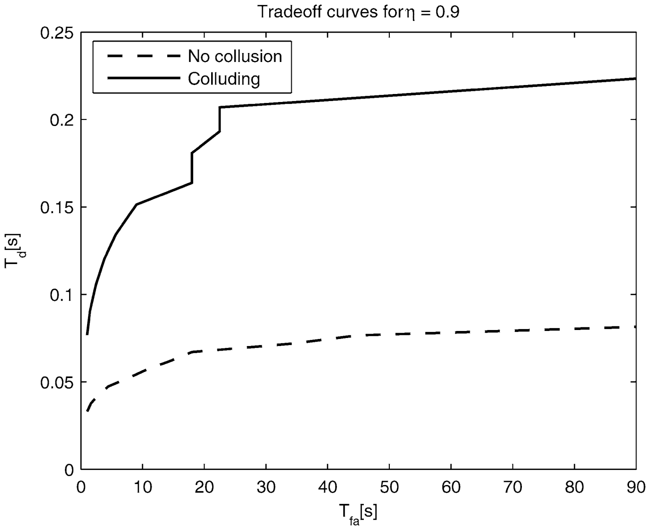

Fig. 7. Tradeoff curves for $\eta = 0.9$: detection times for colluding nodes are up to 3 times longer than for a single node with identical strategy.

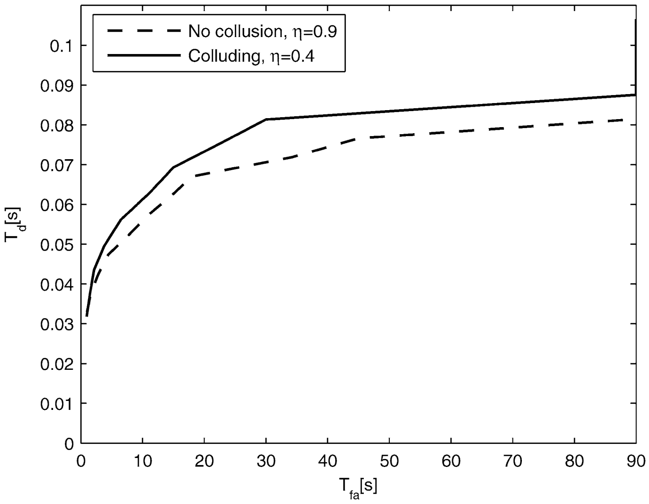

Fig. 8. Tradeoff curves for $\eta = 0.9$ (single attacker) and $\eta = 0.4$ (colluding attackers).

evaluate) and add this value to the current statistic S_n. Therefore, the overhead of the proposed algorithm is low and can be obtained by adding the two previously mentioned values.

7. Conclusions and future work

Misbehavior at the MAC layer achieved by changing the back-off mechanism can lead to performance degradation and even DoS attacks in ad hoc networks. In this paper we have presented an algorithm based on Blum's protocol, in order to prevent misbehavior of non-colluding selfish nodes. A far more challenging problem arises when two or more nodes collude in order to obtain more than fair share of channel access. Our approach encompasses the case of intelligent colluding attackers that adapt their misbehavior strategy with the objective to remain undetected as long as possible. We cast the problem within a minimax robust detection framework, characterize the worst-case misbehavior strategy showing that the optimal detection rule is SPRT. Clearly, if the attacker is ignorant of the detection mechanism, the number of required observations to detect it under the same values of P_D and P_{FA} is smaller than the corresponding value for the adaptive attacker. On the other hand, if the detection system is ignorant of the collusion among two or more protocol participants, this brings significant advantage to the attackers, as seen in Fig. 8. This gives rise to an additional issue in misbehavior detection. An intelligent detection system should perform not only optimal detection of the attacker, but should also be able to *localize* malicious colluding nodes. Our results can thus shed light in the characterization of fundamental performance limits in terms of accuracy or detection delay for misbehavior detection.

Our work constitutes the first step towards building a theoretical framework for studying the structure of network attacks in the presence of colluding nodes. We assume continuously backlogged nodes and use channel access probability as a means of measuring the benefit of the attacker and corresponding performance loss of legitimate nodes. Implicitly, we assume that fair sharing of the medium is reflected by this measure. However, fair sharing also involves the intention of a node to send a packet and therefore it is affected by packet arrivals from higher layers and backlogs at different nodes. This introduces the issue of throughput fairness and throughput benefit. The attacker causes more damage to the system if it prevents legitimate nodes from transmitting their payload. It is important to note that we do not attempt to address the problem of finding hidden terminals in this work. We assume that a monitoring node can only monitor neighboring nodes and cannot detect hidden terminals, even if they are transmitting to the same receiver. Therefore, our solution is best-effort only.

The treatment of more than one attacker in the network presents the first step in quantifying the benefits of co-operation of intelligent attackers and its effects on performance degradation of legitimate nodes.

The effects of the unreliable nature of the wireless medium on the performance of the detection scheme were not addressed in this work. The unreliable nature of the medium affects not only the detection scheme, but also affects the performance of the attackers. For example, either the monitoring nodes or one of the colluding attackers can fail to recognize RTS/CTS signaling due to the low SNR ratio, which consequently delays detection (in case the monitoring nodes fail to hear the transmission) or gives advantage to legitimate nodes (in case one of the colluding nodes fails to transmit). Finally, it would be very interesting to extend our approach and obtain results in the context of more sophisticated MAC protocols such as 802.11e with the special features regarding back-off control and differentiation in channel access opportunities that are incorporated in its enhanced DCF (EDCF) operation mode.

Acknowledgements

Research supported in part by the US Army Research Oce under CIP URI grant No DAAD19-01-1-0494.

References

[1] N. Abramson, The ALOHA system – another alternative for computer communications, *AFIPS* **37** (1970), 281–285.

[2] A. Akella, S. Seshan, R. Karp, S. Shenker and C. Papadimitriou, Selfish behavior and stability of the internet: a game-theoretic analysis of TCP, in: *Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pittsburgh, PA, 2002, pp. 117–130.

[3] L. Anderegg and S. Eidenbenz, Ad Hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents, in: *Proc. of the 9th MobiCom*, San Diego, CA, 2003.

[4] S. Axelsson, The base-rate fallacy and its implications for the difficulty of intrusion detection, in: *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99)*, 1999, pp. 1–7.

[5] J. Bellardo and S. Savage, 802.11 denial-of-service attacks: real vulnerabilities and practical solutions, in: *Proc. of the USENIX Security Symposium*, Washington, DC, 2003.

[6] M. Bellare and P. Rogaway, Random oracles are practical: a paradigm for designing efficient protocols, in: *Proc. of the ACM Conference on Computer and Communications Security*, 1993, pp. 62–73.

[7] D.P. Bertsekas, *Convex Analysis and Optimization*, Athena Scientific, 2003.

[8] V. Bharghavan, A. Demers, S. Shenker and L. Zhang, MACAW: a media access protocol for wireless LAN's, *ACM SIGCOMM Computer Communication Review* **24** (1994), 212–225.

[9] M. Blum, Coin flipping by telephone: a protocol for solving impossible problems, in: *Proc. of the 24th IEEE Spring Computer Conference, COMPCON*, 1982, pp. 133–137.

[10] S. Buchegger and J.-Y. Le Boudec, A robust reputation system for P2P and mobile ad-hoc networks, in: *Second Workshop on the Economics of Peer-to-Peer Systems*, 2004.

[11] S. Buchegger and J.-Y. Le Boudec, Nodes bearing grudges: towards routing security, fairness and robustness in mobile ad hoc networks, in: *Proc. of Tenth Euromicro PDP (Parallel, Distributed and Network-based Processing)*, Gran Canaria, 2002, pp. 403–410.

[12] S. Buchegger and J.-Y. Le Boudec, Performance analysis of the CONFIDANT protocol, in: *Proc.of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Lausanne, Switzerland, 2002, pp. 226–236.

[13] M. Cagalj, S. Ganeriwal, I. Aad and J.-P. Hubaux, On selfish behavior in CSMA/CA networks, in: *Proceedings of the IEEE Infocom*, 2005.

[14] A.A. Cárdenas, J.S. Baras and K. Seamon, A framework for the evaluation of intrusion detection systems, in: *Proc. of the 2006 IEEE Symposium on Security and Privacy*, Oakland, CA, 2006.

[15] J.R. Douceur, The Sybil attack, in: *Proc. of the 1st International Peer To Peer Systems Workshop (IPTPS 2002)*, 2002.

[16] E. Altman, R. El Azouzi and T. Jimenes, Slotted Aloha as a stochastic game with partial information, in: *Proc. of WiOpt*, 2002.

[17] V. Gupta, S. Krishnamurthy and M. Faloutsos, Denial of service attacks at the MAC layer in wireless ad hoc networks, in: *Proc. of 2002 MILCOM Conference*, Anaheim, CA, 2002.

[18] C.W. Helstrom, *Elements of Signal Detection and Estimation*, Prentice-Hall, 1995.

[19] IEEE, IEEE Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[20] P. Karn, MACA – a new channel access method for packet radio, in: *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990, pp. 134–140.

[21] P. Kyasanur and N. Vaidya, Detection and handling of mac layer misbehavior in wireless networks, in: *Proc. of the International Conference on Dependable Systems and Networks*, San Francisco, CA, 2003.

[22] A.B. MacKenzie and S.B. Wicker, Stability of multipacket slotted Aloha with selfish users and perfect information, in: *Proc. of the IEEE Infocom*, San Francisco, CA, 2003.

[23] R. Moriselli, J. Katz and B. Bhattacharjee, A game-theoretic framework for analyzing trust-inference protocols, in: *Workshop on Economics of Peer-to-Peer Systems*, 2004.

[24] S. Radosavac, J.S. Baras and I. Koutsopoulos, A framework for MAC protocol misbehavior detection in wireless networks, in: *WiSe '05: Proceedings of the 4th ACM Workshop on Wireless Security*, Cologne, Germany, 2005, pp. 33–42.

[25] M. Raya, J.-P. Hubaux and I. Aad, DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots, in: *Proc. of the Second International Conference on Mobile Systems, Applications and Services (MobiSys2004)*, Boston, MA, 2004.

[26] R. Rozovsky and P.R. Kumar, SEEDEX: a MAC protocol for ad hoc networks, in: *Proc. of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Long Beach, CA, 2001, pp. 67–75.

[27] V. Srivastava and M. Motani, Cross-layer design: a survey and the road ahead, *IEEE Communications Magazine* **43**(12) (2005), 112–119.

[28] Y. Sun, Z. Han, W. Yu and K.J.R. Liu, A trust evaluation framework in distributed networks: vulnerability analysis and defense against attacks, in: *Proc. of IEEE Infocom*, 2006.

[29] G. Theodorakopoulos and J.S. Baras, Trust evaluation in ad-hoc networks, in: *ACM Workshop of Wireless Security (Wise '04)*, 2004.

[30] A. Wald, *Sequential Analysis*, Wiley, New York, 1947.