SECURE AND SPECTRALLY-EFFICIENT CHANNEL ACCESS IN MULTI-CHANNEL WIRELESS NETWORKS

by

Yan Zhang

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2 0 1 5

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Yan Zhang
entitled Secure and Spectrally-Efficient Channel Access in Multi-Channel Wireless Networks
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Dr. Loukas Lazos — Date: 05/14/2015

Dr. Marwan Krunz — Date: 05/14/2015

Dr. Onur Ozan Koyluoglu — Date: 05/14/2015

Date: 05/14/2015

Date: 05/14/2015

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director: Dr. Loukas Lazos — Date: 05/14/2015

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

Yan Zhang

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisor, Professor Loukas Lazos, for all the help, advice, patience, supports, and encouragements he gave to me during the past six years. Because of him, my graduate experience has been the one that I will cherish forever. I am fortunate to have him as my advisor who taught me how to think and research. He enlightened me through his professional knowledge about where to explore and what is necessary to get there. This dissertation would have been impossible without his guidance.

I would like to thank Professor Marwan Krunz and Professor Onur Ozan Koyluoglu for serving on my Ph.D. committee and for providing all the insightful comments, valuable suggestions, and discussions that make this dissertation better. Your time and effort are greatly appreciated.

I would like to thank my labmates and colleagues for their help and friendship, including Sisi Liu, Diep Nguyen, Alejandro Proano, Bocan Hu, Kai Chen, Swetha Shivaramaiah, Nirnimesh Ghose, Nicholas Fragiskatos, and Jose Carlos Acedo. I would also thank Tami Whelan for handling all the paperwork and giving various forms of support during my graduate study.

To my parents and my husband Yequn Zhang: Thank you for supporting and encouraging me throughout my Ph.D. study. Your love and care give me strength to overcome difficulties throughout this endeavor. You have always been my source of inspiration, confidence, and success.

Finally, I would like to acknowledge the US National Science Foundation (NSF) and the US Army Research Office (ARO) for the financial support provided to conduct the research needed for this dissertation. This research was supported in part by the NSF under grants CNS-1409172, CNS-0844111, and CNS-1016943 and ARO grant W911NF-13-1-0302. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF and the ARO.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF FIGURES – *Continued*

LIST OF TABLES

ABSTRACT

Wireless services have become an indispensable part of our social, economic, and everyday activities. They have facilitated and continue to facilitate rapid access to information and have created a highly-interconnected web of users who are untethered to particular locations. In fact, it is expected that in the very near future, the number of users that access the Internet through their mobile devices will surpass those access the Internet from the fixed infrastructure [1]. Aside from mobile Internet access, wireless technologies enable many critical applications such as emergency response, healthcare and implantable medical devices, industrial automation, tactical communications, transportation networks, smart grids, smart homes, navigation, and weather services.

The proliferation and wealth of wireless applications has created a soaring demand for ubiquitous broadband wireless access. This demand is further fueled by the richness of the information accessed by users. Low-bit rate voice communications and text have been replaced with graphics, high-definition video, multi-player gaming, and social networking. Meeting the growing traffic demand poses many challenges due to the spectrum scarcity, the cost of deploying additional infrastructure, and the coexistence of several competing technologies. These challenges can be addressed by developing novel wireless technologies, which can efficiently and securely manage multi-user access to the wireless medium. The multi-user access problem deals with the sharing of the wireless resource among contending users in an efficient, secure, and scalable manner.

To alleviate contention and interference among the multiple users, contemporary wireless technologies divide the available spectrum to orthogonal frequency bands (channels) [2–5]. The availability of multiple channels has been demonstrated to substantially improve the performance and reliability of wireless networks by alle-

viating contention and interference [6–9]. Multi-channel networks, whether cellular, sensor, mesh, cognitive radio, or heterogeneous ones, can potentially achieve higher throughput and lower delay compared to single-channel networks.

However, the gains from the existence of orthogonal channels are contingent upon the efficient and secure coordination of channel access. Typically, this coordination is implemented at the medium access control (MAC) layer using a multi-channel MAC (MMAC) protocol [7, 10–13]. MMAC protocols are significantly more sophisticated than their single-channel counterparts, due to the additional operations of destination discovery, contention management across channels, and load balancing. A significant body of research has been devoted to designing MMAC protocols. The majority of solutions negotiate channel assignment every few packet transmissions on a default control channel. This design has several critical limitations. First, it incurs significant overhead due to the use of in-band or out-of-band control channels. Second, from a security standpoint, operating over a default control channel constitutes a *single point of failure*. A DoS attack on the control channel(s) would render all channels inoperable. Moreover, MMAC protocols are vulnerable to misbehavior from malicious users who aim at monopolizing the network resources, or degrading the overall network performance.

In this dissertation, we improve the security and spectral efficiency of channel access mechanisms in multi-channel wireless networks. In particular, we are concerned with MAC-layer misbehavior in multi-channel wireless networks. We show that selfish users can manipulate MAC-layer protocol parameters to gain an unfair share of network resources, while remaining undetected. We identify possible misbehavior at the MAC-layer, evaluate their impact on network performance, and develop corresponding detection and mitigation schemes that practically eliminate the misbehavior gains. We extend our misbehavior analysis to MAC protocols specifically designed for opportunistic access in cognitive radio networks. Such protocols implement additional tasks such as cooperative spectrum sensing and spectrum management. We then discuss corresponding countermeasures for detecting and mitigating these misbehavior.

We further design a low-overhead multi-channel access protocol that enables the distributed coordination of channel access over orthogonal channels for devices using a single transceiver. Compared with prior art, our protocol eliminates in-band and out-of-band control signaling, increases spatial channel reuse, and thus achieves significant higher throughput and lowers delay. Furthermore, we investigate DoS attacks launched against the channel access mechanism. We focus on reactive jamming attacks and show that most MMAC protocols are vulnerable to low-effort jamming due to the utilization of a default control channel. We extend our proposed MMAC protocol to combat jamming by implementing cryptographic interleaving at the PHY-layer, random channel switching, and switching according to cryptographically protected channel priority lists. Our results demonstrate that under high load conditions, the new protocol maintains communications despite the jammer's effort. Extensive simulations and experiments are conducted to evaluate the impact of the considered misbehaviors on network performance, and verify the validity of the proposed mechanisms.

CHAPTER 1

INTRODUCTION

1.1 Background

Wireless services have become an indispensable part of our social, economic, and everyday activities. They have facilitated and continue to facilitate rapid access to information and have created a highly-interconnected web of users that is untethered to particular locations. In fact, it is expected that in the very near future, the number of users that access the Internet through their mobile devices will surpass the number of users that access the Internet from the fixed infrastructure [1]. Aside from mobile Internet access, wireless technologies enable many critical applications such as emergency response, healthcare and implantable medical devices, industrial automation, tactical communications, transportation networks, smart grids, smart homes, navigation, weather services, and many others.

The proliferation and wealth of wireless applications has created a soaring demand for ubiquitous broadband wireless access. This demand is further fueled by the richness of the information accessed by users. Low-bit rate voice communications and text have been replaced with graphics, high-definition video, multi-player gaming, and social networking. Meeting the growing traffic demand poses many challenges due to the spectrum scarcity, the cost of deploying additional infrastructure, and the coexistence of several competing technologies. These challenges can be addressed by developing novel wireless technologies, which can efficiently and securely manage multi-user access to the wireless medium. The multi-user access problem deals with the sharing of the wireless resource among contending users in an efficient, secure, and scalable manner. An example of multi-user access for an infrastructure network is shown in Figure 1.1. Six terminals compete for access to the same base station over the same spectrum.

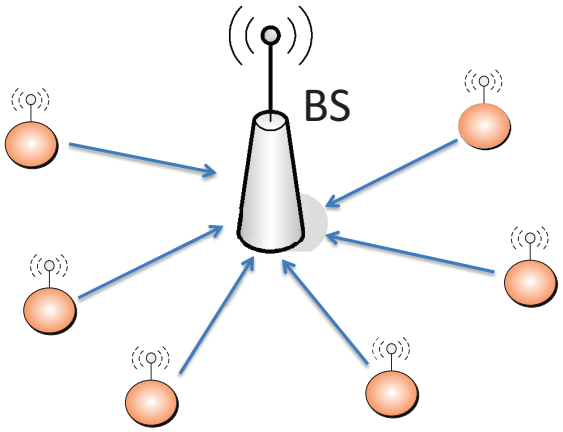


Figure 1.1: Multiple users share access to an infrastructure based network.

To alleviate contention and interference among the users, contemporary wireless technologies divide the available spectrum to multiple orthogonal frequency bands (channels) [2–5]. The availability of multiple channels has been demonstrated to substantially improve the performance and reliability of wireless networks by alleviating contention and interference [6–9]. Multi-channel networks, whether cellular, sensor, mesh, cognitive radio, or heterogeneous ones, can potentially achieve higher throughput and lower delay compared to single-channel networks. Typically, access coordination among multiple users can be classified into two categories: (a) contention-free methods (e.g., FDMA, TDMA, CDMA [14]) and (b) contention-based methods, (e.g., Aloha, slotted Aloha, CSMA [15]). Contention-free MAC protocols allocate the spectrum resource in a deterministic manner. Therefore, they are more suitable for wireless applications with predicable traffic demands. On the other hand, contention-based methods rely on user coordination for accessing the available channels on an on-demand basis. Such access mechanisms improve the spectrum efficiency when traffic demands vary significantly.

In an ideal scenario, we are interested in using N channels to improve the throughput of a single channel network by a factor of N. However, the gains from the existence of non-interfering channels are contingent upon the efficient and secure coordination of channel access. Typically, this coordination is implemented

at the medium access control (MAC) layer using a multi-channel MAC (MMAC) protocol [7,10–13]. MMAC protocols are significantly more sophisticated than their single-channel counterparts, due to the additional operations of destination discovery, contention management across channels, and load balancing. To cope with these challenges, a significant body of research has been devoted to designing throughput-efficient MMAC protocols [7,8,10–13,16–18]. However, there are several limitations exhibited on prior art.

First, most existing MMAC designs assume that participating terminals are protocol-compliant. However, selfish terminals can manipulate protocol parameters in order to gain access to a disproportional amount of bandwidth relative to well-behaved ones. The misbehaving terminals can gain a significant performance advantage, even if they do not violate the protocol specifications. Mechanisms for detecting and mitigating selfish misbehaviors are therefore necessary to secure the MMAC protocols.

Second, to facilitate the channel access coordination, the majority of existing MMAC protocols negotiate channel assignment every few frame transmissions over a default control channel [8,16,19–23]. From a spectral-efficiency point of view, the usage of control channels lowers spectrum utilization as typically no data transmissions are allowed on the control channels. It could be the case that all the data channels become saturated while the control channels remain underutilized. From a security point of view, convergence on a preassigned control channel constitutes a *single point of failure*. An adversary can severely degrade the network performance by launching a Denial of Service (DoS) attack on the control channel, thus negating any gain due to the availability of multiple data channels. One of the most effective DoS attacks is the jamming of the wireless medium. In this attack, an adversary interferes with the set of frequency bands used for communication by transmitting a continuous jamming signal [24], or several short jamming pulses [25]. Control channel jamming is particularly devastating for wireless networks due to their cooperative nature.

In this dissertation, we improve the security and spectral efficiency of distributed

MMAC protocols. We develop several communication mechanisms/protocols that protect distributed multi-channel access process in a spectral efficient manner. Our main contributions are outlined in the following subsection.

1.2 Main Contributions

1.2.1 Detection and Mitigation of Selfish Misbehavior in Multi-Channel MAC Protocols

We analyze selfish misbehavior in distributed MMAC protocols. We identify misbehavior strategies for popular classes of MMAC protocols and develop corresponding detection and mitigation mechanisms to alleviate their impact on performance and fairness. We show that selfish terminals can isolate frequency bands for exclusive use without violating the protocol specifications. We develop corresponding detection and mitigation strategies that practically eliminate the throughput gains due to misbehavior. We validate our results via extensive packet-level simulations for varying traffic load scenarios. Moreover, we identify the vulnerabilities of MMAC protocols specifically designed for CRNs, exploited by selfish/malicious cognitive radio users (CRs). Possible countermeasures for detecting and mitigating these vulnerabilities are also discussed.

1.2.2 Spectrally-Efficient Medium Access without Control Channels

Aside from selfish misbehaviors, medium access in multi-channel networks also faces other security vulnerabilities. In particular, networks deployed in hostile environments are susceptible to DoS attacks by adversaries targeting the functionality of the control channel [26–28]. This is due to the fact that to coordinate parallel transmissions across channels without interference, most existing MMAC protocols require channel assignments between the communicating pairs prior to their actual data transmissions. The channel assignments are typically negotiated over a default control channel [8, 10, 16, 19, 20, 22, 23], which in turn constitutes a single point of failure. If the adversary is successful, transmissions will be prevented on the en-

tire available spectrum even if other frequency bands are still operational. One of the most effective ways for denying access to the control channel is by jamming it. In multi-channel wireless networks without centralized control, control channel jamming is particularly devastating due to their cooperative nature.

Based on the above analysis, to design a secure yet spectrally-efficient MMAC protocol, several critical factors must be taken into consideration. First, the use of dedicated control channels should be avoided for both security and spectrum efficiency reasons. Second, low-overhead mechanisms must be employed at the senders to discover the resident channels of their respective destinations. The delay for destination discovery should be minimized in order to achieve better spectrum utilization. Third, parallel transmissions must be efficiently distributed over all available channels to balance the traffic load and alleviate contention/congestion.

We design a protocol called FD-MMAC that exploits recent advances in full-duplex (FD) communications to coordinate channel access in a distributed manner. Compared to prior MMAC designs, our protocol eliminates the use of dedicated in-band or out-of-band control channels for resolving contention, discovering the resident channel of destinations, and performing load balancing. The elimination of the control channel improves spectral efficiency and alleviates DoS attacks that specifically target the exchange of control information. Moreover, FD-MMAC enables the operation of multi-channel exposed terminals. To achieve these goals, we integrate an advanced suite of PHY-layer techniques, including self interference suppression, error vector magnitude and received power measurements, and signal correlation. We validate the proposed PHY-layer techniques on the NI USRP testbed. Furthermore, we theoretically analyze the throughput performance of FD-MMAC and verify our analysis via packet level simulations. Our results show that FD-MMAC achieves significantly higher throughput compared with prior art.

1.2.3 Jamming-Resistant Medium Access Control in Multi-Channel Wireless Networks

We analyze the jamming attacks against distributed MMAC protocols in multi-channel wireless networks. We define a comprehensive reactive jamming model for the multi-channel domain, based on the cross-layer consideration of the PHY and MAC layers. We show that most MMAC protocols are vulnerable to low-effort jamming attacks due to the utilization of a default control channel for medium access coordination. We further extend our proposed FD-MMAC protocol to combat jamming. The FD-MMAC protocol enables autonomous destination discovery without convergence to a default control channel. Each terminal switches independently between channels based on its own channel state view (idle/busy) and cryptographically-protected priority channel lists. This coordination differs from classical frequency hopping systems because it does not rely on pre-agreed hopping sequences, but is adaptive to the individual channel conditions. We evaluate the performance of FD-MMAC under the reactive jamming attacks through extensive packet-level simulations.

1.3 Dissertation Organization

The remainder of the dissertation is organized as follows. In Chapter 2, we discuss related works. In Chapter 3, we identify selfish misbehavior strategies for existing MMAC protocols, and present our schemes for detecting and mitigating the impact of the considered misbehaviors. In Chapter 4, we propose a spectrally-efficient MMAC protocol based on full-duplex communications that does not rely on a common control channel. In Chapter 5, we analyze the anti-jamming properties of the MMAC protocol proposed in Chapter 4, and explore possible improvements on the protocol to combat jamming. Finally, Chapter 6 summarizes the contributions of this dissertation and suggests several topics for future research.

CHAPTER 2

RELATED WORK

2.1 MAC-layer Misbehavior in Single-Channel Wireless Networks

Contention-based medium access control is primarily mediated using the CSMA family of protocols [2–4,29]. In CSMA, a terminal A with outgoing frames first senses the status of the shared medium before attempting to transmit. If a carrier is sensed which indicates the medium to be busy, A waits for the transmission in progress to finish before initiating its own transmission. If no carrier is sensed, A immediately transmits a frame on the medium. To overcome the hidden terminal problem in wireless networks [30], collision avoidance (CA) mechanism based on RTS/CTS (Request-to-Send/Clear-to-Send) handshaking is implemented in conjunction with CSMA. The most well-known MAC protocol based on CSMA/CA is the *Distributed Coordination Function* (DCF) in IEEE 802.11 family standards.

In DCF, terminals reserve the channel for data transmission by exchanging RTS/CTS messages with respective destination terminals. A sender with outgoing data frames in its transmission queue first initiates an RTS frame to the destination. Upon detection of the RTS frame, the destination replies to sender with a CTS frame if it is available for reception. Both RTS and CTS frames include the NAV (Network Allocation Vector) values, indicating the expected duration for which the channel will be occupied. Neighboring terminals overhearing these frames defer their transmissions for the duration specified by the NAV values. The operation stages of DCF are shown in Figure 2.1. In the depicted scenario, terminal A wants to transmit a data frame to terminal B, while C and D are neighbors of A and B respectively. To avoid collisions in virtual carrier sensing stage, a random backoff procedure based on CSMA/CA is incorporated before A sending its RTS frame to B. Once A successfully decoding the CTS from B, A starts to transmit the data

frame to B. B finally replies to A an acknowledge (ACK) frame to indicate the successful reception of the data frame.

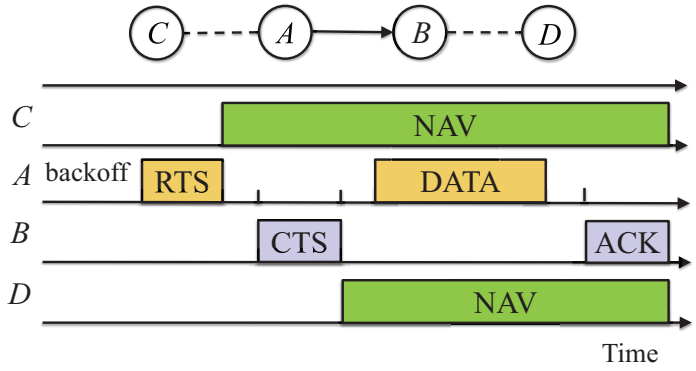


Figure 2.1: Operations of IEEE 802.11 DCF.

Previous work on MAC-layer misbehavior has focused on the IEEE 802.xx family of protocols (primarily IEEE 802.11) [31–38]. Kyasanur and Vaidya showed that misbehaving terminals 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 terminals [31]. This misbehavior strategy can be illustrated using Figure 2.2. Assume that terminals M and C have data frames destined for B and D respectively. All four terminals are within the same collision domain, and thus M and C need to contend for accessing the single available channel. M misbehaves by selecting a zero backoff value every time before initiating its RTS to B. As a result, the channel is solely occupied by M as C always overhears M's RTS during its backoff period and thus defers from transmission. To mitigate such 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 terminals can reside in different channels. For example, consider

terminal 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 terminal C for its next transmission, terminal B can no longer monitor A's behavior.

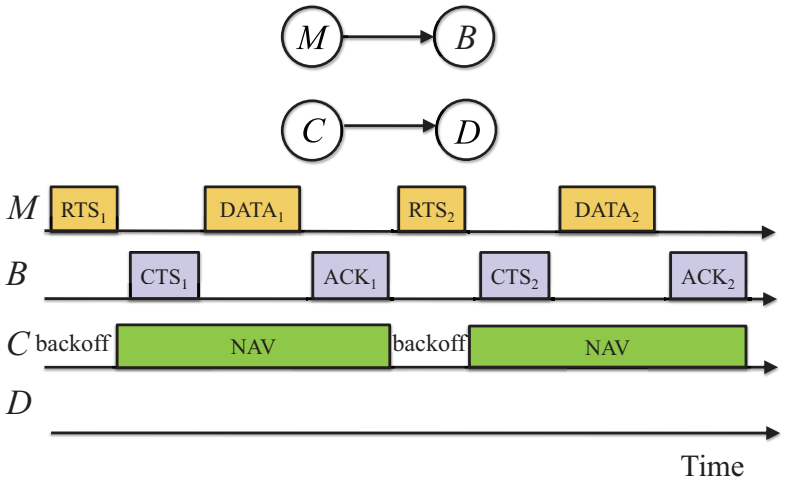


Figure 2.2: Selfish misbehavior that manipulates the backoff rule in 802.11 DCF.

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 [39]. Raya et al. proposed a system called DOMINO that employs a series of statistical detection mechanisms at a trusted access point [32]. A common implicit assumption for [31, 32, 39] is that nearby terminals can infer the backoff values followed by their neighbors via overhearing. This is not easily achieved when transmissions are distributed over multiple channels.

The impact of MAC layer misbehavior in single channel networks was studied using game-theoretic formulations in [36–38,40]. In such formulations, each terminal was modeled as a selfish player who aims at maximizing its own throughput (utility), by manipulating the 802.11 parameters (e.g., setting the CW to a low value). By modeling the selfish misbehavior as a static game [36, 37], two families of Nash equilibria were characterized. In the first family, all players receive zero throughput and it always results in a network collapse. This is because players aggressively select low backoff values, leading to perpetual collisions. In the second family, one

player receives a non-zero throughput while the rest receive zero throughput. This provides incentives for misbehaving terminals to cooperate with others. Dynamic games were also used to model selfish misbehavior in the backoff mechanism of 802.11 [36–38]. Based on this model, misbehavior was addressed by developing detection and punishment techniques. Upon detecting a misbehaving terminal, other terminals penalize it by either selectively jamming its packets [36], or using more aggressive 802.11 parameter configurations [38]. Several works have studied the MAC performance when power control is applied at both selfish and malicious nodes [40, 41]. The terminal behavior has been modeled after one-stage Bayesian games and dynamic repeated games. Finally, game theory inspired protocols have been used to optimize MAC layer throughput in non-adversarial settings [42, 43].

Gupta et al. investigated the effects of DoS attacks at the MAC layer [34]. They showed via simulations that maintaining fairness can help mitigate the effects of such attacks. Zhou et al. proposed countermeasures for MAC layer DoS attacks under colluding adversaries [35]. Guang et al. proposed a system called DREAM that mitigates the effects of frequent timeout of MAC frames at the sender or the receiver [33].

2.2 MAC-layer Protocols for Multi-Channel Wireless Networks

When there are multiple channels available, new MAC protocols are needed as control information exchange is necessary to make channel assignment. Current MMAC protocols for devices equipped with half-duplex transceivers can be classified into three categories: (a) split-phase (SP-MMAC) [8, 16, 23, 44, 45], (b) dedicated control channel (DCC-MMAC) [7, 9–12, 17, 46], and (c) rendezvous [6, 18, 47].

2.2.1 Split-phase

The idea of splitting time into multiple phases to facilitate MAC operations was first explored in MAC protocols proposed for the single-channel domain [44,45]. Acharya et al. proposed the MACA-P protocol which enables simultaneous transmissions in

multi-hop wireless ad hoc networks [44]. In MACA-P, transmissions can proceed in parallel as long as a transmitter and a receiver operating concurrently are not located within the same collision domain. To coordinate parallel transmissions, a control gap (control phase) is scheduled between the RTS/CTS exchange and subsequent DATA/ACK exchange of the terminal pair first seizing the channel. During the control phase, qualified neighboring pairs exchange RTS/CTS packets to reserve the channel for concurrent transmissions to and synchronize the upcoming data transmissions with the first pair.

Muqattash et al. [45] further improved the throughput of single-channel MAC protocols by enabling parallel transmissions over the same channel and collision domain. They proposed a single-channel MAC protocol named POWMAC which follows the split-phase design, similar to the MACA-P protocol. However, unlike MACA-P protocol, POWMAC allows simultaneous transmissions in the vicinity of a receiver even if the transmissions cause some limited interference to that receiver. By adjusting the power of transmitters operating around a receiver, the interference at that receiver is controlled. Information on the power bounds to be followed by nearby transmitters is embedded on CTS packets. Simultaneous transmissions by nearby terminals are negotiated during an access window (control phase). The length of the access window is dynamically adjusted based on localized information to accommodate the competing terminals.

In the multi-channel domain, MMACs following the split-phase design (SP-MMACs) usually divide time to alternating control and data phases [8,16,23,48,49]. During the control phase, all terminals 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, terminals switch to the negotiated channels to perform data transmissions. So et al. proposed an MMAC protocol that addresses the multi-channel hidden terminal problem [8].

When a terminal has a frame 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 terminal. 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 frame that echoes the destination's channel selection. This channel selection is made according to a Preferable Channel List (PCL) individually maintained by each terminal. Typically in SP-MMAC, the PCL records the priority of every channel perceived by a particular terminal. At the beginning of each control phase, the priority of every channel is set to medium (MID). A terminal 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 terminal. 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 Figure 2.3. A set of six terminals located within the same collision domain share three channels. Terminals A, B, and C have data frames for terminals D, E, and F, respectively. During the control phase, terminal C completes a negotiation with F by reserving f_1. Terminals A, B, D, and E lower the priority of f_1 to LOW, while terminals 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.

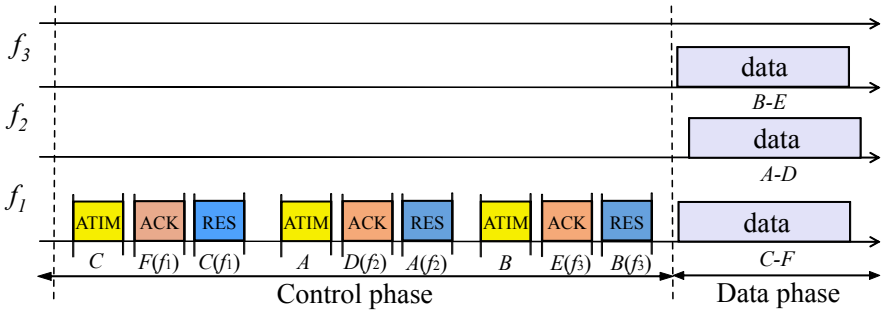


Figure 2.3: Channel negotiation process in MMAC [8]. Within parenthesis, we indicate the channel selection included with each frame.

Chen et al. proposed MAP [23] which extends MMAC to an adjustable data phase according to the number of successful negotiations during the control phase. Zhang et al. proposed TMMAC [16], 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. Additionally, TMMAC not only allows terminals to reserve channels as in [8, 23], it also allows terminals to reserve both channels and corresponding time slots in data phase based on TDMA technique. Chen et al. [49] proposed a traffic aware MMAC protocol (TAMMAC) which adopts a similar design as [8], with the exception that both the control and data phase durations can be dynamically adjusted according to traffic load. A terminal is allowed to request an increase or decrease on the control phase duration to its neighbors once per time interval when one of the adjustment rules is satisfied. Moreover, a terminal is allowed to initiate multiple negotiations to different destinations in a time interval and extend its data transmission to next time interval if needed.

Incel et al. [48] proposed a schedule-based MMAC protocol named MC-LMAC for wireless sensor networks based on a single radio equipped on each sensor node. In MC-LMAC, time is slotted and each timeslot consists of a common frequency (CF) period, a control message (CM) period, and a data transmission (DATA) period. To transmit, a sender first selects a timeslot together with a channel and notifies corresponding receiver during the CF period over a common channel. All

sensor nodes listen to the common channel during CF period in order to be informed about intended transmissions. Upon hearing the invitation from sender, the receiver switches to the sender's channel for communication. The sender then transmits control messages followed by the data frame. MC-LMAC minimizes collisions and achieves high throughput by scheduling parallel interference and contention free transmissions over multiple channels.

Multi-channel hidden terminal problem: The multi-channel hidden terminal problem can be described using the topology of Figure 2.4. Let terminals A and B reside on channel f_1, while terminal C resides on f_2. Topologically, C is a hidden terminal to A. Assume that A performs an RTS-CTS exchange over f_1 before communicating frame P_A to B. Let the transmission of P_A start at time t_0 and terminate at t_1. Assume that C switches to f_1 at t_2 with $t_0 < t_2 < t_1$. Because $t_2 > t_0$, terminal C will not overhear CTS_B. Moreover, the transmission of P_A is ongoing when C switches to f_1. At time $t_3 < t_1$, terminal C causes a collision at B.

SP-MMAC protocols avoid multi-channel hidden terminals by performing channel negotiations on a default control channel during the control phase. However, no data transmissions take place during this phase, thus decreasing the overall spectral efficiency. The control phase can be considerably long under high-contention conditions.

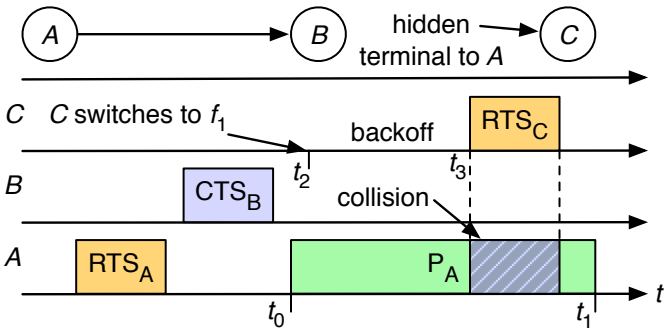


Figure 2.4: The multi-channel hidden terminal problem.

2.2.2 Dedicated Control Channel

In DCC-MMAC designs, one of the available channels is exclusively reserved for exchanging control messages [7,9–12,17,46], while the rest are solely for data transmissions. Several protocols operate under the assumption that terminals are equipped with at least two transceivers, one of which is constantly tuned to the control channel [7, 9, 10, 13, 46, 50]. Typically, access to data channels are also reservation based, similar to SP-MMAC. On the control channel, terminals with outgoing frames contend to reserve data channels for transmissions with their respective destinations using control transceivers. Upon a successful negotiation, both sender and receiver tune their data transceivers to the reserved channel and perform data transmissions. Terminals monitor the traffic on the control channel all the time in order to keep track of the usage status of data channels.

Wu et al. proposed a dynamic channel assignment (DCA) protocol [9] which relies on a dedicated control channel to perform on-demand channel assignment. Because one radio transceiver is always tuned to the control channel, all terminals within the same collision domain overhear the channel assignments. In DCA, each terminal keeps record of the time period during which each data channel is expected to be occupied. Here we use the same terminology for control frames as SP-MMAC to illustrate operation stages of DCA using Figure 2.5 (a). In the considered scenario, six terminals $A-F$ located within same collision domain share access to one control channel and three data channels f_1, f_2, and f_3. Terminals A, C, E have data frames for terminals B, D, F, respectively. Suppose that A captures the control channel first after proper backoff. Before sending an ATIM frame, A prioritizes channels based on their expected release time and builds a list of idle channels. Denote the ordered idle channel list by PCL_A. Since all three data channels are idle at the beginning, PCL_A contains f_1, f_2, f_3 with same priority 0 (0 is deemed as highest priority). A then transmits an ATIM frame to B with PCL_A included. Upon receiving the ATIM, B builds PCL_B in the same way, compares it with PCL_A, and selects a channel idle for both sides. If multiple channels are available, the one with

the highest priority according to both PCL lists is selected, with PCL_B having a higher priority (ties are broken arbitrarily). Since PCL_B is equal to PCL_A at this time, B randomly selects a channel to use, say f_2. B then notifies A about the selected channel f_2 by replying with an ATIM-ACK frame. The ATIM-ACK frame also notifies the reservation on f_2 to neighboring terminals of B. Once receiving the ATIM-ACK from B, A transmits an ATIM-RES frame also indicating channel f_2. The purpose of ATIM-RES is to positively confirm the channel selection at B, as well as to notify A's neighbors about the reservation on f_2. At the same time, A and B both tune their data transceivers to f_2 and start to perform data transmission. Other terminals update the reservation status for the reserved channel f_2. In case no common channel is available, B rejects A's request by sending back a rejection frame, including the minimum estimated waiting time before a data channel will be released. A can then retry after at least this waiting time. Following the same procedure, communicating pairs $E-F$ and $C-D$ complete their data transmissions respectively after successful channel negotiations on the control channel. The PCL updating process at terminal C is shown in Figure 2.5 (b). C timely updates the expected release times for channels and re-prioritizes them accordingly every time it overhears a reservation for a particular channel. After detecting the first channel negotiation between $A-B$ reserving f_2, f_2's priority becomes the lowest at C as it now has the latest release time. Shortly after $E-F$ reserves f_1, C further adjust its PCL table by ordering channel priority as $f_3 > f_2 > f_1$, where f_1 is assigned the least priority (a priority of 2) due to its latest release time. Upon C finally completes its own channel reservation with D, the three channels are reordered as $f_2 > f_1 > f_3$. Other terminals update their PCL lists following the same procedure.

Instead of prioritizing channels based on their release times, terminals can also measure and maintain other relevant information for each data channel, such as signal to interference-plus-noise ratio (SINR), for channel selection purpose. As an example, when channel reuse is taken into consideration [19], each terminal maintains a separate channel reuse index for data channels which participates in building the PCL together with the release time. It is worth to notice that channels do not

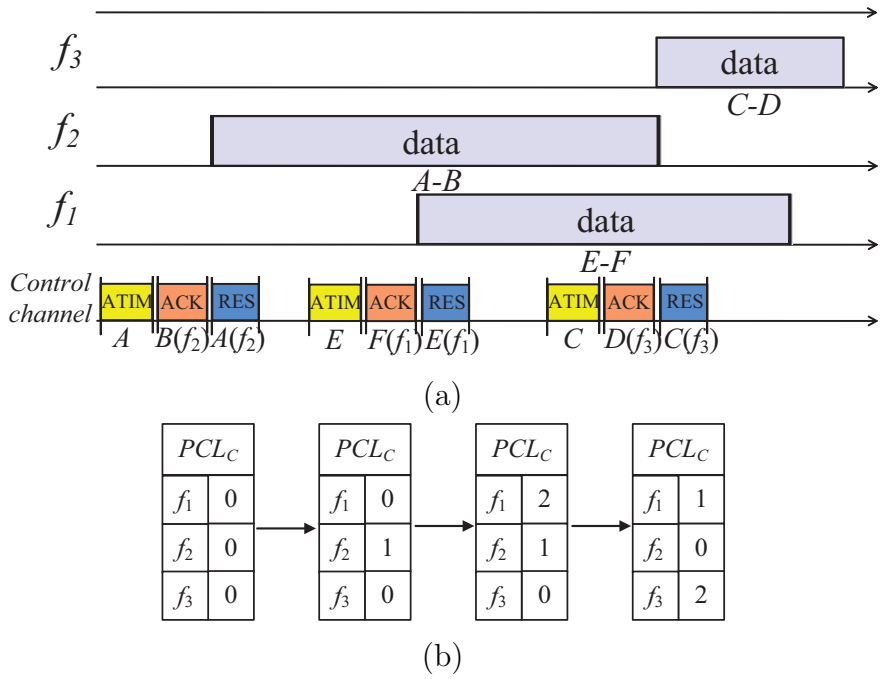


Figure 2.5: (a) Operation stages of DCA [9], (b) PCL update at terminal C.

necessarily to be idle in order to be reserved at the time of negotiation. The PCL lists can simply include all data channels and more complicated rules may be employed to prioritize channels. In case a busy channel is selected, the communicating pair waits until it becomes idle and contends to use it following CSMA/CA like mechanism. Jain et al. [46] proposed a receiver-based channel selection (RBCS) scheme, which makes use of receiver-side channel state information to select the best channel at the sender.

DCC-MMAC can also be built based on a single transceiver [7,12,17,51–53]. Shi et al. proposed an asynchronous protocol called AMCP which aims at mitigating the coordination problems that lead to flow starvation [17]. In AMCP, terminals contend to make channel reservations on the control channel, and once the sender and receiver reserve a channel, they switch to that channel to transmit a data frame. In [7], Luo et al. proposed a single-radio protocol named CAM-MAC which allows neighboring terminals to share information about channel conflicts and deaf terminals. Wang

et al. proposed CMDMAC for multi-channel directional ad hoc networks where neighbors of both sender and receiver act as cooperators to avoid deafness and hidden terminal problems [12]. In CMDMAC, link establishment negotiations are performed on control channel omni-directionally, while data frames are transmitted and received directionally on data channels. Wang et al. [53] proposed a distributed DCC-based MMAC protocol for CRNs by exploiting the "dual -receive" feature of a single half-duplex transceiver. The proposed protocol maximizes throughput by allowing as many simultaneous transmissions as possible while addressing the transmitter deafness problem. Moreover, it performs channel and rate assignment as well as power control to minimize energy consumption. Cross-layer framework for supporting adaptive load control is also proposed based on the proposed MAC protocol.

There are also DCC-MMAC designs in which channel reservation/assignment is not necessary prior to data transmissions. Almotairi et al. proposed an MMAC protocol with hopping reservation (MMAC-HR) [10] which is capable of solving the multi-channel exposed terminal problem. In MMAC-HR, one transceiver is fixed to the control channel, while the other one hops between data channels according to its hopping sequence. Upon a successful negotiation between a sender-destination pair on the control channel, the sender switches to the destination's residing channel and follows its hopping sequence to perform data transmission. Unlike DCA, MMAC-HR employs contention resolving mechanism on both control and data channels to avoid collisions. Such protocol can be considered as a hybrid design of dedicated control channel and frequency hopping. In this dissertation, we will focus the channel reservation/assignment based DCC-MMAC protocols and simply refer them to DCC-MMACs.

In DCC-MMAC protocols, because one radio is always tuned to the DCC, multi-channel hidden terminals are avoided. However, the use of one extra radio increases the device cost. Moreover, the spectral efficiency is decreased due to the dedication of one channel for signaling. The capacity of this channel becomes the performance bottleneck in high-contention scenarios. In addition, from a security standpoint, the

control channel constitutes a single point of failure [54, 55]. An adversary launching a denial-of-service (DoS) attack on the control channel effectively denies communication on all channels. Compared to the SP-MMAC design, time synchronization is not required for DCC-MMAC. Moreover, channels do not remain idle during channel negotiation as in SP-MMACs.

2.2.3 Rendezvous Protocols

In rendezvous protocols, terminals hop between the available channels according to predefined sequences [6, 18, 47, 56, 57]. These sequences are designed to enable the sender-destination rendezvous within a fixed time period. In [47, 56], inactive hosts follow a common predefined sequence. Any pair of terminals wanting to communicate, agree on a private sequence and rejoin the common sequence after the transmission is completed. In parallel rendezvous designs [6, 57], terminals do not converge to a common sequence but follow a unique one called the home sequence. When two sequences overlap, which is guaranteed by design, the receiver shares its seed with the sender and the two synchronously hop until the transmission is terminated. Almotairi et al. proposed a parallel rendezvous based Dynamic Switching Protocol (DSP). Each terminal maintains two hopping sequences on its two transceivers respectively: fast hopping sequence and slow hopping sequence. The former one is dedicated for transmission, the latter one is periodically broadcasted and dedicated for reception. To communicate, sender deviates its fast hopping transceiver to follow the destination's slow hopping sequence, and transmits data frames using legacy IEEE 802.11 MAC strategies. Rendezvous protocols do not address the multi-channel hidden terminal problem.

While the above works only focus on the validity and efficiency of channel access protocols suitable for multi-channel networks, few efforts have been devoted to the security aspects of such protocols. Additionally, existing MMAC designs do not explicitly address the multi-channel exposed terminal problem where exposed terminals lose transmission opportunities when switching to a busy channel in the middle of a data transmission. Referring to the topology of Figure 2.4, assume that

B transmits a data frame to A on f_1 starting at t_0. Terminal C switches to f_1 at t_2 with $t_0 < t_2 < t_1$. C senses the channel busy and defers from transmission, thus losing an opportunity to transmit concurrently with B.

2.3 MAC-layer Protocols for Cognitive Radio Networks

2.3.1 Operations of Cognitive Radio Networks

Cognitive Radio (CR) is a promising enabling technology for realizing opportunistic spectrum access. In cognitive radio networks (CRNs), network users are classified to primary if they are licensed to operate on a particular band, and secondary if they can only access that band when it is free of primary user (PU) activity. CR devices are capable of sensing and coordinating access to the idle portion of the spectrum, while not interfering with PU activity. A general CR system model is depicted in Figure 2.6. The basic functions of a CR system include spectrum sensing, spectrum management, and spectrum access. In spectrum sensing, CRs use signal detection techniques such as energy detection, matched filtering, and cyclostationary feature detection to independently determine the set of idle channels. To combat errors due to shadowing and fading, cooperative sensing is employed. CRs share their sensing observations using explicit messaging over a control channel or by transmitting busy tones on pre-specified frequencies. The sensing observations are fused to reliably determine the idle portion of the spectrum. Information fusion is either centralized or decentralized and the decision rules are based on soft or hard decision combining.

The spectrum management function allocates the idle spectrum to competing CRs. Spectrum allocation can be centrally performed by the fusion center or be coordinated in a distributed manner. Finally, spectrum access is mediated at the Medium Access Control (MAC) layer, which is designed to dynamically allocate the set of idle channels among CRs. To this end, several CR MAC designs that manage access to idle channels have been proposed [58–62]. Typically, these designs integrate the functions of spectrum sensing, spectrum information sharing and spectrum access.

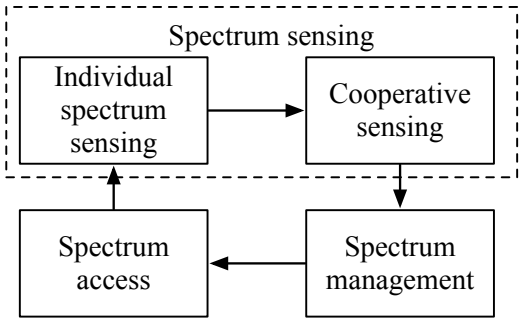


Figure 2.6: A general CR system model.

2.3.2 Cognitive Radio MAC Protocol Designs

The MAC-layer protocols for CRNs (referred to as CR-MAC protocols) can be categorized based on different criteria. For instance, Salameh et al. [63] classify distributed spectrum access mechanisms in multi-hop CRNs based on the employed radio technology (e.g. number of available transceivers and number of simultaneously supported operating channels) and the channel-occupancy model. In this dissertation, we use the same criteria as in Section 2.2 and classify CR-MAC protocols into three categories.

Split-phase CR-MAC Protocols

In split-phase CR-MAC protocols, time is divided to alternating control and data phases. CRs coordinate access to the idle channels during a control phase, before engaging to data transmissions [58, 59]. The control phase is further divided to a spectrum sensing, spectrum information sharing, and channel negotiation phase. During the spectrum sensing phase, CRs individually sense the set of idle channels. The sensing observations are shared during the spectrum information sharing phase by converging to a common control channel. The control phase is completed with the channel negotiations for the upcoming data phase. In the data phase, CRs switch to the agreed channels.

In MMAC-CR [58], sensing information is shared by transmitting busy tones. The spectrum information sharing phase is divided to a fixed number of slots 1,...,k,

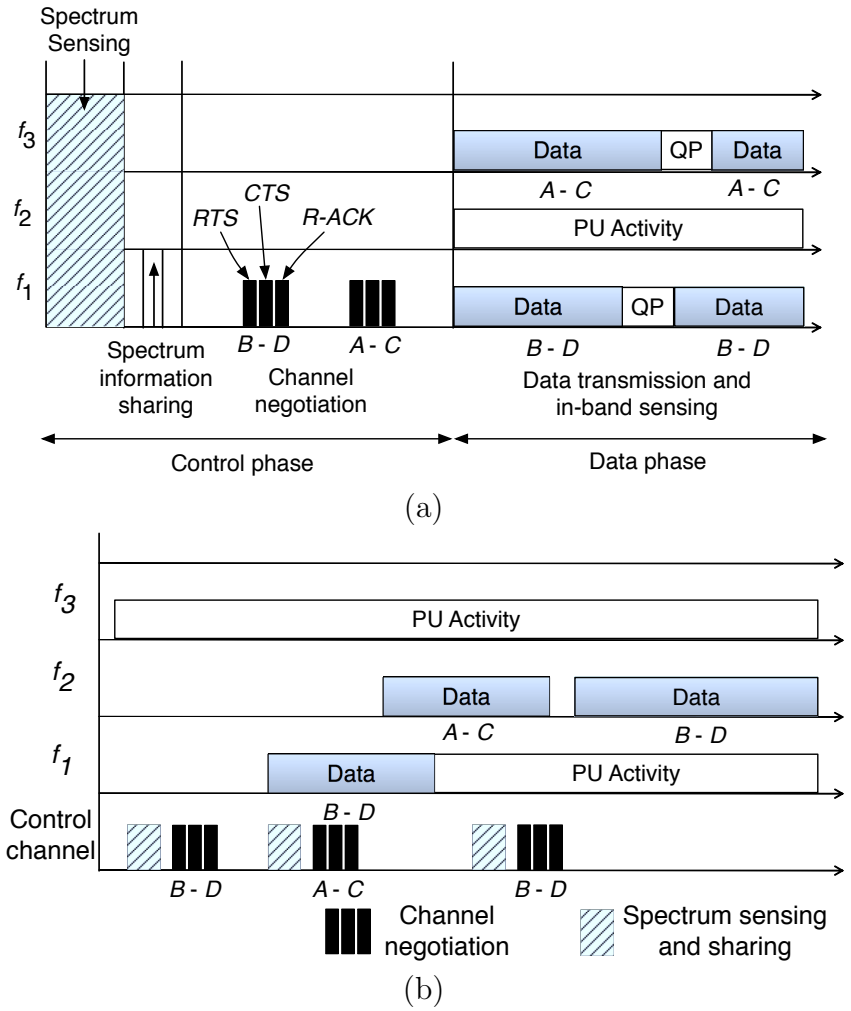


Figure 2.7: (a) A split-phase CR-MAC. During the control phase, CRs sense for idle channels and share their sensing observations by transmitting busy tones on dedicated time slots. CRs negotiate the spectrum allocation for the upcoming data phase. In the data phase, CRs switch to the negotiated channels. In-band sensing is performed to avoid interference with PUs, (b) a dedicated control channel CR-MAC. CRs perform spectrum sensing, information sharing, and channel negotiations on a dedicated band while engaging in data transmissions on other bands.

equal to the number of potentially available bands f_1,...,f_k. If any of the CRs transmits a busy tone at slot i, channel f_i is assumed to be occupied by a PU. CRs then negotiate the channel assignment among PU activity free channels, in a way similar to the MMAC protocol [8]. Figure 2.7(a), depicts all stages of MMAC-CR for four CRs A-D and three channels f_1-f_3. During the spectrum sensing phase, CRs A-D determine that f_2 is occupied by a PU. In the spectrum information sharing phase, CRs transmit a busy tone in slot 2 to indicate that f_2 is occupied. In the channel negotiation phase, CR B performs a channel negotiation with destination D and selects f_1 for the upcoming data phase. Similarly, pair A-C selects f_3. During the data phase, pairs B-D and A-C switch to their selected channels to exchange data. Because a PU may appear at any channel during the data phase, the latter incorporates a periodic quiet period (QP) during which CRs perform in-band sensing. If a PU is detected, CRs abandon the current channel by switching to a back-up one.

Dedicated Control Channel CR-MAC Protocols

Several CR-MAC protocols follow a DCC design [51, 52, 60, 61, 64]. In DCC CR MACs one transceiver is continuously tuned to an the control channel [60, 61]. As illustrated in Figure 2.7(b), spectrum sensing, sharing, and channel negotiations are performed over the DCC, while data transmissions take place over the data channels. These functions are performed in a manner similar to a split-phase design with the exception of executing the control and data phases in parallel rather than sequentially.

In [60], the channel negotiation phase culminates to a single CR gaining access on the entire idle portion of the spectrum. Idle channels are merged using bonding/aggregating technology. The authors in [61] proposed the DOSS CR-MAC protocol for managing access to the idle portion of the spectrum. In DOSS, data channels occupied by CRs are indicated by continuously transmitting a busy tone on a corresponding narrowband channel. Any CR detecting a busy tone on a given busy tone channel will defer from transmission on the corresponding data channel.

In [64], an asynchronous DCC-based CR-MAC protocol named COMAC is proposed. COMAC enables CRs to gain dynamic access to the idle portion of the spectrum, while limiting interference to PRs. The latter is achieved based on stochastic PR-to-PR and PR-to-CR interference models rather than predetermined power masks. COMAC employs a contention-based handshaking mechanism on the DCC to allow control information exchange. To allow more concurrent CR transmission opportunities, the minimum number of channels satisfying the rate demand are assigned to a CR transmission, while CR-to-PR interference is still statistically bounded. COMAC providing soft guarantees on the performance of PUs under different CR traffic loads.

Krunz et al. proposed a distributed CR MAC protocol, where a single transceiver with "dual receive" capability is considered [51]. To alleviate the multi-channel hidden-terminal problem, CRs that do not transmit tune one of the receivers to a DCC. The other receiver could be utilized to receiving data. A cross-layer framework is further proposed to allow for adaptive load control at individual nodes based on the local conditions of the control channel and data channels. The proposed scheme significantly improves the system throughput by alleviating the multi-channel hidden terminal problem and transmitter deafness, and allowing joint channel and transmission rate selection, as well as power control.

Salameh et al. investigated the channel access problem in both single-hop and multi-hop CRNs [52]. For single-hop CRNs, a distributed MAC protocol named AW-MAC is proposed, which incorporates a centralized channel assignment algorithm based on bipartite matching. The channel assignment algorithm aims at maximizing the total number of simultaneous CR transmissions via power management. The authors further extended the channel assignment algorithm and proposed an MMAC protocol named WFC-MAC to coordinate user access to multiple channels in multi-hop CRNs. In WFC-MAC, CRs negotiate for channel usage on the DCC based on an individually maintained Free Channel List (FCL) and the extended channel assignment algorithm. Similar to [51], both AW-MAC and WFC-MAC rely on the "dual receive" capability of the single transceiver at each CR.

Frequency Hopping CR-MAC Protocols

In frequency hopping (FH) CR-MAC protocols, CRs hop between the available channels according to predefined FH sequences (e.g. [62]). These sequences are unique for every CR, but guaranteed to have a minimum degree of overlap (known as rendezvous). Once two CRs rendezvous on a given channel, they can exchange data or agree to synchronously hop for the duration of the data transmission. CRs skip channels that are occupied by PUs to prevent interference with PU transmissions. FH CR-MAC protocols differ from their split-phase and dedicated control channel counterparts in that channel negotiations are not performed in a distributed manner, but rather follow a deterministic design.

2.4 Jamming Attacks in Wireless Networks

The open nature of wireless medium leaves it vulnerable to intentional interference attacks, typically referred to as jamming. In the simplest form of jamming, the adversary interferes with the signal reception by transmitting a continuous jamming waveform [24], or several short jamming pulses [25]. Typically, jamming attacks have been analyzed and addressed as a physical layer vulnerability. There are many different jamming strategies that an adversary can follow to interfere with wireless communications [65–71].

Constant jamming and random jamming: Based on the interval between the state of jamming and not jamming, jamming can be classified into constant jamming and random jamming. Constant jammer continually emits a radio signal without following any specific protocol. Specifically, the constant jammer does not wait for the channel to become idle before transmitting [69]. Instead of continuously sending out radio signal, random jamming alternates between sleeping and jamming.

Reactive jamming: Rather than being continuously active, a jammer with reactive jamming strategy stays quiet when the channel is idle, but starts transmitting a radio signal as soon as it senses activity on the channel. Reactive jamming has been shown to be not only the most difficult to detect, but also the most energy-efficient

approach [72].

Selective jamming: In selective jamming, the adversary targets specific channels, packets, nodes, etc., based on their importance in the overall network operation. As an example, the jammer can target specific packets of high importance. The important packets can be identified by overhearing the headers of the packets before the transmission is complete, or by anticipating the transmission of specific packets, based on protocol semantics.

Countermeasures: Jamming attacks on wireless communications are primarily countered at the physical layer by employing spread spectrum (SS) techniques such as DSSS and FHSS [73, 74]. Using a secret PN sequence, the transmitted signal is spread over a large bandwidth to mitigate the impact of narrowband jammers. SS techniques are effective for systems with sufficiently large bandwidth. Moreover, they are hard to integrate with broadcast communications due to the one-transmitter multiple receivers synchronization problem and the common PN code secrecy problem [75–77]. In the latter, the compromise of a single receiver reveals the commonly-shared PN code, thus rendering any protection offered by SS ineffective.

The jamming impact on MMAC protocols can be addressed by mitigating control-channel jamming. In [26], Chan et al. proposed the replication of control information over multiple control channels according to a binary encoding based key (BBK) assignment. In [27, 78], the BBK scheme was extended to a probabilistic combinatorial design that provides a graceful degradation in performance as a function of the number of inside jammers. Alternative methods eliminate the dependency on shared secrets [28, 76, 77, 79, 80]. Pöpper et al. proposed a DSSS-based method called Uncoordinated DSSS (UDSSS) [77]. In UDSSS, broadcast transmissions are spread according to a PN code, randomly selected from a public set of codes. Receivers must exhaustively apply all codes in the codebook to recover the broadcast message. Liu et. al. showed that UDSSS is vulnerable to a reactive jammer with sufficient computational power to recover the PN code before the end of an ongoing transmission [79]. They proposed RD-DSSS, a randomized differential

DSSS scheme, that expands the public code set and discloses the selected code after the message transmission has ended, thus providing resilience to reactive jammers. The computational efficiency of RD-DSSS was further improved in [81] by encoding the seed of the PN code used to spread a message, at the end of that message. This delayed seed disclosure prevented a jammer from acquiring the PN code before the message was fully received. Note that most of the works on control-channel jamming do not examine the integration with jamming-resistant data communications.

The use of jamming to impact operations in higher layers beyond PHY layer has been studied in [55, 68, 82–87]. Chang. et al. describe the vulnerabilities of existing MAC protocols that rely on centralized control for coordinating channel access. They proposed SimpleMAC, an SS-based protocol that mitigates the impact of intelligent jamming launched from inside attackers by restricting access to control information and hiding the transmitter's activity strategy. Xu et al. described several jamming models for single-channel MAC protocols including a constant, deceptive, random, and reactive jammer. They studied the feasibility of detecting the jamming presence and proposed slow frequency hopping and spatial retreats to evade areas affected by jamming [84]. Cross-layer jamming attacks were addressed by Liu et al. in [85]. The authors proposed an adaptive protocol stack called SPREAD that countered intelligent jammers targeting multiple layers. SPREAD avoids single point of failure by switching between a set of protocols across multiple layers, reducing the adversary's knowledge about the protocol specifications.

Lin et al. investigated jamming attacks on WLAN protocols and showed that data frames can be jammed with relatively low jamming effort [83]. They analyzed the jamming resilience as a function of the ECC capability, interleaving function and SNR. They proposed a combination of cryptographic interleaving and coding to shield data transmissions from jamming. Law et al. identified and evaluated several inside jamming attacks in popular single-channel MAC protocols [82]. Li et al. proposed a game theoretic approach to optimal jamming and anti-jamming strategies at the MAC layer [68]. To best of our knowledge, the integration of anti-jamming techniques with MAC layer protocols has not been thoroughly investigated.

CHAPTER 3

SELFISH MISBEHAVIOR IN CONTENTION-BASED MULTI-CHANNEL MAC PROTOCOLS

3.1 Introduction

3.1.1 Motivation

For multi-user wireless systems with dynamic traffic demands, contention-based channel access protocols have been shown to be superior than schedule-based protocols. Contention-based MAC protocols such as those employed in 802.xx family of protocols, mediate access to the wireless medium 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 [8].

The performance of distributed MMAC protocols has been characterized analytically and via simulations, assuming that terminals remain protocol-compliant [6–9,13,16–18,23]. However, selfish terminals can manipulate protocol parameters to gain access to a disproportional medium share compared to well-behaved terminals. Unfair sharing of the wireless medium is possible even if the misbehaving terminals do not violate the protocol specifications. The problem of selfish misbehavior has been extensively studied in the context of single-channel MAC protocols [31–35]. The majority of prior art has focused on the manipulation of the backoff mechanism of the DCF function, commonly employed to randomize the start of a frame transmission and avoid collisions [88]. In this misbehavior type, selfish terminals deliberately select small backoff values to reserve the wireless medium more often than well-behaved terminals. Common mitigation methods [31,89–98] employ back-

off monitoring mechanisms to detect and penalize the misbehaving terminals.

However, solutions for single-channel MAC protocols cannot be directly ported to the multi-channel domain. In MMAC protocols, misbehavior monitoring is complicated due to the concurrent use of multiple channels by multiple terminals. These channels cannot be continuously and simultaneously monitored. Moreover, MMAC protocols are designed to allow the uneven distribution of resources to accommodate varying traffic demands. This can be exploited by misbehaving terminals to acquire a larger portion of the available resources under high load conditions. To address these shortcomings, we analyze possible misbehavior strategies for MMAC protocols and develop detection and mitigation methods.

3.1.2 Main Contributions and Chapter Organization

The contributions of this chapter are as follows:

- We detail possible misbehavior strategies for two popular classes of contention-based MMAC protocols: the SP-MMACs [8, 16, 23] and the DCC-MMACs [9, 10, 12, 46]. For each class, we demonstrate two attack types: (a) backoff manipulation attacks and (b) multi-reservation attacks. The first type relates to backoff manipulation strategies, adapted to the multi-channel domain. In the second attack type, selfish terminals place multiple reservations for one or more channels in order to gain exclusive access to those channels and reduce contention. We further consider the combination of the two attack types.

- For the SP-MMAC protocol family, we adaptively optimize the multi-reservation attack to minimize the number of control messages that need to be transmitted in order to isolate a desired subset of channels. The minimization of the control messages reduces the exposure of selfish terminals to misbehavior detection.

- To eliminate the throughput gain of selfish behavior, we develop misbehavior detection and mitigation methods that provide fair access opportunities to all

contending terminals and rapidly identify selfish ones. Specifically, we design a backoff value generation and monitoring module which binds each terminal with a publicly available backoff schedule. Deviation from the publicized schedule leads to the detection of misbehavior. We further modify the channel assignment and negotiation rules to prevent exclusive channel reservations for prolonged time periods. Our extensive simulations verify that our methods effectively mitigate the misbehavior gains and detect misbehaving terminals.

- We identify misbehavior strategies for cognitive radio MMAC protocols. We categorize the possible misbehavior actions to three classes: (a) attacks on spectrum sensing, (b) attacks on the channel negotiation process, and (c) DoS attacks. For each class, we present possible countermeasures.

Chapter Organization: The remainder of this chapter is organized as follows. In Section 3.2, we formalize the network and adversary models. Section 3.3 details MMAC misbehaving strategies. We develop methods for mitigating these strategies in Section 3.4. In Section 3.5, we identify vulnerabilities of CR-MAC protocols and present possible countermeasures. In Section 3.6, we evaluate the impact of misbehavior on the network performance and demonstrate the effectiveness of our mitigation mechanisms. Section 3.7 concludes the chapter.

3.2 Model Assumptions

3.2.1 System Model

We consider a wireless network that operates over a set of orthogonal frequency bands (channels), denoted by $F = \{f_1, f_2, \ldots, f_n\}$. Terminals coordinate access to F using an SP-MMAC or DCC-MMAC protocol. Terminals are equipped with one or more half-duplex radio transceivers with negligible channel switching delay. As mandated by the SP-MMAC and DCC-MMAC specifications, the wireless network is time-synchronized to a common slotted system. Terminals who monitor the behavior of their neighbors and detect misbehaving ones can submit recommendations to a

reputation system [99]. Those terminals with low reputation are eventually removed from the network via a credential revocation process. The specifics of the reputation system operation are beyond the scope of the present work. Many such systems have been proposed and extensively studied in the literature (e.g., [99–103]).

3.2.2 Misbehavior Model

We consider selfish terminals that aim at gaining an unfair share of the available spectrum by violating the MMAC specifications. Unfairness is measured in terms of the throughput gains achieved by the misbehaving terminals compared with the throughput of protocol-compliant terminals. Misbehaving terminals are assumed to be independently acting (no collusion). Each terminal has access to his own cryptographic credentials and cannot compromise the credentials of other terminals. Therefore, a terminal cannot launch impersonation attacks such as Sybil attacks, where he assumes identities of other terminals [104].

3.3 Misbehavior Strategies

In this section, we identify possible misbehavior strategies for MMAC protocols. We focus our attention on contention-based protocols such as the DCC-MMAC and SP-MMAC, which are more efficient under bursty traffic conditions from a large number of terminals.

3.3.1 Backoff Manipulation Attack (BMA)

Both DCC-MMAC and SP-MMAC protocols are vulnerable to backoff manipulation attacks (BMAs). In such attacks, selfish terminals systematically select small backoff values to improve their chances of capturing the medium under contention. When multiple channels are available, a BMA can also be used to isolate a channel for exclusive use by the misbehaving terminal.

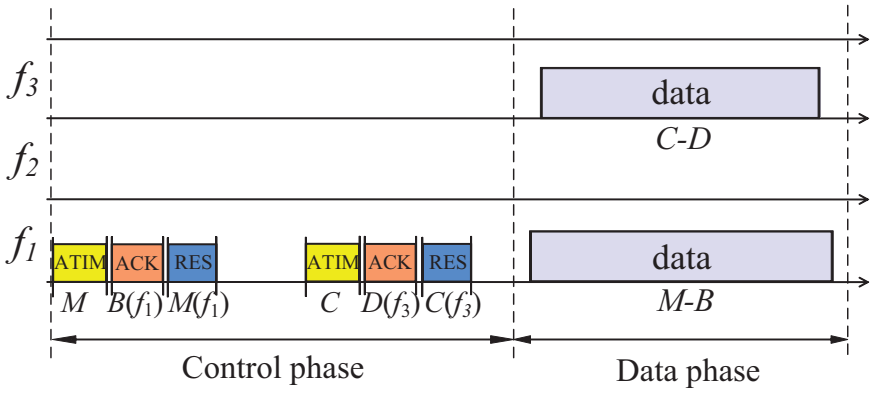
BMA on SP-MMAC

In SP-MMAC, terminals converge to a default control channel during the control phase to negotiate data transmissions for the upcoming data phase. Terminals unable to complete a channel negotiation during the control phase must defer from transmission in the upcoming data phase. A misbehaving terminal M can systematically select small backoff values to increase its chances of completing a channel negotiation, within the allotted control phase period. Following the same misbehavior strategy, M can capture a data channel more often during the data phase, if more than one terminals contend on that channel. The BMA is particularly effective when the control channel is saturated.

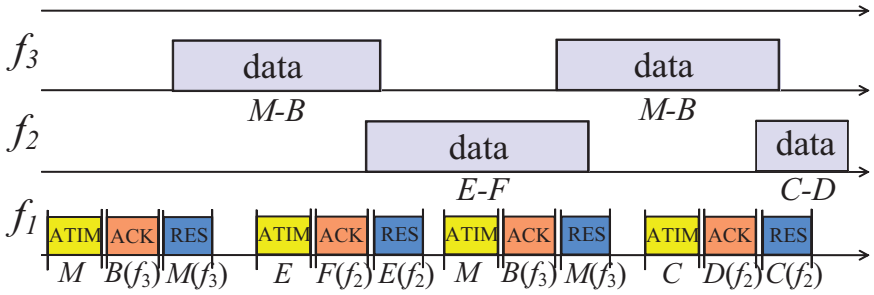
In Figure 3.1(a), we demonstrate the impact of a BMA on SP-MMAC. Similar to the scenario of Figure 2.3, terminals M, C, and E have data frames for terminals B, D, and F, respectively. Terminal M immediately transmits an ATIM frame with the beginning of the control phase and completes a negotiation for channel f_1. Due to the short control phase duration, terminal E is unable to complete a negotiation with F and hence, E defers its transmission for the following data phase. Moreover, channel f_2 remains idle. In the next control phase, terminal M repeats its strategy by capturing the control channel immediately after the initiation of the control phase.

BMA on DCC-MMAC

In DCC-MMAC protocols, data transmissions take place only after a channel negotiation is successfully completed over the DCC. As access to the DCC is contention-based, misbehaving terminals selecting small backoff values capture the control channel more frequently than well-behaved terminals, thus performing more frequent data transmissions. Figure 3.1(b) shows the BMA on DCC-MMAC. One of the three available channels, f_1, is dedicated to control traffic. Six terminals within the same collision domain contend over f_1 for placing reservations to f_2 and f_3. Terminals M, C and E are assumed to have frames for B, D and F, respectively.



(a) BMA on SP-MMAC.



(b) BMA on DCC-MMAC.

Figure 3.1: Backoff manipulation attack on the SP-MMAC and DCC-MMAC.

Misbehaving terminal M continuously selects a zero backoff value before initiating a channel negotiation with B. After the first round channel negotiation between M and B is completed, other terminals update the expected release time of channel f_3 accordingly, and thus its priority in their PCLs is lowered. When E initiates its channel negotiation with F later, channel f_2 is preferred as it has higher priority than f_3. Assuming that M launches the BMA again during its second round negotiation, C and D also end up with selecting f_2 as it has higher priority than f_3 according to the expected release time depicted in the figure. By constantly launching the BMA on the DCC, misbehaving terminal M successfully isolate f_3 for its exclusive use while $C - D$ and $E - F$ have to alternate over f_2.

3.3.2 Multi-reservation Attack (MRA)

DCC-MMAC and SP-MMAC protocols reduce the number of collisions by requiring reservations on the control channel before terminals engage in data transmissions. Terminals overhearing the reservations of other co-located terminals are able to infer which channels will become occupied. This information is used to adjust the channel priority and avoid highly contended channels. When placing its own reservation, a terminal prefers the channel with the least number of reservations.

This adaptive channel priority mechanism can be manipulated by selfish terminals to falsely reduce the contention on target channels by placing multiple reservations for those channels. We call this type of misbehavior as a *multi-reservation attack* (MRA). We emphasize that the placement of multiple reservations by a single terminal is protocol-compliant for most MMAC designs to improve the spectral efficiency. In SP-MMAC, the same terminal can perform multiple transmissions over a single data phase. In DCC-MMAC, a terminal can transmit several back-to-back frames over the same channel, if no other terminal is competing for it.

MRA on SP-MMAC

In SP-MMACs, an MRA is possible via the manipulation of the priority channel list (PCL) maintained by each terminal [8]. When a terminal overhears a reservation for a channel f_i, it lowers the priority of f_i. By placing multiple reservations on a targeted subset of channels, the misbehaving terminal can lower the priorities of those channels so that other terminals defer from them. As a result, the misbehaving terminal does not have to contend during the data phase. We note that a selfish terminal may be interested on the exclusive use of multiple channels to communicate with more than one destination on a single data phase or perform channel bonding/aggregation. The MRA requires the cooperation of multiple terminals that would engage in a reservation process with M. For an independently misbehaving terminal, other terminals may be unwilling to collude with M. However, the misbehaving terminal can still place multiple reservations on one or more channels in several ways.

Reservations with fake terminals: One strategy for M is to broadcast reservation frames to fake terminals. The use of such fake terminals is possible due to the hidden terminal problem. Nearby terminals cannot verify the existence of the fake terminals, as they could be hidden terminals.

An example of the MRA on SP-MMAC is shown in Figure 3.2. Six terminals contend for access to f_1 and f_2. Misbehaving terminal M wants to reserve f_2 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 terminals. After M's first reservation, the priority of f_2 is lowered to LOW(1) in the PCLs of B, C, E, and F. To ensure that no other terminal prefers f_2 in the upcoming data phase, terminal M broadcasts three ATIM-ACK frames as a response to fictitious reservation requests (ATIM frames) originating from fake terminals I_1, I_2, and I_3. All fake ATIM-ACK frames sent by M indicate f_2 as the preferred channel. As a result, terminals B, C, E, and F, lower the priority of f_2 to LOW(4) (the number within the parenthesis indicates the priority level). Note that because none of the terminals within M's range overheard

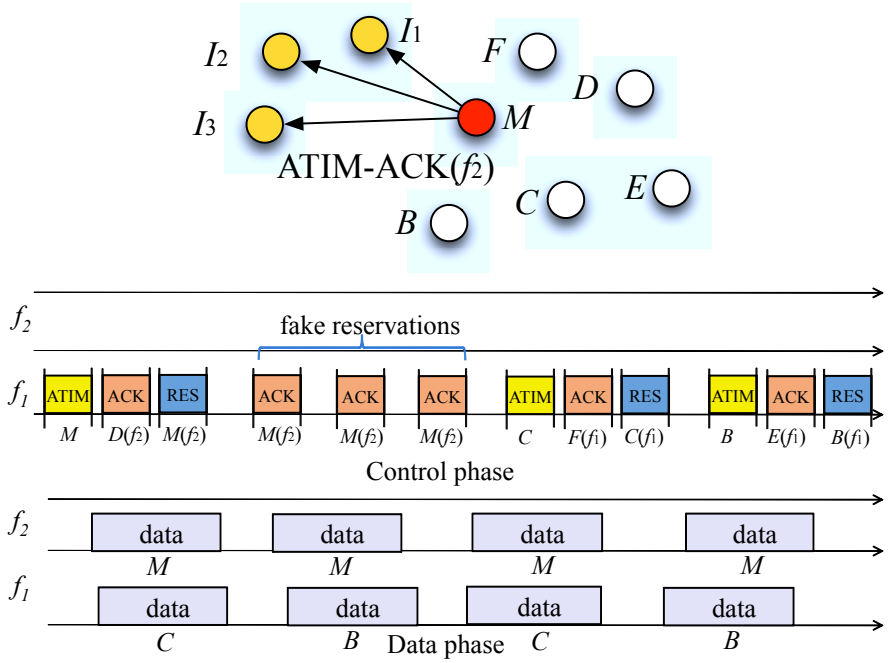


Figure 3.2: In SP-MMAC, M makes four reservations on f_2, by completing one negotiation with D and sending three fake ATIM-ACK frames to terminals I_1, I_2, and I_3. These terminals are presumed to be hidden terminals to terminals B-F.

the frames sent by I_1, I_2, and I_3, they assume that $I_1 - I_3$ are hidden terminals. The PCL update process at terminal B is shown in Figure 3.3. B finally sets the priority of f_1 in its PCL table to HIGH due to f_1 being selected by $B - E$ for data transmission in upcoming data phase. Other terminals update their PCL tables in a way similar to B.

Due to the lowered priority of f_2, communicating pairs B-E and C-F prefer channel f_1 during the upcoming data phase. Thus, M monopolizes the use of f_2, while pairs B-E and C-F have to contend on f_1. Equivalently, M can transmit sequences of ATIM/ATIM-RES frames to fake destinations. A requirement for a successful MRA is that M completes the fake reservations before competing terminals are able to place their own reservations. This can be achieved by combining the MRA with a BMA.

Incomplete negotiations: A misbehaving terminal can launch an MRA by

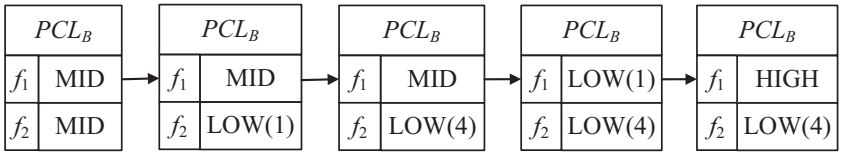


Figure 3.3: Evolving PCL table at terminal B.

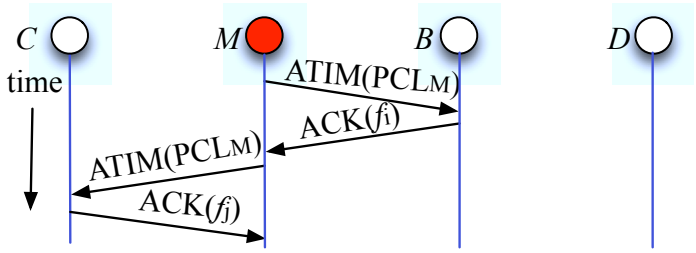


Figure 3.4: In SP-MMAC, M performs incomplete channel negotiations with B and C.

engaging in a series of incomplete channel negotiations with its one-hop neighbors. According to the SP-MMAC protocol, the sender must respond to an ATIM-ACK frame with an ATIM-RES frame that verifies the receiver's channel selection. If the sender does not reply with an ATIM-RES frame, the negotiation is not completed. However, terminals in the communication range of the receiver consider the channel contained in the ATIM-ACK frame as reserved. This is because they could be hidden terminals to the sender and therefore unable to overhear the ATIM-RES frames. Multiple incomplete negotiations lead to the distortion of the PCL at all terminals 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 reservations that are only known to one of the two terminals.

An example of an incomplete negotiation MRA is shown in Figure 3.4. Terminal M initiates negotiations with B and C, but does not respond to the ATIM-ACKs transmitted by each terminal. As a result, the priorities of channels f_i and f_j are lowered at the PCL of any terminal that overhears the ATIM-ACKs from B and C.

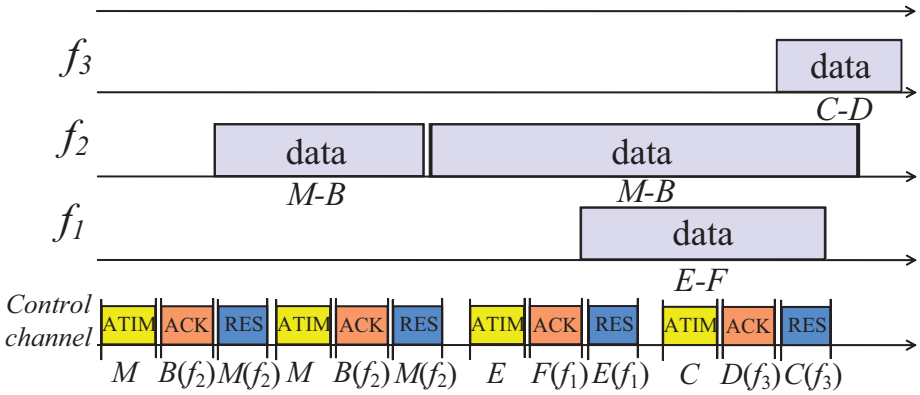
MRA on DCC-MMAC

In DCC-MMAC protocols, the number of reservations placed for a particular channel f_i affects the priority of f_i through the expected release time (time that f_i is expected to become idle). Channels with late release times are less preferable during channel negotiations. A misbehaving terminal M can launch an MRA to inflate the expected release time of a target channel f_i. The multiple reservations can be placed while M is active on f_i to ensure the perpetual use of f_i from the misbehaving terminal for any future data transmission. To be more efficient, M can combine the MRA with the BMA so that it is more likely to place reservations for the target channel before other contending pairs. We note that some DCC-MMAC protocols allow reservations to be placed for channels while they are still occupied to reduce the delay until the channel becomes re-occupied [10].

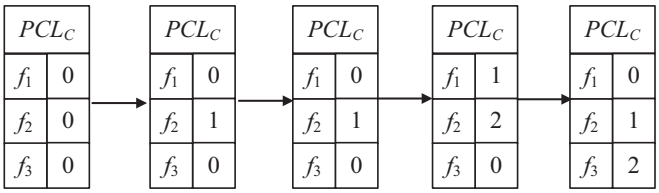
To demonstrate an MRA on DCC-MMAC consider the example shown in Figure 3.5 (a). Initially, M completes a negotiation with B over the DCC for channel f_2. While transmitting a data frame to B on f_2, M continues to place reservations for f_2 over the DCC. The misbehaving terminal prefers f_2 even if other channels have earlier release times. The continuous placement of reservations for f_2 significantly inflates the release time of f_2. Contending terminals C, D, E and F refrain from using f_2, thus leaving it for exclusive use to the misbehaving terminal. The PCL table at C upon every successful channel negotiation overheard by C is shown in Figure 3.5 (b). Other terminals update their PCL tables in a similar way to C.

3.3.3 Optimal Misbehavior Strategies under SP-MMAC

In this section, we analytically derive the optimal MRA strategy for guaranteeing exclusive access of the misbehaving terminal M to a desired subset of channels in the SP-MMACs. 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,



(a) Operating example of DCC-MMAC under the MRA.



(b) Evolving PCL tables at terminal C.

Figure 3.5: Illustration of the MRA on DCC-MMAC.

ATIM-ACK and ATIM-RES frames) 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 terminal M can exclusively use 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 terminal can place a reservation. Parameter l defines the number of reservations to be placed by the contending well-behaved terminals 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 terminals 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 terminals 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 terminals 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. □

Proposition 1 suggests that the best strategy for M is to place d consecutive reservations before any contending pair has 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 SP-MMAC control phase contention process after a rooted tree T with a vertex set $\mathcal{U}$. For a vertex $u_{i,j} \in \mathcal{U}$ 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 Figure 3.6. The senders of the two well-behaved pairs are denoted by A and B. For each vertex, the corresponding set of terminals 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 Figure 3.6, the pmf of S is given by the following proposition.

Proposition 2. *The pmf of S is given by,*

$$\Pr[S = d + \ell] = \sum_{u_{\ell+1,j} \in \mathcal{L}_{\ell+1}} Pr[R(u_{\ell+1,j})],\ \ell \geq 0, \tag{3.1}$$

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,

$$\Pr[R(u_{\ell+1,j})] = \prod_{u_{w,k} \in R(u_{\ell+1,j})} \Pr[u_{w,k}],\ 1 \leq w \leq \ell + 1, \tag{3.2}$$

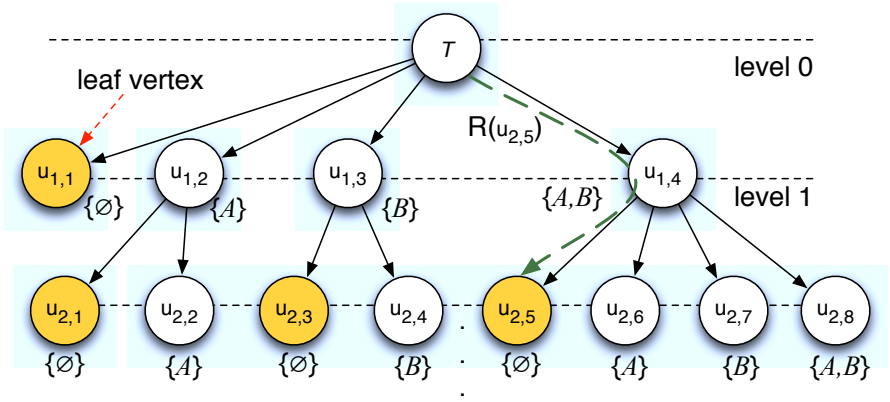


Figure 3.6: Representation of the SP-MMAC control phase as a rooted tree.

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

$$\Pr[u_{w,k}] = \left(\frac{1}{\min\{cw_{w-1}, cw_{\max}\}}\right)^{n_u} \cdot \left(1 - \frac{1}{\min\{cw_{w-1}, cw_{max}\}}\right)^{n_p - n_u}, \quad (3.3)$$

where n_u *and* n_p *are the number of terminals 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 terminals 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 terminals 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. (3.1) 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. (3.2).

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 terminals in $u_{w,k}$, and n_p the number of transmitting terminals in the parent terminal $p(u_{w,k})$. The CW of a transmitting terminal at level w is equal to $\min\{cw_{w-1}, cw_{max}\}$ since that terminal must have collided $(w-1)$ times with M in order to be transmitting at reservation round w. Moreover, the probability of a terminal 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. (3.3) follows by noting that it expresses the probability of exactly n_u terminals being in transmit state at stage w when n_p terminals 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 terminals choose their backoff values independently of each other, independently of previous rounds, and in a random fashion. Therefore, a terminal 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. (3.3), corresponds to one unique combination of transmit and backoff states for the participating senders, and *not any combination* that yields n_u transmitting terminals. 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 terminals 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 terminal 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 terminal 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 terminal 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 terminals. 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.

Adaptive strategy performance of multiple misbehaving pairs

When more than one terminals misbehave, the possibility of colliding during the backoff operation increases due to the consistent selection of small CW sizes. These collisions may reduce the number of reservations that can be placed during the control phase. Therefore, we expect that the overall network throughput will decrease with the number of misbehaving terminals. In the presence of other misbehaving terminals, a misbehaving terminal M will still aim in isolating one of the available channels by following the adaptive reservation strategy. The misbehavior advantage of M will not be affected as long as it is successful in isolating a single channel by performing the required number of reservations. The misbehaving terminals will target the isolation of different channels to avoid contention between them.

On the other hand, well-behaved terminals will get the opportunities to place reservations only after misbehaving terminals complete their reservations. As a

result, less time will be left for well-behaved pairs to place reservations due to the presence of multiple misbehaving pairs. Figure 3.7 illustrates a scenario with two misbehaving terminals M_1 and M_2. Initially, M_1 and M_2 select the same backoff value for placing a reservation and therefore collide. Following the collision, M_2 wins the channel and successfully places its first reservation for target channel f_2. M_1 refrains from transmitting until M_2's reservation is completed. According to the adaptive strategy, M_2 then allows two reservations from other pairs before its next reservation. After M_2's reservation, misbehaving terminal M_1 successfully places a reservation for f_1 and allows two additional reservations from other pairs. After both M_1 and M_2 complete their reservations, well-behaved pairs get the opportunity to negotiate and f_3 is preferred based on their PCLs. By repeating the above steps, M_1 and M_2 successfully isolate f_1 and f_2 respectively, while well-behaved terminals share f_3 during the data phase. Note that due to limited duration of the control phase, fewer well-behaved pairs are able to reserve a channel and transmit during the upcoming data phase. Therefore, the throughput of well-behaved terminals is expected to be reduced further.

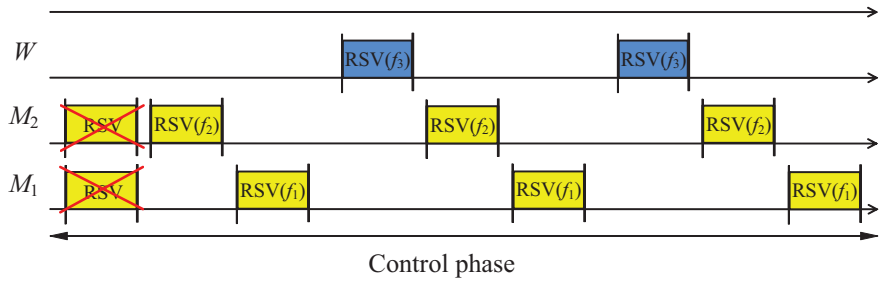


Figure 3.7: The control phase operations when two misbehaving terminals adopt the adaptive misbehavior strategy.

3.4 Mitigating MMAC Misbehavior

3.4.1 Mitigation of the Backoff Manipulation Attack

We first consider the manipulation of the backoff mechanism in MMACs. For simplicity, we focus our discussion on the mitigation of the BMA in SP-MMAC. The

process is also applicable to the BMA in DCC-MMAC as the two attacks are similar in nature. As explained in Section 2.1, the BMA detection mechanisms proposed for single-channel networks (e.g., [31,32]) are not adequate for multi-channel networks, because they are not designed to monitor parallel transmissions over 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. Our scheme consists of two modules: the backoff generation module and the backoff monitoring module.

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 terminal X uses a public pseudo-random number generator and a seed s_X (e.g., the unique terminal id) to generate a sequence of numbers uniformly distributed $[0, 1]$. Suppose these numbers are denoted by $b_X(1), b_X(2), \ldots$ The seed s_X is published to all one-hop neighbors of X, denoted by $\mathcal{N}_X^1$. Each number is used to calculate the backoff of X during the i^{th} transmission cycle. A transmission cycle consists of a series of frames transmitted by X after the expiration of its backoff counter. These include sequences of ATIM/RES/ACK frames over the control channel or RTS/CTS/DATA/ACK frames over a data channel. Finally, let r denote the number of times that the last frame has been retransmitted. This number is set to zero after every successful frame transmission, when the retransmission limit is reached, or at the beginning of each phase. Parameter r is incremented by one every time terminal X retransmits a frame due to a timeout. The backoff period for the i^{th} transmission cycle of X is given by:

$$T_b(i, r) = \lceil b_X(i) \cdot \min\{2^r cw_0, cw_{\max}\} \rceil - 1. \tag{3.4}$$

In (3.4), the range of $T_b(i, r)$ is in $[0, \min\{2^r cw_0, cw_{\max}\})$, as mandated by the backoff mechanisms of MMAC. One-hop neighbors of terminal X that are aware of (i, r)

and s_x can individually compute the backoff period $T_b(i, r)$ for the i^{th} transmission cycle.

Backoff Monitoring Module

The backoff monitoring module identifies misbehaving terminals that use smaller backoff values compared with their public schedule. To achieve this, terminals monitoring a misbehaving terminal M keep track of the transmission cycles and the number of attempted retransmissions that allow the computation of $T_b(i, r)$. The two parameters are included with every ATIM, ATIM-RES and RTS frame transmitted by M. We focus the analysis of the backoff monitoring module on scenarios where terminals are backlogged. This is because in non-backlogged scenarios, the BMA does not yield significant throughput gains to the misbehaving terminal due to lower contention.

Suppose terminal A is a one-hop neighbor of M and monitors M's backoff behavior. The backoff process at M for two transmission cycles $P_M(i-1)$ and $P_M(i)$ is illustrated in Fig. 3.8. Let $t_s(i)$ denote the start of the i^{th} transmission cycle and $t_e(i)$ denote its end. When M is backlogged, it initializes its backoff counter for $P_M(i)$ at $t_e(i-1)$, immediately after completing the $(i-1)^{st}$ transmission cycle. The period between $P_M(i-1)$ and $P_M(i)$ consists of periods $T_b(i, r)$ and T_{fr}. The latter denotes the time for which the backoff counter of M remains frozen due to the occurrence of other transmissions within M's collision domain. For the monitor A, terminal M misbehaves if

$$t_s(i) - t_e(i-1) < T_b(i, r) + T_{fr} + \delta. \tag{3.5}$$

Parameter δ denotes an error margin in the computation of M's backoff which accounts for timing errors due to the propagation delay and node mis-synchronization. As such errors are relatively small, δ is not expected to be longer than one slot. We now describe how monitoring terminal A can compute all parameters involved in (3.5).

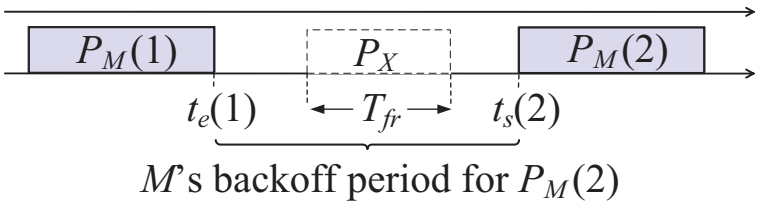


Figure 3.8: Backoff process at the monitored terminal M.

Computation of $t_e(i-1)$: Monitor A can compute $t_e(i-1)$ by keeping track of M's transmission cycles. As A is a neighbor of M, it does not transmit or receive while M is active. Therefore, A can know the exact time slots on which M's transmission cycles are terminated. These can be estimated by either overhearing M's transmissions or through the NAV value included in control packets preceding a data transmission. Note that the backoff counter of M is also restarted with the initiation of every control phase and data phase (for SP-MMAC protocols). Monitoring terminals set $t_e(i-1)$ to the first slot of each phase to monitor M's first transmission cycle within that phase.

Computation of $t_s(i)$: The starting time for M's i^{th} transmission is computed in a straightforward manner by overhearing the transmissions of M.

Computation of T_{fr}: According to the backoff mechanism specifications, terminals must freeze their backoff counters when another transmission occurs within their collision domain. The ability to correctly estimate the backoff counter freezing period depends on the topology of the involved terminals (monitors, monitored terminals, transmitter/receiver pair). The topology determines which terminals can sense (either physically or virtually) the ongoing transmission. Let a transmission from S to D occur within the vicinity of A and M. We classify the possible topology configurations of S, D, A, and M to the following three cases.

Case 1: The $S-D$ transmission can be sensed by both M and A. This scenario is depicted in Fig. 3.9(a). Terminal M must freeze its backoff counter for a time equal to the duration of the $S-D$ transmission cycle. Because A is also able to sense the $S-D$ transmission, it can determine the backoff counter freezing period T_{fr}.

Case 2: The $S-D$ transmission can be sensed by A, but not M. This scenario is depicted in Fig. 3.9(b). In this scenario, terminal M shall not freeze its backoff counter and the T_{fr} should be set to zero. Note that A can monitor the behavior of M only if it can virtually sense the $S-D$ transmission (i.e., A is in the range of D but not of S). If A can physically sense S's transmission and M initiates its own transmission (M's backoff becomes zero), A will experience a collision, thus being unable to monitor M. However, even if M cannot be correctly monitored by A, possible misbehavior by M does not cause an unfairness at A, as A's backoff counter is already frozen due to the $S-D$ transmission. That is, A is not currently competing for the channel because it is occupied by other terminals. In fact, if M reduces the delay until it reserves the channel, A will have the opportunity to initiate its own transmission faster.

Case 3: The $S-D$ transmission can be sensed by M, but not A. This scenario is depicted in Fig. 3.9(c). In this case, A cannot know that M must freeze its backoff counter, because it cannot sense the $S-D$ transmission, either physically or virtually. The monitor A will set T_{fr} to zero. This gives M the opportunity to select a small backoff without being detected. However, M would still have to defer from transmission (freeze its backoff counter) until the $S-D$ transmission cycle is completed to avoid causing a collision at $S-D$ and its own transmission. In a backlogged scenario, terminal A continues decrementing its own backoff counter while M's counter remains frozen. Thus, A is more likely to seize the channel before M. As a result, the BMA does not yield significant advantages for M in its contention with A. Terminal M can still maintain an advantage related to other terminals that also freeze their counters due to the $S-D$ transmission. However, these terminals will belong to either Case 1 or Case 2 and can accordingly monitor M.

Backoff Monitoring Algorithm: The pseudocode of the backoff monitoring module employed by a monitor A for a neighboring node M is summarized in Algorithm 1.

In Algorithm 1, the monitor A must be aware of M's one-hop neighborhood $\mathcal{N}_M^1$,

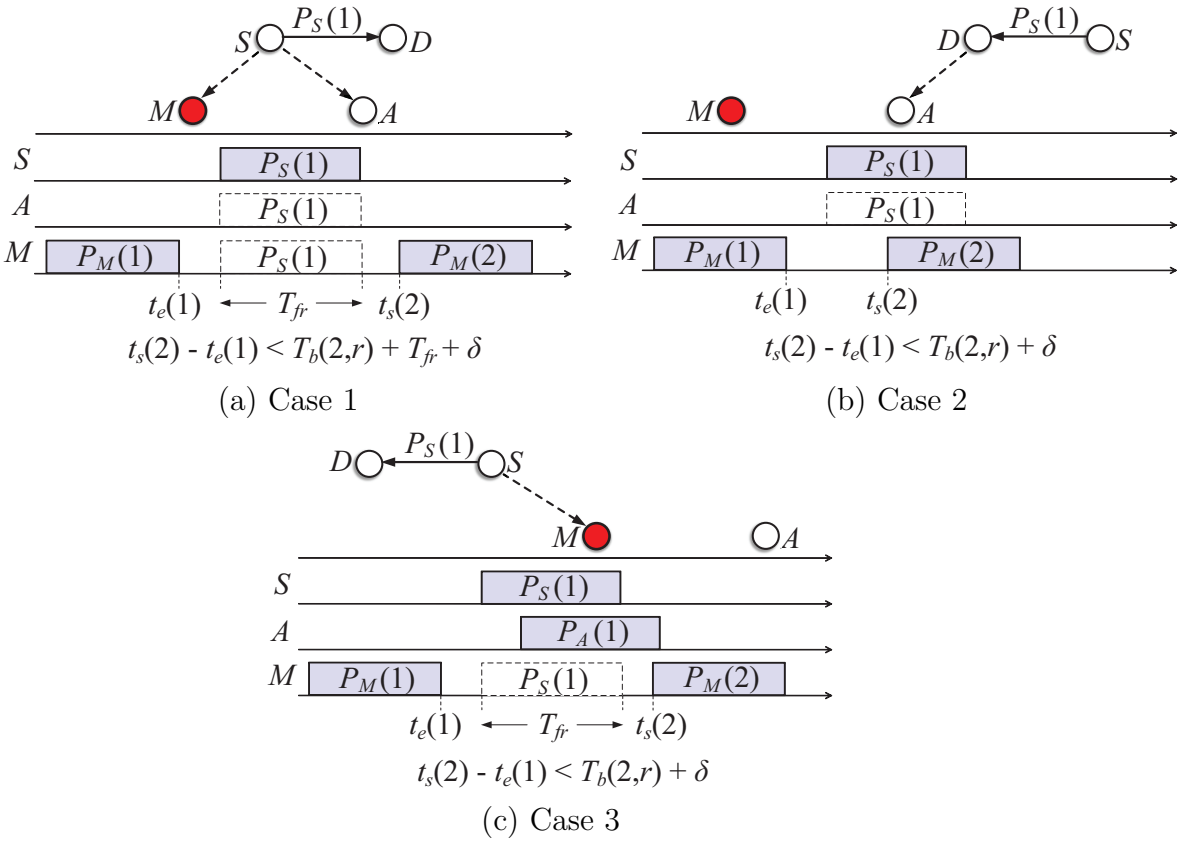


Figure 3.9: Classification of backoff monitoring scenarios for the computation of the backoff counter freezing period T_{fr}.

Algorithm 1 Backoff Monitoring Module

Input: $i, r, M, \mathcal{N}_1^M$
Output: Evaluation of M's behavior for the i^{th} transmission
Let C_M be a backoff counter held by A
At the end of M's $(i-1)^{st}$ transmission or start of a data/control phase set $C_M \leftarrow 0$, $T_{fr} \leftarrow 0$
while i^{th} transmission of M has not started **do**
  if Case 1: a transmission is sensed by M and A **then**
    $T_{fr} \leftarrow T_{fr} + 1$
  else
    $C_M \leftarrow C_M + 1$
  end if
end while
if $C_M < T_b(i, r) + T_{fr}$ **then**
  M is misbehaving
else
  M is behaving
end if

to classify nearby transmissions to cases 1, 2, or 3. In the next section, we describe an efficient secure neighbor discovery protocol for determining the two-hop topology (and hence, the one-hop topology of any neighbor). Moreover, we note that during the control phase, all one-hop neighbors of M can monitor M's compliance with the publicized backoff schedule. This is because all nodes converge on the control channel. When transitioning to the data phase, only the subset of one-hop neighbors $\mathcal{K}_M^1 \subseteq \mathcal{N}_M^1$ that hop to the channel reserved by M can continue to monitor M. This subset contains at least the terminal who will be communicating with M during the data phase.

When terminals converge to the control channel at the end of a data phase, all terminals in $\mathcal{K}_M^1$ remain synchronized to the correct values (i, r), based on the transmissions that occurred during the data phase. However, terminals in $\mathcal{N}_M^1 \backslash \mathcal{K}_M^1$ are not aware of M's data transmissions, as they hopped to other channels. For the first control phase transmission, terminal M is monitored solely by terminals in $\mathcal{K}_M^1$. The remaining one-hop neighbors synchronize with parameters (i, r) after the first successful transmission cycle by M.

Secure neighbor discovery

To facilitate the backoff monitoring module and the detection of MRAs, 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 terminal collusion, deployment of wormholes, and others [105, 106]. For our purposes, we develop a simple scheme that is efficient to implement and suits the misbehavior model. Our scheme involves the following steps.

Step 1: Every terminal X broadcasts a hello message

$$X \mid\mid sig_{sk_X}(X),$$

where $sig_{sk_X}(X)$ is the signature on X with private key sk_X known only to X. Using the signed hello messages, every terminal X builds a one-hop neighbor list $\mathcal{N}_X^1$.

Step 2: Every terminal X broadcasts

$$X, \mathcal{N}_X^1 \mid\mid sig_{sk_X}(X, \mathcal{N}_X^1).$$

Using $\mathcal{N}_Y^1, \forall\ Y \in \mathcal{N}_X^1$, each terminal X creates its two-hop neighbor list $\mathcal{N}_X^2$. A terminal $Z \in \mathcal{N}_Y^1$ is a two-hop neighbor of terminal X, if $Z \notin \mathcal{N}_X^1$.

Step 3: Every terminal X broadcasts

$$X, \mathcal{N}_X^2 \mid\mid sig_{sk_X}(X, \mathcal{N}_X^2).$$

Every terminal X creates a common neighbor list $P_X(Y)$, for each two-hop terminal $Y \in \mathcal{N}_X^2$. That is, $P_X(Y) = \{Z : Z \in \mathcal{N}_X^1, Y \in \mathcal{N}_Z^1\}$.

In the secure neighbor discovery mechanism, a malicious terminal M can choose to omit certain terminals from the one, or two-hop neighbor lists, which are broadcasted in Steps 1 and 2. However, this behavior can be easily detected based on the inconsistency in the neighbor lists of one-hop neighbors. If $M \in \mathcal{N}_X^1$ this implies

that $X\mathcal{N}_M^1$, and hence, X cannot be omitted from the one-hop list of M. Moreover, a terminal X can prove that M is its one-hop neighbor by presenting the hello message broadcasted by M during Step 1. This message is signed by M. The same argument can be extended to the two-hop neighbor lists broadcasted in Step 2.

The lists generated during the secure neighbor discovery can be used by a monitoring terminal A to compute T_{fr} for a monitored terminal M when an $S - D$ transmission occurs in the vicinity of A. If $S \in N_M^1$ or $D \in N_M^1$ then this is classified as Case 1 and the backoff counter of M must remain frozen. Otherwise, the transmission of $S - D$ is classified to belong to Case 2. Note that Case 3 cannot be detected by A as it involves a transmission that cannot be sensed (either physically or virtually by A. However, as explained in Section 3.4.1, Case 3 provides limited opportunity for a terminal to gain throughput advantages from misbehaving.

Parameter manipulation

We now consider the scenario where a misbehaving terminal M manipulates (i, r) to select a random number $b_M(i)$ that leads to a short backoff time $T_b(i, r)$. This is possible because monitoring terminals cannot attribute collided frames to their senders. Hence, a misbehaving sender can: (a) avoid the incrementation of i and r if its transmission collided and (b) take advantage of other collisions to advance i to a future value $i + k, k > 1$ that yields a smaller $T_b(i + k, r)$.

For case (a), assume that M does not increment (i, r) after one of its frames has collided. This strategy will prevent the increase of the CW due to the increase of r. However, since collisions are receiver-dependent, a misbehaving sender cannot know with certainty that a collision has indeed occurred at all monitoring terminals. This scenario is depicted in Figure 3.10 using the SP-MMAC design. M broadcasts an ATIM frame for terminal B. Terminals C and D act as monitors of M. The ATIM frame sent by M collides at B due to a concurrent reception of an ATIM frame from F. Monitoring terminals C and D still receive M's ATIM frame, as they are located outside the interference range of F. A retransmission of an ATIM frame with the same (i, r) is detected by C and D as misbehavior. Moreover, the intended

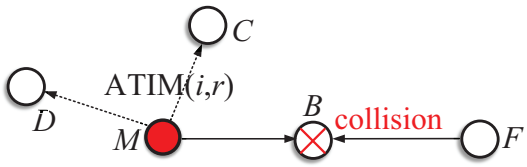


Figure 3.10: Detection of the manipulation of the retransmission number r during the control phase in SP-MMAC.

receiver may choose to intentionally drop correctly received frames in order to detect a misbehaving sender that does not increase i or r. For case (b), Proposition 3 shows that it eventually leads to either the identification of misbehavior or does not reduce the backoff time.

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

Proof. Assume that M successfully completed the i^{th} transmission cycle. For the $(i+1)^{st}$ transmission cycle, let M skip $k>1$ values from the pseudo-random sequence and use $b_M(i+k)$ instead of $b_M(i+1)$, because $b_M(i+k) \ll b_M(i+1)$. In order for M to be protocol-compliant, (a) the advertised value r must increase by $(k-1)$ and (b) M's transmission must start no earlier than

$$(k-1)T_0 + \sum_{j=1}^{k} \lceil b_M(i) \min\{2^{(j-1)} cw_0, cw_{\max}\} \rceil - k \tag{3.6}$$

slots after the i^{th} transmission cycle, where T_0 denotes the timeout period in slots. Condition (a) holds true since i is considered to be the last successful transmission for which the retransmission number r is reset to zero. In order for M to use $(i+k)$ in its next successful attempt, $(k-1)$ unsuccessful ones must have preceded, thus setting the value of r to $(k-1)$. In this case, terminal i must have followed all intermediate backoff intervals as indicated by the values $\{T_b(i+1,0), T_b(i+2,1), \ldots, T_b(i+k,k-1)\}$. Moreover, an additional period equal to the $(k-1)$ previous unsuccessful trials of transmitting a frame and waiting for a timeout (period 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 ($T_b(i+1,0)$ is a factor of the sum in (3.6)). □

3.4.2 Mitigation of Multi-reservation Attacks in SP-MMAC

In this subsection, we mitigate the MRA in SP-MMAC by modifying its PCL update rules and detecting fictitious terminals.

PCL Update Rules

In the original MMAC [8], a terminal 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 Figure 3.2. Terminal M places four reservations on f_2, while terminal C places one reservation on f_1. Both f_1 and f_2 have the same priority in the PCLs of the terminals overhearing the channel negotiations. Hence, terminal B may choose f_1 or f_2 with equal probability, were it to communicate with another terminal. 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 terminal 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 terminals 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 terminal i overhears an ATIM-ACK or ATIM-RES frame indicating the selection of channel f_ℓ with priority LOW, it checks whether the frame 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.

We emphasize that the modified PCL rules are designed to balance the number of pairs that occupy each channel during the data phase. The priority of a channel is modified according to the number of pairs that are scheduled to communicate on that channel. This leads to the uniform distribution of the communicating pairs over the available channels. Figure 3.11 shows the distribution of data transmissions under the modified PCL rules, for a scenario with five pairs and three channels. Let pairs $A-B$, $C-D$, $E-F$, $G-H$, and $I-J$ be within the same collision domain and contend for channels f_1, f_2, and f_3. Initially, all channels have the same priority. When a reservation is placed on a channel, its priority is demoted. After the first three reservations, all three channels have the same priority. The next two reservations are distributed on two out of the three channels. During the data phase, pair $A-B$ occupies f_2, pairs $C-D$ and $E-F$ occupy f_1, while pairs $I-J$ and $G-H$ occupy f_3.

Note that when a pair has a low traffic load, it is possible that it reserved a channel f_i during the control phase but does not transmit during the upcoming data phase. In this case, the rest of the pairs scheduled to transmit on f_i will seize channel, without impacting the actual load achieved on f_i. As illustrated in Figure 3.11, when $G-H$ pair does not have any outgoing frame, pair $I-J$ performs successive transmissions on f_3.

Detection of Fictitious Terminals

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 frames by broadcasting ATIM-ACKs. Any terminal that overhears those ATIM-ACKs assumes that the source of the corresponding ATIM/ATIM-RES frames is a hidden terminal. We employ the secure neighbor discovery mechanism presented in Section 3.4.1 to detect the existence of such fictitious sources as follows. Assuming that for every terminal X, its common neighbor list $P_X(Y)$ ($Y \in \mathcal{N}_X^2$) is already obtained after executing the secure neighbor discovery

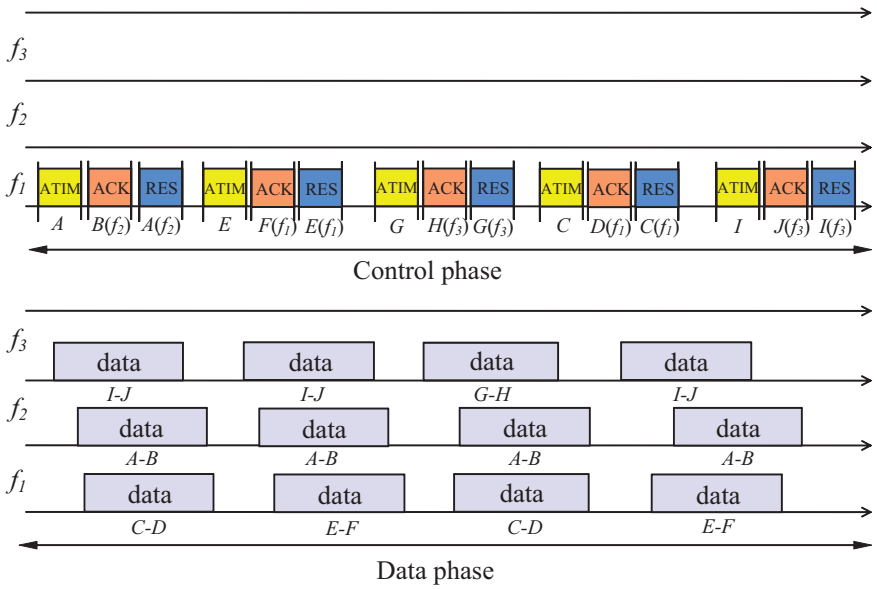


Figure 3.11: Load balancing during the data phase is achieved under the modified PCL rules.

algorithm. Two-hop neighbors are hidden terminals.

Step 1: If $|P_X(Y)| = 1$, terminal X suspects terminal Y to be fictitious and terminal $P_X(Y)$ to be misbehaving. This is because no other terminal but $P_X(Y)$ can verify the existence of j.

Step 2: For a suspected fictitious terminal Y, terminal X issues a signed challenge c, sent via $P_X(Y)$. Terminal Y must reply to the challenge with a signed response.

The application of our secure neighbor discovery protocol is shown in Figure 3.12. Tables N_i show the combined 1-hop and 2-hop topological information. For terminal $F \in \mathcal{N}_A^2$, it holds that $P_A(F) = \{M\}$. That is, terminal F appears to be a hidden terminal to all neighbors of A, except M and hence, M is suspected of misbehavior. Terminal 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.

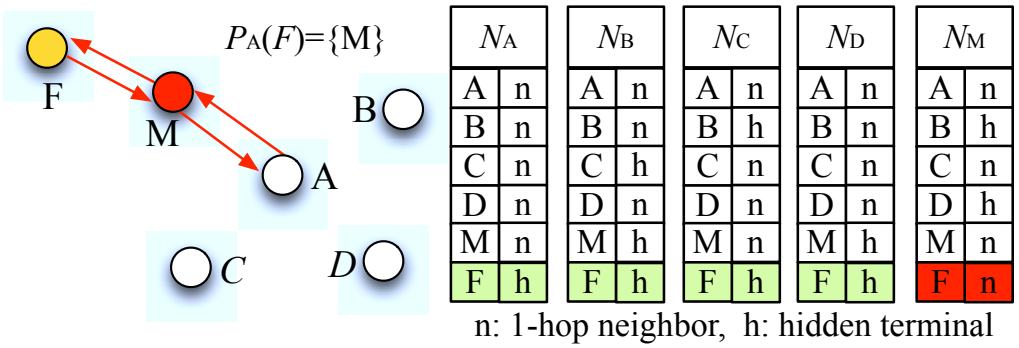


Figure 3.12: Secure neighbor discovery protocol for SP-MMAC. Tables N_i show combined 1-hop and 2-hop topological information.

3.4.3 Mitigation of Multi-reservation Attacks in DCC-MMAC

As discussed in Section 3.3.2, an MRA can be combined with a BMA in order for misbehaving node M to monopolize a target channel in an efficient manner. In this section, we propose a mechanism to mitigate the effect of such MRA by modifying the DCC-MMAC operating rules and incorporating a multi-reservation monitoring module. The monitoring module is capable of detecting the misbehaving terminals which do not comply with the proposed operating rules.

Modified Operating Rules in DCC-MMAC

We improve the DCC-MMAC design by modifying its operating rules related to the channel negotiation on the control channel. In the modified set of rules, a sender terminal S involved in an active data transmission on channel f_i is prohibited from initiating channel negotiations on the control channel until the present transmission nears the end. At the mean time, other contending terminals are allowed to negotiate and reserve f_i before it being released by S, given that f_i has higher priority in their PCLs.

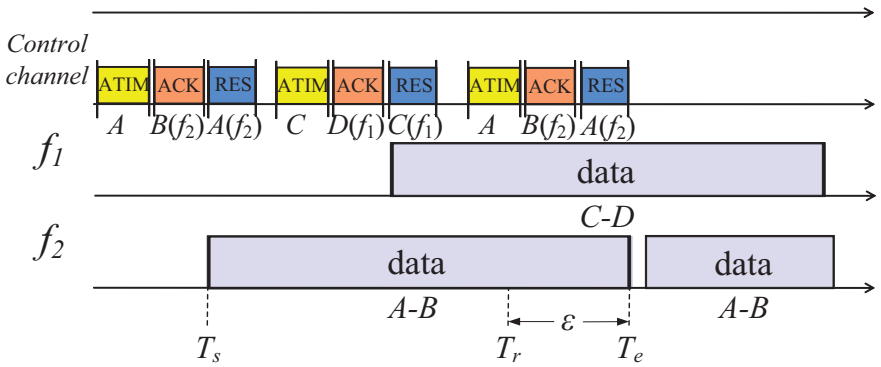
We use Figure 3.13 to illustrate this rule. Four terminals A, B, C, and D within same collision domain contend to use two data channels. Suppose A initiates a data transmission to B on channel f_2 upon their successful negotiation on the control channel. Let the starting and ending time of this transmission be denoted by T_s

and T_e respectively, as depicted in Figure 3.13 (a). While the $A-B$ transmission is ongoing, A keeps monitoring the data channels that have been reserved by its neighbors. If A detects that f_2 has not been reserved ε time units before the end of the ongoing transmission, then A is allowed to further negotiate for channel f_2 (or any other channel). The purpose of allowing a reservation towards the end of an active transmission, is to allow an active terminal to perform back-to-back transmissions without any negotiation delay under low contention, but also provide transmission opportunities to contending terminals, under high contention. If no other terminal is interested in f_2, it is wise for $A-B$ to continue reserving early instead of spending another contention period together with a channel negotiation for every transmission. We denote the time when A is allowed to attempt another negotiation as

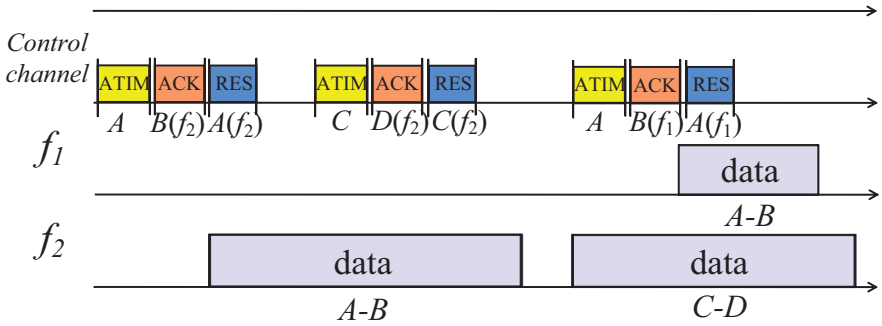
$$T_r = T_e - \varepsilon \tag{3.7}$$

where ε is an adjustable parameter. The value of ε is selected to guarantee that the active terminal can place a reservation before the ongoing transmission is completed. Note that for short data frames, this opportunity may not exist. To avoid collision, A needs to sense the channel to be idle for DIFS duration at T_r before transmitting an ATIM frame.

On the other hand, if any neighboring terminal reserved f_2 during $A-B$ transmission as shown in Figure 3.13 (b), then A is not allowed to make further reservation until it completes the ongoing transmission. It has to start competing for control channel to negotiate for its next transmission after $A-B$ transmission is finished. It is worth noticing that f_2 could still be preferred by $C-D$ pair during $A-B$'s ongoing transmission, despite its later release time compared to f_1. For instance, if C and D detect much better SINR on f_2 than f_1 which is used as channel prioritizing criteria, they would reserve f_2 even if f_1 is unoccupied. This may give opportunities to other terminals who receive better SINR on f_1 to make reservation for it.



(a) Scenario where A is allowed to make further negotiation during its data transmission on f_2.



(b) Scenario where A is not allowed to make further negotiation during its data transmission on f_2.

Figure 3.13: Illustration of the modified operating rules for DCC-MMAC.

Multi-reservation Monitoring Module

The multi-reservation monitoring module identifies misbehaving terminals that violate the above operating rules, i.e., the terminals which attempt to negotiate channel before T_r. To achieve this, each terminal maintains a Channel Reservation Table (CRT) where the reservation status of the n data channels f_1, f_2,...,f_n are being tracked of. The CRT contains n entries, each entry CRT_k ($1 \leq k \leq n$) records the reservation status of channel f_k, including the starting time (T_s) and the ending time (T_e) during which f_k is reserved, as well as the identity of the terminal that reserved it. All one-hop neighbors of terminal i, denoted by $\mathcal{N}_i^1$, are able to monitor i's behavior. When a terminal $j \in \mathcal{N}_i^1$ overhears an ATIM frame from i at time T_c, j checks to see if i is associated with any entry in CRT_j. If an existing entry of i is found, j further checks corresponding T_s and T_e and compare them with T_c. Terminal i is identified as misbehaving if $T_s < T_c < T_r + DIFS + T_{ATIM}$, where T_{ATIM} is the duration of an ATIM frame. Upon detection, i will be reported to the reputation system by j and any further reservation made by i will not be updated in CRT_j. In this way, every terminal is monitored by its one hop neighbors and any terminal that violates the above operating rules can be detected.

3.5 Vulnerabilities of CR-MAC Protocols and Countermeasures

Compared to multi-channel MAC protocols, CR-MACs implement additional tasks including cooperative spectrum sensing, spectrum information sharing and spectrum management. Cooperative CR-MAC protocols are designed to provide fair access opportunities to all participating CRs, if CRs remain protocol-compliant. However, selfish or malicious CRs violating the CR-MAC protocol specifications can gain an unfair share of the idle spectrum (selfish), or deny spectrum access to other CRs (malicious). Such selfish or malicious activities could significantly degrade the performance of CRNs, or render them inoperable for large periods of time.

In this section, we identify possible CR-MAC vulnerabilities. According to the different function modules of CRNs discussed in Section 2.3.1, we categorize these

vulnerabilities to three classes: (a) attacks on spectrum sensing, (b) attacks on the channel negotiation process, and (c) denial-of-service attacks. For each class, we present possible countermeasures.

3.5.1 Spectrum Sensing Vulnerabilities

Distortion of Spectrum Availability

CR-MAC protocols rely on cooperative sensing mechanisms to determine the set of idle channels. A malicious CR can report false sensing observations to distort the spectrum availability. False information is particularly harmful when an "AND" rule is used to combine sensing observations. In this case, a single false report can prevent access to idle channels.

Spectrum distortion can be easily achieved in spectrum information sharing techniques that utilize busy tones [58,61]. Such tones are unauthenticated and could be transmitted by any CR without reflecting the true channel state. As an example, referring to Figure 2.7(a), malicious CR D could transmit a busy tone on every slot during the spectrum information sharing phase, thus indicating that channels f_1-f_3 are occupied by PUs. CRs A, B, and C will defer from communicating in the upcoming data phase. A similar attack can be mounted when the set of idle channels is reported via explicit messaging.

Primary User Emulation (PUE) Attacks

In a PUE attack, malicious CRs emulate the transmission characteristics of a PUs to distort the spectrum sensing process. This attack is possible because the signals transmitted by a PU are detected using signal detection techniques that do not provide any form of authentication [107]. Using the software defined radio engine, a CR can emulate PU signals that conform to the characteristics of the detectors. Referring to Figure 2.7(a), malicious CR D emitting emulated PU signals during the spectrum sensing phase could lead CRs A, B, and C in reporting the presence of an incumbent signal on all three channels during the spectrum information sharing

phase. As a result, A, B, and C defer from transmiting during the upcoming data phase.

3.5.2 Attacks on the Channel Negotiation Process

Cognitive Radio Backoff Manipulation Attacks (CR-BMA)

The CR-BMA is a variant of the BMA discussed in Section 3.3.1 for regular MMAC protocol designs. Similarly, CRs engage in a channel negotiation process for coordinating access to the set of idle channels [58–61, 108] in split-phase and dedicated control channel CR-MAC protocols. This negotiation is contention-based, following variants of the CSMA/CA protocol. Malicious terminals that manipulate the contention protocol parameters can gain exclusive and/or more frequent access to a subset of available channels, thus occupying a disanalogous portion of the available spectrum. This can be achieved by manipulating the backoff mechanism of CSMA/CA.

In a CR-BMA, a selfish terminal systematically selects small backoff values to increase its chances of reserving an idle channel compared to protocol-compliant terminals [31]. This attack is particularly effective when the control channel becomes saturated due to the large number of contending CRs, or the entire idle spectrum is assigned to a single CR [60]. In this case, CRs unable to complete a channel negotiation during the control phase, defer from transmission during the upcoming data phase.

We use Figure 3.14 to highlight the severity of a CR-BMA in CR-MAC protocols. In this scenario, CR A selects a small backoff value in order to seize the control channel before any other CR. Because the entire spectrum is bonded and allocated to a single CR, CRs B and D are deprived of channel access. Similar illustrations can be shown for other CR MAC protocols relying on CSMA/CA for control channel contention.

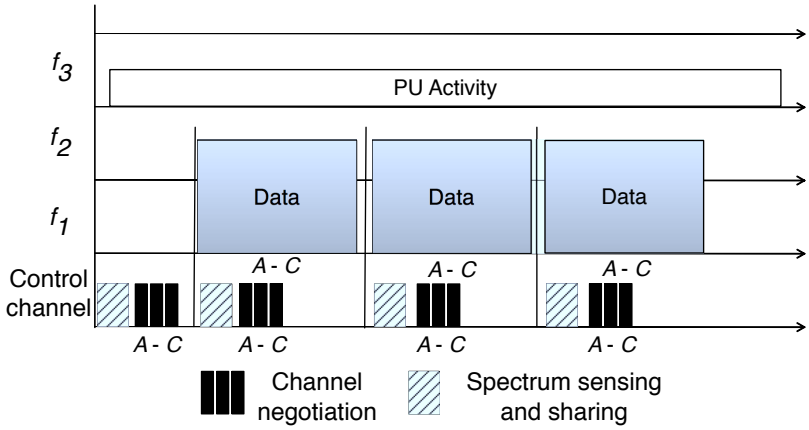


Figure 3.14: Backoff manipulation attack for the CR-MAC in [60]. Misbehaving CR A systematically selects small backoff values during the channel negotiation phase. All idle spectrum is bonded as one channel and assigned to the A-C communicating pair.

Cognitive Radio Multi-Reservation Attacks (CR-MRA)

The CR-MRA is also a variant of the MRA as discussed in Section 3.3.2. Similarly in CR-MAC protocols, channel selection during the channel negotiation process is based on the expected traffic load on each of the available channels. This selection is facilitated by overhearing control messages, as various CR pairs negotiate their channel assignments. In the channel negotiation, the communicating CRs select the idle channel with the least number of reservations. At the same time, nearby CRs lower the priority of the selected channel. However, this strategy creates the opportunity for launching a CR-MRA. In this attack, a malicious CR places multiple reservations for one or several targeted channels to lower their priority in the channel preference lists of contending CRs. As a result, protocol-compliant CRs defer from selecting the targeted channels, thus providing exclusive use of those channels to the malicious CR. A realization of the CR-MRA can be referred to Figure 3.2 where the selfish CR M appears to be legally engaged in several channel negotiations with fictitious CRs by taking advantage of the hidden terminal problem.

3.5.3 Countermeasures

Countering Attacks on Spectrum Sensing

Countering Distortion of Spectrum Availability: Cooperative sensing is vulnerable to spectrum availability distortion attacks due to the conservative nature of the hard decision combining mechanism. To avoid interference with PUs, channel availability is determined by following an "AND" rule. A channel is considered to be free of PU activity if all cooperating CRs agree on its idle state. A single report from a malicious CR is sufficient to discard an idle channel from further use.

To mitigate the impact of such attacks, decision combining mechanisms using *threshold voting* rules can be employed. In threshold voting, a channel is deemed to be occupied by a PU, if at least τ out of n CRs report it to be busy, where τ is a system-defined parameter. Under threshold voting, a small number of colluding CRs cannot distort the spectrum availability. The caveat of a threshold rule is that it does not always account for the spatial variations of PU activity. As an example, an occupied channel detected by a small number of CRs could be falsely declared as idle. To alleviate this drawback, parameter τ must be adaptive to the spacial variations of PU activity.

Threshold voting is easily implemented when spectrum information sharing is realized via the exchange of authenticated messages. However, several CR-MAC protocols employ simpler forms of information sharing such as busy tones [60, 61]. Busy tones are not authenticated, nor do they account for the number of CRs reporting on the channel state. One candidate solution could be the measurement of the busy tone power. If multiple CRs transmit a busy tone on the corresponding slot, the power of that tone is expected to be high. However, a malicious CR may intentionally increase the power of its busy tone to defeat a power-based busy tone threshold voting technique.

Countering Primary User Emulation Attacks: Even if threshold voting is selected as the cooperation rule, the spectrum availability can still be distorted under a PUE attack. When a malicious CR emulates PU activity on a channel

f_i, all nearby CRs detect f_i to be busy. Hence, f_i is declared to be busy under either an "AND" or a threshold voting rule. Defending against a PUE attack is challenging because the energy or feature detectors used during spectrum sensing cannot verify the authenticity of a PU signal. Moreover, current regulations prohibit any modifications on legacy systems.

Several mechanisms have been proposed for authenticating PU activity without imposing any modifications on the PU network. If the locations of PUs are known a priori, PU signals can be authenticated by determining the position of the PU transmitter [109]. This can be achieved by estimating the distance between the PU and several receiving CRs using the received signal strength (RSS) and computing the PU location using trilateration. Manipulation of the transmission power by a malicious CR for emulating the fixed PU position becomes challenging if the malicious CR is not within less than a few meters from the legitimate PU. PU signal authentication can also be achieved by constructing an RF signature of the PU-CR channel [110, 111]. RF signatures capture unique characteristics of the RF channel (channel and frequency response) between two stationary terminals, based on random multipath components. These characteristics cannot be emulated unless the malicious CR is located within a few wavelengths from the emulated PU. Assuming that PU terminals are physically protected, mounting a PUE attack that emulates the RF channel becomes challenging.

Countering Attacks on the Control Channel

Countering Cognitive Radio Backoff Manipulation Attacks: BMA attacks can be mitigated by regulating and monitoring the backoff schedule of contending terminals. As a reminder here, in [31], the backoff value of a sender is assigned by the corresponding receiver. The receiver is responsible of monitoring the sender's compliance with the assigned backoff value. If the sender deviates from that value, the receiver "punishes" the sender by assigning larger backoff values for future transmissions. Repeated violations lead to the characterization of the violating terminal as misbehaving, and eventually to its removal from the network.

As discussed in previous sections, receiver-based backoff assignment mechanism is not effective in the multi-channel domain, and thus not directly applicable in CRNs. To counter CR-BMA, mechanisms proposed in Section 3.4.1 can be incorporated in CR-MAC designs: forcing every CR publish its backoff schedule ahead of time. Every CR could broadcast the unique seed of a publicly known pseudorandom number generator used for the generation of the backoff values. Neighboring CRs can then monitor the backoffs selected by their peers and detect misbehaving CRs that violate their backoff schedules.

Countering Cognitive Radio Multi-reservation Attacks: CR-MAC protocols are vulnerable to CR-MRAs due to: (a) the adjustment of channel priorities based on the number of reservations placed on each channel, and (b) the exploitation of the hidden terminal problem for introducing fictitious terminals. The former vulnerability can be countered by modifying the channel priority rules such that the priority of a channel f_i is lowered only if new CR pairs place reservations on f_i as discussed in Section 3.4.2. Referring to the attack scenario presented in Figure 3.2, multiple reservations placed by malicious CR M on channel f_2 would only lower the priority of f_2 by one. Thus, CR M would not be able to isolate f_2 from the rest of the CRs.

Communication with fictitious CRs for the purpose of placing multiple reservations can be defeated by employing secure two-hop neighbor discovery protocols, such as the one proposed in Section 3.4.2. Such protocols are executed during the network setup phase and are periodically repeated if the CRs are mobile. If the two-hop neighborhood is securely known, CRs are aware of the identities of all CRs that are hidden terminals. Hence, malicious CRs cannot pretend to communicate with fictitious CRs.

3.6 Performance Evaluation

In this section, we evaluate the impact of MMAC misbehavior on fairness and network performance. We further evaluate the improvements achieved by our detection

and mitigation methods.

3.6.1 Simulation Setup

We performed our experiments using the OPNET^{TM} Modeler packet-level simulator [112]. We considered a single-hop network topology of multiple terminal pairs communicating over three orthogonal channels of capacity 2Mbps. Both the SP-MMAC and DCC-MMAC protocol families were implemented and evaluated. For SP-MMAC, we implemented the MMAC protocol in [8]. The control phase was fixed to 20ms and the data phase was fixed to 80ms unless otherwise stated. For DCC-MMAC, we implemented the DCA protocol in [9]. 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 and results were averaged over 40 simulation runs.

3.6.2 Impact of the Backoff Manipulation Attack

Impact of the BMA on SP-MMAC

In this set of experiments, we evaluated the impact of the BMA on SP-MMAC protocols. We considered a misbehaving terminal M that uniformly selected its backoff from $[0, 4)$ during both the control and data phases, and contended with 10 well-behaved pairs. Well-behaved pairs conformed to the MMAC protocol specifications. Figure 3.15 shows the average throughput (T) of M compared with the average per-flow throughput of well-behaved terminals and the average per-flow throughput in the absence of misbehavior. We observe that for low λ, the throughput of all flows is identical. However, in high load conditions, the misbehaving terminal gains a significant advantage (about 20%) compared to well-behaved pairs. This significant throughput gain is due to the following. With the initialization of the control phase, all terminals choose small backoff values which leads to collisions and increase of the CW for the well-behaved terminals. However, M continues to select small backoff

values. leading to the frequent capturing of the control channel. This guarantees a data transmission for the upcoming data phase.

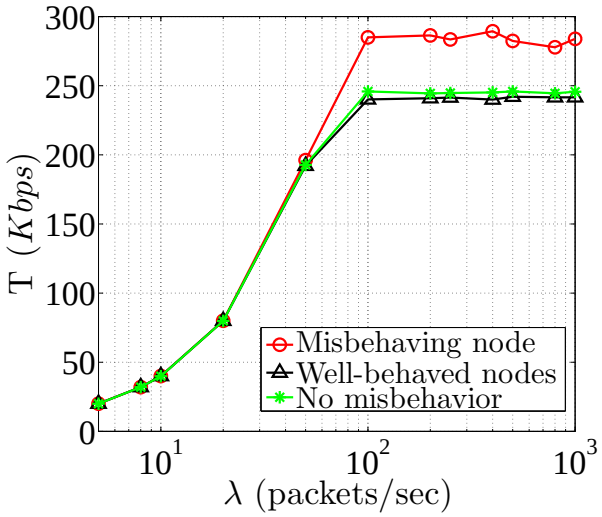


Figure 3.15: Throughput as a function of the packet arrival rate when one terminal launches a BMA in SP-MMAC.

Impact of the BMA on DCC-MMAC

In this set of experiments, we evaluated the impact of the BMA on DCC-MMAC protocols. We considered a misbehaving terminal M that used the same misbehavior strategy as in Figure 3.15, i.e., M selected its backoff counter uniformly from $[0, 4)$. Figure 3.16 shows the average throughput achieved by M compared with the average per-flow throughput of well-behaved terminals and the average per-flow throughput in the absence of misbehavior. We observe that in high load scenarios, M achieves approximately 6 times the throughput of any other well-behaved terminal. Compared with the impact of the BMA on the SP-MMAC protocol, the misbehaving terminal achieves a significantly higher throughput by launching a BMA on DCC-MMAC. This is because in SP-MMAC, well-behaved terminals can still reserve the same channel as M, even if M was the first to select a channel. M still has to contend with other terminals during the data phase. However, in DCC-MMAC, once M completes a negotiation over the control channel, it isolates one of the data

channels without experiencing further contention. As a consequence, well-behaved terminals have fewer data channels to share.

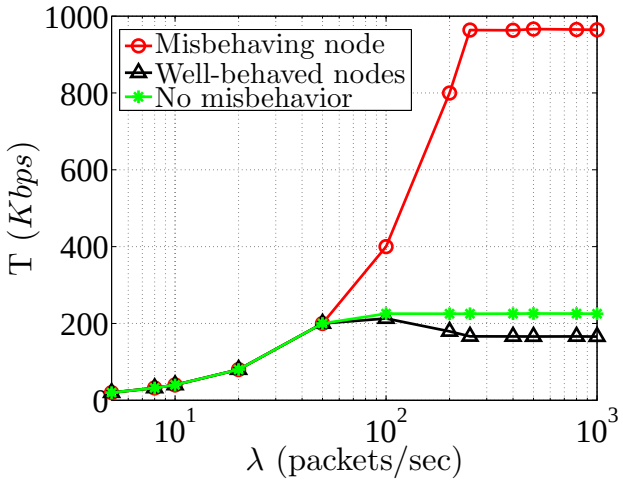


Figure 3.16: Throughput as a function of the packet arrival rate when one terminal launches a BMA in DCC-MMAC.

3.6.3 Impact of the Multi-reservation Attack

Impact of the MRA on SP-MMAC

In this set of experiments, we evaluated the impact of the MRA on SP-MMAC protocols. We considered one misbehaving terminal aiming at isolating a single channel ($n_M = 1$) when contending with 10 well-behaved pairs. According to Proposition 1, the misbehaving terminal placed five reservations (one real and four fake ones) on the targeted channel to guarantee exclusive use of that channel. To ensure the placement of the required reservations, the four fake reservations were transmitted back-to-back after the first real reservation (i.e., the backoff counter was always set to zero).

Figure 3.17 shows the average throughput of the misbehaving terminal vs. the average throughput of well-behaved terminals and the average per-flow throughput in the absence of misbehavior. We observe that for low λ, the throughput of all flows is identical and similar to Figure 3.15. Under high traffic load ($\lambda \geq 100$ packets/sec),

the misbehaving terminal achieves approximately 1.6 times the throughput of any other well-behaved pair. This is because as long as the misbehaving terminal seizes the control channel, it can isolate one of the channels for exclusive use during the upcoming data phase. In this case, the well-behaved terminals have to contend on the remaining two channels. Comparing the throughput under a BMA (Figure 3.15) and under an MRA (Figure 3.17), we observe that the misbehaving terminal achieves higher throughout in the MRA case. This is because under an MRA, the misbehaving terminal manages to isolate one of the data channels during the data phase. However, under a BMA, the misbehaving terminal may still have to contend with other terminals.

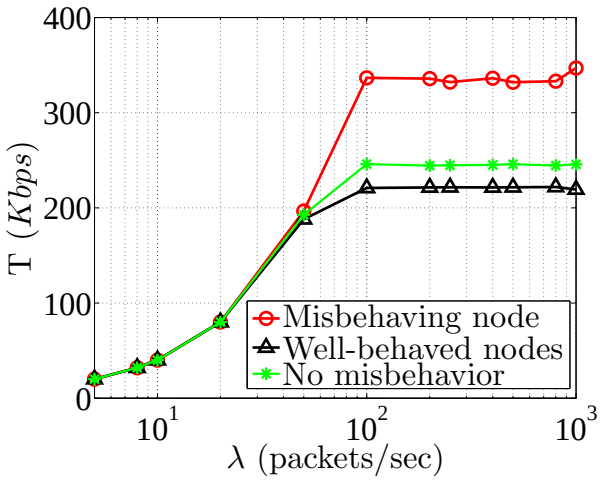


Figure 3.17: Throughput as a function of the packet arrival rate when one terminal launches an MRA in SP-MMAC.

We also evaluated the impact of a combined BMA/MRA attack. We again considered one misbehaving terminal M aiming at isolating a single channel when contending with 10 well-behaved pairs. M placed five reservations on the targeted channel (one real and four fake ones) and applied a BMA towards the first real reservation.

Figure 3.18 shows the average throughput of the misbehaving terminal vs. the average throughput of well-behaved terminals and the average per-flow throughput

in the absence of misbehavior. We observe that the misbehaving terminal achieves almost three times the throughput of well-behaved terminals under high traffic load conditions. This is because by combining the MRA with the BMA, the misbehaving terminal is able to reserve and isolate the targeted channel before any other contending pair on almost all control phases. Therefore, it does not have to share its channel during the data phase. Moreover, the throughput of well-behaved terminals is reduced by 25% (approximately 60Kbps) compared to the scenario where all terminals follow the MMAC protocol.

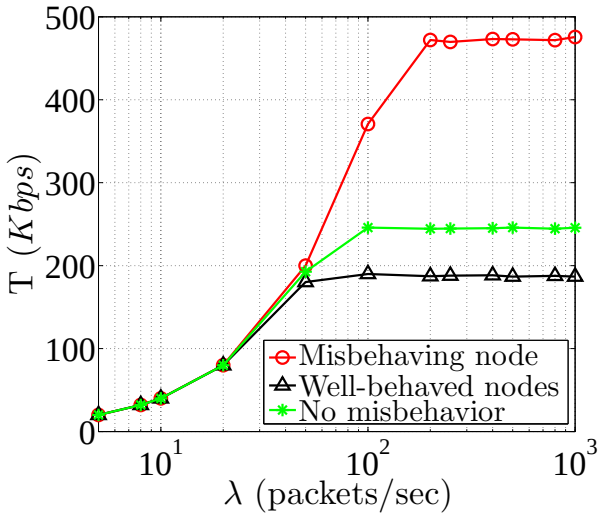


Figure 3.18: Throughput as a function of the packet arrival rate when one terminal launches an MRA together with a BMA in SP-MMAC.

In Figure 3.19, 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 Figure 3.20, we show the average throughput of contending pairs under an MRA and a BMA, but for a control phase duration of 30ms. A longer control phase allows terminals 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.

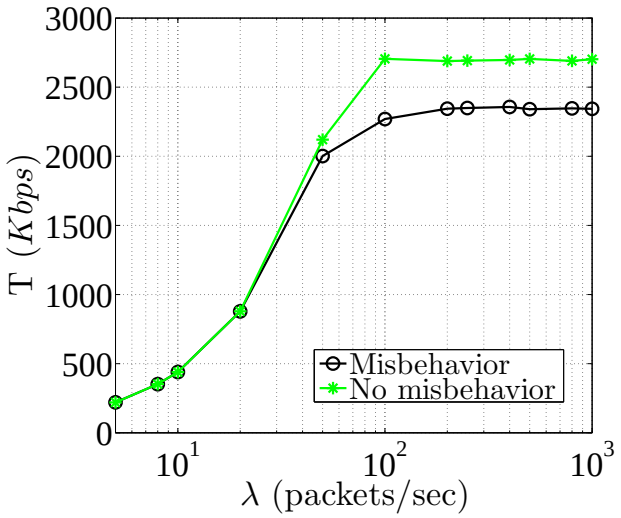


Figure 3.19: Aggregate throughput for all contending pairs in the presence and absence of misbehavior in SP-MMAC.

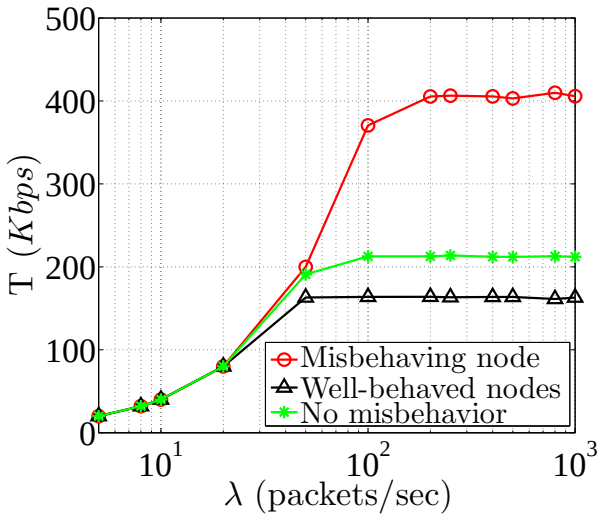


Figure 3.20: Throughput as a function of the packet arrival rate when one terminal launches an MRA together with a BMA in SP-MMAC with a 30ms control phase duration.

To further study the impact of BMA/MRA on the performance of multi-channel networks, we simulated the same scenario as in Figure 3.18, but with 2, 3, 4, and 5 channels available. Figure 3.21 shows the throughput advantage achieved by the misbehaving terminal as a function of number of available channels in high load conditions. The throughput advantage is presented in terms of the ratio of the average throughput achieved by the misbehaving terminal over the average throughput of well-behaved terminals. Results show that the throughout advantage of misbehavior decreases with the number of channels. The misbehaving terminal achieves around 3.4 times the throughput of well-behaved terminals when 2 channels are available. The throughput ratio decreases to 1.6 when 5 channels are available. The throughput ratio decrease is due to the throughput increase of well-behaved terminals when more channels are available. Independent of the number of channels, the misbehaving terminal is able to isolate a single channel, leaving the remaining channels free for well-behaved terminals. Contention in the remaining channels decreases with the number of available channels.

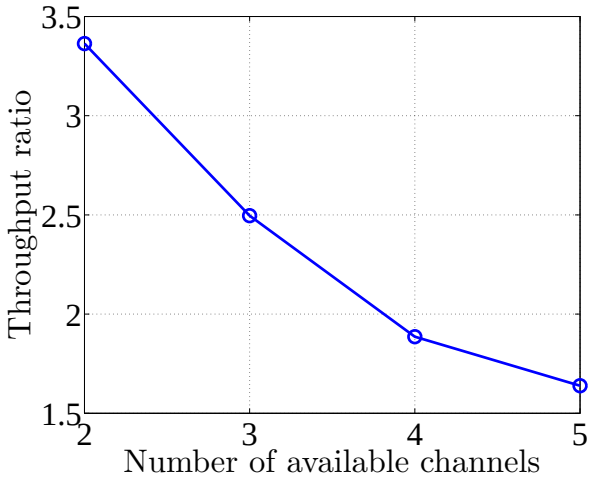


Figure 3.21: The ratio of misbehaving flow's throughput to well-behaved per-flow throughput when 2, 3, 4, and 5 channels are available under SP-MMAC.

Impact of MRA on DCC-MMAC

In this set of experiments, we evaluated the impact of the MRA combined with a BMA on DCC-MMAC. We considered a misbehaving terminal M aiming at isolating one of the two available data channels when contending with 10 well-behaved flows. To enable M launch the MRA as described in Section 3.3.2, we modified the DCC-MMAC implemented in Section 3.6.2 to allow reservations to be placed for channels while they are still occupied. Terminal M continuously placed reservations for a given channel to isolate it and eliminate contention. M fixed its CW to four. Figure 3.22 shows the average throughput achieved by M compared with the average per-flow throughput of well-behaved terminals and the average per-flow throughput in the absence of misbehavior. We observe that in high load conditions, the M achieve an eight-fold increase in throughput compared to well-behaved terminals. When combining the MRA with a BMA, terminal M is able to use one data channel without interruption for most of the time. The misbehaving terminal achieves the highest throughput gain among all evaluated scenarios.

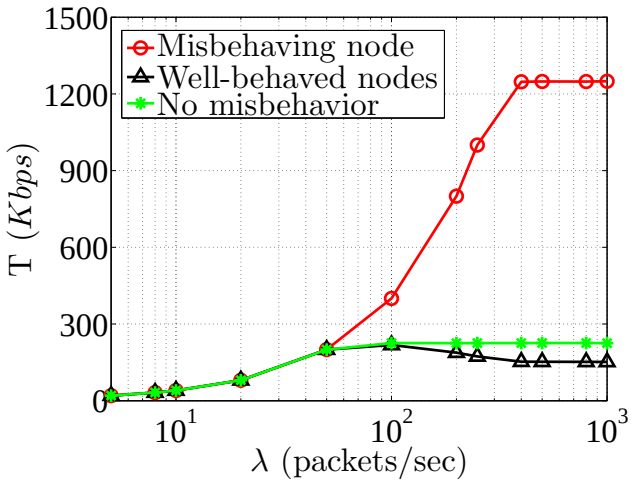


Figure 3.22: Throughput as a function of the packet arrival rate when one terminal launches an MRA together with a BMA in DCC-MMAC.

3.6.4 Evaluation of the Adaptive Misbehavior in SP-MMAC

In this set of experiments, we evaluated the adaptive reservation strategy proposed in Section 3.3.3. The misbehaving terminal placed multiple reservations adaptively, depending on the reservations of other contending pairs to gain exclusive use of a single channel. Figure 3.23(a) 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 terminal reduces because well-behaved terminals 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 terminal faces higher levels of contention.

Furthermore, we validated the analytical result obtained in Proposition 2 via simulations. Figure 3.23(b) 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 Figure 3.23(b), 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 terminal successfully places the required reservations within the first few slots.

3.6.5 Mitigation of Terminal Misbehavior in MMAC

Mitigation of SP-MMAC Misbehavior

In this set of experiments, we evaluated the effectiveness of our mitigation methods proposed for SP-MMAC misbehavior. Figure 3.24(a) shows the average throughput

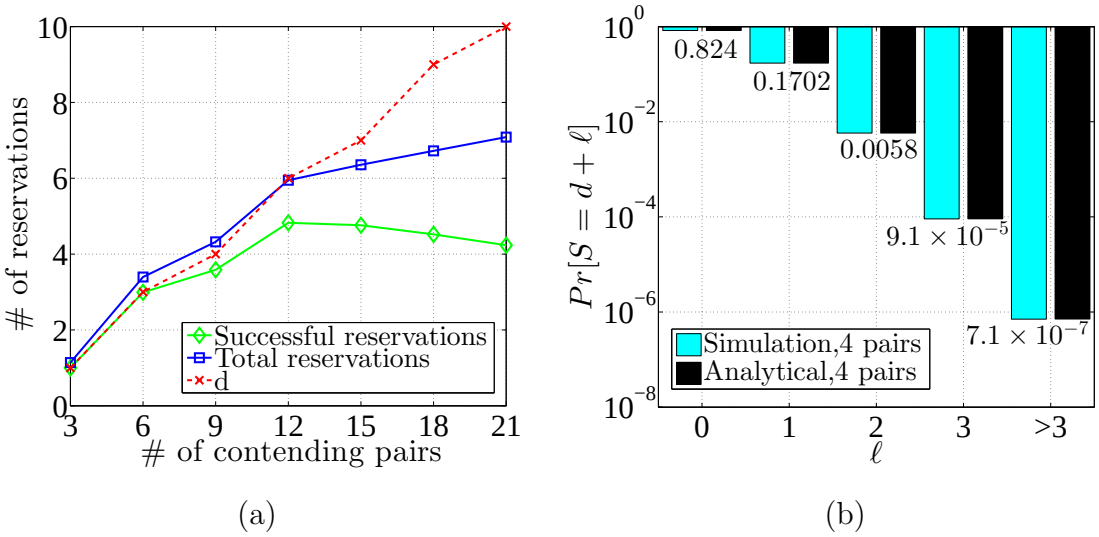


Figure 3.23: (a) Number of reservations needed to isolate a single channel as a function of the number of contending pairs, (b) pmf of the total number of reservation attempts for isolating a single channel (theoretical and simulation), as a function of ℓ.

of the misbehaving terminal and the average per-flow throughput of well-behaved terminals under the modified PCL update rules listed in Section 3.4.2. 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 terminal 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 3.4.1.

Moreover, the effect of a BMA is practically eliminated if we consider a control phase with longer duration. In Figure 3.24(b), we show the throughput of the misbehaving terminal for a control phase of 30ms. With the application of the modified PCL rules, the throughput of the misbehaving terminal becomes equal to that of well-behaved ones, even if the misbehaving terminal 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.

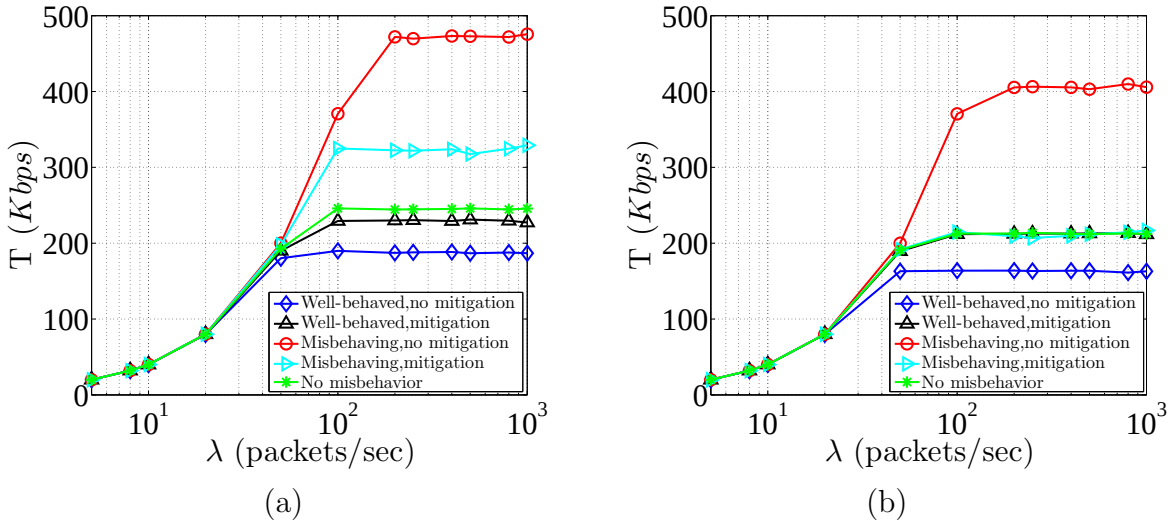


Figure 3.24: (a) Throughput as a function of the packet arrival rate for the misbehaving and well-behaved terminals, under the modified PCL rules in SP-MMAC, (b) throughput as a function of the packet arrival rate under the modified PCL rules in SP-MMAC, for a control phase duration equal to 30ms.

Mitigation of DCC-MMAC Misbehavior

In this set of experiments, we evaluated the effectiveness of the mitigation methods proposed for DCC-MMAC for the misbehavior scenario evaluated in Section 3.6.3. Figure 3.25 shows the average throughput of the misbehaving terminal M and the average per-flow throughput of well-behaved terminals under the modified operating rules presented in Section 3.4.3. We observe that M's throughput drops by 400Kbps in high load scenarios, while the throughput of well-behaved terminals increases by 50Kbps per flow. When M complies with the modified rules and does not attempt to seize a channel before time T_r has passed, other pairs are able to reserve the channel used by M. This forces M to wait until the end of current data transmission to initiate another channel reservation. As a result, M's advantage is significantly

reduced. M's compliance towards the modified rules is guaranteed by the multi-reservation monitoring module.

Similar to Figure 3.24(a), M still gains a throughput advantage due to the BMA, but this advantage is also reduced when comparing to Figure 3.16. This is because under moderate or high load scenarios, M's residing channel will most likely be reserved by another pair before T_r and thus M is forced to negotiate and switch to another data channel. Remaining data channels may not be immediately available when M completes its current data transmissio. Therefore, M has to wait until at least one data channel becomes idle to attempt its next data transmission. The reduced throughput gap is due to the extra waiting period imposed at M. We also note that the throughput of well-behaved flows is slightly improved compared with the scenario where only a BMA attack is launched. This is because the control channel is better utilized under the modified rules.

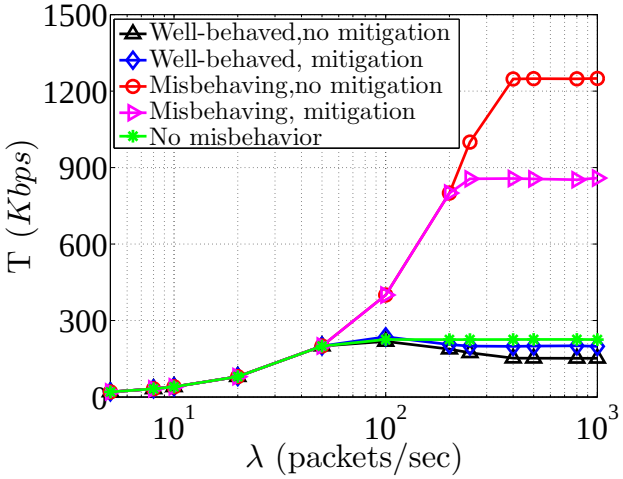


Figure 3.25: Throughput as a function of the packet arrival rate of the misbehaving and well-behaved terminals, under the modified operating rules in DCC-MMAC.

3.7 Chapter Summary

We addressed the problem of MAC layer misbehavior in multi-channel wireless networks. For MMACs following the split-phase and dedicated control channel designs,

we described possible misbehaving strategies that yield a significant throughput advantage to misbehaving terminals. We showed that misbehaving terminals 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 terminals. We further extended our misbehavior analysis to cognitive radio MAC protocols. We examined various vulnerabilities of existing CR-MAC protocols exploited by selfish/malicious CR users. We then discussed possible countermeasures for detecting and mitigating these vulnerabilities. Finally, we verified the effectiveness of our mitigation methods via extensive packet-level simulations and showed that the throughput of misbehaving terminals is equalized to the throughput of well-behaved terminals.

CHAPTER 4

SPECTRALLY-EFFICIENT MULTI-CHANNEL MEDIUM ACCESS WITHOUT CONTROL CHANNELS

4.1 Introduction

4.1.1 Motivation

Besides misbehavior, medium access in multi-channel networks is also vulnerable to other types of attacks. This is because in order to coordinate parallel transmissions across channels without interference, most existing MMAC protocols negotiate channel assignment over a default control channel [8, 10, 16, 19, 20, 22, 23], which in turn constitutes a single point of failure. If the adversary is successful, transmissions will be prevented on the entire available spectrum even if other frequency bands are still operational. One of the most effective ways for denying access to the control channel is by jamming it. In multi-channel wireless networks without centralized control, control channel jamming is particularly devastating due to their cooperative nature. On the other hand, the use of default control channels (either in-band or out-of-band) decreases spectrum efficiency as no actual data transmissions can take place on control channels. In certain scenarios, the data channels could become congested while the control channels remain underutilized.

To improve the spectral efficiency and jamming resilience of MMAC protocols, we design an MMAC protocol and eliminate the use of control channels by exploiting recent advances in full duplex (FD) communications over a *single* channel [113–116]. In certain low-power wireless environments, sophisticated self interference suppression (SIS) techniques allow for concurrent transmission and reception over a single channel. This is achieved by suppressing a significant portion of the self interference (up to 110 dB) [116], using a combination of antenna-based SIS [115], signal inver-

sion [114], and RF/digital interference cancelation [117,118]. The integration of FD communications in the MMAC design provides unique opportunities for reducing the control overhead, increasing the spatial channel reuse, and improving resilience to jamming.

4.1.2 Main Contributions and Chapter Organization

We design an MMAC protocol called FD-MMAC that coordinates multi-channel access in a distributed fashion. Compared with prior MMAC designs, FD-MMAC exhibits the following attractive features.

- It improves spectral efficiency by reducing the in-band and out-of-band control signaling for combating the multi-channel hidden terminal problem, discovering the resident channel of destinations, and performing load balancing.
- It increases the spatial channel reuse by enabling the operation of multi-channel exposed terminals.
- It achieves load balancing and fairness autonomously.
- It is less vulnerable to DoS attacks launched against the control channel [55, 76], due to the elimination of the use of default control channels.

To achieve these goals, we integrate an advanced suite of PHY-layer techniques, including self interference suppression, error vector magnitude and received power measurements, and signal correlation. We theoretically analyze the saturation throughput of FD-MMAC and verify our analysis via extensive simulations. Our results show that FD-MMAC achieves significantly higher throughput compared to prior art.

Chapter Organization: The remainder of this chapter is organized as follows. In Section 4.2, we describe the system model. Section 4.3 details the FD carrier sensing operation. In Section 4.4, we address the multi-channel hidden and exposed terminal problems. In Section 4.5, we present the operational details of FD-MMAC. In Section 4.6, we analytically evaluate the saturation throughput of FD-MMAC.

We compare the performance of FD-MMAC with existing MMAC designs in Section 4.7 and conclude the chapter in Section 4.7.3.

4.2 System Model

Network model

We consider a wireless network that operates over N orthogonal channels, denoted by $\mathcal{F} = \{f_1, f_2, ..., f_N\}$. For simplicity, we assume that all channels have the same bandwidth and propagation characteristics. Terminals are equipped with a single radio transceiver and are assumed to be time-synchronized to a common slotted system. Time synchronization can be achieved using out-of-band solutions such as GPS [119], or any of the readily available in-band methods [120]. We note that time-slotted synchronization is not a necessary FD-MMAC requirement. It is assumed here to facilitate legacy operations used by FD-MMAC, such as the slotted CSMA algorithm. However, FD-MMAC can also operate in an asynchronous mode.

FD Communications and SIS

Terminals can operate in single channel FD mode as illustrated in Figure 4.1, where *a terminal can receive while simultaneously transmitting over the same channel.* In the depicted scenario, terminal 1 and terminal 2 are transmitting signals S_1 and S_2 over the same channel simultaneously. While receiving its desired signal S_2, terminal 1 is also receiving a version of S_1 which is considered as self-interference. As a result, the actual signal received by terminal 1 is the superposition of S_2 and self-interference. The self-interference is usually millions of times stronger than S_2. To enable terminal 1 receiving S_2 correctly while transmitting S_1, the significant self-interference that results from its own transmission must be canceled. This can be achieved by applying a combination of analog and digital SIS techniques [113–116].

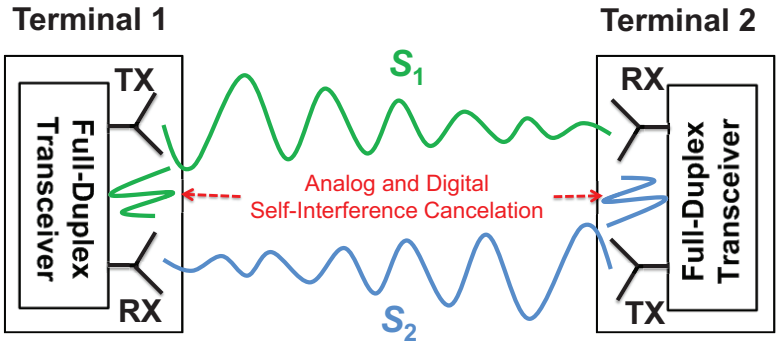


Figure 4.1: Two terminals communicate in single channel FD mode by applying SIS techniques.

Signal Correlation

Terminals apply signal correlation techniques for detecting the transmission of known bit patterns. These techniques are common in frame detection, even in the presence of collisions [118]. The concept of signal correlation is shown in Figure 4.2. Consider the concurrent reception of frames P_A and P_B at C. Terminal C is interested in detecting whether $P_B = P$, where P is a known bit pattern. Let the sampled signal representing P be $\mathcal{L}$ samples long. C computes the signal correlation between $P_A + P_B + w$ and P (w denotes the noise component at the receiver) by aligning the $\mathcal{L}$ samples of P with the first $\mathcal{L}$ samples of $P_A + P_B + w$. It then shifts the alignment of P by one sample and recomputes the correlation until the end of $P_A + P_B + w$. Formally, let $x[i]$ denote the i^{th} sample of P and $y[j]$ the j^{th} sample of the received signal. The correlation at the j^{th} position of $y[j]$ is:

$$C[j] = \sum_{i=1}^{\mathcal{L}} x^*[i]y[j+i], \tag{4.1}$$

where $x^*[i]$ is the complex conjugate of $x[i]$. The correlation value peaks when P is aligned with P_B. Using this method, C can identify if P_B is transmitted, despite the concurrent transmission of P_A. In practice, C must compensate $C[j]$ for the frequency offset of B. The frequency offset can be estimated in advance from prior frame exchanges between B and C. One limitation of the signal correlation method

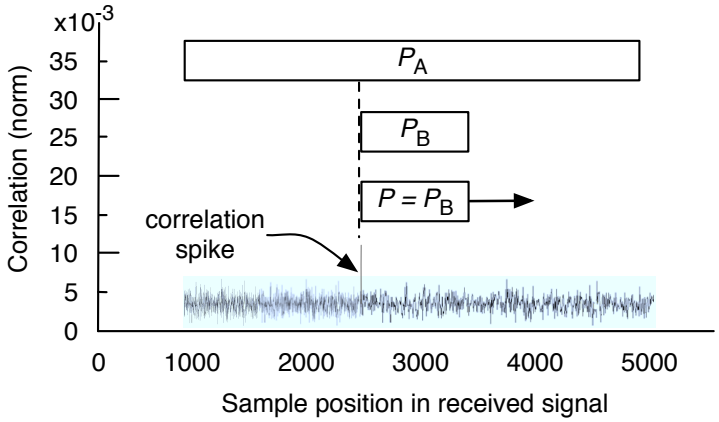


Figure 4.2: Detecting a known bit pattern P when two frames collide using the signal correlation technique.

is that P has to exhibit desirable cross-correlation properties.

4.3 FD Carrier Sensing

To combat multi-channel hidden terminals, we extend the physical carrier sensing function to the receiver's collision domain by operating the receiver in FD mode. We refer to this mechanism as *FD carrier sensing.* In FD-MMAC, we improve FD carrier sensing by integrating a suite of PHY-layer techniques. Our techniques extend beyond the estimation of the carrier state (idle or busy) to determining a terminal's operational state relative to an ongoing transmission. The state information is used to create transmission opportunities for exposed terminals, avoid collisions caused by hidden terminals, and discover the resident channel of a destination.

4.3.1 Operation in FD Mode

An example of the FD operation is shown in Figure 4.3. Sender A initiates the transmission of P_A to B. Terminal B decodes the MAC header of P_A and determines it is the destination, upon which it transmits a *beacon frame* BCN_B while receiving P_A. This mechanism was demonstrated in [121] for a single channel MAC. Terminal A receives BCN_B by also operating in FD mode. Upon receiving BCN_B, terminal A verifies that B is receiving P_A and continues the transmission of P_A. Lack of

a BCN reply implies that either B is unavailable or that the MAC header of P_A got corrupted. The sender uses the lack of a BCN as an early collision detection mechanism and aborts further transmission of the data frame.

Generally, a data frame P is expected to be longer than a BCN frame. To account for this difference, BCNs are transmitted back-to-back until the reception of P is completed. The reception ending time t_e is known to the destination based on the network allocation vector (NAV) included in P's MAC header. The BCN contains the destination's id, the time slot t_{ACK} at which the ACK transmission is to be completed, and a CRC code. If the reception of P is successful, the destination replies with an acknowledgement (ACK).

The use of BCN frames for performing virtual carrier sensing is similar to the well-known busy-tone approach proposed to address the hidden terminal problem in single-channel CSMA-based networks [30, 122–124]. In a busy tone based MAC protocol, a receiver transmits a busy tone signal on a separate narrowband channel while receiving a data frame on the data channel. The busy tone informs all terminals around the receiver about the ongoing frame reception, thus avoiding collisions due to the operation of hidden terminals. This approach can be directly extended to a multi-channel setup by associating each data channel with one busy-tone channel. However, this lowers the spectral efficiency, as the number of required busy tone channels grow linearly with the number of data channels. Moreover, busy tones do not convey any additional information about about an ongoing transmission. The use of in-band BCN frames in the FD-MMAC protocols allows nearby terminals to evaluate their location relative to an ongoing transmission, and potentially operate as exposed terminals.

Finally, busy-tone based MAC protocols are particularly susceptible to DoS attacks. The narrowband channels used to convey the state of the data channels can be easily jammed, indicating that the corresponding data channels are occupied. The jammer can deny communications by focusing his energy only on a small portion of the spectrum. On the other hand, FD-MMAC distributes the carrier sensing operation over all data channels. As a result, the jammer has to spread his energy

over the entire spectrum to potentially deny channel access.

4.3.2 Operation State Classification

To determine their operational state, terminals perform a region classification on their resident channel. We divide the collision domains of A and B to the three regions shown in Figure 4.3: (a) the receiver-only (RO) region, (b) the collision region (CO), and (c) the transmitter-only (TO) region. Referring to Figure 4.3, a terminal C can determine its region using the following rules.

1. If C can decode BCN_B, it infers that it is in the RO region (hidden terminal).
2. if C cannot decode the received signal due to the collision of P_A with BCN_B, it infers it is in the CO region.
3. If C can decode P_A, it infers that it is in the TO region (exposed terminal).

When located in the CO/RO regions, C defers from transmission to prevent a collision at B. Otherwise, C explores transmission opportunities as an exposed terminal.

4.3.3 Practical Issues

Several practical issues complicate the proposed region classification rules. First, when C is in the TO region (position C_1 in Figure 4.3), it cannot verify the correct decoding of P_A until P_A's transmission is completed and the CRC code is checked. Similarly, if C switches to a busy channel in the middle of P_A's transmission, the CRC code cannot be checked. To evaluate the decodability of P_A, terminal C computes the error vector magnitude (EVM) on the received symbols. The RMS EVM value (dB) is given by [125]:

$$EVM_{RMS}(dB) = 20\log\left(\sqrt{\frac{\frac{1}{n}\sum_{k=1}^{n}|\mathbf{s}[k]-\mathbf{r}[k]|^2}{\frac{1}{M}\sum_{i=1}^{M}|\mathbf{s}_i|^2}}\right), \tag{4.2}$$

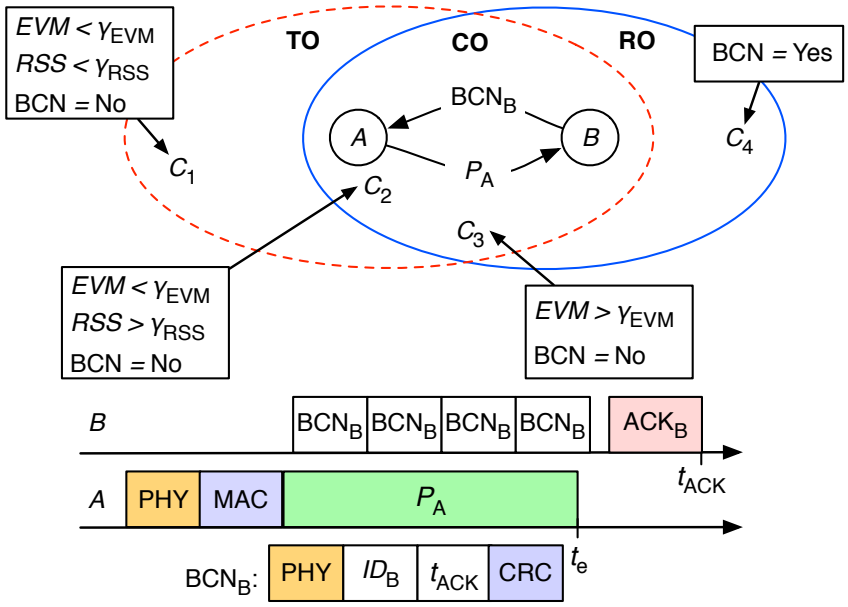


Figure 4.3: The three regions for a terminal C relative to a transmission $A \rightarrow B$.

where $\mathbf{s}[k]$ is the k^{th} transmitted symbol, $\mathbf{r}[k]$ is the k^{th} received symbol, n is the window size (in symbols) over which the EVM is computed, $\mathbf{s}_i$ is the i^{th} modulation symbol, and M is the modulation order. The EVM serves as a measure of the signal quality and is strongly correlated to the bit error rate [125]. Note that for arbitrary frames, the $\mathbf{s}[k]$'s *are not known to the receiver*. To compute an EVM estimate using formula (4.2), the receiver matches $\mathbf{s}[k]$ to the constellation symbol closest to $\mathbf{r}[k]$. We use this approach as it is expected that $\mathbf{r}[k]$'s will be closest to the actual transmitted $\mathbf{s}[k]$'s if a frame is correctly decoded. On the other hand, in a collision scenario, the distance between the closest $\mathbf{s}[k]$ and $\mathbf{r}[k]$ is expected to be large, yielding a larger EVM. Figure 4.4 demonstrates the computation of the EVM vector as a function of the received symbol $\mathbf{r}[k]$ and the closest symbol $\mathbf{s}[k]$ for the QPSK modulation scheme. Finally, we set the window size n equal to the duration of two BCN frames. This is to differentiate between the TO and the RO regions, where terminals are expected to have a lower EVM compared with the CO region. If terminal C is located in the RO region, it is likely to decode at least one BCN frame within two BCN frame durations (recall that C can switch to a busy

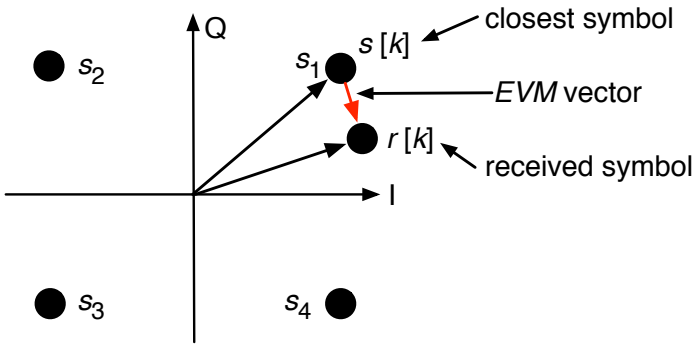


Figure 4.4: EVM vector computation for QPSK modulation.

channel at any time). Otherwise, if C is in the TO region, it will not decode a BCN frame, but will have a low EVM value. Therefore, a terminal switching to a busy channel has to attempt decoding for two BCN durations to determine if a BCN is decodable (first classification rule). We utilize this time to measure the EVM more accurately and compare it with a threshold γ_{EVM}.

Note that the third classification rule could also be satisfied due to the capture effect [126]. When C is in the CO region but very close to A (position C_2 in Figure 4.3), it can measure low EVM values. In this case, C will assume that it is in the TO region and could cause a collision at B. To prevent this scenario, we incorporate received signal strength (RSS) measurements. If the RSS at C is higher than a threshold γ_{RSS}, terminal C concludes that it is in the CO region and defers from transmission, despite measuring a low EVM. Finally, if C is in the CO region, but can decode the BCN due to its proximity to B, we allow C to falsely infer that it is in the RO region. This is because C defers from transmission, whether inside the CO or the RO region.

The computation of the EVM and RSS could also be affected by the presence of noise and nearby interference sources. Such interference could increase the EVM and RSS values computed by a terminal performing region classification. In this circumstance, the terminal determines itself to be in the CO region (C_3 location) and refrains from transmission. This is the right decision for the terminal as the chances

of a successful parallel transmission in the presence of strong interference are low (the transmission will likely fail when interference is high). Under such conditions, the terminal switches to another channel to seek other transmission opportunities. The region classification rules used by FD-MMAC are summarized in Table 4.1. In Section 4.7, we perform testbed experiments to determine γ_{EVM} and γ_{RSS}, based on measurements at locations C_1–C_4.

Table 4.1: Region classification rules

	BCN	$EVM < \gamma_{EVM}$	$RSS < \gamma_{RSS}$	Region
C_1	No	Yes	Yes	TO
C_2	No	Yes	No	CO
C_3	No	No	-	CO
C_4	Yes	-	-	RO

4.4 Combating Hidden/Exposed Terminals

In this section, we show how FD carrier sensing addresses the multi-channel hidden and exposed terminal problems. Consider the frame exchange sequence illustrated in Figure 4.5(a), for the topology of Figure 4.3 (C is a hidden terminal to A). Terminal A transmits P_A to B over f_1 at time t_0. Terminal B decodes the PHY and MAC headers and infers that it is the destination. Terminal B replies with BCN_B that is repeated for the duration of P_A, which terminates at t_1. Terminal C switches to f_1 at t_2 with $t_0 < t_2 < t_1$. First, C senses f_1 to be busy due to the BCN_B transmissions. Second, C decodes BCN_B and infers it is in the RO region. Therefore, it defers from transmission.

4.4.1 Early Collision Detection

A collision due to hidden terminals is still possible during the transmission of the PHY and MAC headers of P. In a collision scenario, the destination is unable to decode the MAC header and therefore, does not reply with a BCN. If the sender

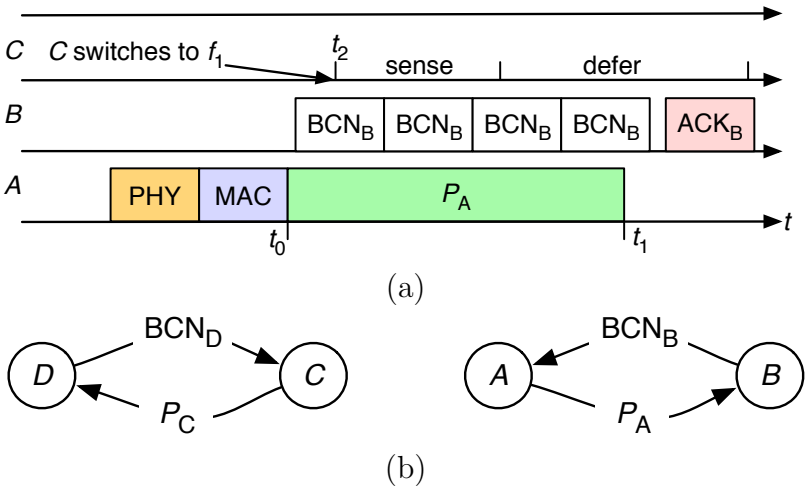


Figure 4.5: (a) Combating the multi-channel hidden terminal problem, (b) exposed terminal operation. Transmission $C \rightarrow D$ occurs in parallel with transmission $A \rightarrow B$ on the same channel.

does not receive a BCN reply, it assumes that P has collided or the destination is unavailable. The sender aborts further transmission of P without waiting for the expiration of the ACK timer.

4.4.2 Enabling Exposed Terminal Transmissions

An exposed terminal C located in the TO region of an ongoing transmission $A \rightarrow B$ could attempt to communicate P_C to a candidate destination D. If D can decode the MAC header of P_C, it will respond with BCN_D by operating in FD mode. Terminal C will continue the transmission of P_C if it detects BCN_D, and will abort otherwise. The destination D will not be able to respond with BCN_D if one of the following occurs: (a) D is in the collision domain of another transmission and hence, cannot decode the MAC header of P_C or, (b) D resides on another channel. The exposed terminal operation for transmissions $A \rightarrow B$ and $C \rightarrow D$ is shown in Figure 4.5(b).

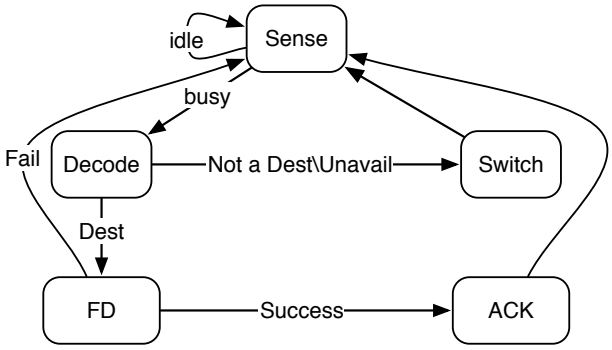
4.4.3 Receiving BCNs/ACKs in the Presence of Exposed Terminals

Exposed terminal transmissions may prevent the correct decoding of BCNs and ACKs. In the example of Figure 4.5(b), terminals A and C cannot decode BCN_B and BCN_D, respectively, due to mutual interference. Similarly, terminals A and C cannot decode ACK_B and ACK_D, respectively, due to the interfering transmissions of P_C and P_A. To enable the parallel operation of $A \rightarrow B$ and $C \rightarrow D$, terminals detect BCNs and ACKs using signal correlation [118].

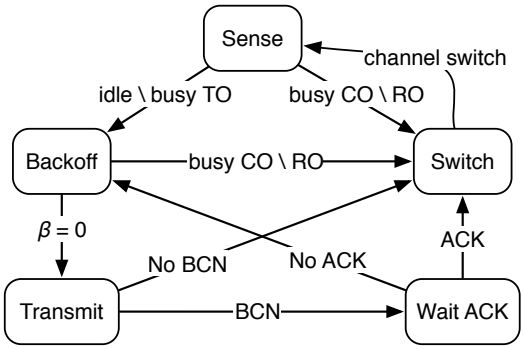
Terminal C applies signal correlation to detect BCN_D and ACK_D in the presence of P_A. Similarly, terminal A applies signal correlation to detect BCN_B and ACK_B when P_C is concurrently transmitted. Note that a sender is aware of the exact bit pattern of the BCN and ACK frames based on the data frame it has transmitted. Moreover, the sender is aware of the approximate time that a BCN (or ACK) is expected, based on the data frame transmission time. Hence, it can limit the signal correlation within only a few sample shifts. One limitation of the signal correlation is that frames have to exhibit low cross-correlation. To satisfy this condition, BCNs and ACKs are hashed (except the PHY header) with a uniform hash function to produce a random but known output.

4.5 The FD-MMAC Protocol

We design FD-MMAC as a time-slotted protocol based on CSMA/CA. To improve spectral efficiency, FD-MMAC eliminates the message overhead associated with virtual carrier sensing. Moreover, to mitigate DoS attacks against the control channel, destination discovery and channel assignment are performed independently by senders and destinations, without converging to a common channel. The key idea behind FD-MMAC is for destinations to switch to an idle channel as soon as their resident channel becomes busy. This makes them available to receive transmissions from senders while distributing traffic across all channels. We now present the operational details of FD-MMAC. The destination and sender state diagrams are shown in Figure 4.6.



(a) The state diagram of an FD-MMAC destination.



(b) The state diagram of an FD-MMAC sender.

Figure 4.6: Operational details of FD-MMAC protocol.

4.5.1 Destination Operation

When a terminal's transmission queue is empty, it operates as a destination. A destination selects a resident channel such that it can be discovered by candidate senders. Referring to the state diagram of Figure 4.6(a), a destination transitions between the following states.

Sense state: In the "Sense" state, the destination continuously senses the resident channel. If the resident channel becomes busy, the destination transitions to the "Decode" state.

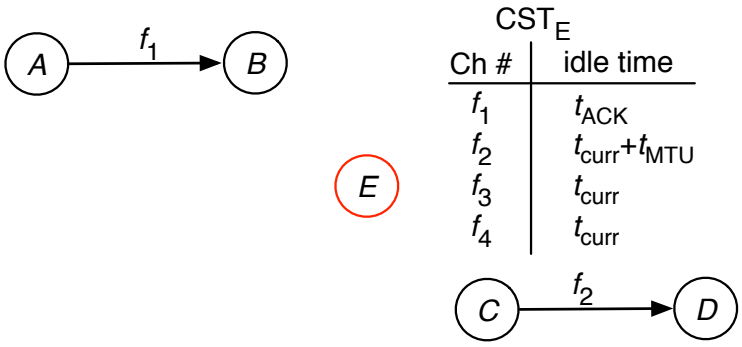


Figure 4.7: The CST table for terminal E.

Decode state: In the "Decode" state, the destination attempts to decode the received signal. It transitions to the "FD" state if it is the intended destination and available for reception. Otherwise, it transitions to the "Switch" state.

FD state: In the "FD" state, the destination operates in FD mode. Based on the MAC header of the frame P that is being received, the destination determines the t_{ACK} and the number of BCNs that need to be successively transmitted until the reception of P is completed. Then, it transmits BCNs while receiving P. The destination checks the CRC code of P. If P is successfully received, it transitions to the "ACK" state. Otherwise, it returns to the "Sense" state.

ACK state: After a successful frame reception, the destination replies with an ACK and returns to the "Sense" state.

Switch state: In the "Switch" state, the destination autonomously determines its resident channel. This decision is based on a *channel state table (CST)* that records the expected time that each channel becomes idle (*idle time*). The CST is updated according to the following rules:

1. If the resident channel f_i is idle, update the idle time for f_i to the current slot t_{curr}.

2. If the resident channel f_i is busy and the destination is in the RO region (BCN is decodable), update the idle time for f_i to t_{ACK} (contained in the BCN).

3. If the resident channel f_i is busy and the destination is in the CO or TO regions (BCN not decodable), update the idle time for f_i to $t_{curr} + T_{MTU}$, where T_{MTU} is the transmission duration of the maximum transmission unit (MTU) plus the corresponding ACK.

After the CST update, the destination switches to the channel with the earliest idle time. If several channels are tied, the destination selects the next channel according to a channel priority list. This list could be a simple channel ordering rule (e.g., based on channel index). Based on the channel priority list, candidate destinations always select the highest-priority channel and remain on that channel until the channel priority changes. When residing on an idle channel, this can only occur if the channel becomes occupied. In this case, idle destinations switch to the next channel in the channel priority list and update the CST based on their individual sensing results.

The proposed switching mechanism achieves several desirable properties. First, senders and destinations switch following the same rules, thus facilitating destination discovery. Second, load balancing is indirectly achieved, as idle destinations avoid busy channels. Both properties are achieved without exchanging control messages.

As an example, consider the topology of Figure 4.7. Assume that destination E resides on f_1. Initially, E sets the idle time for all channels to t_{curr}. When the $A \rightarrow B$ transmission occupies f_1, terminal E decodes BCN_B because it is a hidden terminal to A. E updates the idle time for f_1 to t_{ACK} and switches to f_2, because f_2 has the lowest index among the channels with the earliest idle time. Assume that transmission $C \rightarrow D$ is ongoing on f_2 when E switches to f_2. E cannot decode BCN_D because it is in the TO region. Terminal E uses the the worst-case estimate for the idle time of f_2 and sets the idle time to $t_{curr} + T_{MTU}$. It then switches to f_3 which is currently idle. From our example, it becomes evident that the information stored in the CST does not reflect the true channel state for all channels. This is because destinations do not sense the state of a channel unless switching to it. Despite the inaccuracy of the CST, destinations quickly discover idle channels due to the low delay overhead of the physical carrier sensing operation.

4.5.2 Sender Operation

For the sender, we adapt the CSMA backoff mechanism to the multi-channel environment. A sender in backoff state retains his selected backoff value when switching channels and continues the countdown once it reaches an idle channel. When the backoff counter reaches zero, the sender maintains this value until it discovers the destination. This implements a global contention mechanism that extends to all channels. Referring to Figure 4.6(b), a sender operates as follows.

Sense state: In the "Sense" state, the sender senses its resident channel f_i. If f_i is idle, it transitions to the "Backoff" state. If f_i is busy, it classifies its operation state using the region classification rules of Section 4.3. If the sender is in the TO region (exposed terminal), it transitions to the "Backoff" state. Otherwise, it transitions to the "Switch" state.

Backoff state: In the "Backoff" state, the sender selects a backoff value β for a frame P, by using the following rules:

1. In the first transition to the "Backoff" state for P, the sender draws β uniformly from $[0, cw_0]$, where cw_0 is the minimum contention window (CW).

2. In any following transition from the "Sense" state to the "Backoff" state, the sender retains the current β value (backoff is resumed from the current value).

3. In a transition from the "Wait ACK" state to the "Backoff" state, the sender doubles the CW and draws β uniformly. The CW is capped at cw_m.

In the "Backoff" state, the sender decrements β by one unit with every idle slot. Here, a slot is assumed to be idle if: (a) no channel activity is detected, or (b) the channel is busy but the sender is in the TO region. When $\beta = 0$, the sender transitions to the "Transmit" state. If the channel becomes busy before $\beta = 0$ (and the sender is not in the TO region), the sender transitions to the "Switch" state and freezes β.

Transmit state: In the "Transmit" state, the sender initiates the transmission of P. If the destination responds with a BCN, the sender continues the transmission

of P. With the completion of P's transmission, the sender transitions to the 'Wait ACK" state. If a BCN is not detected, the sender aborts the transmission of P and transitions to the "Switch" state.

Wait ACK state: With the completion of P's transmission, the sender waits for an ACK by the destination. The sender transitions to the "Backoff" state if an ACK is not received by the expiration of the ACK timer, without transitioning to the "Switch" state. This is because the sender is aware that the destination resides on the current channel due to the reception of the BCN during the "Transmit" state. If the ACK reception is successful, the sender transitions to the "Switch" state.

Switch state: In the "Switch" state, the sender performs two operations. First, it updates the CST information and second, it decides on the next channel using the same switching rules as the destination. The CST is updated using the following rules:

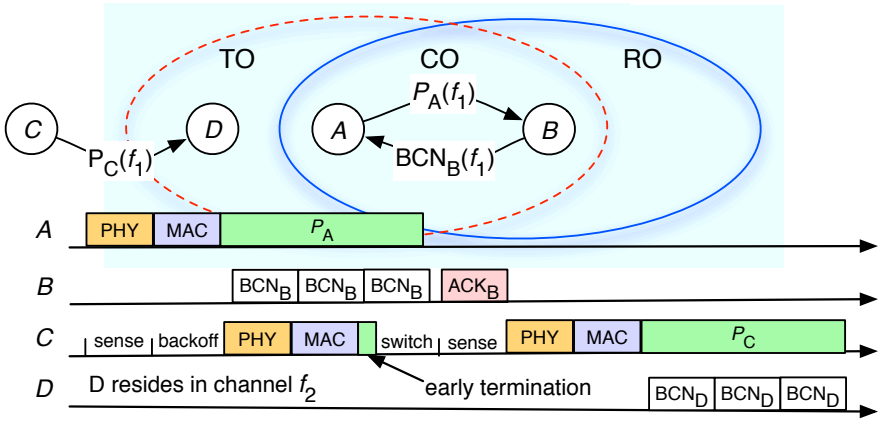
1. If the sender is in the RO region of a transmission on f_i (BCN is decodable), it sets the idle time of f_i to t_{ACK}

2. If the sender is in the CO/TO region of a transmission on f_i (BCN is not decodable), it sets the idle time of f_i to $t_{curr} + T_{MTU}$.

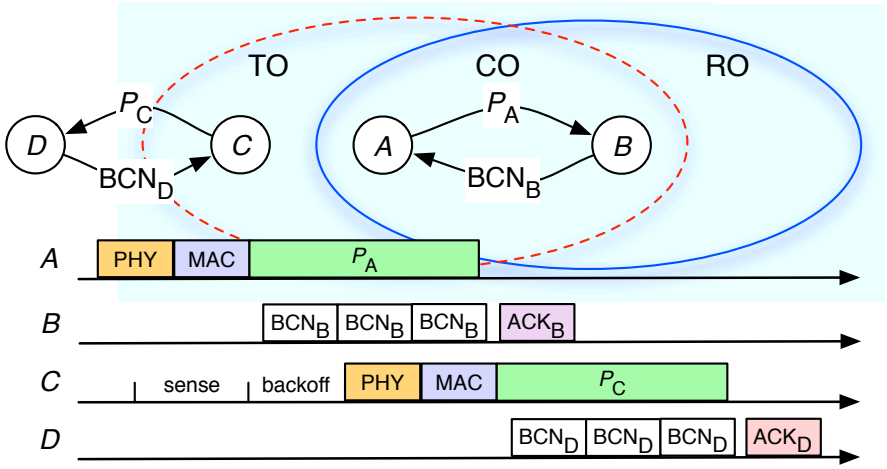
3. If a sender transmitted a frame P on f_i, but did not receive a BCN response, it sets the idle time of f_i to $t_{curr} + T_{MTU}$. This update leads to a channel switch to continue the destination discovery process.

4.5.3 FD-MMAC Operational Examples

To ease the understanding of the FD-MMAC protocol, we present two operational examples for the topology of Figure 4.8. These examples demonstrate the destination discovery process performed by senders, the channel switching operation of destinations, and the operation of exposed terminals.



(a) Sender/Destination operation.



(b) Exposed terminal operation.

Figure 4.8: Two operational examples of FD-MMAC.

Sender/Destination Operation

In the example of Figure 4.8(a), we demonstrate the destination discovery performed by a sender. Initially, terminals A, B, C, and D reside on channel f_1. Terminal A initiates a data frame transmission to terminal B, which replies with BCN_B. Terminal D detects that f_1 is busy and switches to f_2, in order to be available for reception. Terminal C, who resides on f_1, has a frame P_C for D. Operating according to the sender state diagram of Figure 6(c), terminal C transitions to the "Sense" state. Since f_1 is idle from C's perspective, C transitions to the "Backoff"

state and uniformly selects a β from $[0, cw_0 - 1]$. When $\beta = 0$, terminal C transitions to the "Transmit" state and initiates the transmission of P_C. Because D resides on f_2, C does not receive BCN_D. Terminal C aborts further transmission of P_C and transitions to the "Switch" state. In this state, it updates the idle time for f_1 to $t_{curr} + T_{MTU}$ in the CST and switches to f_2, which has the lowest index among the idle channels. Once in f_2, terminal C transitions to the "Sense" state and senses f_2 to be idle. It then transitions to the "Backoff" state for a second time and retains $\beta = 0$. Terminal C retransmits P_C and completes the communication with D.

Exposed Terminal Operation

In the example of Figure 4.8(b), we demonstrate the operation of an exposed terminal. Terminal C has a data frame for terminal D while being in the TO region of the $A \rightarrow B$ transmission. While in the "Sense" state, terminal C determines that it is in the TO region by measuring a low EVM value and a low RSS value. Terminal C can therefore operate as an exposed terminal. Terminal C transitions to the "Backoff" state, selects β uniformly from $[0, cw_0 - 1]$, and initiates the backoff countdown. When $\beta = 0$, terminal C transitions to the "Transmit" state, and transmits data frame P_C. Because D is currently idle, it replies with BCN_D. Terminal C detects BCN_D using the signal correlation method and continues with the transmission of P_C. For the $A \rightarrow B$ communication, upon termination of the P_A transmission, terminal B transmits ACK_B. The acknowledgment ACK_B is detected at A using the signal correlation technique. Note that ACK_B is not decodable due to the concurrent transmission of P_C from C. Upon termination of the P_C transmission, D replies with ACK_D which is decodable at C, because the transmission of P_A is already completed.

4.6 Throughput Analysis of FD-MMAC

In this section, we analytically evaluate the *saturation* throughput of FD-MMAC using a three-dimensional discrete-time Markov model. We follow similar formula-

tions and assumptions to those proposed for single-channel [127] and multi-channel MACs [128]. Consider M senders within the same collision domain, contending over N channels. The senders are always backlogged. We model the state of a single sender, referred to as the *tagged sender*, using three discrete-time stochastic processes $\{F_n, S_n, B_n\}$. Here, F_n represents the sender's resident channel index (1,2,...,N), S_n represents the backoff stage, with $S_n \in [0, m]$, and B_n represents the sender's backoff counter, with $B_n \in [0, 2^m cw_0 - 1]$.

Stochastic processes F_n and B_n (and as a result, S_n) are non-Markovian, as they depend on the channel history and transmission history of the sender. To ease our analysis, we assume that a sender switches to a channel f_i with fixed probability $p(f_i)$, which is independent of the current resident channel. Moreover, we approximate the probability of attempting a transmission at slot n with a *constant* probability p_{tr}, referred to as the *transmission probability* [127]. These two approximations become more accurate with the increase of n and if an equal number of senders contend on every channel. We later verify that FD-MMAC tends to uniformly distribute sender-destination pairs on all available channels (see Section 4.7). Finally, we denote by $p_d(f_i)$ the probability of discovering a destination on f_i and approximate the number of senders contending on a channel by $\frac{M}{N}$. Under independent F_n and B_n, we can model the three-dimensional process $\{F_n, S_n, B_n\}$ as a discrete-time Markov chain, with one-step transition probability from state $\langle u_1, k_1, \beta_1 \rangle$ to state $\langle u_2, k_2, \beta_2 \rangle$ as:

$$p_{(u_2,k_2,\beta_2|u_1,k_1,\beta_1)} = P\{F_{n+1} = u_2, S_{n+1} = k_2, B_{n+1} = \beta_2 | F_n = u_1, S_n = k_1, B_n = \beta_1\}. \tag{4.3}$$

For the tagged sender, a slot n for which he defers from transmission can be: (a) idle, if no other sender transmits during slot n, (b) successful, if exactly one other sender transmits during that slot, and (c) collision, if more than one of the remaining senders attempt to transmit during slot n. We denote the probabilities of an idle, successful, and collision slot by p_I, p_S, and p_C, respectively. Given that each

sender transmits during a slot independently with probability p_{tr}, the slot events occur with probability:

$$p_I = (1-p_{tr})^{\frac{M}{N}-1}, \quad p_S = (\frac{M}{N}-1)p_{tr}(1-p_{tr})^{\frac{M}{N}-2}, \quad p_C = 1-p_I-p_S. \tag{4.4}$$

Based on the FD-MMAC state diagram of Figure 4.6(b), there are four non-zero one-step transition probabilities:

1) The tagged sender is at state $\langle u,k,\beta\rangle$, with $\beta \geq 1$ and the current slot is idle. In this case, the tagged sender decrements the backoff counter by one and transitions to state $\langle u,k,\beta-1\rangle$. This occurs with probability:

$$p_{(u,k,\beta-1|u,k,\beta)} = p_I, \quad 1 \leq u \leq N,\ 0 \leq k \leq m,\ 1 \leq \beta < cw_k. \tag{4.5}$$

2) The tagged sender is at state $\langle u,k,\beta\rangle$ with $\beta \geq 1$ and the current slot is busy. The channel could be busy due to the successful transmission of another sender (with probability p_S), or due to a collision (with probability p_C). In this case, the tagged sender freezes his counter and switches to channel $f_{u'}$ with probability $p(f_{u'})$. The state transition to $\langle u',k,\beta\rangle$ occurs with probability

$$\begin{aligned} &p_{(u',k,\beta|u,k,\beta)} = (p_S+p_C)\,p(f_{u'}), \\ &1 \leq u,u' \leq N,\ u \neq u',\ 0 \leq k \leq m,\ 1 \leq \beta < cw_k. \end{aligned} \tag{4.6}$$

3) The tagged sender transmits a data frame at state $\langle u,k,0\rangle$ and successfully detects a BCN reply. Once the transmission is completed, the sender transitions to state $\langle u,k,\beta\rangle$ by selecting a new backoff counter in $[0,cw_0)$ (each backoff value is selected with probability $\frac{1}{cw_0}$).

$$p_{(u,0,\beta|u,k,0)} = \frac{p_I \cdot p_d(f_u)}{cw_0}, \quad 1 \leq u \leq N,\ 0 \leq k \leq m,\ 0 \leq \beta < cw_0. \tag{4.7}$$

4) The tagged sender transmits a data frame at state $\langle u,k,0\rangle$, but does not detect a BCN reply. In this cace, the sender aborts the data transmission, switches to a new

channel $f_{u'}$, and sets his backoff counter to one. The transition to state $\langle u', k, 1\rangle$ occurs with probability

$$\begin{aligned} p_{(u',k,1|u,k,0)} &= p(f_{u'})\left(1 - p_d(f_u) + p_d(f_u)(p_S + p_C)\right) \\ &= p(f_{u'})\left(1 - p_d(f_u)p_I\right), \quad 1 \le u, u' \le N,\ 0 \le k \le m. \end{aligned} \tag{4.8}$$

Using the one-step transition probabilities, we derive the transition matrix $\mathbf{P}$ for the Markov model. Because the Markov model is three-dimensional, the matrix elements of $\mathbf{P}$ are also matrices, given as follows

$$\mathbf{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & N \end{array} \\ \begin{pmatrix} D^{11} & Z^{12} & Z^{13} & \cdots & Z^{1N} \\ Z^{21} & D^{22} & Z^{23} & \cdots & Z^{2N} \\ Z^{31} & Z^{32} & D^{33} & \cdots & Z^{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Z^{N1} & Z^{N2} & Z^{N3} & \cdots & D^{NN} \end{pmatrix} \end{array}, \tag{4.9}$$

where D^{ii} $(i = 1, 2, ..., N)$ and Z^{ij} $(1 \le i, j \le N,\ i \ne j)$ are matrices of dimensions $\sum_{i=0}^{m} 2^i \cdot cw_0 \times \sum_{i=0}^{m} 2^i \cdot cw_0$. In $\mathbf{P}$, the index of each row/column corresponds to the channel index. Thus, $\mathbf{P}$ has a total of N^2 matrix elements. Matrices D^{ii} $(i = 1, 2, ..., N)$ in the diagonal of $\mathbf{P}$ correspond to those state transitions for which the tagged sender does not switch channels. Matrices Z^{ij} $(1 \le i, j \le N,\ i \ne j)$ correspond to those state transitions for which the sender switches channels.

For the steady-state distribution π, it holds that $\pi\mathbf{P} = \pi$ and $\sum_{j \in S} \pi_j = 1$. This non-linear system can be numerically solved for p_{tr}, for known $p(f_i)$ and $p_d(f_i)$. In turn, knowledge of p_{tr} allows us to compute the aggregate system throughput by defining the following slot events for the entire system (not a tagged sender): (a) idle, if no sender transmits during slot n, (b) success, if exactly one sender transmits during slot n, (c) collision, if two or more senders attempt to transmit during slot n. Denoting the event probabilities for an idle, successful, and collision slot by p'_I, p'_S, p'_C, the network throughput is given by Proposition 4.

Proposition 4. *The aggregate FD-MMAC throughput for M terminals contending over N channels under saturation is:*

$$T = \sum_{u=1}^{N} \frac{p'_S \cdot p_d(f_u) \cdot payload}{E[\tau_{slot}(f_u)]}, \tag{4.10}$$

where $E[\tau_{slot}(f_u)]$ denotes the average slot duration for channel f_u (derivation will be given in the proof), and payload denotes the data frame length in bits.

Proof. In a single-hop network with N available channels, the aggregate network throughput can be computed by summing the throughput of individual channels. To obtain the throughput of a channel f_u, we first compute the probabilities for the following events. Let p'_I denote the idle slot probability if no senders attempt to transmit during slot n. Let also p'_S denote the probability of a successful transmission, if exactly one sender transmits during slot n. Finally, let p'_C denote the collision probability if two or more senders attempt to transmit during slot n. The probability for each event is given by:

$$\begin{aligned} p'_I &= (1 - p_{tr})^{\frac{M}{N}} \\ p'_S &= (\frac{M}{N}) p_{tr} (1 - p_{tr})^{\frac{M}{N} - 1} \\ p'_C &= 1 - p'_I - p'_S, \end{aligned} \tag{4.11}$$

where p_{tr} is sender's transmission probability at a time slot, and $\frac{M}{N}$ approximates the number of senders contending on one channel. Based on the Markov model described in Section 4.6, the transmission probability p_{tr} can be computed by summing over the probabilities of all states with a backoff counter value equal to zero on any of the N channels. This is because a saturated sender will initiate transmission on any channel, once its backoff counter reaches zero. Therefore, we have:

$$p_{tr} = \sum_{k=0}^{m} \pi_{\langle u,k,0 \rangle}, \quad 1 \leq u \leq N. \tag{4.12}$$

From (4.12), p_{tr} is also dependent on p_I given $p_d(f_u)$ and $p(f_u)$. Therefore, p_{tr} can be uniquely determined by finding a value satisfying the following equations:

$$\begin{cases} p_I = (1 - p_{tr})^{\frac{M}{N}-1} \\ p_{tr} = \sum_{k=0}^{m} \pi_{\langle u,k,0 \rangle}, \quad 1 \leq u \leq N. \end{cases} \tag{4.13}$$

Based on (p'_I, p'_S, p'_C), the throughput of channel f_u can be computed by [129]:

$$T_{f_u} = \frac{p'_S \cdot p_d(f_u) \cdot payload}{E[\tau_{slot}(f_u)]}. \tag{4.14}$$

To derive the average slot duration for a channel, we need to know the actual length of a success, collision, and idle slot. Let them be denoted by τ_S, τ_C, and τ_I respectively. The length of a successful slot is defined as the duration of a successful transmission. FD-MMAC complies with the basic access mechanism of IEEE 802.11 DCF, a successful transmission includes the transmission of a data frame, a SIFS (Short Interframe Space) period, an ACK frame, and a subsequent DIFS (DCF Interframe Space) period before which the backoff process is resumed. Therefore, $\tau_S = \tau_p + \tau_{SIFS} + \tau_{ACK} + \tau_{DIFS}$, where τ_p and τ_{ACK} denote the transmission duration of data frame and ACK frame respectively, and τ_{SIFS} and τ_{DIFS} denote the SIFS and DIFS period respectively. The length of a collision slot in FD-MMAC is defined as the duration of a corrupted transmission or a transmission for which the destination is not detected. Recall that FD-MMAC employs an early collision detection mechanism, in which the sender uses the lack of BCN reply as an indication of a collision or of a failed destination discovery attempt. Thus, we have $\tau_C = \tau_{PHY} + \tau_{MAC} + \tau_{BCN} + \tau_{timeout}$, where τ_{PHY} and τ_{MAC} denote the length of PHY-layer and MAC-layer header, respectively, τ_{BCN} denotes BCN frame transmission duration, and $\tau_{timeout}$ is the time out period. The average slot duration for channel f_u is derived as:

$$E[\tau_{slot}(f_u)] = p'_I \cdot \tau_I + p'_S \cdot p_d(f_u) \cdot \tau_S + (1 - p'_I - p'_S \cdot p_d(f_u)) \cdot \tau_C. \tag{4.15}$$

Once $E[\tau_{slot}(f_u)]$ is obtained, we are able to compute the throughput of f_u using (4.14). The aggregate network throughput is then derived as follows:

$$T = \sum_{u=1}^{N} T_{f_u} = \sum_{u=1}^{N} \frac{p'_S \cdot p_d(f_u) \cdot payload}{E[\tau_{slot}(f_u)]}. \tag{4.16}$$

□

4.7 Testbed Experiments and Simulations

In this section, we experimentally verify the PHY-layer techniques used by FD-MMAC. Furthermore, we validate the throughput analysis of Section 4.6 and compare FD-MMAC with prior art via packet-level simulations. Finally, we evaluate the impact of various jamming strategies.

4.7.1 Validation of the PHY-layer Techniques

Testbed: We performed our experiments on NI USRPs devices [130], over the 2.4 GHz band. The signal processing blocks were implemented in Labview [130]. Transmissions were modulated using Quadrature Phase Shift Keying (QPSK). The radios applied phase/frequency offset correction and time synchronization using 88-bit preamble sequences.

Operation State Classification

To validate the operation state classification rules presented in Section 4.3, we replicated the topology of Figure 4.3. Terminals A and B were placed 7ft apart and transmitted concurrently. Terminal A transmitted 100 data frames carrying a 500-bit payload, while B transmitted 500 BCNs with a 50-bit payload. We placed terminal C at positions C_1-C_4 of Figure 4.3 and measured the EVM, RSS, and the decodability of BCNs. Figure 4.9 shows the CDF of the EVM for each position of C. The RO and TO curves were combined, as they resulted in similar values. We observe that the EVM in the CO region (position C_3) is significantly

higher compared with all other locations due to the collision of P with the BCN. The difference allows us to select the threshold γ_{EVM} for the EVM classification rule. In our experiments, we set $\gamma_{EVM} = -18$ dB to achieve a false positive rate of 2% ($EVM < \gamma_{EVM}$ when in the CO region) and a false negative rate of 4% ($EVM \geq \gamma_{EVM}$ when in the TO/RO region).

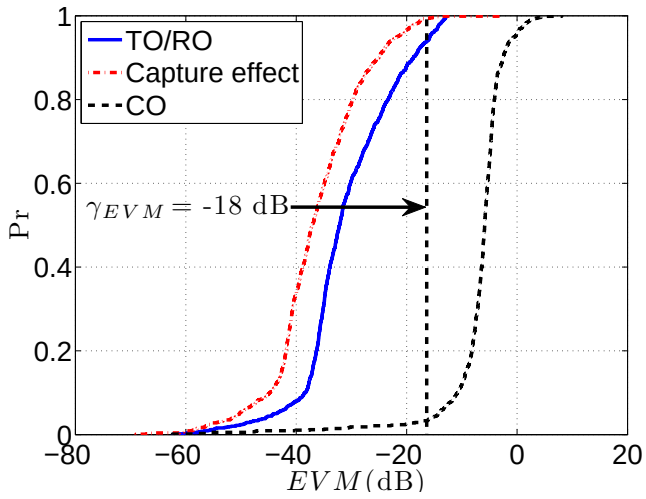


Figure 4.9: The EVM CDF at the RO, CO, and TO regions.

For position C_2, $EVM < \gamma_{EVM}$ due to the capture effect [126]. To avoid the classification of a terminal located at C_2 as an exposed terminal, we use the mean RSS value. Figure 4.10 shows the mean RSS value for different receiver locations, averaged over the experiment duration. For the RSS classification rule, we set γ_{RSS} to 1dBm. We observe that for location C_2 (within 2ft from A), C has an RSS value significantly higher than γ_{RSS}, and therefore infers that it is located in the CO region, despite having an $EVM < \gamma_{EVM}$. Also, for exposed terminal locations (more than 5ft from A), the EVM and RSS are below γ_{EVM} and γ_{RSS}, respectively.

Finally, we measured the fraction of BCNs that can be decoded by C over ten repeated experiments (500 BCNs each run). We recorded zero decodable BCN at locations C_1, C_2, and C_3, while 100% of the BCNs were recovered at C_4. We also placed C in the vicinity of B but within the CO. For this position, terminal C was

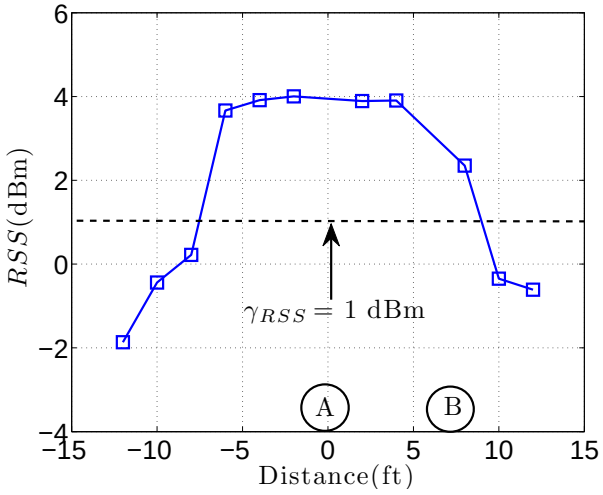


Figure 4.10: Average RSS at different positions.

able to decode a large fraction of BCNs and falsely assume it is in the RO region. However, this error does not impact the correct FD-MMAC operation because, for all practical purposes, a terminal in the RO region defers from transmission.

Signal Correlation

We experimentally evaluated the signal correlation technique for the exposed terminal topology of Figure 4.5(b). Terminal A transmitted 500-bit long data frames continuously while terminal D transmitted 50-bit long BCNs. Terminal C applied the signal correlation method to detect BCN_D frames. Figure 4.11 shows the normalized correlation for a snapshot of ten BCNs, when C is placed between A and D, at a 7ft distance from each. The correlation peaks correspond to the BCN transmissions and can be clearly distinguished. In our experiments, we set the detection threshold to 0.005. Furthermore, we placed C in three discrete positions between A and D and measured the percentage of BCN_D frames that can be detected by correlating the received signal with the known BCN pattern (preamble + payload). Terminal D transmitted 1,000 BCNs. The results are shown in Table 4.2. Distances are measured from D.

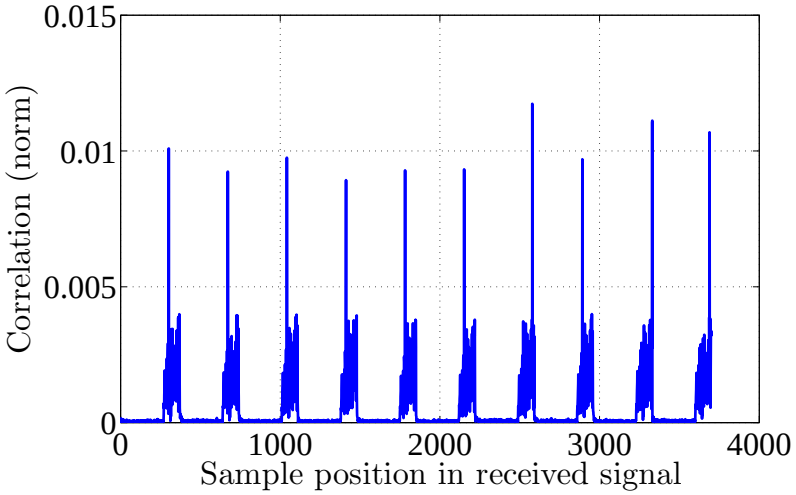


Figure 4.11: Normalized correlation values for 10 BCN frames.

Table 4.2: Fraction of detected BCN frames

Distance from D	3 ft	5 ft	7 ft
Percentage	100%	99%	94%

Table 4.2 shows that a terminal in the collision domain of two transmitters can reliably detect a frame with known pattern using the signal correlation technique.

4.7.2 Performance Evaluation of FD-MMAC

In this section, we evaluate the performance of FD-MMAC and compare it with prior art via simulations.

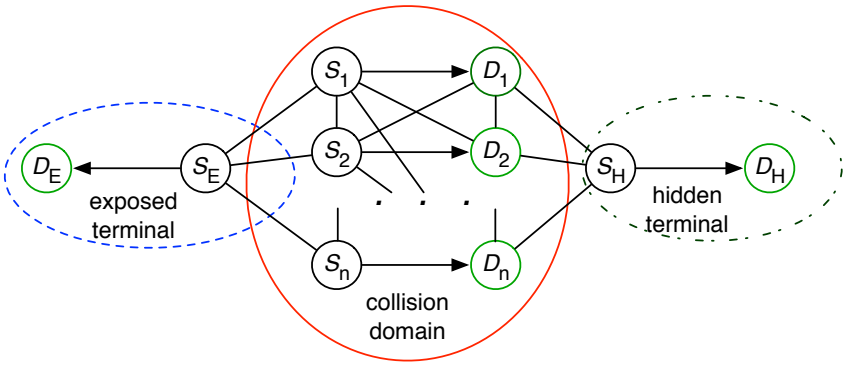


Figure 4.12: The network topology used in the simulation experiments.

Simulation setup: We performed packet-level simulations using OPNET^{TM} [112]. In our setup, multiple sender-destination pairs (flows) were organized in the topology of Figure 4.12 and shared orthogonal channels with 2Mbps capacity. The frame arrival process at each sender followed the Poisson distribution with an average arrival rate equal to λ frames per second, unless otherwise specified. Each frame was 512 bytes long. Every sender generated traffic for at least two destinations, so more than one senders contended for the same destination. The average switching delay and slot duration were set to $20\mu s$ each. Simulations were run for 40 sec and results were averaged over 10 simulation runs.

Throughput (T)

In the first set of experiments, we compared FD-MMAC's throughput with the throughput of the SP-MMAC in [8] and the DCC-MMAC in [19]. The control and data phase of SP-MMAC were set to 20ms and 80ms, respectively. Figures 4.13 and

4.14 compare the aggregate throughput for a varying number of flows contending over three channels and co-located in the same collision domain (senders S_E and S_H were idle). For low λ's, all protocols achieve similar throughput due to low contention. However, in high load conditions, FD-MMAC achieves significantly higher aggregate throughput due to the elimination of signaling for channel negotiation and virtual carrier sensing. The maximum FD-MMAC throughput is close to 5.5MBps under high load (total capacity of the three channels is 6Mbps). Figures 4.15(a) and 4.15(b) show the average per-flow throughput of FD-MMAC and SP-MMAC. FD-MMAC significantly outperforms SP-MMAC in high load conditions.

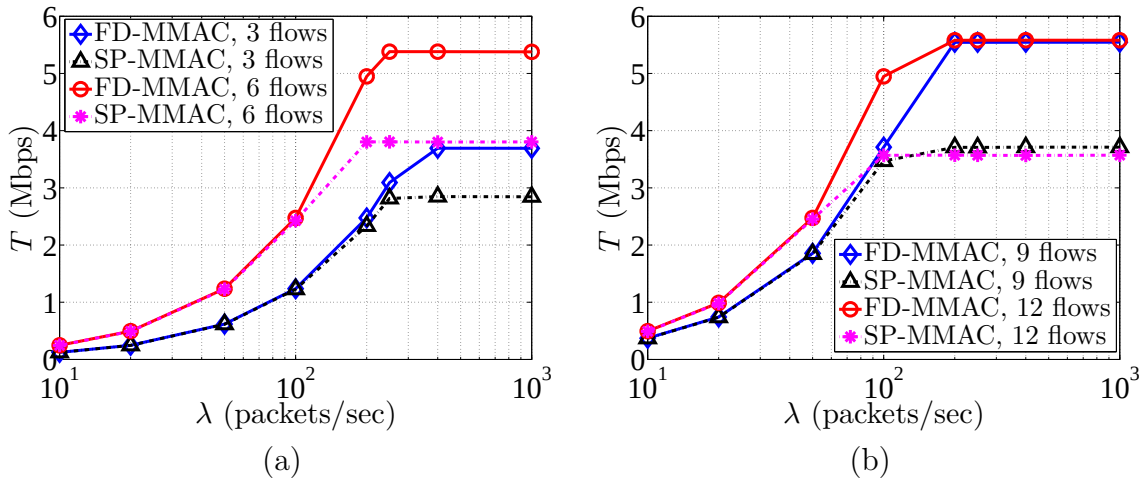


Figure 4.13: (a) Aggregate T of FD-MMAC and SP-MMAC when 3 and 6 flows are within same collision domain, (b) aggregate T of FD-MMAC and SP-MMAC when 9 and 12 flows are within same collision domain.

In the second set of experiments, we placed five flows $S_1 \rightarrow D_1$,...,$S_5 \rightarrow D_5$ in the same collision domain, while S_E operated as an exposed terminal to S_1-S_5. For FD-MMAC, we considered two scenarios. In the first scenario, BCNs and ACKs were perfectly detected using signal correlation. In the second scenario, 5% of BCNs and 5% of ACKs were undetectable by the intended recipients. Figure 4.16(a) shows the aggregate throughput for varying λ. We observe that in high load conditions, FD-MMAC achieves an aggregate throughput that is 83% higher compared

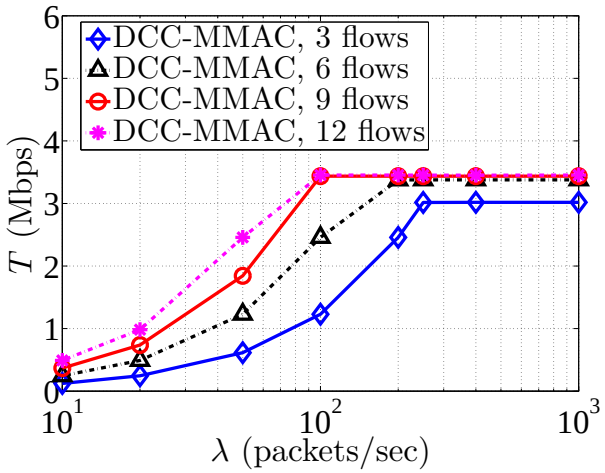


Figure 4.14: Aggregate T of DCC-MMAC when 3, 6, 9, and 12 flows are within same collision domain.

with SP-MMAC and 92% higher compared with DCC-MMAC under ideal operating conditions. The throughput improvement drops to 70% and 78%, respectively, when 5% of BCNs and ACKs are lost.

The superior performance of FD-MMAC is due to the parallel operation of S_E with any of the S_1-S_5. In fact, the individual throughput of S_E was 63% higher than the throughput of S_1-S_5 because S_E did not contend with other senders/ At every transmission attempt, S_E classified its state as an exposed terminal and continued the backoff countdown. We also evaluated the concurrent operation of exposed and hidden terminals (both S_E and S_H were active senders). Figure 4.16(b) shows that FD-MMAC achieves 56% and 53% higher throughput in high load conditions compared with SP-MMAC and DCC-MMAC, respectively.

Delay

In the third set of experiments, we evaluated the frame delay for bursty frame arrivals. We loaded the transmission queue of each sender with 100 data frames and measured the delay until all 100 frames were delivered to their respective destinations. All flows were within same collision domain. Figure 4.17 shows the average

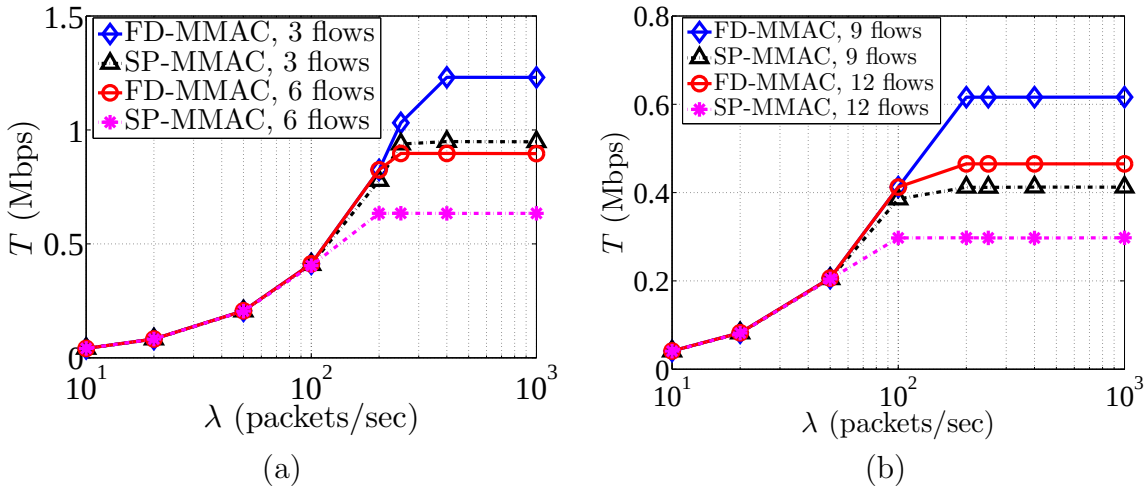


Figure 4.15: (a) Per-flow average T for FD-MMAC and SP-MMAC when 3 and 6 flows are within same collision domain, (b) per-flow average T for FD-MMAC and SP-MMAC when 9 and 12 flows are within same collision domain.

delay as a function of the number of competing flows. We observe that FD-MMAC reduces the delay due to the elimination of the control message exchange before frame transmissions. The delay increases almost linearly with the number of contending flows for all protocols, because the available channels are shared by more flows in a fair manner.

Validation of the Theoretical Throughput Analysis

To validate the Markov model proposed in Section 4.6, we compared the saturation throughput computed via Proposition 4 with the throughput measured in simulations. To simulate saturated traffic, we implemented backlogged queues at all senders by employing a deterministic traffic model with fixed frame arrival rate of 1000 frames per second. For the analytical model, we varied the number of flows to saturate nine channels and computed the aggregate throughput when: (a) a channel priority list is employed to resolve ties in the CST (best case), and (b) ties are broken arbitrarily (worst case). For the first scenario, we set $p(f_i) = p_d(f_i) = 1$. That is, the first channel in priority list is always preferred ($p_(f_i) = 1$) and the destination

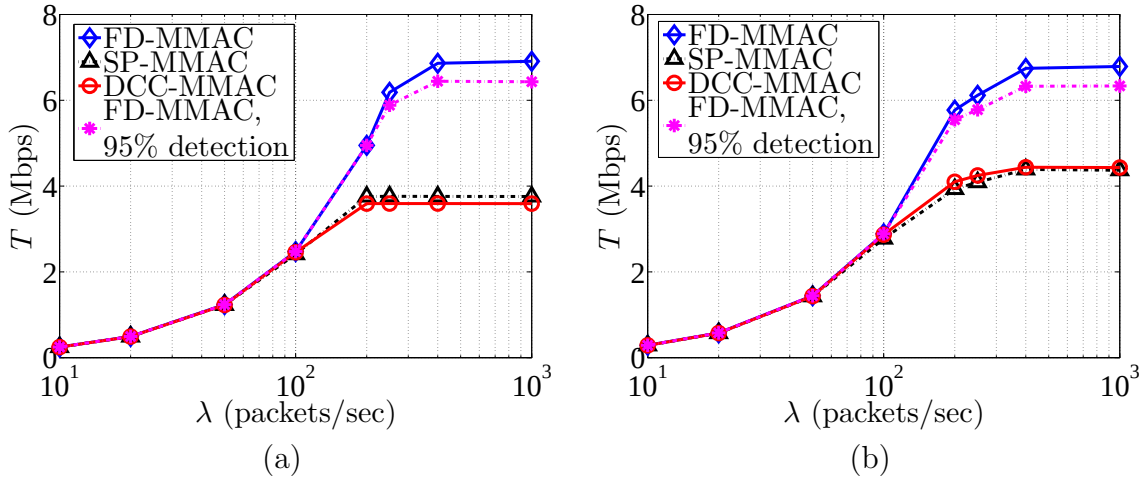


Figure 4.16: (a) Aggregate T in the presence of an exposed terminal, (b) aggregate T in the presence of one exposed and one hidden terminal.

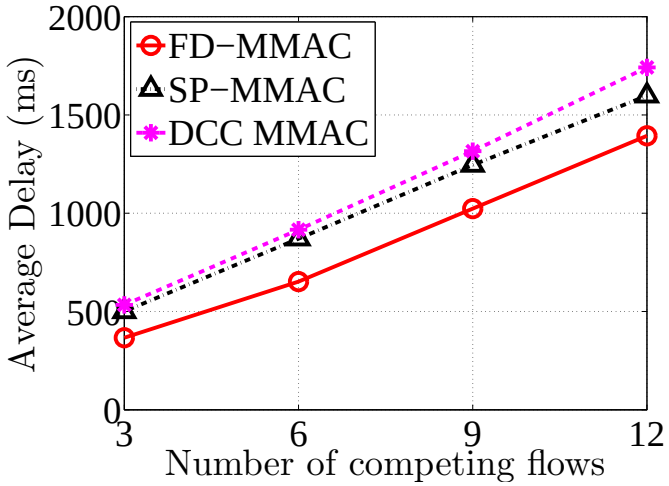


Figure 4.17: Average delay for transmitting a batch of 100 data frames.

is always found on that channel ($p_d(f_i) = 1$). For the second scenario, terminals switch at any channel with equal probability ($p(f_i) = \frac{1}{N-1}$) and discover the destination with equal probability ($p_d(f_i) = \frac{1}{N-1}$). We observe from Figure 4.18 that the throughput obtained via simulations lies between the best-case and worst-case scenarios. Under low contention, the achievable throughput is better approximated by $p_(f_i) = p_d(f_i) = 1$, as senders are likely to find their destination when switching according to the CST. On the other hand, increased contention causes frequent channel switching making the CST view of each terminal obsolete faster. Therefore, the probabilities of switching to a channel and finding the destination approximate the uniform distribution. We note that the mismatch between the simulation and analytical results are due to several simplifying assumptions stated in Section 4.6. Nevertheless, the two theoretical scenarios yield useful best-case and worst-case performance indicators under saturation conditions.

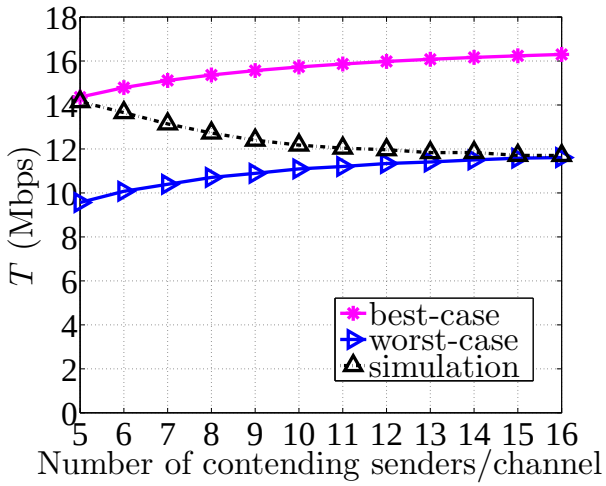


Figure 4.18: Comparison of the analytical aggregate throughput with the simulated throughput.

Fairness and Load Balancing

We also examined the fairness and load balancing properties of FD-MMAC under different traffic load conditions. To evaluate fairness, we use Jain's *Fairness Index*

(FI):

$$FI = \frac{(\sum_{i=1}^{n} T_i)^2}{n \times \sum_{i=1}^{n} (T_i)^2}, \tag{4.17}$$

where T_i is the throughput of the i^{th} flow and n is the total number of flows.

The FI equaled 0.91 for a topology with one exposed and one hidden terminal active. This was due to the higher throughput attained by the exposed and hidden terminal flows. As an exposed terminal, S_E did not contend with other senders (it could operate in parallel with any other sender), thus achieving higher throughput. Moreover, S_E did not experience any destination discovery delay because D_E stayed on a particular channel and was always available for reception (D_E always perceives its resident channel idle). Similarly, D_H did not switch channels, making the destination discovery delay for S_H negligible. The FI increased to 0.99 for a topology with six flows in the same collision domain, indicating that FD-MMAC achieves fair distribution of resources among competing flows.

We note that although FD-MMAC employs a CSMA/CA-like backoff mechanism to resolve contention, it does not exhibit the well-known unfairness of the exponential backoff process [131]. This is because collisions are rare due to the availability of several channels and the use of BCNs as a virtual carrier sensing mechanism. Moreover, collisions that corrupt the first BCN (most probable collision scenario) are interpreted by the sender as a failure to discover the destination and cause a channel switch without doubling the contention window. As a result, the colliding terminals does not have an unfair advantage in accessing the new channel after a switch.

We also evaluated the traffic load carried by each channel by computing the *Load Balancing Index* (LBI) under high load:

$$LBI = \frac{(\sum_{i=1}^{N} T_{f_i})^2}{N \times \sum_{i=1}^{N} (T_{f_i})^2}, \tag{4.18}$$

where T_{f_i} is the aggregate throughput on channel f_i.

The LBI equaled 0.86 for a topology with one exposed and one hidden terminal.

This was due to the concurrent operation of the exposed terminal on the same channel with another flow and the use of the single channel by the hidden terminal destination (no channel switching). The LBI increased to to 0.99 for a topology with all flows in the same collision domain.

4.7.3 Chapter Summary

We proposed FD-MMAC, a distributed MMAC protocol that exploits FD communications to coordinate channel access at low control overhead. FD-MMAC eliminates control signaling over a common control channel to improve spectral efficiency and to mitigate DoS attacks launched against control channels. The FD-MMAC properties are achieved by utilizing an advanced suite of PHY-layer techniques, including SIS, EVM and RSS measurements, and signal correlation techniques. We analytically evaluated the saturation throughput of FD-MMAC using a three-dimensional Markov model. Finally, we experimentally validated the PHY layer techniques employed by our protocol on the NI USRP testbed and measured its performance via simulations. Our simulations showed that FD-MMAC achieves significantly higher throughput compared with prior MMAC designs.

CHAPTER 5

JAMMING RESISTANT MULTI-CHANNEL MEDIUM ACCESS CONTROL

5.1 Introduction

5.1.1 Motivation

As discussed in Chapter 4, FD-MMAC was primarily designed to eliminate the need for a common control channel. As such, it is capable of preventing the network from DoS attacks on the control channel. We illustrate the impact of control channel jamming on different MMAC protocol types using Figures 5.1, 5.2, and 5.3. For the SP-MMAC and DCC-MMAC designs, which were not designed with security in mind, communications on all channels can be easily denied by jamming the control channel used for coordinating channel access. For SP-MMACs, jamming the control channel during the control phase is sufficient to prevent transmissions on all channels during the upcoming data phase. This attack is demonstrated in Figure 5.1. It is highly efficient since the adversary jams a single channel and only during the control phase. Similarly, in DCC-MMACs, jamming the DCC is sufficient for blocking the communications on all available channels, as shown in Figure 5.2. In FD-MMAC, a jammer without a single target must spread his resources over all channels, thus allowing for jamming-free transmission opportunities. This scenario is depicted in Figure 5.3. In this chapter, we study the anti-jamming properties of FD-MMAC and extend FD-MMAC to combat jamming.

5.1.2 Main Contributions and Chapter Organization

We define a comprehensive reactive jamming model for MMAC protocols based on the cross-layer consideration of the PHY and MAC layers. Our model analyzes

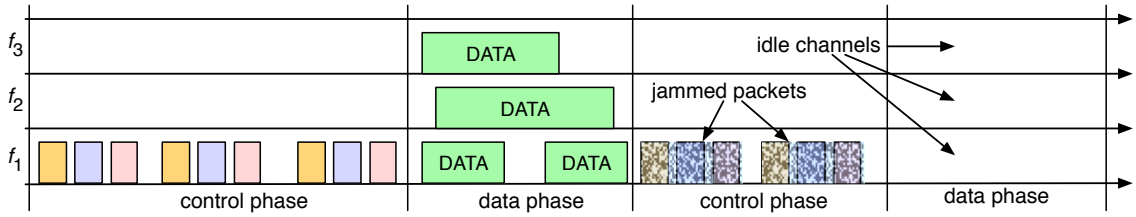


Figure 5.1: Control-channel jamming on SP-MMAC protocol type [7, 8, 16].

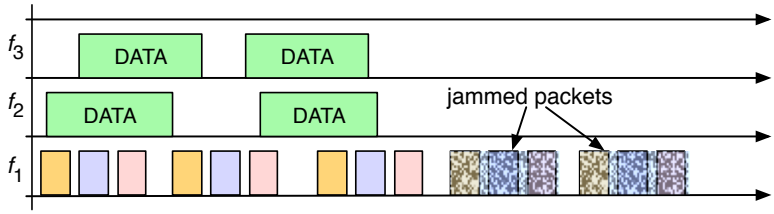


Figure 5.2: Control-channel jamming on DCC-MMAC protocol type [19–21].

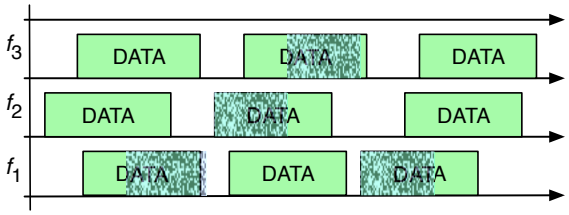


Figure 5.3: Control-channel jamming on FD-MMAC.

the jamming period (time spent jamming a transmission) as a function of the error correction capability (ECC), modulation order, and interleaving function. We further analyze various jamming attacks on FD-MMAC based on the jammer's targeted frames, channel dwell period (time he resides on a target channel), and channel switching strategy. We extend FD-MMAC to combat jamming by investigating cryptographic interleaving at the PHY-layer, random channel switching, and switching according to a common secret channel priority list. We evaluate FD-MMAC's resilience to jamming by studying the relationship between the jammer's effort, the channel dwell period, the adopted PHY layer parameters, and the throughput and goodput.

Chapter Organization: In Section 5.2, we define the jamming model. Section 5.3 describes various jamming attacks on FD-MMAC. In Section 5.4, we discuss possible modifications of FD-MMAC for improving its anti-jamming properties. In Section 5.5, we evaluate the performance of FD-MMAC under the jamming attacks presented in Section 5.3. Section 5.6 concludes the chapter.

5.2 Jamming Model

We consider a fast-hopping jammer, denoted by J, with negligible channel switching delay. The jammer can interfere with one channel at any time. The jammer is reactive and operates in two phases: the *sensing phase* and the *jamming phase*. During the sensing phase, the jammer senses the current channel for a sensing period τ_s to estimate the channel state. During the jamming phase, the jammer transmits an interfering signal for a jamming period τ_j, with sufficient power to corrupt interfered symbols. The selection of τ_s, τ_j, and of the channel switching pattern form a *jamming strategy*. Such strategies differ in sophistication, resource requirements, and effectiveness. To capture these differences we use the following metrics.

Definition 1. Jamming effort $\mathcal{A}$: The fraction of time that the adversary jams any of the N available channels.

$$\mathcal{A} = \frac{1}{NT} \sum_{i=1}^{N} \alpha(f_i), \tag{5.1}$$

where $\alpha(f_i)$ is the time that f_i is jammed over period T.

Definition 2. Effective hopping rate $\mathcal{R}$: The inverse of the channel dwell period $\tau_d = \tau_s + \tau_j$, which is the period spent by the jammer on a channel for performing channel sensing and/or jamming. That is, $\mathcal{R} = \frac{1}{\tau_d}$.

5.2.1 Determining the Jamming Period τ_j

Consider the transmission of frame P from A to B. Let P be encoded with a channel coding scheme that can correct *up to any e bit errors*. Encoded frame P is modulated to complex symbols, which are transmitted every T_s (T_s denotes the symbol duration). For symbol $\mathbf{s}[k]$ transmitted by A, the received symbol $\mathbf{r}[k]$ at B, when corrupted by a jamming signal $\mathbf{j}[k]$, can be expressed as:

$$\mathbf{r}[k] = \mathbf{H}\mathbf{s}[k] + \mathbf{G}\mathbf{j}[k] + \mathbf{w}[k], \tag{5.2}$$

where $\mathbf{H} = he^{j\theta}$ is the channel response of the A-B channel, $\mathbf{G} = ge^{j\phi}$ is the channel response of the J-B channel and $\mathbf{w}[k]$ is random complex noise. Here, h, g refer to the channel attenuation and θ, ϕ refer to the channel phase shift. During the demodulation process, the receiver compensates for $\mathbf{H}$ and attempts to recover $\mathbf{s}$ by mapping $\mathbf{r}$ to the closest symbol $\mathbf{s}'$ in the Euclidean distance sense[1]. However, $\mathbf{s}'$ may differ from $\mathbf{s}$, due to $\mathbf{G}\mathbf{j}$ and $\mathbf{w}$.

The jammer could attempt to design $\mathbf{j}$ such that $\mathbf{r}$ is mapped to a desired symbol $\mathbf{s}'$. However, to craft $\mathbf{j}$, the jammer must know a priori $\mathbf{H}$, $\mathbf{G}$, $\mathbf{w}$, and the transmitted symbol $\mathbf{s}$. From these parameters, $\mathbf{s}$ cannot be known before it is transmitted, while the rest vary with time. Therefore, the jammer has no advantage in constructing $\mathbf{j}$ to

[1]For convenience, we drop the "[k]" notation, when unnecessary.

fall within a specific region in the constellation diagram. Given independent values for $\mathbf{Gj}$, $\mathbf{w}$, $\mathbf{s}$, and $\mathbf{H}$, we can assume that the received symbol $\mathbf{s}'$ takes any of the q symbol values equiprobably. Let random variable (RV) $\mathbf{X}$ denote the number of flipped bits, when a symbol $\mathbf{s}$ is jammed and decoded to a symbol $\mathbf{s}'$. The probability mass function (PMF) of $\mathbf{X}$ is given in Proposition 5.

Proposition 5. *For a q-order modulation, the PMF of $\mathbf{X}$ is:*

$$\Pr[\mathbf{X} = x] = \frac{1}{q}\binom{\log_2 q}{x}. \tag{5.3}$$

Proof. For random values of $\mathbf{j}$ and $\mathbf{w}$, a transmitted symbol $\mathbf{s}$ is mapped to any of the q symbols of the constellation with equal probability. Therefore, the probability that x out of the $\log_2 q$ bits of $\mathbf{s}$ are flipped is:

$$\begin{aligned} \Pr[\mathbf{X} = x] &= \sum_{\mathbf{s}'} \Pr[\mathbf{r} \rightarrow \mathbf{s}' : \mathcal{H}(\mathbf{s}, \mathbf{s}') = x] \\ &= \frac{1}{q}\binom{\log_2 q}{x}, \end{aligned} \tag{5.4}$$

where $\mathcal{H}(\mathbf{s}, \mathbf{s}')$ is the Hamming distance between the bit pattern (Gray codeword) assigned to $\mathbf{s}$ and $\mathbf{s}'$, respectively. Proposition 5 follows immediately by noting that for any $\mathbf{s}$, there are exactly $\binom{\log_2 q}{x}$ symbols s' with a Hamming distance equal to x and $\mathbf{r}$ is decoded to one of these symbols with probability $\frac{1}{q}$. □

Using Proposition 5, we can compute the probability of corrupting P, when P is jammed for τ_j symbol periods.

Proposition 6. *Let RV $\mathbf{S_y} = \mathbf{X_1} + \mathbf{X_2} + \ldots + \mathbf{X_y}$ be the number of flipped bits when y symbols are jammed. Here, $\mathbf{X_1}$, $\mathbf{X_2}, \ldots, \mathbf{X_y}$ are i.i.d.s, following the distribution in* (5.3)*. The complementary cumulative probability mass function (CCMF) of $\mathbf{S_y}$ is:*

$$\Pr[\mathbf{S_y} > e] = 1 - \left(\frac{1}{q}\right)^y \sum_{i=0}^{e} \binom{y \log_2 q}{i}. \tag{5.5}$$

Proof. To compute the CCMF of $\mathbf{S_y}$, we first note that $\mathbf{S_y}$ is the sum of y i.i.d.'s following the distribution of Proposition 5. The distribution of the sum of y i.i.d. RVs is given by the discrete convolution formula. We compute this formula for $y = 2$ and then extend to the general case by induction. For RV $\mathbf{S_2} = \mathbf{X_1} + \mathbf{X_2}$, it follows that:

$$\begin{aligned}
\Pr[\mathbf{S_2} = e] &= \sum_{i=0}^{e} \Pr[\mathbf{X_1} = i] \Pr[\mathbf{X_2} = e - i] \\
&= \sum_{i=0}^{e} \frac{1}{q} \binom{\log_2 q}{i} \frac{1}{q} \binom{\log_2 q}{e-i} \\
&= \left(\frac{1}{q}\right)^2 \sum_{i=0}^{e} \binom{\log_2 q}{i} \binom{\log_2 q}{e-i} \\
&= \left(\frac{1}{q}\right)^2 \binom{2\log_2 q}{e}. \qquad (5.6)
\end{aligned}$$

In (5.6), we have used the Vandermonde convolution theorem for the computation of the summation of binomial coefficients. For an arbitrary y, by induction (convolution of $\mathbf{S_{y-1}}$ with $\mathbf{X_y}$) the PDF of $\mathbf{S_y}$ is:

$$\Pr[\mathbf{S_y} = e] = \left(\frac{1}{q}\right)^y \binom{y \log_2 q}{e} \qquad (5.7)$$

It is straightforward to verify that (5.7) is a valid probability distribution as:

$$\begin{aligned}
\sum_{e=0}^{y\log_2 q} \Pr[\mathbf{S_y} = e] &= \left(\frac{1}{q}\right)^y \sum_{e=0}^{y\log_2 q} \binom{y \log_2 q}{e} \\
&= \left(\frac{1}{q}\right)^y 2^{y \log_2 q} \\
&= 1.
\end{aligned}$$

From (5.7), it immediately follows:

$$\Pr[\mathbf{S_y} > e] = 1 - \left(\frac{1}{q}\right)^y \sum_{i=0}^{e} \binom{y \log_2 q}{i}. \tag{5.8}$$

□

Using Proposition 6, the jammer can determine the jamming period $\tau_j = yT_s$, such that a frame P protected from up to e errors and modulated with a q-order modulation is corrupted beyond recovery with a desired probability. In Figure 5.4(a), we show the probability of corrupting P as a function of number of jammed symbols (y) and for different modulation orders, when $e = 10$. Based on Figure 5.4(a), to drop P with probability 0.9 when $q = 4$, the jammer has to jam $y = 13$ symbols, yielding a $\tau_j = 13T_s$. In Figure 5.4(b), we show the CCMF of $\mathbf{S_y}$ as a function of y for different ECC thresholds, when $q = 4$.

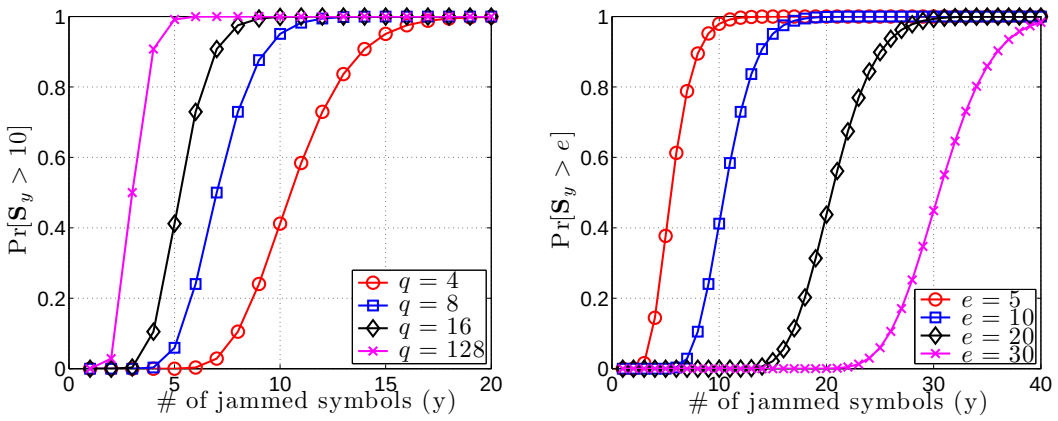
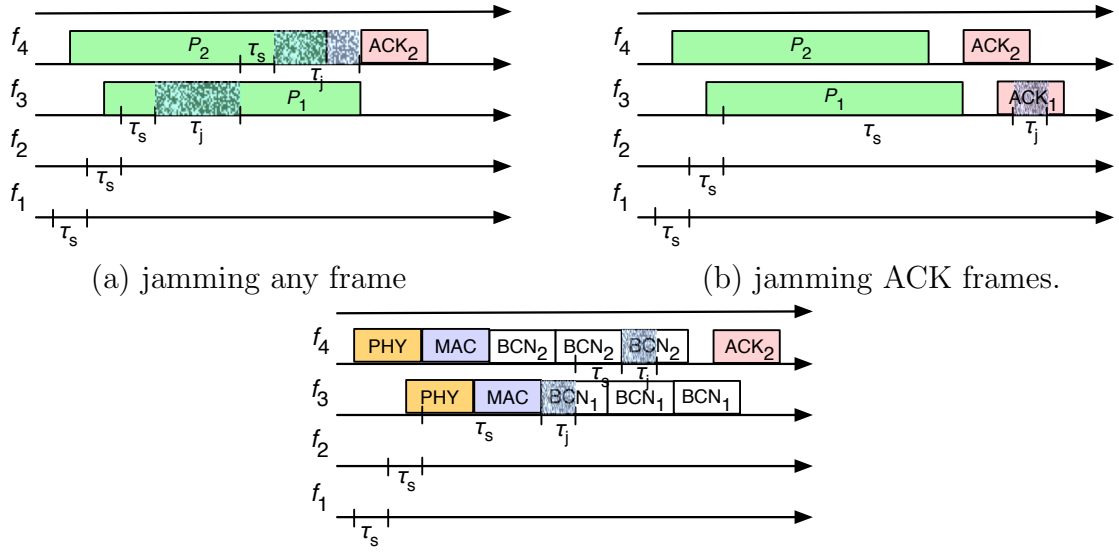


Figure 5.4: The CDF of corrupting e bits when jamming y symbols for (a) $e = 10$ and varying modulation order q and, (b) $q = 4$ and varying e.

5.3 Jamming Attacks on FD-MMAC

In this section, we describe reactive jamming attacks on FD-MMAC. The jammer's strategy is defined by the targeted frames, the selection of τ_j and τ_s, and the channel



(a) jamming any frame

(b) jamming ACK frames.

(c) jamming BCN frames

Figure 5.5: Jamming attacks on FD-MMAC.

switching strategy. We examine the jamming of a) any frame, b) ACK frames, and c) BCN frames.

5.3.1 Jamming Any Frame

When the jammer does not target a particular frame type, it can initiate his jamming attack immediately after a channel is detected to be busy. This approach minimizes τ_s for determining the channel state to one slot. In Figure 5.5(a), we show a jammer applying the reactive jamming strategy independent of the transmitted frame. The jammer detects a transmission on f_3 after sensing f_1 and f_2 idle. It jams f_3 for τ_j and corrupts P_1. It then hops to f_4 and jams the transmission of P_2. However, the receiver is able to recover P_2 because only a small portion of P_2 is jammed.

5.3.2 Jamming ACK Frames

The jammer can choose to target the ACKs arriving at the sender. The ACK jamming strategy is presented in Figure 5.5(b). The jammer detects the transmission of P_1 and extends the sensing period until the ACK transmission is initiated. It

then jams the transmission of ACK_1. Jamming ACKs is equivalent to jamming the corresponding data frames, as it forces data retransmission. However, in FD-MMAC, ACKs can be detected even if they are not correctly decoded by applying the signal correlation method. Moreover, ACK jamming requires a significantly longer sensing period τ_s. This is because the jammer has to continuously sense his resident channel until the ACK transmission is initiated. The average sensing period is given by the following proposition.

Proposition 7. *For an adversary switching to a channel with an active data transmission, the average sensing period $E[\tau_s]$ until an ACK frame can be jammed is:*

$$E[\tau_s] = \frac{\tau_p}{2} + \tau_{SIFS}, \tag{5.9}$$

where τ_p denotes the data frame transmission duration and τ_{SIFS} denotes the short inter-frame space (SIFS) period between the data frame and the ACK.

Proof. To jam an ACK over a channel f_i, the adversary J must switch to f_i while a data frame P is transmitted. Since J switches over various channels in arbitrary fashion (channels are selected at random and the jammer's channel dwell time depends on current traffic patterns), we can assume that the time at which J switches to f_i while P is being transmitted follows the uniform distribution $\mathcal{U}(0, \tau_p)$. Consequently, the average time until the end of P's transmission is equal to $\frac{\tau_p}{2}$. Adding the SIFS time τ_{SIFS} that separates a data frame from an ACK transmission proves Proposition 7. □

Note that for most realistic PHY layer parameters, τ_j is significantly smaller than $\frac{\tau_p}{2}$. Therefore, targeting ACKs reduces the effective channel hopping rate $\mathcal{R}$ compared to targeting any frame.

5.3.3 Jamming BCN Frames

The jammer can target the BCN frames sent by the destination during a data transmission. Recall that the first BCN is used by the sender to verify that the

destination resides on the same channel and is receiving. If the first BCN is jammed, the sender will abort the transmission of the data frame and switch to another channel. Figure 5.5(c) shows a BCNs jamming scenario. The jammer jams the first BCN_1 because it switched to f_3 during the transmission of P_1's PHY header. However, it missed the first BCN_2 for P_2. The probability of hitting the first BCN for a jammer that switches to a busy channel is given in following proposition.

Proposition 8. *The probability of jamming the first BCN of duration τ_{BCN} when switching to a busy channel is:*

$$\Pr[BCN = jam] = 1 - \frac{\tau_{PHY} + \tau_{MAC} + \tau_{BCN} - \tau_j}{\tau_p}. \tag{5.10}$$

where τ_{PHY} and τ_{MAC} denote the PHY header and MAC header duration, respectively.

Proof. Let the adversary switch to channel f_i during the transmission of P. The jammer's switching time and the transmission start time are independent events. Hence, without loss of generality, we assume that switching time follows a uniform distribution $\mathcal{U}(0, \tau_p)$[2]. To have the opportunity of jamming the first BCN frame, the jammer must switch to f_i before the end of the BCN transmission, which is equal to $\tau_{PHY} + \tau_{MAC} + \tau_{BCN}$. Moreover the jammer must have time to corrupt sufficient number of bits from the BCN. If the jamming period to drop BCN is equal to τ_j, the jammer must switch to f_i before time $\tau_{PHY} + \tau_{MAC} + \tau_{BCN} - \tau_j$ has elapsed from the beginning of the frame transmission. The proof follows by noting that the switching time follows the uniform distribution $\mathcal{U}(0, \tau_p)$. □

Note that the sender may still be able to detect a jammed BCN using signal correlation. Verification of the BCN transmission is not based on frame decoding, but on the correlation of the received signal with the known BCN bit pattern. Moreover, if the jammer misses the first BCN transmission, he has no incentive of

[2]In our analysis, we have ignored the case where the jammer switches to channel that is sensed to be busy due to the transmission of an ACK frame.

jamming subsequent BCN frames. This is because those frames are only used to occupy the channel in the receiver's collision domain such that hidden terminals sense this channel to be busy. The superposition of a jamming signal with the BCN maintains the busy channel state.

Best strategy for the reactive jammer: Based on our reactive strategy analysis, we conclude that the best jamming strategy against FD-MMAC is to target any frames. This strategy maximizes the effective channel hopping rate and requires relatively small jamming effort. Moreover, ACK and BCN frames are better protected than data frames due to the application of the signal correlation technique for their detection. Furthermore, to jam ACKs, the jammer must prolong the sensing period to the end of the frame transmission. Finally, BCN jamming is only effective if the first BCN arriving at the sender is jammed.

5.3.4 Channel Switching

To quickly discover occupied channels, the jammer can take advantage of the channel priority list and the CST. Similar to any other terminal, the jammer can keep track of the channel state of all channels he senses and construct a CST, following the rules presented in Section 4.5.1. He can then hop between the channels according to the CST, using the channel priority list to break ties. In the next section, we describe several techniques for mitigating the jammer's effectiveness in discovering occupied channels.

5.4 Improving FD-MMAC Resilience to Jamming

In this section, we discuss possible modifications of FD-MMAC for improving its anti-jamming properties. Specifically, we investigate cryptographic interleaving at the PHY-layer, random channel switching, and switching according to a common secret channel priority list. The latter method differs from classical FH in several ways. First, terminals do not continuously hop in a synchronous fashion. Second, when a terminal is in "Switch" state, it selects the next hop independently from

other terminals based on its individual CST. Despite the use of a common secret channel priority list, the FH sequences formed by each terminal's switching decisions are unique. The proposed improvements are at the expense of key management for establishing and maintaining secrets among terminals. The key management problem is a well-studied one, and is beyond the scope of this article.

5.4.1 Cryptographic Interleaving

In most scenarios, a frame P may consist of several codewords, which are interleaved to combat burst errors. For simplicity, consider a block interleaver of *depth* Δ that permutes Γ symbols of Δ codewords ($\Delta \times \Gamma$ denotes the interleaving *period*) using a permutation function Π:$\{1 \ldots \Delta\Gamma\} \rightarrow \{1 \ldots \Delta\Gamma\}$. The interleaver depth Δ denotes the minimum separation in symbol periods at the interleaver output (and hence, the wireless channel) between any two adjacent symbols at the interleaver input. A block interleaver with $\Delta = 5$ applied to 10-symbol codewords is shown in Figure 5.6. Codewords are arranged row-wise and symbols are transmitted column-wise.

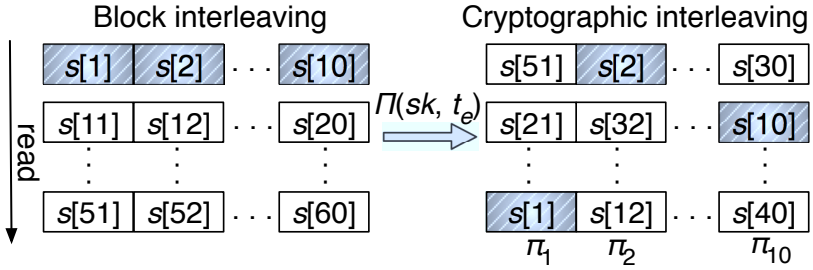


Figure 5.6: A block interleaver of depth Δ and period $\Delta \times \Gamma$, applied to codewords of length Γ symbols.

Interleaving does not reduce the total number of symbols that must be jammed to corrupt a frame. However, it can potentially prolong the jamming period τ_j. Let a codeword be corrupted if more than y symbols are jammed. As y symbols of *any* codeword are spread over time $y\Delta T_s$, the jammer must remain on the same channel for $y\Delta T_s$ to corrupt the targeted frame. This is a Δ-fold increase on τ_j compared to non-interleaved communications. We note, however, that the interleaver

permutation Π is typically publicly known. A sophisticated jammer with negligible channel switching delay could selectively target y symbols from the same codeword and switch to other channels when symbols from other codewords are transmitted. For instance, the jammer could target symbols $\{s[1], s[2], \ldots, s[10]\}$ of the first codeword. Following this strategy, the jammer can jam more than one channels over time $y\Delta T_s$.

To prevent the jammer from exploiting the known Π, we can apply cryptographic interleaving [25]. In cryptographic interleaving, Π becomes a function of a secret key sk shared between the sender and the destination and of the current time t_e, quantized to epochs. Without access to sk, the jammer cannot know the symbol positions of a codeword within an interleaved block. Moreover, the use of the current epoch t_e allows the sender and the destination to update $\Pi(sk, t_e)$ periodically. A random permutation could violate the minimum symbol separation requirements, leading to poor interleaving performance. To address this issue, we construct Π from random column sub-permutations of the original interleaved block. Let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_\Gamma\}$, where π_i is the sub-permutation applied to column i. Each π_i is a random permutation of $\{1 \ldots \Delta\}$, indicating a symbol rearrangement column-wise. The resulting interleaved block after the application of Π is shown in Figure 5.6. Under cryptographic interleaving, the required jamming period for corrupting y symbols of the same codeword is given by the following proposition.

Proposition 9. *Let a $\Delta \times \Gamma$ cryptographic interleaver be constructed using random column sub-permutations $\Pi = \{\pi_1, \pi_2, \ldots, \pi_\Gamma\}$. The adversary is guaranteed to jam y consecutive symbols from the same codeword, if he jams $(y-1)\Delta + 1$ consecutive symbols.*

Proof. We first show that jamming $(y-1)D+1$ consecutive symbols is sufficient to jam y symbols from one codeword and for any permutation Π. Consider the jamming of $(y-1)D$ consecutive symbols, starting from any symbol $s[i]$. The $(y-1)D$ symbols span across at least $(y-1)$ columns of the interleaving block. By construction of Π, every column contains one symbol from each of the D codewords. Therefore,

at least $(y-1)$ symbols from each codeword are jammed. Jamming one additional symbol guarantees that y symbols that belong to a single codeword are jammed.

It can be easily shown by an example that jamming fewer than $(y-1)D+1$ consecutive symbols does not guarantee the jamming of y symbols from one codeword. Consider any Π that maintains a fixed symbol separation of D (symbols of any codeword are separated by $(D-1)$ other symbols). Such a Π can be obtained by simply rearranging the rows of the interleaving block. Because of the fixed symbol separation D, jamming $(y-1)D$ consecutive symbols leads to the jamming of exactly $(y-1)$ symbols from each of the D codewords. Hence, an additional symbol needs to be jammed to guarantee the jamming of y symbols from at least one codeword. □

By combining Propositions 6 and 9, the jammer can determine the appropriate jamming period τ_j that leads to the irrecoverable corruption of a frame P, when cryptographic interleaving is applied. This is done as follows. Let P be interleaved with a cryptographic interleaver of depth D, be modulated with a q-order modulation, and be protected by a channel code that can correct up to e errors. Using Proposition 6, the jammer chooses the number of symbols y that must be corrupted to drop P with the desired probability $\Pr[\vec{S}_y > e]$. He then jams $(y-1)D+1$ consecutive symbols to corrupt y symbols from at least one codeword in P. As a numerical example, when $e = 30$bits and $q = 4$, the jammer must corrupt $y = 35$ symbols to drop P with probability 0.8 (see Figure 5.4(b)). Taking into account a cryptographic interleaver of $D = 5$, the jamming period must extend to 171 symbols to corrupt P with probability 0.8.

5.4.2 Randomizing the Channel Priority List

To further improve the resilience of FD-MMAC to jamming, we consider the randomization of the channel priority list used to break ties in the CST. A jammer could exploit this list to jam high-priority channels with higher probability. Two possible improvements can be adopted. First, senders and destinations could eliminate the

channel priority list in the presence of jamming and break channel ties arbitrarily. This will increase the destination discovery delay, but prevent the jammer from accurately guessing the next hop.

An alternate approach is to incorporate cryptographically-protected channel priority lists. In the case of a tie, the involved channels are ordered based on a *pre-agreed secret permutation* $\rho(gk, t_e)$, where gk is a globally shared secret[3] and t_e is the current epoch.

The secrecy of the channel priority list prevents the jammer from targeting those channels that are assigned a higher priority, which are more likely to host transmissions. Although the permutation $\rho(gk, t_e)$ that determines the channel priority remains secret, the adversary could still infer it by profiling the traffic on each channel. We prevent this profiling attack by periodically updating $\rho(gk, t_e)$ at every epoch. The epoch duration typically spans many frame transmissions.

5.5 Performance Evaluation of FD-MMAC under Jamming Attacks

In this section, we evaluate the performance of FD-MMAC under various jamming strategies. We studied the relationship between the jamming effort, the jammer's hopping strategy, the adopted PHY-layer parameters, and achievable throughput/goodput. We did not evaluate SP-MMACs and DCC-MMACs, as these protocols were not designed to operate under jamming. In fact, these protocols are expected to achieve zero throughput for a reactive jammer targeting solely the control channel. For FD-MMAC, we focused our attention to a jammer that targets any frame to minimize the sensing period τ_s and therefore, maximize the effective hopping rate. We considered both cryptographically protected and publicly known channel priority list. All simulations were run for 40 sec and results were averaged over 10 simulation runs. We placed senders and destinations in the same collision domain. Senders were always backlogged with data frames. We varied the proba-

[3]A locally agreed group key could be used instead of a global key to prevent the key exposure with a single node compromise. Note that gk is never exposed to cryptanalysis, as it is not used to encrypt data. Key gk could also be refreshed on every epoch t_e.

bility of corrupting a jammed frame by varying the jamming period τ_j (see Section 5.2 for the relation between the frame corruption probability and the jamming period). We used the following metrics to evaluate the FD-MMAC performance under jamming.

(a) *Jamming effort and effective hopping rate*: The jamming effort $\mathcal{A}$ (%) and effective hopping rate $\mathcal{R}$ (channels/ms), as specified in Definitions 1 and 2.

(b) *Normalized throughput*: The average sender throughput, normalized over the per-sender throughput in the absence of jamming.

(c) *Normalized goodput*: The average sender goodput, normalized over the sender goodput in the absence of jamming. We use the Gilbert-Varshamov (GV) lower bound [132] to translate the achieved throughput to goodput, for varying ECC thresholds.

5.5.1 Jamming Effort and Effective Hopping Rate

We evaluated the jamming effort $\mathcal{A}$ and effective hopping rate $\mathcal{R}$ for varying τ_j. Figure 5.7(a) compares the jamming effort of a reactive jammer for 12-channel/3-flow and 12-channel/12-flow scenarios. As expected the jammer expends more effort in the 12-channel/12-flow scenario ($\sim$8%). This is consistent with a jammer who is always active on one channel (jamming one out of 12 channels). Figure 5.7(b) shows the effective hopping rate of the jammer for varying τ_j. As expected, the jammer hops faster under light traffic (12-channel/3-flow) to discover the occupied channels. Our results are consistent with Figure 5.7(a), because the effective hopping rate is roughly the inverse of the jamming effort, for a short sensing period τ_s.

5.5.2 Impact of Error Correction Capability

We evaluated the impact of the ECC on the jammer's effectiveness under the cryptographically protected channel priority list. Figure 5.8(a) shows the normalized

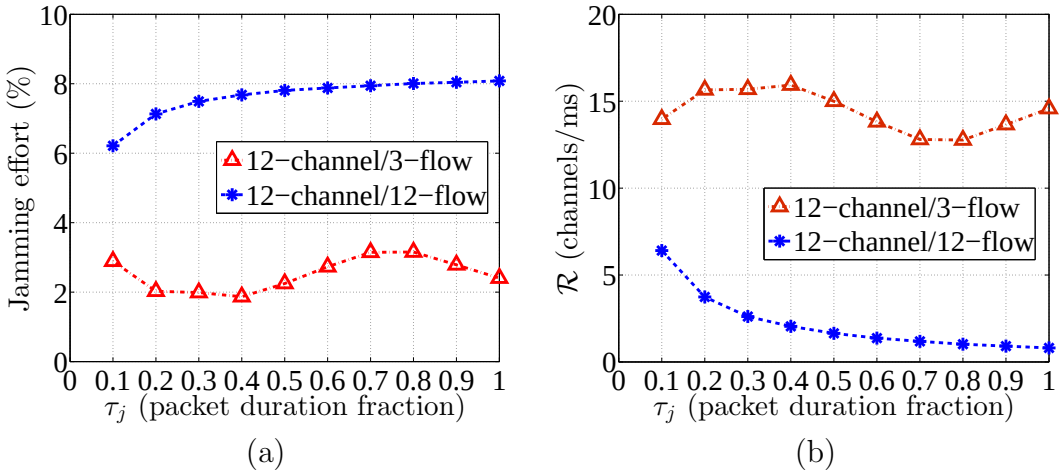


Figure 5.7: (a) Jamming effort (%) when 3 and 12 flows contend over 12 channels, (b) effective hopping rate (channels/ms) when 3 and 12 flows contend over 12 channels.

throughput when 12 senders/destinations contend over 12 channels. It is interesting to note that FD-MMAC maintains high throughput when ECC=0.1 and 0.2 independent of τ_j, due to the tradeoff between τ_j and $\mathcal{R}$. A higher τ_j increases the probability of corrupting jammed frames beyond recovery, but reduces the effective hopping rate, thus reducing the number of frames that can be jammed per unit of time. The reduced effective hopping rate justifies the high throughput for large τ_j values, even when ECC=0. Figure 5.8(b) shows the normalized throughput for a 3-flow scenario. In light traffic conditions, the jammer's effective hopping rate is increased, allowing him to discover the channels occupied by the three flows. Longer jamming periods increase the frame corruption probability and hence, reduce throughput.

5.5.3 Impact of Channel Priority List Knowledge

We further evaluated the impact of jamming when the jammer is aware of the channel priority list used to break ties. In this set of experiments, we set the jammer to switch channels according to its own CST and to break ties in the channel idle times according to the publicly-known channel priority list. Under this switching

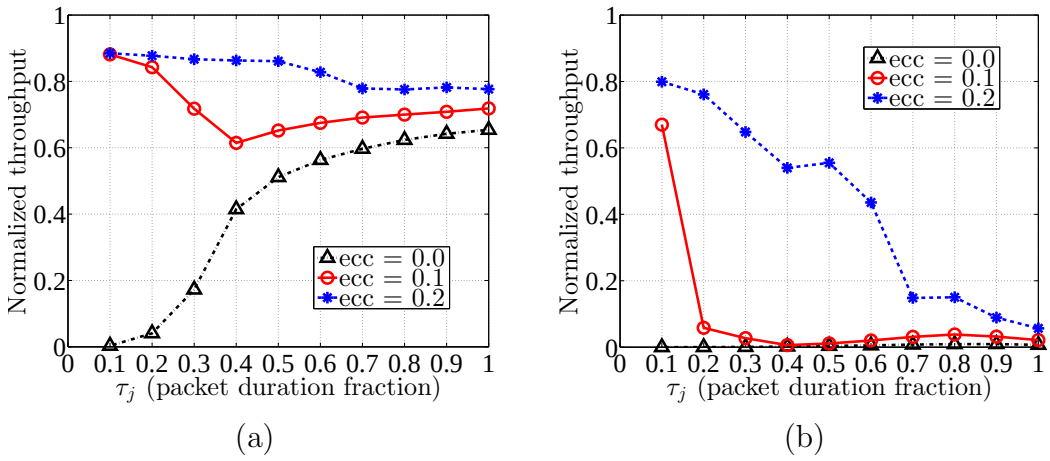


Figure 5.8: (a) Normalized throughput as a function of τ_j for varying ECC for 12 flows contending over 12 channels with secret channel priority list, (b) normalized throughput as a function of τ_j for varying ECC for 3 flows contending over 12 channels with secret channel priority list.

strategy, it is expected that the jammer would discover occupied channels faster. Figure 5.9 shows the normalized throughput for a 12-channel/12-flow scenario and a 12-channel/3-flow scenario, for varying ECC. For the first scenario, we observe that the FD-MMAC throughput is similar to the case where the channel priority list remains secret (Figure 5.8(a)). This is because all channels are occupied and therefore, the jammer's switching strategy does not impact the jammer's success in discovering active transmissions. On the other hand, the jammer improves his effectiveness in the 3-flow scenario, because it scans through the available channel in an order similar to that used by the terminals.

5.5.4 Impact of the Number of Available Channels

We evaluated the impact of the number of available channels. Figure 5.10 shows the normalized throughput for the reactive jammer for varying number of channels and ECC. As expected, the jammer's effectiveness decreases as ECC increases. The jammer performs the worst when six flows contend over six channels. Under the latter scenario, the destination discovery delay and contention levels remain relatively

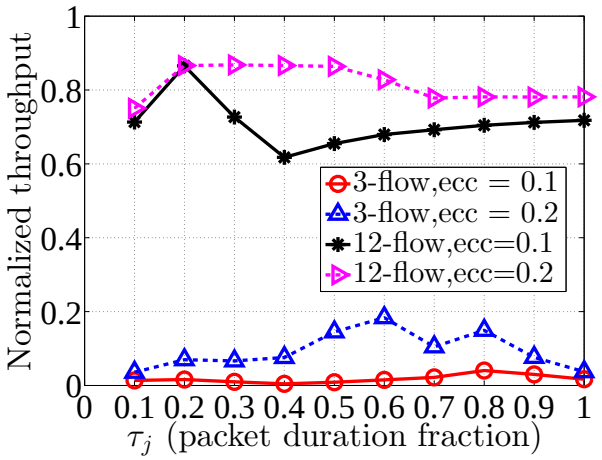


Figure 5.9: Normalized throughput as a function of τ_j for varying ECC for 12 and 3 flows contending over 12 channels with public channel priority list.

low.

5.5.5 Goodput Evaluation

We also evaluated the normalized goodput under the reactive jamming strategy for varying ECC capability. We used the GV lower bound [132] to convert throughput to goodput, by finding an achievable code rate for a given relative distance. Figure 5.11(a) shows the normalized goodput for the scenario simulated in Figure 5.8(a) (the goodput is equal to the throughput, scaled by the code rate). We note that for ECC=0.1, the goodput remains close to 20% for all values of τ_j. Moreover, although ECC=0.2 yields the highest throughput in the experiments of Figure 5.8(a), the achievable goodput is only about 2% due to the low achievable code rate. On the other hand, lack of any ECC protection maximizes the goodput when the jammer dwells on channels for long time periods due to the increased τ_j.

Figure 5.11(b) shows the normalized goodput for the scenario simulated in Figure 5.10. Although ECC=0 yields the lowest throughput in the experiments of Figure 5.10, the performance without coding is higher when the available channels are less than nine. For larger number of available channels, ECC=0.1 yields the best

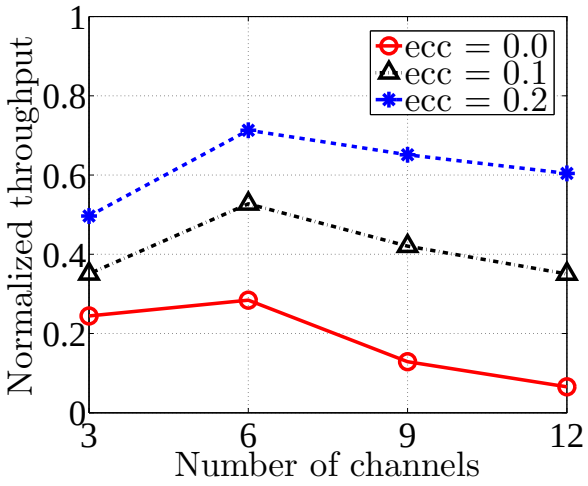


Figure 5.10: Normalized throughput as a function of the number of available channels for a 6-flow scenario, for varying ECC when $\tau_j = 0.4$.

goodput performance. As in the case of Figure 5.11(a), the goodput of ECC=0.2 is the lowest due to the low code rate of coding schemes with such ECC capability.

The goodput results depicted in Figures 5.11(a) and 5.11(b) indicate that the anti-jamming properties of FD-MMAC are primarily due to avoiding the jammer rather than correcting jammed frames. For jammers with low effective hopping rate, eliminating coding overall yields better performance. This strategy maximizes the per-frame goodput for jamming-free frames. For more aggressive jammers with faster effective hopping rates, offering moderate jamming protection using coding yields better goodput results.

5.6 Chapter Summary

We analyzed the anti-jamming properties of FD-MMAC protocol. We defined a comprehensive reactive jamming model for MMAC protocols based on the cross-layer consideration of the PHY and MAC layers and extended the jamming strategies in the multi-channel domain. We showed that by coordinating medium access without requiring a control channel, FD-MMAC effectively mitigates the impact of jamming attacks. We further explored possible improvements on FD-MMAC for improving its

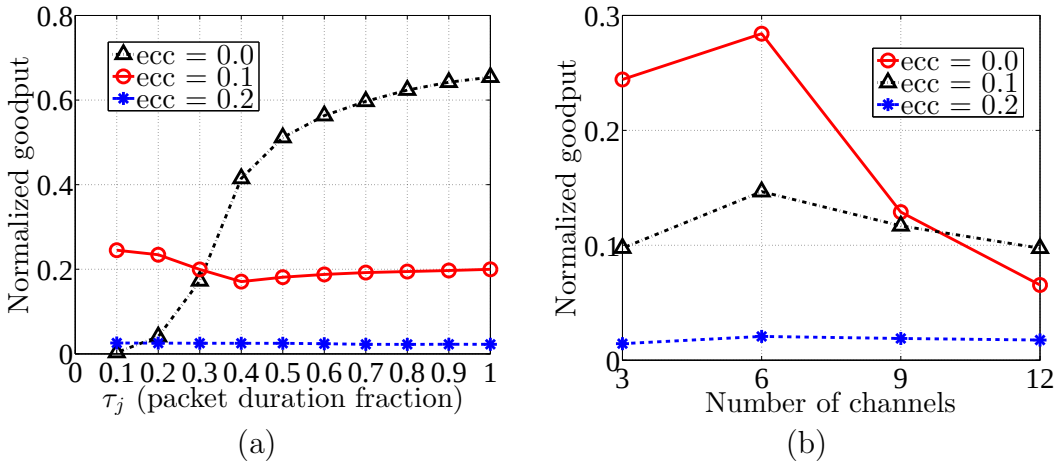


Figure 5.11: (a) Goodput as a function of τ_j for varying ECC capability, (b) goodput as a function of the number of available channels for varying ECC capability.

jamming resilience. Without a default control channel, the jammer must spread his resources over all channels, thus allowing for jamming-free transmission opportunities. We showed that FD-MMAC maintains communications, despite the jammer's efforts.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

6.1 Conclusions

In this dissertation, we developed methods for improving the security and spectral efficiency of medium access in multi-channel wireless networks. In particular, we focused on mechanisms for detecting and mitigating selfish MAC-layer misbehavior, coordinating multi-channel access in a spectral-efficient manner, and enabling data transmissions in the presence of reactive jamming. Our main achievements and finding are summarized as follows.

We studied the problem of MAC layer misbehavior in multi-channel wireless networks. We focused on MMACs following the split-phase and dedicated control channel designs, and identified possible misbehaving strategies that yield a significant throughput advantage to misbehaving terminals. We showed that misbehaving terminals can isolate a significant portion of the available bandwidth by placing multiple reservations on the available channels in a timely manner. We further proposed countermeasures that mitigate the impact of misbehavior and lead to the detection of misbehaving terminals. We also extended our misbehavior analysis to cognitive radio MAC protocols and discussed possible countermeasures for detecting and mitigating the identified vulnerabilities. Finally, we verified the effectiveness of our mitigation mechanisms via extensive packet-level simulations and showed that the throughput of misbehaving terminals is equalized to the throughput of well-behaved terminals.

We investigated the control channel jamming problem to which most existing MMAC protocols are susceptible. We designed a distributed MMAC protocol named

FD-MMAC that exploits FD communications to coordinate channel access in multi-channel domain. FD-MMAC eliminates control signaling over a common control channel so that communications can be maintained in the presence of jamming. It also effectively solves the multi-channel hidden terminal problem and enables parallel non-interfering operations from multi-channel exposed terminals. The FD-MMAC properties are achieved by utilizing an advanced suite of PHY-layer techniques, including SIS, EVM and RSS measurements, and signal correlation techniques. We analytically evaluated the saturation throughput of FD-MMAC using a three-dimensional Markov model. Finally, we experimentally validated the PHY layer techniques employed by our protocol on the NI USRP testbed and measured its performance via simulations. Our simulations showed that FD-MMAC achieves significantly higher throughput compared with prior MMAC designs.

We analyzed anti-jamming properties of MMAC protocols. We defined a comprehensive reactive jamming model based on the cross-layer consideration of the PHY and MAC layers and extended the jamming strategies in the multi-channel domain. We considered the application of these strategies to our proposed FD-MMAC protocol. By coordinating medium access without relying on a control channel, FD-MMAC effectively mitigates the impact of jamming attacks. We further explored possible improvements on FD-MMAC for improving its jamming resilience. Without a default control channel, the jammer must spread his resources over all channels, thus allowing for jamming-free transmission opportunities. We showed that under high load conditions, FD-MMAC achieves significant throughput despite the jammer's efforts. On the other hand, under light traffic conditions, the jammer is effective in targeting the few ongoing flows.

6.2 Future Research Directions

This dissertation presents a full-duplex communication based MAC protocol (FD-MMAC) which coordinates terminals' access to multiple channels in a different manner compared to existing MMACs. To mitigate DoS attacks against the control

channel, destination discovery and channel assignment are performed independently by senders and destinations based on their individual views of channel status, without converging to a common channel. As such, the performance of FD-MMAC is largely dependent on the efficiency of the employed destination discovery mechanism. Therefore, investigation of other time and spectrally efficient sender-destination convergence mechanisms can lead to improved network throughput and lower delay for frame delivery. For example, one other possible solution is to assign a unique predefined hopping sequence to each terminal to facilitate destination discovery. Terminals hop between the available channels according to their hopping sequences instead of the channel priority lists. Convergence of a sender-destination pair occurs when their hopping sequences overlap, which can be guaranteed by design. Upon convergence, one party of the communicating pair may share its hopping sequence with the other one, and the two parties hop synchronously and communicate using the mechanisms proposed in Chapter 4 until data transmission is completed. Such mechanism also does not rely on a common control channel, and thus the anti-jamming properties of FD-MMAC can be sustained. To further protect the destination discovery process against jamming, the hopping sequences can be refreshed on every epoch. The key challenge here is to design proper hopping sequences and mechanism that provide a tradeoff between short convergence time for any two terminals and adequate rendezvous time for a particular pair to perform data transmission. This mechanism is similar to the rendezvous MMAC design as described in Chapter 2, with the exception that the pure rendezvous MMAC is not capable of solving the multi-channel hidden and exposed terminal problem as FD-MMAC does.

REFERENCES

[1] Morgan Stanley. The mobile internet report. *Morgan Stanley Research*, 2009.

[2] IEEE 802.15 for wireless personal area networks (WPAN), http://www.ieee802.org/15/.

[3] IEEE 802.11 for wireless local area networks (WLANs), http://www.ieee802.org/11/.

[4] IEEE 802.16 for wireless metropolitan area networks (WMAN), http://www.ieee802.org/16/.

[5] S. Sesia, I. Toufik, and M. Baker. *LTE: the UMTS long term evolution*. Wiley Online Library, 2009.

[6] 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.

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

[8] J. So and N.H. 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.

[9] S.L. Wu, C.Y. Lin, Y.C. Tseng, and J.L. 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.

[10] K.H. Almotairi and X. Shen. Multichannel medium access control for ad hoc wireless networks. *Wireless Communications and Mobile Computing*, 13(11):1047–1059, 2013.

[11] C. Han, M. Dianati, R. Tafazolli, X. Liu, and X. Shen. A novel distributed asynchronous multichannel MAC scheme for large-scale vehicular ad hoc networks. *IEEE Transactions on Vehicular Technology*, 61(7):3125–3138, 2012.

[12] Yu Wang, Mehul Motani, Hari Krishna Garg, Qian Chen, and Tie Luo. Multichannel directional medium access control for ad hoc networks: A cooperative approach. In *Communications (ICC), 2014 IEEE International Conference on*, pages 53–58. IEEE, 2014.

[13] B. Yang, B. Li, Q. Qu, and Z. Yan. A new multi-channel MAC protocol based on multi-step channel reservation. In *Proc. of the ICSPCC Conference*, pages 603–607, 2014.

[14] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[15] N. Abramson. THE ALOHA SYSTEM: another alternative for computer communications. In *Proc. of the American Federation of Information Processing Societies (AFIPS)*, volume 70, pages 281–285, 1970.

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

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

[18] Khaled H Almotairi and Xuemin Sherman Shen. A distributed multi-channel MAC protocol for ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 14(1):1–13, 2015.

[19] S. L. Wu and J. Y. Yang. A novel channel assignment scheme for improving channel reuse efficiency in multi-channel ad hoc wireless networks. *Computer Communications*, 30(17):3416–3424, 2007.

[20] X. Xing, K Liu, and H. Lu. A multichannel MAC protocol to solve exposed terminal problem in multihop wireless networks. In *Proc. of the CCNC Conference*, pages 1–2, 2009.

[21] K. H. Almotairi and X. S. Shen. Multichannel medium access control for ad hoc wireless networks. *Wireless Communications and Mobile Computing*, pages 1–14, 2011.

[22] T. Shu, S. Cui, and M. Krunz. Medium access control for multi-channel parallel transmission in cognitive radio networks. In *Proc. of the GLOBECOMM Conference*, pages 1–5, 2006.

[23] J. Chen, S.T. Sheu, and C.A. 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.

[24] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt. *Spread Spectrum Communications Handbook*. McGraw-Hill, 2001.

[25] G. Noubir and G. Lin. Low-power DoS attacks in data wireless LANs and countermeasures. *ACM SIGMOBILE Mobile Computing and Communications Review*, 7(3):29–30, 2003.

[26] A. Chan, X. Liu, G. Noubir, and B. Thapa. Broadcast control channel jamming: resilience and identification of traitors. In *Proc. of the ISIT Symposium*, pages 2496–2500, 2007.

[27] P. Tague, M. Li, and R. Poovendran. Probabilistic mitigation of control channel jamming via random key distribution. In *Proc. of the PIMRC Symposium*, pages 1–5, 2007.

[28] L. Lazos, S. Liu, and M. Krunz. Mitigating control-channel jamming attacks in multi-channel ad hoc networks. In *Proc. of the ACM WiSec Conference*, pages 169–180, 2009.

[29] IEEE 802.3 for ethernet, http://www.ieee802.org/3/.

[30] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part ii–the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Transactions on Communications*, 23(12):1417–1433, 1975.

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

[32] M. Raya, J.P. 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.

[33] 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.

[34] 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.

[35] Y. Zhou, D. Wu, and S.M. 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.

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

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

[38] A. Banchs, A. Garcia-Saavedra, and P. Serrano. A game theoretic approach to selfishness in 802.11e WLANs. *IEEE/ACM Transactions on Networking*, 2011.

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

[40] Y. E. Sagduyu, R. Berry, and A. Ephremides. MAC games for distributed wireless network security with incomplete information of selfish and malicious user types. In *Proc. of the GameNets Conference*, pages 130–139, 2009.

[41] Y. E. Sagduyu and A. Ephremides. SINR-based MAC games for selfish and malicious users. In *Proc. of the Information Theory and Applications Workshop*, 2007.

[42] F. Wang, O. Younis, and M. Krunz. Throughput-oriented MAC for mobile ad hoc networks: A game-theoretic approach. *Ad Hoc Networks*, 7(1):98–117, 2009.

[43] M. Ghazvini, N. Movahedinia, and K. Jamshidi. A game theory based contention window adjustment for IEEE 802.11 under heavy load. *International Journal of Communication Networks and Information Security (IJCNIS)*, 5(2), 2013.

[44] A. Acharya, A. Misra, and S. Bansal. MACA-P: A MAC for concurrent transmissions in multi-hop wireless networks. In *Proc. of the PerCom Conference*, pages 505–508, 2003.

[45] A. Muqattash and M. Krunz. POWMAC: A single-channel power-control protocol for throughput enhancement in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(5):1067–1084, 2005.

[46] N. Jain, S.R. 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.

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

[48] O. D. Incel, L. van Hoesel, P. Jansen, and P. Havinga. MC-LMAC: A multi-channel MAC protocol for wireless sensor networks. *Ad Hoc Networks*, 9(1):73–94, 2011.

[49] W.-T. Chen, J.-C. Liu, T.-K. Huang, and Y.-C. Chang. TAMMAC: An adaptive multi-channel MAC protocol for MANETs. *IEEE Transactions on Wireless Communications*, 7(11):4541–4545, 2008.

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

[51] M. Krunz and D. Manzi. Channel access and traffic control for dynamic-spectrum networks with single-transmit, dual-receive radios. *Computer Communications*, 34(8):935–947, 2011.

[52] H. B. Salameh and M. Krunz. Adaptive power-controlled MAC protocols for improved throughput in hardware-constrained cognitive radio networks. *Ad Hoc Networks*, 9(7):1127–1139, 2011.

[53] F. Wang and M. Krunz. Multi-channel spectrum-agile MAC protocol with adaptive load control. In *Proc. of the WoWMoM Symposium*, pages 1–9, 2009.

[54] S. Liu, L. Lazos, and M. Krunz. Thwarting control-channel jamming attacks from inside jammers. *IEEE Transactions on Mobile Computing*, 11(9):1545–1558, 2012.

[55] A. Proano and L. Lazos. Selective jamming attacks in wireless networks. In *Proc. of the ICC Conference*, pages 1–6, 2010.

[56] A. Tzamaloukas and JJ Garcia-Luna-Aceves. Channel-hopping multiple access. In *Proc. of the ICC Conference*, volume 1, pages 415–419, 2000.

[57] W. So, J. Walrand, and J. Mo. McMAC: a parallel rendezvous multi-channel MAC protocol. In *Proc. of the WCNC Conference*, pages 334–339, 2007.

[58] M. Timmers, S. Pollin, A. Dejonghe, L. V. Perre, and F. Catthoor. A distributed multichannel MAC protocol for multihop cognitive radio networks. *IEEE Transactions on Vehicular Technology*, 59(1):446–459, 2010.

[59] J. Zhao, H. Zheng, and G.H. Yang. Spectrum sharing through distributed coordination in dynamic spectrum access networks. *Wireless Communications and Mobile Computing*, 7(9):1061–1075, 2007.

[60] H. Su and X. Zhang. Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks. *IEEE Journal on Selected Areas in Communications*, 26(1):118–129, 2008.

[61] L. Ma, X. Han, and C.C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *Proc. of the IEEE DySPAN Symposium*, pages 203–213, 2005.

[62] K. Bian and J.M. Park. Asynchronous channel hopping for establishing rendezvous in cognitive radio networks. In *Proc. of the INFOCOM Conference*, pages 236–240, 2011.

[63] H. B. Salameh and M. Krunz. Channel access protocols for multihop opportunistic networks: challenges and recent developments. *IEEE Network*, 23(4):14–19, 2009.

[64] H. B. Salameh, M. Krunz, and O. Younis. MAC protocol for opportunistic cognitive radio networks with soft guarantees. *IEEE Transactions on Mobile Computing*, 8(10):1339–1352, 2009.

[65] M. Çakiroglu and A. T. Özcerit. Jamming detection mechanisms for wireless sensor networks. In *Proc. of the ICST InfoScale Conference*, page 4, 2008.

[66] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. *ACM Transactions on Sensor Networks*, 5(1):6, 2009.

[67] W. Xu, K. Ma, W. Trappe, and Y. Zhang. Jamming sensor networks: attack and defense strategies. *IEEE Network*, 20(3):41–47, 2006.

[68] M. Li, I. Koutsopoulos, and R. Poovendran. Optimal jamming attacks and network defense policies in wireless sensor networks. In *Proc. of the INFOCOM Conference*, pages 1307–1315, 2007.

[69] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proc. of the MobiHoc Conference*, pages 46–57, 2005.

[70] L. Lazos and M. Krunz. Selective jamming/dropping insider attacks in wireless mesh networks. *IEEE network*, 25(1):30–34, 2011.

[71] A. D. Wood, J. A. Stankovic, and G. Zhou. DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks. In *Proc. of the SECON Conference*, pages 60–69, 2007.

[72] D. Giustiniano, V. Lenders, J. B. Schmitt, M. Spuhler, and M. Wilhelm. Detection of reactive jamming in DSSS-based wireless networks. In *Proc. of the WiSec Conference*, pages 43–48, 2013.

[73] D. Adamy. *EW 101: A first course in electronic warfare*. Artech House Publishers, 2001.

[74] B. Sklar. *Digital communications*, volume 1099. Prentice-Hall, 2001.

[75] S. Liu, L. Lazos, and M. Krunz. Thwarting inside jamming attacks on wireless broadcast communications. In *Proc. of the ACM WiSec Conference*, pages 29–40, 2011.

[76] S. Liu, L. Lazos, and M. Krunz. Cluster-based control channel allocation in opportunistic cognitive radio networks. *IEEE Transactions on Mobile Computing*, 11(10):1436–1449, 2012.

[77] C. Popper, M. Strasser, and S. Capkun. Anti-jamming broadcast communication using uncoordinated spread spectrum techniques. *IEEE Journal on Selected Areas in Communications*, 28(5):703–715, 2010.

[78] P. Tague, M. Li, and R. Poovendran. Mitigation of control channel jamming under node capture attacks. *IEEE Transactions on Mobile Computing*, 8(9):1221–1234, 2009.

[79] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential DSSS: Jamming-resistant wireless broadcast communication. In *Proc. of the INFOCOM Confernece*, pages 1–9, 2010.

[80] L. Xiao, H. Dai, and P. Ning. Jamming-resistant collaborative broadcast using uncoordinated frequency hopping. *IEEE Transactions on Information Forensics and Security*, 7(1):297–309, 2012.

[81] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang. Defending DSSS-based broadcast communication against insider jammers via delayed seed-disclosure. In *Proc. of the Annual Computer Security Applications Conference (ACSAC)*, pages 367–376, 2010.

[82] Y. W. Law, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. In *Proc. of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 76–88, 2005.

[83] G. Lin and G. Noubir. On link layer denial of service in data wireless LANs. *Wireless Communications and Mobile Computing*, 5(3):273–284, 2005.

[84] W. Xu, T. Wood, W. Trappe, and Y. Zhang. Channel surfing and spatial retreats: Defenses against wireless denial of service. In *Proc. of the WiSe Conference*, pages 80–89, 2004.

[85] X. Liu, G. Noubir, R. Sundaram, and S. Tan. SPREAD: Foiling smart jammers using multi-layer agility. In *Proc. of the INFOCOM Conference*, pages 2536–2540, 2007.

[86] R. Negi and A. Perrig. Jamming analysis of MAC protocols. *Carnegie Mellon Technical Memo*, 2003.

[87] S. Chang, Y. Hu, and N. Laurenti. SimpleMAC: a jamming-resilient mac-layer protocol for wireless channel coordination. In *Proc. of the MOBICOM Conference*, pages 77–88, 2012.

[88] M. Gast. *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc., 2005.

[89] A. L. Toledo and X. Wang. Robust detection of selfish misbehavior in wireless networks. *IEEE Journal on Selected Areas in Communications*, 25(6):1124–1134, 2007.

[90] S. Radosavac, J. S. Baras, and I. Koutsopoulos. A framework for MAC protocol misbehavior detection in wireless networks. In *Proc. of the WiSe Workshop*, pages 33–42, 2005.

[91] A. A. Cárdenas, S. Radosavac, and J. S. Baras. Evaluation of detection algorithms for MAC layer misbehavior: theory and experiments. *IEEE/ACM Transactions on Networking*, 17(2):605–617, 2009.

[92] F. Shi, J. Baek, J. Song, and W. Liu. A novel scheme to prevent MAC layer misbehavior in IEEE 802.11 ad hoc networks. *Telecommunication Systems*, 52(4):2397–2406, 2013.

[93] L. Guang and C. Assi. Mitigating smart selfish MAC layer misbehavior in ad hoc networks. In *Proc. of the WiMob Conference*, pages 116–123, 2006.

[94] M. Raya, I. Aad, J-P Hubaux, and A. El Fawal. DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots. *IEEE Transactions on Mobile Computing*, 5(12):1691–1705, 2006.

[95] V. N. Lolla, L. K. Law, S. V. Krishnamurthy, C. Ravishankar, and D. Manjunath. Detecting MAC layer back-off timer violations in mobile ad hoc networks. In *Proc. of the ICDCS Conference*, pages 63–63, 2006.

[96] S. Djahel, Z. Zhang, F. Nait-Abdesselam, and J. Murphy. Fast and efficient countermeasure for MAC layer misbehavior in MANETs. *IEEE Wireless Communications Letters*, 1(5):540–543, 2012.

[97] M. Chinipardaz and M. Dehghan. DCF/RCB: A new method for detection and punishment of selfish nodes in IEEE 802.11. In *Computer Networks and Distributed Systems*, pages 23–36. 2014.

[98] F. Shi, W. Liu, D. Jin, and J. Song. A cluster-based countermeasure against MAC layer attacks in IEEE 802.11 MANETs. In *Applied Mechanics and Materials*, volume 284, pages 2662–2666, 2013.

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

[100] Y. Zhang, L. Lazos, and W. Kozma. AMD: Audit-based misbehavior detection in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, PP(99):1, 2012.

[101] S. Ganeriwal, L. Balzano, and M. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks*, 4(3):15, 2008.

[102] Q. He, D. Wu, and P. Khosla. SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. In *Proc. of the WCNC conference*, volume 2, pages 825–830, 2004.

[103] S. Soltanali, S. Pirahesh, S. Niksefat, and M. Sabaei. An efficient scheme to motivate cooperation in mobile ad hoc networks. In *Proc. of the ICNS conference*, pages 92–98, 2007.

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

[105] 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.

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

[107] R. Chen, J.M. Park, Y. T. Hou, and J. H. Reed. Toward secure distributed spectrum sensing in cognitive radio networks. *IEEE Communications Magazine*, 46(4):50–55, 2008.

[108] H. Su and X. Zhang. CREAM-MAC: An efficient cognitive radio-enabled multi-channel MAC protocol for wireless networks. In *Proc. of the WoWMoM Symposium*, pages 1–8, 2008.

[109] R. Chen, J.M. Park, and J. H. Reed. Defense against primary user emulation attacks in cognitive radio networks. *IEEE Journal on Selected Areas in Communications*, 26(1):25–37, 2008.

[110] Y. Liu, P. Ning, and H. Dai. Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures. In *Proc. of IEEE 2010 Symposium on Security and Privacy*, pages 286–301, 2010.

[111] S. Chandrashekar and L. Lazos. A primary user authentication system for mobile cognitive radio networks. In *Proc. of 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–5, 2010.

[112] Riverbed. OPNET. `http://www.riverbed.com/`, 2015.

[113] A. Sahai, G. Patel, and A. Sabharwal. Pushing the limits of full-duplex: Design and real-time implementation. Technical Report TREE1104, Rice University, February 2011.

[114] M. Jain, J. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. Practical, real-time, full duplex wireless. In *Proc. of the MobiCom Conference*, pages 301–312, 2011.

[115] J. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving single channel, full duplex wireless communication. In *Proc. of the MobiCom Conference*, pages 1–12, 2010.

[116] Dinesh Bharadia, Emily McMilin, and Sachin Katti. Full duplex radios. In *Proceedings of the ACM SIGCOMM Conference*, pages 375–386, 2013.

[117] B. Radunovic, D. Gunawardena, P. Key, A. Proutiere, N. Singh, V. Balan, and G. Dejean. Rethinking indoor wireless mesh design: Low power, low frequency, full-duplex. In *IEEE Workshop on Wireless Mesh Networks*, pages 1–6, 2010.

[118] S. Gollakota and D. Katabi. ZigZag decoding: combating hidden terminals in wireless networks. In *Proc. of the ACM SIGCOMM Conference*, pages 159–170, 2008.

[119] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. *IEEE/ACM Transactions on Networking*, 17(6):1846–1859, 2009.

[120] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. Strogatz. Distributed synchronization in wireless networks. *IEEE Signal Processing Magazine*, 25(5):81–97, 2008.

[121] N. Singh, D. Gunawardena, A. Proutiere, B. Radunovic, H. Balan, and P. Key. Efficient and fair MAC for wireless networks with self-interference cancellation. In *Proc. of the WiOpt Symposium*, pages 94–101, 2011.

[122] C.-S. Wu and V. Li. Receiver-initiated busy-tone multiple access in packet radio networks. *ACM SIGCOMM Computer Communication Review*, 17(5):336–342, 1987.

[123] K. Xie, K. Xie, S. He, D. Zhang, J. Wen, and J. Lloret. Busy tone-based channel access control for cooperative communication. *Transactions on Emerging Telecommunications Technologies*, 2014.

[124] S. G. Sayed, Y. Yang, and J. Xu. BTAC: A busy tone based cooperative MAC protocol for wireless local area networks. *Mobile Networks and Applications*, 16(1):4–16, 2011.

[125] R. A. Shafik, S. Rahman, and R. Islam. On the extended relationships among EVM, BER and SNR as performance metrics. In *Proc. of the ICECE Conference*, pages 408–411, 2006.

[126] J. Lee, W. Kim, S. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. An experimental study on the capture effect in 802.11a networks. In *Proc. of ACM WiNTECH Conference*, pages 19–26, 2007.

[127] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[128] K.J. Kim, J.S. Park, Y.H. Bae, and B.D. Choi. Performance analysis of a slotted multi-channel MAC protocols for cognitive radio networks. In *Proc. of the 5th International Conference on Queueing Theory and Network Applications*, pages 148–155, 2010.

[129] C. Hu, H. Kim, and J.C. Hou. An analysis of the binary exponential backoff algorithm in distributed MAC protocols. Technical report, University of Illinois at Urbana-Champaign, 2005.

[130] NI. National instruments. `http://www.ni.com/usrp/`, 2015.

[131] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LAN's. In *Proc. of ACM SIGCOMM Conference*, volume 24, pages 212–225, 1994.

[132] W. Ryan and S. Lin. *Channel Codes: Classical and Modern*. Cambridge University Press, 2009.