Countering Selfish Misbehavior in Multi-channel MAC Protocols

Yan Zhang and Loukas Lazos
Dept. of Electrical and Computer Engineering, University of Arizona
Email: yanzhang@email.arizona.edu, llazos@ece.arizona.edu

***Abstract*—We address the problem of MAC-layer misbehavior in distributed multi-channel MAC protocols. We show that selfish users can manipulate the protocol parameters to gain an unfair share of the available bandwidth, while remaining undetected. We identify optimal misbehavior strategies that can lead to the isolation of a subset of frequency bands for exclusive use by the misbehaving nodes and evaluate their impact on performance and fairness. We develop corresponding detection and mitigation strategies that practically eliminate the misbehavior gains. To the best of our knowledge, this is the first attempt in characterizing the impact of misbehavior on multi-channel MAC protocols.**

I. Introduction

The availability of multiple orthogonal frequency bands has been demonstrated to substantially improve the performance of wireless networks by alleviating contention and interference [1], [12], [18], [21]. Multi-channel wireless networks can potentially achieve higher throughput and lower delay by allowing nodes located within the same collision domain to engage in parallel transmissions. However, possible performance gains are contingent on the efficient coordination of access to the available channels. Typically, this coordination is implemented at the medium access control (MAC) layer using a multi-channel MAC (MMAC) protocol (e.g., [1], [12], [18], [23]).

For networks without centralized control such as ad-hoc, sensor, and mesh networks, multi-channel access is coordinated in a distributed fashion. Distributed MMAC protocols are significantly more sophisticated than their single-channel counterparts. Efficient mechanisms are necessary for discovering the residing frequency bands of the destinations, distributing parallel transmissions over those bands, and addressing the multi-channel hidden terminal problem [18]. To address these challenges, a significant body of research has been devoted to designing MMAC protocols [1], [5], [12], [17], [18], [21]–[23], assuming that participating nodes remain protocol-compliant. However, selfish nodes that manipulate protocol parameters can gain access to a disproportional amount of bandwidth relative to well-behaved ones.

Selfish and malicious misbehavior has been extensively studied in the context of single-channel MAC protocols [6], [7], [10], [16], [24]. However, solutions for single-channel MAC protocols cannot be directly ported to MMAC protocols. In this paper, *we address the problem of identifying and mitigating possible misbehaviors in MMAC protocols.* To the best of our knowledge, this is the first work that considers the vulnerability aspects of channel access specific to the multi-channel domain.

Contributions–We detail several possible misbehavior strategies for a class of MMAC protocols that follow the split-phase design [5], [18], [23]. We adaptively optimize such strategies and show that misbehaving nodes can gain exclusive access to a desired subset of channels. As a result, selfish nodes achieve significantly higher throughput compared to protocol-compliant nodes. To mitigate the impact of misbehavior, we develop misbehavior detection and mitigation methods that provide fair access opportunities to all contending nodes and rapidly identify selfish ones. Our extensive simulations verify that our methods effectively mitigate the advantages of misbehavior and detect misbehaving nodes.

Paper Organization–The remainder of this paper is organized as follows: Section II discusses related work. In Section III, we formalize the network and adversary models. Section IV details MMAC misbehaving strategies.We develop methods for mitigating these strategies in Section V. In Section VI, we evaluate the impact of misbehavior on the network performance and demonstrate the effectiveness of our mitigation mechanisms. In Section VII, we conclude.

II. Related Work

Single-channel MAC misbehavior: Previous work on MAC misbehavior has focused on the IEEE 802.xx family of protocols (primarily IEEE 802.11) [6], [7], [10], [16], [24]. Kyasanur and Vaidya showed that misbehaving nodes that violate the CSMA/CA backoff rules of IEEE 802.11 by systematically selecting small backoff values, achieve significantly higher throughput compared to protocol-compliant nodes [10]. To mitigate this misbehavior, the authors proposed the assignment of the backoff value to a sender by a corresponding receiver, who then monitors the sender's compliance. If the sender deviates from the assigned value, it is penalized by the assignment of a larger backoff value. This solution is more suitable for infrastructure-based networks where a trusted access point assigns backoff values to possibly selfish clients. Moreover, it does not directly apply to multi-channel networks where the monitoring and monitored nodes can reside in different channels. For example, consider node A being assigned a backoff value after communicating with receiver B over channel f_1. If A switches to channel f_2 to communicate with node C for its next transmission, node B can no longer monitor A's behavior.

Cardenas et al. proposed a mechanism for detecting backoff manipulation under collusion by devising statistical tests for the mean and entropy of the backoff value distribution [4]. Raya et

al. proposed a system called DOMINO that employs a series of statistical detection mechanisms at a trusted access point [16]. A common implicit assumption for [4], [10], [16] is that nearby nodes can infer the backoff values followed by their neighbors via overhearing. This is not easily achieved when transmissions are distributed over multiple channels. Game-theoretic formulations of the impact of MAC layer misbehavior were presented in [3], [9].

Multi-channel MAC Protocols: Multi-channel MAC protocols can be classified into three categories: (a) split-phase [5], [18], [23], (b) dedicated control channel [8], [12], [17], [21], and (c) frequency hopping [1], [20]. We limit our related work description to the fist category, as our misbehavior strategies are primarily applicable to split-phase MMACs.

So et al. proposed an MMAC protocol that addresses the multi-channel hidden terminal problem [18]. In MMAC, time is divided to alternating control and data phases. During the control phase, all nodes converge to a default channel to negotiate the channel assignment for the upcoming data phase using a variant of the Distributed Coordinated Function (DCF) of IEEE 802.11. In the data phase, nodes switch to the negotiated channels to perform data transmissions. When a node has a packet for transmission, it initializes a backoff counter to a random value within $[0, cw_0]$, where cw_0 denotes the minimum contention window (CW) in slots. For every elapsed idle slot, the sender decrements its backoff counter by one unit, while the counter remains frozen when slots are sensed to be busy. When the backoff counter becomes zero, the sender transmits an *Ad hoc Traffic Indication Message* (ATIM), used as a communication request for the desired destination node. If a collision is detected (based on the timeout of an ATIM-ACK), the sender chooses a new backoff value from $[0, cw_1]$, where $cw_1 = 2cw_0$. The CW is doubled with every consecutive collision up to $cw_{\max}$, and is reset to cw_0 after a success.

If the ATIM transmission is successful, the destination selects a channel for the upcoming data phase and replies with an ATIM-ACK. The sender confirms the reservation by broadcasting an ATIM-RES packet that echoes the destination's channel selection. This channel selection is made according to a Preferable Channel List (PCL) individually maintained by each node. At the beginning of each control phase, the priority of every channel is set to medium (MID). A node i promotes the priority of channel f_j to HIGH if it reserves f_j for the following data phase, and demotes the priority of f_j to LOW if f_j is reserved by any other node. The priority of a channel can be demoted multiple times (indicated by an associated counter) if multiple reservations are placed on the same channel. The channel with the highest priority according to the sender's and receiver's PCL lists is selected, with the receiver's PCL having a higher priority than the sender's (ties are resolved arbitrarily). Channel access during the data phase is contention-based using the DCF function, as it is possible that the same channel is selected by multiple communicating pairs.

The stages of MMAC are shown in Fig. 1. A set of six nodes located within the same collision domain share three channels.

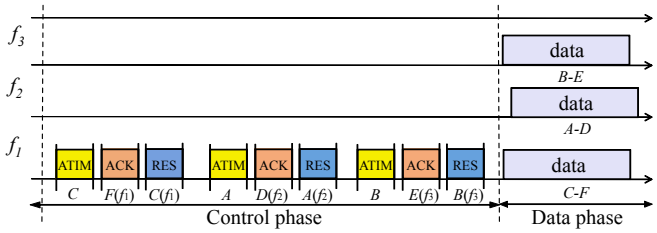

Fig. 1. Channel negotiation process in MMAC [18]. Within parenthesis, we indicate the channel selection included with each packet.

Nodes A, B, and C have data packets for nodes D, E, and F, respectively. During the control phase, node C completes a negotiation with F by reserving f_1. Nodes A, B, D, and E lower the priority of f_1 to LOW, while nodes C and F promote the priority of f_1 to HIGH. At subsequent negotiations, pairs A-D and B-E choose channels f_2 and f_3, respectively. During the data phase, all pairs engage in parallel transmissions.

Chen et al. proposed MAP [5] which extends MMAC to an adjustable data phase according to the number of successful negotiations during the control phase. Zhang et al. proposed TMMAC [23], a TDMA based multi-channel MAC protocol with a split-phase design. Unlike MMAC and MAP, in TMMAC the control phase is also dynamically adjusted to accommodate varying traffic loads.

III. Model Assumptions

System model: We consider a wireless network that operates over a set of orthogonal frequency bands (channels), denoted by $\mathcal{F} = \{f_1, f_2, ..., f_n\}$. All channels are assumed to have the same bandwidth and propagation characteristics. Nodes are equipped with a single half-duplex radio transceiver with a fixed communication range r, independent of the operating frequency. Nodes are assumed to be time-synchronized to a common slotted system. This is a common requirement for both split-phase and channel-hopping multi-channel MAC designs [18], [20]. We consider that nodes coordinate access to the set of channels using a split-phase MMAC (e.g., [5], [18], [23]). Upon detection of misbehavior, we assume that monitoring nodes can make recommendations to a reputation system [2]. Nodes with low reputation can eventually be removed from the network via a credential revocation process. Finally, communications among nodes can be authenticated using standard cryptographic techniques such as public key cryptography or symmetric key methods [19].

Adversary model: We assume an adversary aiming at gaining an unfair share of the available bandwidth by violating the MMAC specifications. Unfairness is measured in terms of the throughput achieved by the misbehaving node compared with the throughput of protocol-compliant nodes. The adversary is assumed to be independently acting (no collusion). Moreover, he only has access to his own cryptographic credentials and cannot compromise the credentials of other nodes. Therefore, he cannot launch impersonation attacks such as Sybil attacks, where he assumes identities of other nodes [11]. Physical layer attacks such as selective jamming of MMAC packets are beyond the scope of the present work.

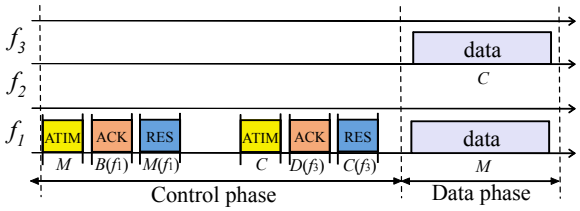

Fig. 2. Backoff manipulation attack: misbehaving node M systematically selects small backoff values, thus reserving one channel at every data phase.

IV. Multi-channel MAC Misbehavior

In this section, we identify several misbehavior strategies for the split-phase MMAC design proposed in [5], [18], [23].

A. Backoff Manipulation Attack (BMA)

In MMAC, nodes unable to complete a channel negotiation during the control phase must defer from transmission in the upcoming data phase. A misbehaving node M can manipulate the backoff mechanism of CSMA/CA by systematically selecting small backoff values to increase its chances of completing a channel negotiation. Following the same misbehavior strategy, M can capture a data channel more often than contending nodes during the data phase. The backoff manipulation attack (BMA) is particularly effective when the control channel is saturated.

We use Fig. 2 to illustrate the impact of a BMA. Similar to the scenario of Fig. 1, nodes M, C, and E have data packets for nodes B, D, and F, respectively. Node M misbehaves by selecting small backoff values. Node M immediately transmits an ATIM packet with the beginning of the control phase and completes a negotiation for channel f_1. Due to the short control phase duration, node E is unable to complete a negotiation with node F and hence, E has to defer its transmission for the following data phase. Moreover, channel f_2 remains idle. While in essence, the BMA for MMAC is similar to its single channel counterpart [10], [16], the example of Fig. 2 shows that this attack impacts MMAC in a more severe manner since not only the misbehaving node gains priority access to a channel, but other channels may be left unused.

B. Multi-reservation Attack (MRA)

In a multi-reservation attack (MRA), the misbehaving node places multiple reservations on one or several channels to make them appear less attractive to contending communicating pairs. In MMAC, this is possible via the manipulation of the PCL stored at each node. When a node overhears a reservation for a particular channel, it lowers the priority of that channel. By placing multiple reservations on a targeted subset of channels, the misbehaving node lowers the priorities of those channels so that other nodes defer from them. As a result, the misbehaving node does not have to contend during the data phase. We note that a selfish node may be interested on the exclusive use of multiple channels when acting as a transmitter to communicate on more than one channels during a single data phase. The MRA requires the cooperation of multiple nodes that would engage in a reservation process with M. Under an independent misbehaving scenario, other nodes may be unwilling to collude with M. However, the misbehaving node can place multiple reservations on one or more channels in several ways.

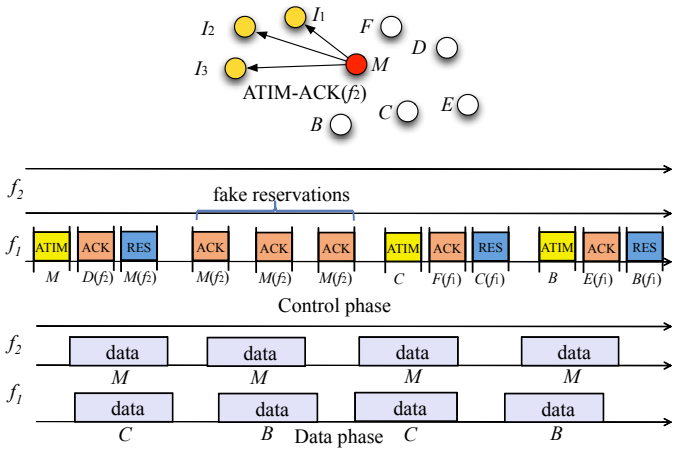

Fig. 3. M makes four reservations on f_2, by completing one negotiation with D and sending three fake ATIM-ACK packets to nodes I_1, I_2, and I_3. These nodes are presumed to be hidden terminals to nodes B-F.

Reservations with fictitious nodes: One possible strategy for M is to broadcast reservation packets to fictitious nodes. The use of such fictitious nodes becomes possible due to the hidden terminal problem. Nearby nodes cannot verify the existence of those nodes, since they could be hidden terminals.

An example of the MRA is shown in Fig. 3 where six nodes are assumed to contend for access over f_1 and f_2. Misbehaving node M wants to reserve one of the channels for exclusive use in his communication with D. Initially, the PCL values for both f_1 and f_2 are set to MID for all nodes. After M's reservation, the priority of f_2 is lowered to LOW in the PCLs of B, C, E, and F. To ensure that no other node prefers channel f_2, M broadcasts three ATIM-ACK packets as a response to fictitious reservation requests (ATIM packets) originating from imaginary nodes I_1, I_2, and I_3. All fake ATIM-ACK packets sent by M indicate f_2 as the preferred channel. As a result, nodes B, C, E, and F, lower the priority of f_2 to LOW(3) (the number within the parenthesis indicates the priority level). Note that because none of the nodes within M's range overheard the packets sent by I_1, I_2, and I_3, they assume that these nodes are hidden terminals. Because of the lowered priority of f_2, communicating pairs B-E and C-F prefer channel f_1. Thus, M monopolizes the use of f_2, while B and C have to contend on f_1. Equivalently, M can transmit sequences of ATIM/ATIM-RES packets to fictitious destinations. A requirement for a successful MRA is that M is able to place its reservations before other nodes are able to place their own. This can be achieved by combining the MRA with a BMA.

Incomplete negotiations: A misbehaving node may also be capable of placing multiple reservations without utilizing fictitious nodes. This can be achieved by engaging in a series of incomplete channel negotiations with its one-hop neighbors. According to the MMAC protocol, the sender must respond to an ATIM-ACK packet with an ATIM-RES packet that verifies the receiver's channel selection. If the sender does not reply with an ATIM-RES packet, the negotiation is not completed.

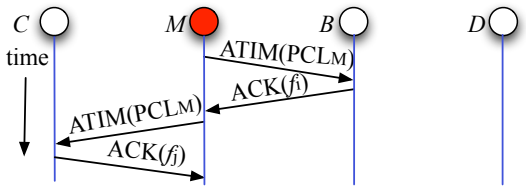

Fig. 4. M performs incomplete channel negotiation with B and C.

However, nodes in the communication range of the receiver consider the channel contained in the ATIM-ACK packet as reserved, since they could be hidden terminals to the sender. Multiple incomplete negotiations lead to the distortion of the PCL at all nodes around the receiver. We emphasize that incomplete negotiations can occur under benign conditions if the sender and the receiver do not agree on the channel selection. This could be due to other existing reservations that are only known to one of the two parties.

An example of an incomplete negotiation is shown in Fig. 4. Node M initiates negotiations with B and C, but does not respond to the ACKs received by each node. As a result, the priorities of channels f_i and f_j are lowered at the PCL of any node that overhears the ACKs from B and C.

C. Optimal Misbehavior Strategies

In this section, we analytically derive the optimal MRA strategy for guaranteeing exclusive access of the misbehaving node M to a desired subset of channels. In our analysis, we consider M contending with well-behaved pairs that want to place l reservations during one control phase. For the simplification of our analysis, we model the control phase as a series of reservation rounds. A reservation round consists of the three-way handshake process (exchange of ATIM, ATIM-ACK and ATIM-RES packets) detailed by the MMAC protocol.

Number of reservations: The number of reservations d needed by M during the control phase to guarantee the exclusive use of n_M out of n available channels during the data phase is given by the following proposition.

Proposition 1: A misbehaving node M can gain exclusive use of n_M out of n available channels when it successfully places $d = \lceil \frac{l}{n-n_M} \rceil n_M$ consecutive reservations before any well-behaved node can place a reservation. Parameter l defines the number of reservations to be placed by the contending well-behaved nodes within the same collision domain.

Proof: Without loss of generality, assume that M wants to isolate channels $\{f_1, f_2, \ldots, f_{n_M}\}$. To prevent well-behaved nodes from placing a reservation on $\{f_1, f_2, \ldots, f_{n_M}\}$, the priority of each of those channels in the PCLs of the well-behaved nodes must be lower than the priorities of all remaining $(n-n_M)$ channels at any time. Let M lower the priority of each of the n_M channels by x, by placing $d = n_M x$ consecutive reservations, evenly distributed on the n_M channels. It takes $(n - n_M)x$ reservations until the priorities of the remaining $(n - n_M)$ channels become equal to x. Equating $(n - n_M)x$ to the l reservations to be placed by well-behaved nodes and solving for d yields the desired result (d is an integer). It is straightforward to show via example, that placing d reservations is a sufficient but not necessary condition for isolating n_M channels. ■

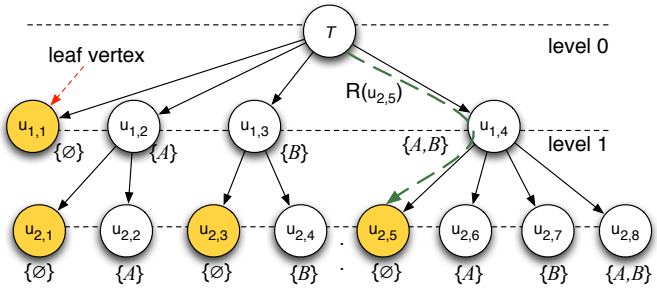

Fig. 5. Representation of the MMAC control phase as a rooted tree.

Proposition 1 suggests that the best strategy for M is to place d consecutive reservations before any contending pair obtains the opportunity to reserve a channel. This can be achieved by attempting a reservation in every reservation round until d reservations are successfully placed and the priority of each targeted channel is lowered by $\lceil \frac{l}{n-n_M} \rceil$.

Number of reservation rounds: We now evaluate the number of reservation rounds required until M successfully places d consecutive reservations. This number is computed under the assumption that M follows the optimal strategy of attempting a reservation on every reservation round by selecting a zero backoff value.

Let S denote the random variable representing the number of reservation rounds until d reservations are successfully placed by M. To find the probability mass function (pmf) of S, we model the MMAC control phase contention process after a rooted tree T with a vertex set $\mathcal{V}$. For a vertex $u_{i,j} \in \mathcal{V}$ located at the i^{th} level of T, let $p(u_{i,j})$ denote the parent of $u_{i,j}$. Let also $R(u_{i,j})$ denote the path traversed from the root of T to $u_{i,j}$. The i^{th} tree level represents the i^{th} reservation round. At each reservation round, a well-behaved pair could be in either transmit state or backoff state, depending on the backoff value selected by the sender of the pair. A vertex $u_{i,j}$ at level i represents a unique combination of the possible states of the senders of well-behaved pairs contending for the set of channels. We denote the probability of occurrence of that combination as $\Pr[u_{i,j}]$. The set of leaf vertices at level i, denoted by $\mathcal{L}_i$, corresponds to all possible state combinations for which well-behaved senders do not transmit at reservation round i, thus allowing M to seize the control channel.

The first two levels of the tree model for three contending pairs (one misbehaving and two well-behaved ones) is shown in Fig. 5. The senders of the two well-behaved pairs are denoted by A and B. For each vertex, the corresponding set of nodes in transmit state is indicated within brackets. For instance, vertex $u_{1,4}$ at level 1 with set $\{A, B\}$, represents the state where both A and B transmit in reservation round 1 (due to selecting a backoff value equal to zero). Based on the tree model of Fig. 5, the pmf of S is given by the following proposition.

Proposition 2: The pmf of S is given by,

$$\Pr[S = d + \ell] = \sum_{u_{\ell+1,j} \in \mathcal{L}_{\ell+1}} Pr[R(u_{\ell+1,j})],\ \ell \geq 0, \quad (1)$$

where $\mathcal{L}_{\ell+1}$ is the set of leaf vertices at level $\ell+1$ of T, $\Pr[R(u_{\ell+1,j})]$ is the probability of events expressed by the vertices along path $R(u_{\ell+1,j})$ of T given by,

$$\Pr[R(u_{\ell+1,j})] = \prod_{u_{w,k} \in R(u_{\ell+1,j})} \Pr[u_{w,k}],\ 1 \leq w \leq \ell+1, \quad (2)$$

and $\Pr[u_{w,k}]$ is given by,

$$\Pr[u_{w,k}] = \left(\frac{1}{\min\{cw_{w-1}, cw_{\max}\}}\right)^{n_u} \cdot \left(1 - \frac{1}{\min\{cw_{w-1}, cw_{max}\}}\right)^{n_p - n_u}, \quad (3)$$

where n_u and n_p are the number of nodes in transmit state for $u_{w,k}$ and $p(u_{w,k})$ respectively, and $cw_i = 2^{i-1}cw_0, i \geq 1$.

Proof: We first show that if M adopts the optimal strategy of attempting a reservation at every reservation round (i.e. always selecting a zero backoff value), then after M's first successful reservation, M will successfully place the remaining $(d-1)$ reservations in the next $(d-1)$ reservation rounds, without contending. In order for M to have a successful reservation on a given reservation round, all senders of the well-behaved pairs must have a backoff counter greater than zero. Because M chooses a zero backoff value on all consecutive rounds, well-behaved nodes do not get the opportunity to decrement their backoff counter, once they have selected a backoff value greater than zero. Thus, M successfully places the remaining $(d-1)$ reservations in the following $(d-1)$ rounds. For the $(\ell+1)^{th}$ $(\ell \geq 0)$ reservation round, the event that none of the well-behaved nodes is in transmit state is represented by the leaf vertices at level $(\ell+1)$. Summing over all probabilities of arriving at the leaf vertices of level $(\ell+1)$, yields the probability of having M's first successful reservation at the $(\ell+1)^{th}$ reservation round. Or equivalently, this event yields the probability of finishing d reservations at reservation round $(\ell+1)+(d-1) = d+\ell$. Eq. (1) follows by noting that all events at a given level are mutually exclusive.

The probability of arriving at any vertex $u_{i,j}$ is equal to the probability product of all vertices along the path $R(u_{i,j})$. This path corresponds to a unique combination of events (transmit states) at reservation rounds 1 to $(i-1)$ that lead to the unique combination of transmit/backoff states expressed by $u_{i,j}$. Because backoff values are independently selected at each round, $\Pr[R(u_{i,j})]$ is equal to the product of the vertex probabilities, which yields eq. (2).

Finally, we compute the probability $\Pr[u_{w,k}]$ of the unique combination of transmit/backoff states expressed by vertex $u_{w,k} \in \mathcal{V}$. Let n_u denote the number of transmitting nodes in $u_{w,k}$, and n_p the number of transmitting nodes in the parent node $p(u_{w,k})$. The CW of a transmitting node at level w is equal to $\min\{cw_{w-1}, cw_{max}\}$ since that node must have collided $(w-1)$ times with M in order to be transmitting at reservation round w. Moreover, the probability of a node transmitting at reservation round w is equal to the probability of selecting a zero backoff value at that round, which is equal to $\frac{1}{\min\{cw_{w-1}, cw_{max}\}}$. Eq. (3) follows by noting that it expresses the probability of exactly n_u nodes being in transmit state at stage w when n_p nodes were in transmit state in stage $(w-1)$. This is equivalent to a particular subset of size n_u choosing a zero backoff value, while the complementary subset of size $(n_p - n_u)$ chooses any other value. All nodes choose their backoff values independently of each other, independently of previous rounds, and in a random fashion. Therefore, a node is in transmit state with probability $\frac{1}{cw_{w-1}}$, or $\frac{1}{cw_{\max}}$ if the CW has reached its maximum value. We note that the event expressed by eq. (3), corresponds to one unique combination of transmit and backoff states for the participating senders, and *not any combination* that yields n_u transmitting nodes. Therefore, it is not given by a binomial distribution. ■

Adaptive reservation strategy: In realistic scenarios, the number of reservations l to be placed by contending nodes at a given control phase is not known a priori. To account for an unknown l, we propose an adaptive strategy for capturing n_M channels which is as follows.

Step 1: Misbehaving node M lowers the priority of the n_M targeted channels by one unit, by selecting a zero backoff value and placing n_M reservations before any other node can place a reservation. It then defers from further reservations.

Step 2: M repeats Step 1 every time $(n - n_M)$ reservations are placed on the remaining $(n - n_M)$ channels.

It is straightforward to show that the n_M targeted channels always have a lower priority than the remaining $(n-n_M)$ ones. Therefore, based on the PCL rules, a well-behaved node will always select one of the $(n - n_M)$ channels, giving exclusive use of the n_M channels to M. Moreover, this adaptive strategy is optimal in the number of reservations needed for capturing n_M channels. If the number of reservations placed during Step 1 is less than n_M, some of the targeted channels will have equal priority as the remaining $(n - n_M)$ ones and therefore, be equally likely preferred by well-behaved nodes. The adaptive reservation strategy shows that the condition of Proposition 1 is a sufficient but not necessary condition for the exclusive use of n_M channels by M.

V. Mitigating MMAC Misbehavior

A. Mitigation of the Backoff Manipulation Attack

We first consider the manipulation of the backoff mechanism which is adopted during both the control and data phases. As explained in Section II, mechanisms proposed for single-channel networks (e.g., [10], [16]) are not adequate for multi-channel networks, because they are not designed to consistently monitor parallel transmissions on multiple channels. To address this limitation, we propose a BMA detection scheme that utilizes a priori publicized backoff sequences to detect deviations from a random backoff strategy during both the control and the data phase. Our scheme consists of the backoff generation and the backoff monitoring modules.

Backoff generation module: The backoff generation module is responsible for generating a *public* random sequence that is used to compute the random backoff times for each transmission. A node i uses a public pseudo-random number generator G and a seed s_i (e.g., the unique node id) to generate a random sequence of numbers in $[0, 1]$. The seed s_i is published to all one-hop neighbors of i, denoted by $\mathcal{N}_i^1$. Let q_i denote the number of packets sent by node i. The value of q_i is initialized to zero with the publishing of s_i and incremented by one with every packet sent by i (including retransmitted ones). Here, we only consider packets transmitted after the expiration of the backoff counter such as ATIM packets (control phase) and RTS packets (data phase), and ignore all other packets (ACKs, data, etc.) that are transmitted at fixed times relative to the expiration of a backoff counter. Let also r_i denote the number of times that the last packet has been retransmitted. This number is set to zero at the beginning of each phase or after every successful packet transmission and is incremented by one every time node i retransmits a packet due to a timeout. Using the values (q_i, r_i), the backoff for the current transmission is:

$$b_i(q_i, r_i) = \lceil G(q_i, s_i) \times \min\{2^{r_i} cw_0, cw_{\max}\} \rceil, \qquad (4)$$

where $G(q_i, s_i)$ is the q_i^{th} value of the pseudo-random sequence generated based on seed s_i. Using eq. (4), one-hop neighbors of node i that know (q_i, r_i) can verify compliance with backoff value b_i for the q_i^{th} transmission.

Backoff monitoring module: The backoff monitoring module identifies misbehaving nodes that adopt smaller backoff values compared with their public schedule. To achieve this, monitoring nodes keep track of parameters (q_i, r_i) that allow the computation of $b_i(q_i, r_i)$. Values (q_i, r_i) are included in every ATIM, ATIM-RES and RTS packet transmitted by each sender i. During the control phase, all one-hop neighbors $\mathcal{N}_1^i$ of i can verify compliance with the correct backoff value $b_i(q_i, r_i)$. When transitioning to the data phase, only the subset of one-hop neighbors $\mathcal{K}_i^1 \subseteq \mathcal{N}_i^1$ that hop to the channel reserved by i can continue to monitor i. This subset contains at least the node who will be communicating with i during the data phase.

When nodes converge to the control channel at the end of a data phase, all nodes in $\mathcal{K}_i^1$ remain synchronized to the correct values (q_i, r_i), based on the transmissions that occurred during the data phase. However, nodes in $\mathcal{N}_i^1 \backslash \mathcal{K}_i^1$ are not aware of i's data transmissions, since they hopped to other channels. For the first control phase transmission, node i is monitored solely by nodes in $\mathcal{K}_i^1$. The remaining one-hop neighbors synchronize with parameters (q_i, r_i) after the first successful reception of an ATIM or ATIM-RES packet.

Parameter manipulation: We now consider the scenario where a misbehaving node M manipulates (q_M, r_M) to select a random number $G(q_M, s_M)$ that leads to a short backoff time. This is possible because monitoring nodes cannot attribute collided packets to their senders. Hence, a misbehaving sender can: (a) avoid the incrementation of q_M and r_M if its transmission collided and (b) take advantage of other collisions to advance q_M to a future value $q_M + k, k > 1$ that yields a smaller $G(q_M + k, s_M)$.

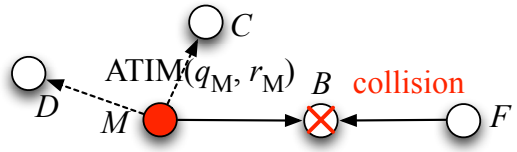

Fig. 6. Detection of the manipulation of the retransmission number r_M during the control phase.

For case (a), assume that M does not increment (q_M, r_M) after one of its packets has collided. This strategy will prevent the increase of the CW due to the increase of r_M. However, since collisions are receiver-dependent, a misbehaving sender cannot know with certainty that a collision has indeed occurred at all monitoring nodes. This scenario is depicted in Fig. 6. M broadcasts an ATIM packet for node B. Nodes C and D act as monitors of M. The ATIM packet sent by M collides at B due to a concurrent reception of an ATIM packet from F. Monitoring nodes C and D still receive M's ATIM packet, as they are located outside the interference range of F. A retransmission of an ATIM packet with the same (q_M, r_M) is detected by C and D as misbehavior. Moreover, the intended receiver may choose to intentionally drop correctly received packets in order to detect a misbehaving sender that does not increase q_M or r_M. For strategy (b), Proposition 3 shows that it eventually leads to either the identification of misbehavior or does not reduce the backoff time.

Proposition 3: Let q_M be the sequence number of the last successful transmission of node M. Advancing the sequence number of the next transmission to $q_M + k, k > 1$ strictly increases the backoff time compared with $k = 1$.

Proof: Assume that M successfully completed a transmission with sequence number q_M. For its next transmission, let M skip $k > 1$ values from the pseudo-random sequence and use $G(q_M + k, s_M)$ instead of $G(q_M + 1, s_M)$, because $G(q_M + k, s_M) \ll G(q_M + 1, s_M)$. In order for M to be protocol-compliant, (a) the advertised value r_M must increase by $(k-1)$ and (b) M's transmission must start no earlier than

$$(k-1)t_0 + \sum_{j=1}^{k} \lceil G(q_M + j, s_M) \min\{2^{(j-1)} cw_0, cw_{\max}\} \rceil \qquad (5)$$

slots after the q_M^{th} transmission, where t_0 denotes the timeout period in slots. Condition (a) holds true since q_M is considered to be the last successful transmission for which the retransmission number r_M is reset to zero. In order for M to use $(q_M + k)$ in its next successful attempt, $(k-1)$ unsuccessful ones must have preceded, thus setting the value of r_M to $(k-1)$. In this case, node i must have followed all intermediate backoff intervals as indicated by the values $\{b_M(q_M + 1, 0), b_M(q_M + 2, 1), \ldots, b_M(q_M + k, k-1)\}$. Moreover, an additional period equal to the $(k-1)$ previous unsuccessful trials of transmitting a packet and waiting for a timeout (time t_0) must be added to the overall delay. Adding the $(k-1)$ timeout times and the $(k-1)$ backoff periods yields condition (b). From (b), the proposition immediately follows ($b_M(q_M + 1, 0)$ is a factor of the sum in (5)). ■

B. Mitigation of Multi-reservation Attacks

In this section, we mitigate the MRA by modifying the MMAC PCL update rules and detecting fictitious nodes.

PCL update rules: In the original MMAC, a node placing n reservations on a channel f_i lowers the priority of f_i by n units. We modify the MMAC PCL rules such that the priority of any channel is based on the number of distinct sources scheduled to operate on a channel, rather than the number of reservations. For example, consider the attack shown in Fig. 3. Node M places four reservations on f_2, while node C places one reservation on f_1. Both f_1 and f_2 have the same priority in the PCLs of the nodes overhearing the channel negotiations. Hence, node B may choose f_1 or f_2 with equal probability, were it to communicate with another node. For multiple reservations across multiple channels, we modify the PCL rules such that only the first reservation affects the priority of the corresponding channel. Subsequent reservations placed by the same node do not have any effect on the PCL. The new PCL rules are as follows:

a) The priority of all channels is set to MID at the beginning of each control phase.

b) If a pair of nodes i, j agrees to communicate on channel f_ℓ, the priority of f_ℓ in PCL$_i$ and PCL$_j$, is promoted to HIGH.

c) The priority of a channel in HIGH state cannot be changed.

d) If a node i overhears an ATIM-ACK or ATIM-RES packet indicating the selection of channel f_ℓ with priority LOW, it checks whether the packet originator has placed another reservation within the same control phase. If so, i does not update its PCL; otherwise the priority of f_ℓ is demoted.

Detection of fictitious nodes: When the modified PCL rules are followed, M can manipulate the PCL of its neighbors only if he emulates the existence of fictitious sources. This can be achieved if M responds to imaginary ATIM packets by broadcasting ATIM-ACKs. Any node that overhears those ATIM-ACKs assumes that the source of the corresponding ATIM/ATIM-RES packets is a hidden terminal. In order to detect the existence of such fictitious sources, we employ a secure neighbor discovery mechanism that determines the two-hop network topology. While several secure neighbor discovery mechanisms have been proposed in the literature, typically they consider strong adversary models that include node collusion, deployment of wormholes, and others [14], [15]. For our purposes, we develop a simple scheme that is efficient to implement. Our scheme involves the following steps.

Step 1: Every node i broadcasts a hello message $HELLO_i$, signed by i. Every node i builds a one-hop neighbor list $\mathcal{N}_i^1$.

Step 2: Every node i broadcasts $\mathcal{N}_i^1$, signed by i. Using $\mathcal{N}_j^1, \forall j \in \mathcal{N}_i^1$, each node i creates its two-hop neighbor list $\mathcal{N}_i^2$. A node $k \in \mathcal{N}_j^1$ is a two-hop neighbor of node i, if $k \notin \mathcal{N}_i^1$. Two-hop neighbors are hidden terminals.

Step 3: Every node i broadcasts $\mathcal{N}_i^2$, signed by i. Every node i creates a parent list $P_i(j)$ for each node $j \in \mathcal{N}_i^2$, containing the set of nodes $P_i(j) = \{m : m \in \mathcal{N}_i^1, j \in \mathcal{N}_m^1\}$. List $P_i(j)$ indicates the one-hop neighbors of i directly connected to hidden terminal j. If $|P_i(j)| = 1$, node i suspects node j to be fictitious and node $P_i(j)$ to be misbehaving. This is because no other node but $P_i(j)$ can verify the existence of j.

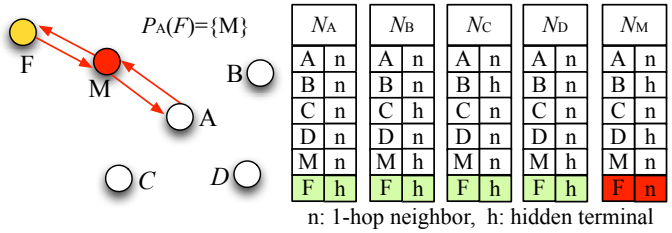

Fig. 7. Secure neighbor discovery protocol. Tables N_i show combined 1-hop and 2-hop topological information.

Step 4: For a suspected fictitious node j, node i issues a signed challenge c, sent via $P_i(j)$. Node j must reply to the challenge with a signed response.

The application of our secure neighbor discovery protocol is shown in Fig. 7. Tables N_i show the combined 1-hop and 2-hop topological information. For node $F \in \mathcal{N}_A^2$, it holds that $P_A(F) = \{M\}$. That is, node F appears to be a hidden terminal to all neighbors of A, except M and hence, M is suspected of misbehavior. Node A challenges F via M. If F cannot reply with an authentic response, M is accused of misbehavior. This is true since M cannot sign on behalf of F.

Note that our secure neighbor discovery protocol is vulnerable to the collusion of two or more nodes. Two colluding nodes can place a fictitious node F to their one-hop neighbor lists thus avoiding the challenge-response phase. Resilience to collusion can be achieved by challenging every two-hop neighbor regardless of the number of parent nodes, at the expense of increased communication overhead. We leave the detailed exposition on collusion as future work.

VI. Performance Evaluation

A. Simulation Setup

We performed our experiments using the OPNET$^{\text{TM}}$ Modeler packet-level simulator [13]. We considered a single-hop network topology of multiple node pairs communicating over three orthogonal channels of capacity 2Mbps. We implemented the MMAC protocol as a MAC layer module that was embedded in every node. Unless otherwise stated, the control phase was fixed to 20ms and the data phase was fixed to 80ms [18]. The arrival process at the MAC layer of each source was assumed to follow the Poisson distribution with parameter λ. Each data packet was assumed to be 512 bytes. Misbehavior strategies were implemented on a single sender. The simulation duration was set to 40s (equivalent to 400 rounds of control/data phases) and results were averaged over 40 simulation runs.

B. Impact of the Backoff Manipulation Attack

In the first set of experiments, we evaluated the impact of the BMA. We considered a misbehaving node M that fixed its CW to four during both the control and data phases, and contended with 10 well-behaved pairs. Fig. 8(a) shows the average throughput (T) of M compared to the per-flow throughput of well-behaved nodes and the per-flow throughput

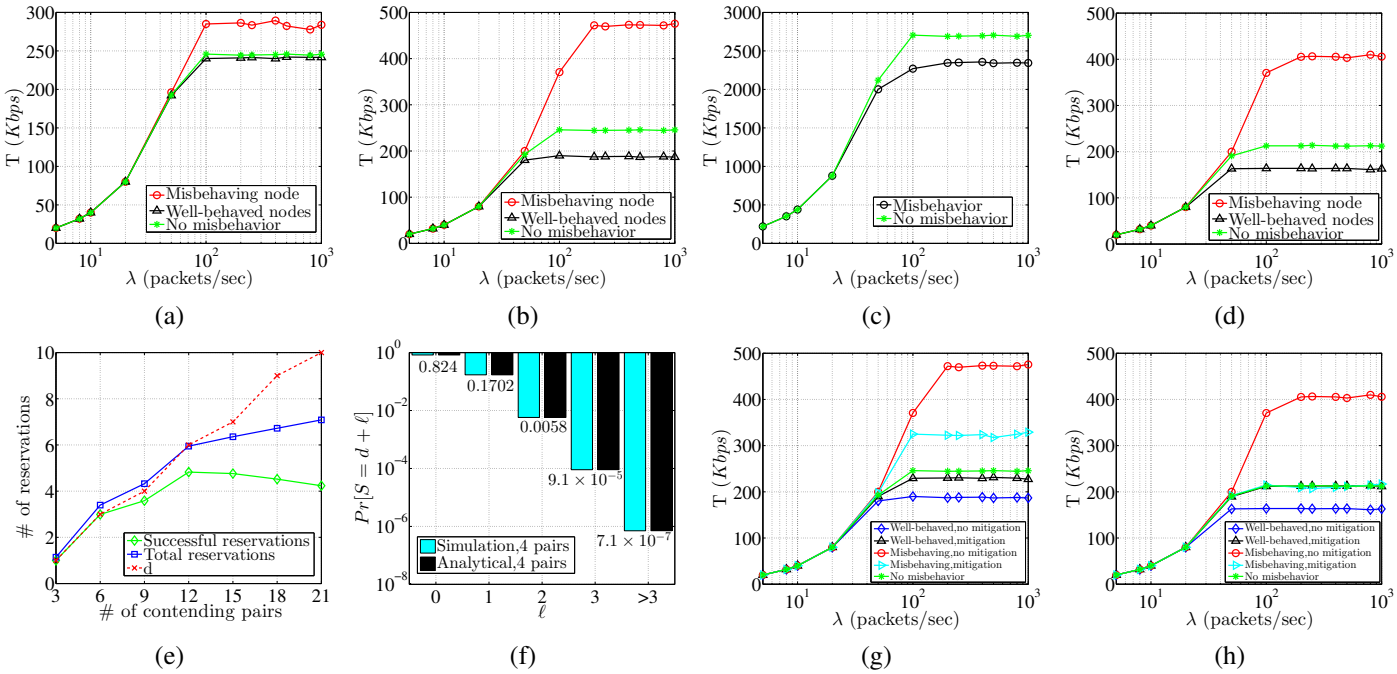

Fig. 8. (a) T as a function of λ under a BMA, when the misbehaving node fixes CW with four compared to the T of well-behaved nodes and in the absence of misbehavior, (b) T as a function of λ under an MRA and BMA, (c) aggregate T for all contending pairs in the presence and absence of misbehavior, (d) T as a function of λ under an MRA and BMA, when the control phase duration is increased to 30ms, (e) number of reservations needed to isolate a single channel as a function of the number of contending pairs, (f) pmf of the total number of reservation attempts for isolating a single channel (theoretical and simulation), as a function of ℓ, (g) T as a function of λ for the misbehaving and well-behaved nodes, under the modified PCL rules, (h) T as a function of λ under the modified PCL rules, for a control phase duration equal to 30ms.

in the absence of misbehavior, as a function of the packet arrival rate λ. We observe that for low λ, the throughput of all flows is identical. However, as the control channel becomes saturated, the misbehaving node gains a significant advantage (about 20%) compared to well-behaved pairs.

C. Impact of the Multi-reservation Attack

In the second set of experiments, we evaluated the impact of an MRA when combined with a BMA. We considered one misbehaving node aiming at isolating a single channel ($n_M = 1$) when contending with 10 well-behaved pairs. According to Proposition 1, the misbehaving node placed five reservations (one real and four fake ones) on the targeted channel to guarantee exclusive use of that channel.

Fig. 8(b) shows the average throughput of the misbehaving node vs. the average throughput of well-behaved nodes and the average per-flow throughput in the absence of misbehavior. We observe that the misbehaving node achieves almost three times the throughput of any other source under traffic saturation conditions. This is because the misbehaving node does not have to share its channel during the data phase, while well-behaved pairs have to contend in the remaining two channels. Moreover, the throughput of well-behaved nodes is reduced by 25% (approximately 60Kbps) compared to the scenario where all nodes follow the MMAC protocol. In Fig. 8(c), we show the aggregate network throughput in the presence and in the absence of misbehavior. We observe that selfish misbehavior significantly degrades the overall network performance.

In Fig. 8(d), we show the average throughput of contending pairs under an MRA, but for a control phase duration of 30ms. A longer control phase allows nodes more time for negotiating channel assignments, but reduces the number of data phases that can fit within our simulation period. We observe that misbehavior has a similar impact on the throughput of contending pairs, although all sources achieve lower throughput due to the increased duration of the control phase.

D. Evaluation of the Adaptive Misbehavior

In the third set of experiments, we evaluated the adaptive reservation strategy proposed in Section IV-C. The misbehaving node placed multiple reservations adaptively, depending on the reservations of other contending pairs to gain exclusive use of a single channel. Fig. 8(e) compares the number of successful reservations of the adaptive strategy, the number of reservations d required for guaranteeing exclusive use of one channel according to Proposition 1, and the total number of attempted reservations (including collisions), as a function of the number of contending pairs. We observe that the adaptive strategy requires significantly less successful reservations than d when contention increases. In fact, when the control channel becomes saturated (more than 12 contending pairs), the number of successful reservations needed by the misbehaving node reduces because well-behaved nodes place fewer reservations during the control phase due to contention. Hence, reaching the theoretical limit that guarantees exclusive channel use is unnecessary. On the other hand, the total number of attempted reservations increases with the number of contending pairs, since the misbehaving node faces higher levels of contention.

Furthermore, we validated the analytical result obtained in Proposition 2 via simulations. Fig. 8(f) shows the pmf of S as a function of ℓ for a topology with four contending pairs (the y-axis is in logarithmic scale). The numbers indicated below the bars are the exact analytical pmf values for $\ell = 0, 1, 2, 3$ and $\ell > 3$ respectively. From Fig. 8(f), we observe that the pmf rapidly decreases to negligible probability values with the increase of ℓ. This indicates that when launching a BMA with a backoff value equal to zero, the misbehaving node successfully places the required reservations within the first few slots.

E. Mitigation of MMAC Misbehavior

In this set of experiments, we evaluated the effectiveness of our mitigation methods. Fig. 8(g) shows the average throughput of the misbehaving node and the average per-flow throughput of well-behaved nodes under the modified PCL update rules listed in Section V-B. We observe that when multiple reservations do not affect the PCL, the adversary's throughput drops by 150Kbps while the throughput of well-behaved sources increases by 40Kbps per flow. The misbehaving node still gains a throughput advantage due to the selection of small backoff values, but this advantage is significantly reduced. The throughput gap is attributed to the short duration of the control phase that does not always allow all 11 contending pairs to complete their channel negotiations. However, the backoff manipulation attack is easily detectable by the scheme developed in Section V-A.

Moreover, the effect of a BMA is practically eliminated if we consider a control phase with longer duration. In Fig. 8(h), we show the throughput of the misbehaving node for a control phase of 30ms. With the application of the modified PCL rules, the throughput of the misbehaving node becomes equal to that of well-behaved ones, even if the misbehaving node is allowed to select small backoff values. This is because the control phase is long enough for all communicating pairs to make reservations for the data phase. These reservations are equally distributed across all available channels.

VII. Conclusion

We addressed the problem of MAC layer misbehavior in multi-channel wireless networks. For MMACs following a split-phase design, we described possible misbehaving strategies that yield a significant throughput advantage to misbehaving nodes. We showed that misbehaving nodes can isolate a significant portion of the available bandwidth by placing multiple reservations on the available channels in a timely manner. We developed countermeasures that mitigate the impact of misbehavior and lead to the detection of misbehaving nodes. We verified the effectiveness of our mitigation methods via extensive packet-level simulations and showed that the throughput of misbehaving nodes is equalized to the throughput of well-behaved nodes.

VIII. Acknowledgments

This research was supported in part by the National Science Foundation (under grants CNS-0844111, CNS-1016943, and CNS-1145913). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] P. Bahl, R. Chandra, and J. Dunagan. SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In *Proc. of the MOBICOM conference*, pages 216–230, 2004.

[2] S. Buchegger and J. Le Boudee. Self-policing mobile ad hoc networks by reputation systems. *IEEE Communications Magazine*, 43(7):101–107, 2005.

[3] M. Cagalj, S. Ganeriwal, I. Aad, and J. Hubaux. On selfish behavior in CSMA/CA networks. In *Proc. of the INFOCOM conference*, volume 4, pages 2513–2524, 2005.

[4] A. Cardenas, S. Radosavac, and J. Baras. Detection and prevention of MAC layer misbehavior in ad hoc networks. In *Proc. of the WiSe conference*, pages 17–22, 2004.

[5] J. Chen, S. Sheu, and C. Yang. A new multichannel access protocol for IEEE 802.11 ad hoc wireless LANs. In *Proc. of the PIMRC conference*, volume 3, pages 2291–2296, 2003.

[6] L. Guang, C. Assi, and Y. Ye. DREAM: A system for detection and reaction against MAC layer misbehavior in ad hoc networks. *Computer communications*, 30(8):1841–1853, 2007.

[7] V. Gupta, S. Krishnamurthy, and M. Faloutsos. Denial of service attacks at the MAC layer in wireless ad hoc networks. In *Proc. of the MILCOM conference*, volume 2, pages 1118–1123, 2002.

[8] N. Jain, S. Das, and A. Nasipuri. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proc. of the ICCCN conference*, pages 432–439, 2001.

[9] J. Konorski. A game-theoretic study of CSMA/CA under a backoff attack. *IEEE/ACM Trans. on Networking (TON)*, 14(6):1167–1178, 2006.

[10] P. Kyasanur and N. Vaidya. Selfish MAC layer misbehavior in wireless networks. *IEEE Trans. on Mobile Computing*, pages 502–516, 2005.

[11] B. Levine, C. Shields, and N. Margolin. A survey of solutions to the sybil attack. Technical Report 052, University of Massachusetts Amherst, 2006.

[12] T. Luo, M. Motani, and V. Srinivasan. Cooperative asynchronous multichannel MAC: Design, analysis, and implementation. *IEEE Trans. on Mobile Computing*, 8(3):338–352, 2009.

[13] OPNET. OPNET technologies inc. http://www.opnet.com, 2009.

[14] R. Poovendran and L. Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. *Wireless Networks*, 13(1):27–59, 2007.

[15] M. Poturalski, P. Papadimitratos, and J. Hubaux. Secure neighbor discovery in wireless networks: formal investigation of possibility. In *Proc. of the ASIACCS conference*, pages 189–200, 2008.

[16] M. Raya, J. Hubaux, and I. Aad. DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots. In *Proc. of the MobiSys conference*, pages 84–97, 2004.

[17] J. Shi, T. Salonidis, and E. Knightly. Starvation mitigation through multi-channel coordination in CSMA multi-hop wireless networks. In *Proc. of the MOBIHOC conference*, pages 214–225, 2006.

[18] J. So and N. Vaidya. Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *Proc. of the MOBIHOC conference*, pages 222–233, 2004.

[19] D. Stinson. *Cryptography: theory and practice*. CRC press, 2006.

[20] Z. Tang and J. Garcia-Luna-Aceves. Hop reservation multiple access for multichannel packet radio networks. *Computer Communications*, 23(10):877–886, 2000.

[21] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proc. of the I-SPAN conference*, pages 232–237, 2002.

[22] S. Wu, Y. Tseng, C. Lin, and J. Sheu. A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. *The Computer Journal*, 45(1):101–110, 2002.

[23] J. Zhang, G. Zhou, C. Huang, S. Son, and J. Stankovic. TMMAC: An energy efficient multi-channel mac protocol for ad hoc networks. In *Proc. of the ICC conference*, pages 3554–3561, 2007.

[24] Y. Zhou, D. Wu, and S. Nettles. Analyzing and preventing MAC-layer denial of service attacks for stock 802.11 systems. In *Proc. of the Workshop on Broadband Wireless Services and Applications*, 2004.