Multi-channel Medium Access Without Control Channels: A Full Duplex MAC Design

Yan Zhang, Loukas Lazos, Kai Chen, Bocan Hu, and Swetha Shivaramaiah
Dept. of Electrical and Computer Engineering, University of Arizona
Email: {yanzhang, llazos, chenkai, bocanhu, sshivaramaiah}@email.arizona.edu

Abstract—We address the problem of improving the throughput and security of multi-channel MAC (MMAC) protocols. 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 with 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 mitigates denial-of-service 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. Finally, we analyze the resilience of FD-MMAC to reactive jamming attacks.

Index Terms—Wireless networks, media access control, multi channel, full duplex, security, jamming, denial-of-service.

1 Introduction

Modern wireless technologies accommodate parallel transmissions over orthogonal frequency bands, herein referred to as *channels*, to alleviate contention and interference. Access to those channels is coordinated at the medium access control (MAC) layer, which is also responsible for reliable frame delivery, destination discovery, contention management, and load balancing. In the simplest form of medium access control (e.g., 802.11a/b/g), terminals remain tuned to the same channels for long periods of time. As a result, some channels become saturated while others remain underutilized.

The design of efficient multi-channel MAC (MMAC) protocols poses significant fundamental challenges [1], [24], [27], [31]–[33], [35]. These challenges are unique in the multi-channel domain and thus absent in the single-channel MAC counterparts. First, senders must employ low-overhead mechanisms for discovering the resident channel of their respective destinations. Second, parallel transmissions must be efficiently distributed over all channels to alleviate contention. Load balancing across channels must be achieved so that capacity of those channels is fully exploited. In a single-channel domain, no such destination discovery nor load balancing is necessary as terminals share a single channel. Furthermore, the availability of multiple channels provides an opportunity to design MAC protocols that are resilient to DoS attacks. Single channel MACs are hard to defend against these types of attacks, because only one channel is available. However, the majority of existing MMAC protocols rely on the Common Control Channel (CCC) for addressing the above challenges. This leaves them vulnerable to the same types of attacks as in single-channel MACs, as the CCC constitutes a single point of failure. Finally, most existing MMAC protocols fail to address the multi-channel exposed terminal problem [32], whereby a sender switching to a busy channel, but being exposed to a transmitter, cannot proceed with a parallel non-interfering transmission.

A preliminary version of this work appeared in INFOCOM 2014.

To improve the spectral efficiency and jamming resilience of MMAC protocols, we exploit recent advances in full duplex (FD) communications over a *single* channel [3], [6], [10], [21]. 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 self interference (up to 110 dB) [3], using a combination of antenna-based SIS, signal inversion, and RF/digital interference cancellation. The integration of FD communications provides unique opportunities for reducing the control overhead, increasing the spatial channel reuse, and improving resilience to jamming.

Our Contributions: 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 control signaling for coordinating transmissions.
- It increases the spatial channel reuse by enabling the operation of multi-channel exposed terminals.
- It mitigates jamming attacks by eliminating the use of default control channels.

FD communications have already been integrated to single-channel MAC protocols [7], [26], [29], [36]. These works use the same carrier sensing principle as FD-MMAC for avoiding hidden terminals. The destination operates in FD mode to inform hidden terminals of an eminent reception. Moreover, the FD mode enables secondary transmissions, either from the primary sender, or an exposed terminal [36]. With respect to FD-MAC protocols, FD-MMAC integrates the following functions, which are unique to the multi-channel setting.

- It enables the autonomous discovery of the destination's resident channel. This function is unnecessary in the single-channel domain, as all terminals reside on the same channel.
- It performs load balancing by distributing parallel transmissions over all channels without explicit coordination.

- It extends the single-channel backoff mechanism to a global and fair multi-channel backoff mechanism.

We further perform various improvements with respect to state-of-the-art FD-MAC protocols. In particular, we integrate an advanced suite of PHY-layer techniques, including error vector magnitude measurements, received power measurements, and signal correlation to improve spectral efficiency. These methods allow the initiation of exposed terminal transmissions at any time, without requiring synchronization with the primary transmission. This is particularly useful in multi-channel networks, because an exposed terminal may switch to a busy channel *at any time*. We exploit signal correlation for detecting ACKs without requiring decoding. Therefore, secondary transmissions are not limited to a fixed data frame length and the respective ACKs need not be transmitted synchronously, as in single-channel FD-MACs [26], [36]. We theoretically analyze the saturation throughput of FD-MMAC and verify our analysis via simulations.

Paper Organization: In Section 2, we discuss related work. The system model is described in Section 3. Section 4 describes the FD carrier sensing operation. In Section 5, we address the multi-channel hidden and exposed terminal problems. In Section 6, we present the operational details of FD-MMAC. In Section 7, we analytically evaluate the saturation throughput of FD-MMAC. We study the anti-jamming properties of FD-MMAC in Section 8. We compare the performance of FD-MMAC with existing MMAC designs in Section 9 and conclude in Section 10.

2 Motivation - Related Work

MMAC protocols: MMAC protocols can be broadly categorized to three classes: (a) split-phase [27], [30], [33], (b) dedicated control channel (DCC) [1], [31], [34], and (c) rendezvous [2], [9], [28]. In split-phase MMACs, time is divided to alternating control and data phases. During the control phase, all terminals converge to a default channel to negotiate the channel assignment for the upcoming data phase. During the data phase, terminals exchange data on the assigned channels. In DCC MMACs, terminals are equipped with at least two radios. One radio is always tuned to a DCC to perform channel assignment and virtual carrier sensing. Other radios switch between the remaining channels to perform data transmissions. Finally, in rendezvous protocols, nodes hop between channels using predefined hopping sequences. These sequences are designed to enable the sender-destination rendezvous within a fixed time period.

The multi-channel hidden terminal problem: Consider the topology of Fig. 1. Let A and B reside on channel f_1, while 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 P_A to B. Let the transmission of P_A start at 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 $t_3 < t_1$, C causes a collision at B.

Although hidden terminals may appear less frequently when terminals are distributed over multiple channels, they can still

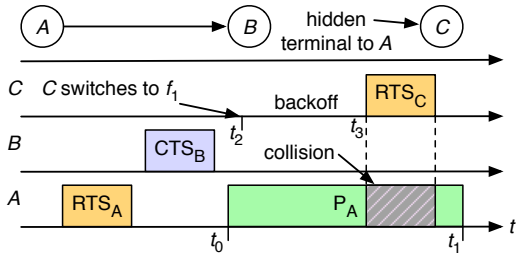


Fig. 1: The multi-channel hidden terminal problem.

cause significant throughput degradation in applications with high user density such as deployments in metropolitan areas, conference halls, stadiums, etc. Split-phase and DCC MMACs avoid multi-channel hidden terminals by performing channel negotiations on a default control channel. However, no data transmissions take place during channel negotiations, thus decreasing the overall spectral efficiency. In addition, from a security standpoint, the control channel constitutes a single point of failure [14], [18]. An adversary launching a denial-of-service (DoS) attack on the control channel effectively denies communication on all channels. In FD-MMAC, we address the multi-channel hidden terminal problem without incurring any control overhead. Moreover, we eliminate the control channel.

The multi-channel exposed terminal problem: Multi-channel exposed terminals lose transmission opportunities when switching to a busy channel in the middle of a data transmission. Referring to the topology of Fig. 1, assume that B transmits a data packet 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.

Destination discovery and load balancing: In a multi-channel setting, a destination must be discoverable by a candidate sender. Moreover, transmissions must be distributed over all channels to balance the traffic load and alleviate contention. In DCC and split-phase MMAC protocols, these two functions are coordinated by exchanging control messages, thus decreasing the spectral efficiency. Rendezvous protocols utilize a priori known hopping sequences to reduce the coordination overhead. However, in some designs, an initial discovery delay is incurred until the sender's and destination's hopping sequences overlap [2]. In FD-MMAC, destination discovery and load balancing are achieved without the exchange of control information. Nodes independently switch to idle channels by tracking the state of each channel.

Full-duplex single-channel MACs: Prior works have explored the use of FD in single-channel networks [7], [26], [29], [36]. Zhou et al. [36] proposed the RCTC protocol that enables three operation modes: (1) a bi-directional transmission mode, where a primary receiver simultaneously transmits a packet to the primary transmitter, (2) a secondary transmission mode in which the primary receiver simultaneously transmits to a secondary receiver, and (3) a unidirectional mode in which a busy tone is transmitted by the primary receiver. In the latter mode, an exposed terminal can operate in parallel with the primary transmitter. The mode selection is enabled by the transmission of PHY-layer signatures before data transmissions are initiated. The same operation modes are adopted by the ContraFlow

protocol proposed by Singh et al. [26].

Choi et al. proposed an FD-MAC protocol that eliminates hidden terminals [7]. To do so, the receiver transmits an ACK upon successfully decoding the header of a data frame. If a collision is detected, the receiver transmits periodic NACKs until the sender stops its transmission. This method does not extend to multi-channel networks. A hidden terminal could switch to a busy channel after the ACK has been transmitted by the receiver, thus being unaware of the ongoing reception.

The FD-MMAC protocol integrates additional functions, which are unique to the multi-channel setting. These include destination discovery and load balancing across multiple channels. Such functions are implemented by instituting a distributed and autonomous destination discovery mechanism, without control channels, and by establishing a global backoff mechanism. Moreover, FD-MMAC integrates various improvements with respect to prior FD-MACs such as the elimination of hidden terminals for secondary transmissions, the transmission of arbitrary length packets for both primary and secondary transmissions, and elimination of synchronization requirements for the respective ACKs.

3 System Model

We consider a wireless network that operates over N orthogonal channels $f_1, f_2, ..., f_N$. Terminals are equipped with a single radio transceiver and are assumed to be time-synchronized to a common slotted system. Terminals can operate in FD mode by applying a combination of analog and digital SIS techniques [3], [6], [10], [21].

Moreover, 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 [8]. They have also been used for early collision avoidance, whereby the receiver transmits a special notification if it experiences a collision [22]. The concept of signal correlation is shown in Fig. 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 $P_A + P_B + w$. 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 the correlation for the frequency offset of B, which can be estimated in advance from prior frame exchanges between B and C.

4 FD Carrier Sensing

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*. FD carrier sensing extends beyond the estimation of the carrier state (idle or busy) to determining a terminal's operational

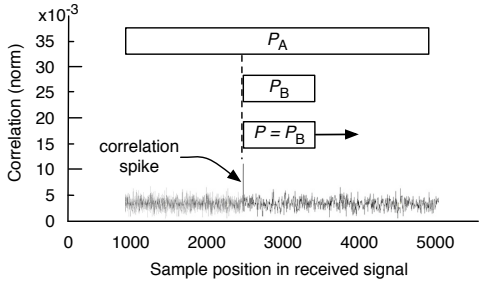


Fig. 2: Detecting a known bit pattern P when two frames collide using the signal correlation technique.

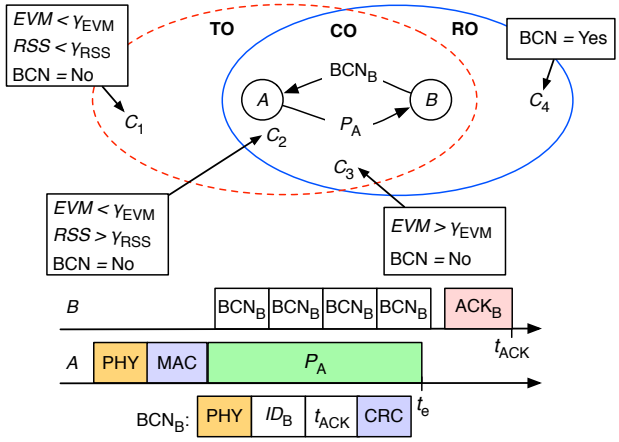


Table 1: Region Classification Rules

	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

Fig. 3: The three regions for a terminal C relative to a transmission $A \rightarrow B$.

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.

Operation in FD mode: An example of the FD operation is shown in Fig. 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 [26] 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).

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 Fig. 3: (a) the receiver-only (RO) region, (b) the collision region (CO), and (c) the transmitter-only (TO) region. Referring to Fig. 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.

Practical issues: Several practical issues complicate the proposed region classification rules. First, when C is in the TO region (position C_1 in Fig. 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, node C computes the error vector magnitude (EVM) on the received symbols. The RMS EVM value (dB) is given by [23]:

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

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 [23]. Note that for arbitrary frames, the $\mathbf{s}[k]$'s *are not known to the receiver*. To compute an EVM estimate using formula (1), 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. Finally, we set the window size n equal to the duration of two BCN frames. This is because 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. When C is in the CO region but very close to A (position C_2 in Fig. 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.. 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 region classification rules used by FD-MMAC are summarized in Table 1. In Section 9, we perform testbed experiments to determine γ_{EVM} and γ_{RSS}.

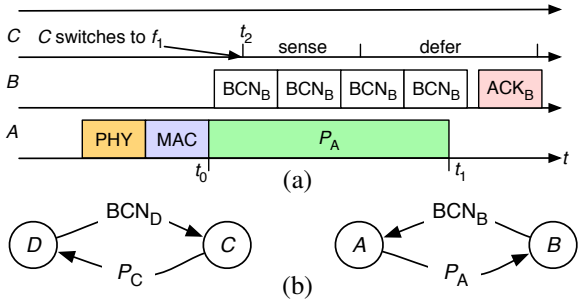


Fig. 4: (a) Combating the multi-channel hidden terminal problem, (b) exposed terminal operation. Transmission $C \rightarrow D$ occurs in parallel with transmission $A \rightarrow B$ on the same channel.

5 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 Fig. 4(a), for the topology of Fig. 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 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 $t_0 < t_2 < t_1$. Terminal C senses f_1 to be busy due to the BCN$_B$ transmissions and defers from transmission.

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 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. Our method relies in the absence of BCNs, as opposed to detecting NACKs, as proposed in [22].

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 Fig. 4(b).

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 Fig. 4(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. In prior works, this problem is

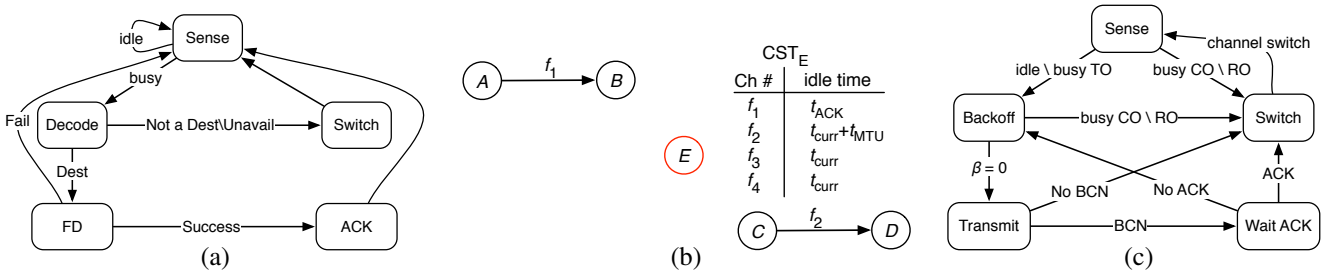


Fig. 5: (a) The state diagram of an FD-MMAC destination, (b) the CST table for node E, and (c) the state diagram of an FD-MMAC sender.

avoided by synchronizing the ACKs from the primary and secondary receiver and equalizing the data frame lengths [36]. In FD-MMAC, senders detect BCNs and ACKs using signal correlation. 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.

6 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 jamming attacks on 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.

6.1 Destination Operation

When a terminal's transmission queue is empty, it operates as a destination. Referring to the state diagram of Fig. 5(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.

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 region (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). If the resident channel is involved in the tie, it is always given the highest priority to prevent unnecessary channel switches. 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.

As an example, consider the topology of Fig. 5(b). 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 if the carrier sensing and channel switching delays are kept low. Carrier sensing is in the order of one slot in popular 802.11 protocols [11], whereas channel switching widely varies with the hardware implementation. Platforms optimized to swiftly switching channels can achieve a switching delay in the order of 50μs [11], whereas devices aimed at infrequent switching could exhibit switching delays up to 0.5s. Assuming a packet transmission time of approximately 350μs (for a packet of maximum size at 56Mbps), a terminal can sense 4 channels (at 75μs carrier sensing and switching delay), before a packet transmission is completed.

6.2 Sender Operation

For the sender, we adapt the CSMA backoff mechanism to the multi-channel environment. A sender 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. In Appendix 1.1, we present an operational example of FD-MMAC that illustrates the fairness achieved by the global backoff process. Moreover, we analyze the protocol fairness via simulations in Section 9.2. Referring to Fig. 5(c), 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. 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.

For rule # 3, a sender experiencing a collision during the "Wait ACK" state, does not switch channels. This is because a transition to the "Wait ACK" state occurs only if a BCN from the intended destination is successfully received at the sender. In this case, the sender is aware that the destination resides on the same channel and has no reason to switch channels.

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 follows the same rules as the destination in updating the CST, except rule #1. If a sender transmitted P on f_i, but did not receive a BCN, 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. On the other hand, if the data frame was successfully received, the sender sets the idle time of f_i to t_{curr}. This prevents an unnecessary channel switch every time a transmission is successful. Even if the idle time of the resident channel is later than t_{curr}, a channel switch will be prevented if this channel has the earliest idle time in the CST. To ease the understanding of the FD-MMAC protocol, we present two operational examples in Appendix 1.2 and 1.3.

7 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 formulations and assumptions to those proposed for single-channel [5] and multi-channel MACs [12]. 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, B_n and S_n are non-Markovian, as they depend on the channel and transmission history of the sender. To ease our analysis, we assume that a sender switches to f_i with fixed probability $p(f_i)$, which is independent of the 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* [5]. 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 9). 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:

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

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:

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

Based on the FD-MMAC state diagram of Fig. 5(c), 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:

$$p_{(u,k,\beta-1|u,k,\beta)} = p_I, \\ 1 \leq u \leq N,\ 0 \leq k \leq m,\ 1 \leq \beta < cw_k. \tag{3}$$

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

$$p_{(u',k,\beta|u,k,\beta)} = (p_S+p_C)\,p(f_{u'}), \\ 1 \leq u,u' \leq N,\ u \neq u',\ 0 \leq k \leq m,\ 1 \leq \beta < cw_k. \tag{4}$$

3) The tagged sender transmits a data packet 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 $\frac{1}{cw_0}$).

$$p_{(u,0,\beta|u,k,0)} = \frac{p_I \cdot p_d(f_u)}{cw_0}, \\ 1 \leq u \leq N,\ 0 \leq k \leq m,\ 0 \leq \beta < cw_0 \tag{5}$$

4) The tagged sender transmits a data packet 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

$$\begin{aligned} p_{(u',k,1|u,k,0)} &= p(f_{u'})\,(1-p_d(f_u)+p_d(f_u)(p_S+p_C)) \\ &= p(f_{u'})\,(1-p_d(f_u)p_I) \\ &\quad 1 \leq u,u' \leq N,\ 0 \leq k \leq m. \end{aligned} \tag{6}$$

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 by

$$\mathbf{P} = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & N \end{array} \\ \begin{pmatrix} D^{11} & Z^{12} & \cdots & Z^{1N} \\ Z^{21} & D^{22} & \cdots & Z^{2N} \\ \vdots & \vdots & \ddots & \vdots \\ Z^{N1} & Z^{N2} & \cdots & D^{NN} \end{pmatrix} \end{array}, \tag{7}$$

where D^{ii} $(i=1,2,...,N)$ and Z^{ij} $(1 \leq i,j \leq N,\ i \neq j)$ 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} $(1 \leq i,j \leq N,\ i \neq j)$ 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 1.

Proposition 1: The aggregate FD-MMAC throughput for M terminals contending over N channels under saturation is:

$$T = \sum_{u=1}^{N} \frac{p'_S \cdot p_d(f_u) \cdot payload}{E[\tau_{slot}(f_u)]}, \tag{8}$$

where $E[\tau_{slot}(f_u)]$ denotes the average slot duration for f_u, and *payload* denotes the data frame length in bits.

Proof: The proof is provided in Appendix 2. □

8 Anti-jamming Properties of FD-MMAC

A common attack on spectrum access is to intentionally interfere with wireless transmissions. For the split phase and DCC MMAC designs, communications on all channels can be easily denied by jamming the control channel. Jamming attacks can be mitigated by spread spectrum (SS) communications [25]. However, for several popular wireless technologies (e.g., the 802.xx family), the available spectrum is not sufficient to assign non-overlapping sequences to many contending terminals. Moreover, the disclosure of commonly shared sequences used to broadcast over the control channel nullifies the SS gains. Several anti-jamming schemes that do not rely on common sequences have recently been proposed [13]–[15]. However, these schemes are meant as emergency mechanisms and exhibit poor spectral efficiency. On the other hand, FD-MMAC has inherent anti-jamming properties due to the elimination of the control channel, while maintaining high throughput in the absence of jamming. In this section, we define a comprehensive reactive jamming model for MMAC protocols and study the anti-jamming properties of FD-MMAC.

8.1 Jamming Model

We consider a fast-hopping jammer, denoted by J, with negligible channel switching delay. 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*. This strategy is captured by the following metrics.

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

$$\mathcal{A} = \frac{1}{NT}\sum_{i=1}^{N}\alpha(f_i), \tag{9}$$

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, $\mathcal{R} = \frac{1}{\tau_d}$.

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:

$$\mathbf{r}[k] = \mathbf{H}\mathbf{s}[k] + \mathbf{G}\mathbf{j}[k] + \mathbf{w}[k], \tag{10}$$

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 demodulation, the receiver compensates for $\mathbf{H}$ and attempts to recover $\mathbf{s}$ by mapping $\mathbf{r}$ to the closest symbol $\mathbf{s}'$. However, $\mathbf{s}'$ may differ from $\mathbf{s}$, due to $\mathbf{G}\mathbf{j}$ and $\mathbf{w}$.

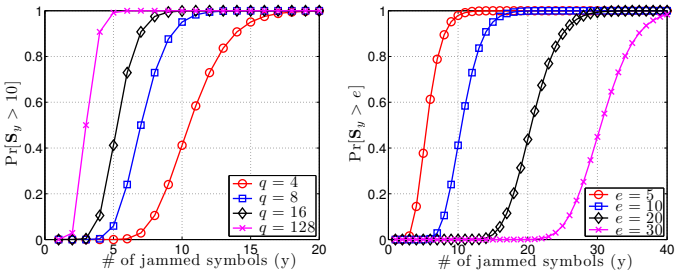
The jammer could attempt to design $\mathbf{j}$ such that $\mathbf{r}$ is mapped to a desired symbol $\mathbf{s}'$. However, to craft $\mathbf{j}$, the jammer must know a priori $\mathbf{H}$, $\mathbf{G}$, $\mathbf{w}$, and the transmitted symbol $\mathbf{s}$. From these parameters, $\mathbf{s}$ cannot be known before it is transmitted, while the rest vary with time. Therefore, the jammer has no advantage in constructing $\mathbf{j}$ to fall within a specific region in the constellation diagram. Given independent values for $\mathbf{G}\mathbf{j}$, $\mathbf{w}$, $\mathbf{s}$, 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 2.

Proposition 2: For a q-order modulation, the PMF of $\mathbf{X}$ is:

$$\Pr[\mathbf{X} = x] = \frac{1}{q}\binom{\log_2 q}{x}. \tag{11}$$

Proof: The proof is provided in Appendix 3. □

Using Proposition 2, we can compute the probability of corrupting P, when P is jammed for τ_j symbol periods.



(a) (b)

Fig. 6: 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.

Proposition 3: 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 (11). The complementary cumulative probability mass function (CCMF) of $\mathbf{S_y}$ is:

$$\Pr[\mathbf{S_y} > e] = 1 - \left(\frac{1}{q}\right)^{y}\sum_{i=0}^{e}\binom{y\log_2 q}{i}. \tag{12}$$

Proof: The proof is provided in Appendix 4. □

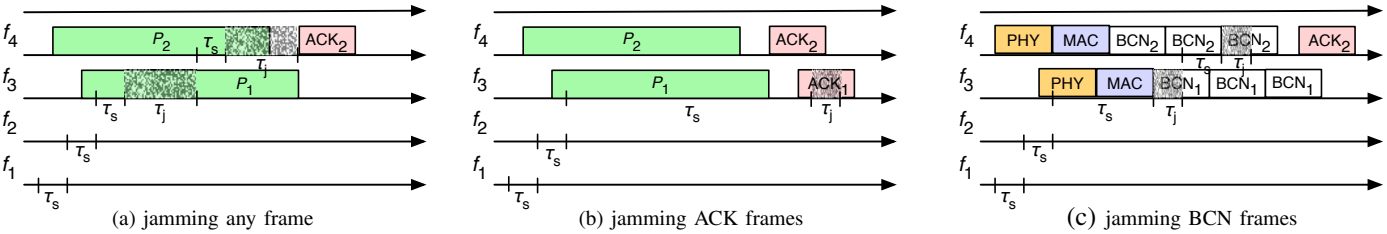
Using Proposition 3, the jammer can determine the jamming period $\tau_j = yT_s$, such that a frame P protected from up to e errors and modulated with a q-order modulation is corrupted beyond recovery with a desired probability. In Fig. 6(a), we show the probability of corrupting P as a function of number of jammed symbols (y) and for different modulation orders, when $e = 10$. Based on Fig. 6(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 Fig. 6(b), we show the CCMF of $\mathbf{S_y}$ as a function of y for different ECC thresholds, when $q = 4$.

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

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 Fig. 7(a), we show a jammer applying the reactive jamming strategy independent of the transmitted packet. 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 fewer than e bits were jammed.

Jamming ACK frames: The jammer can choose to target the ACKs arriving at the sender. The ACK jamming strategy is presented in Fig. 7(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



(a) jamming any frame (b) jamming ACK frames (c) jamming BCN frames

Fig. 7: Jamming attacks on FD-MMAC.

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.

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. Fig. 7(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 4: The probability of jamming the first BCN of duration τ_{BCN} when switching to a busy channel is:

$$\Pr[\text{BCN} = jam] = 1 - \frac{\tau_{PHY} + \tau_{MAC} + \tau_{BCN} - \tau_j}{\tau_p}. \qquad (13)$$

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

Proof: The proof is provided in Appendix 5. □

Note that the sender may still be able to detect a jammed BCN using signal correlation. Moreover, if the jammer misses the first BCN, he has no incentive of jamming subsequent BCNs. This is because thosBCNs are only used to occupy the channel in the receiver's collision domain to avoid hidden terminals. The superposition of a jamming signal with the BCN maintains the busy channel state.

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 6.1. 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.

8.3 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.

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 Fig. 8. Codewords are arranged row-wise and symbols are transmitted column-wise.

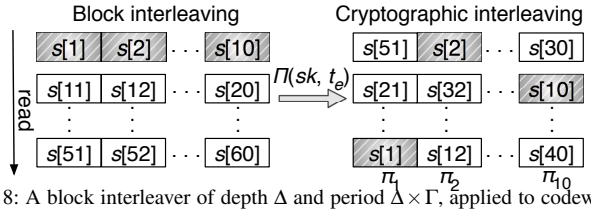
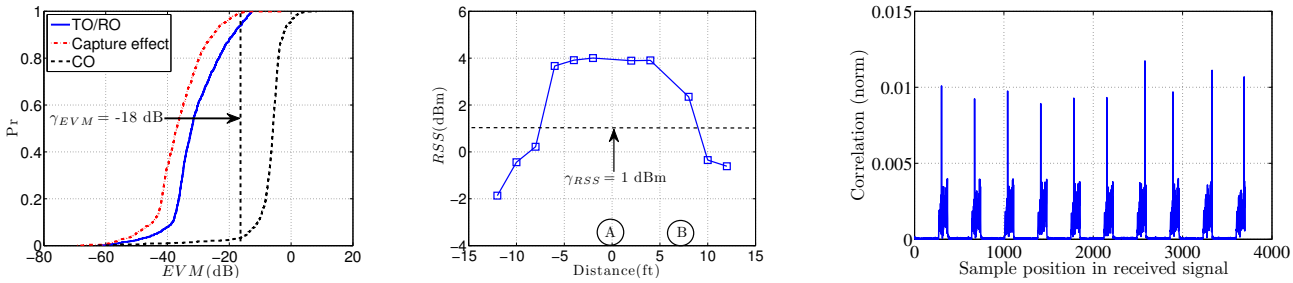


Fig. 8: A block interleaver of depth Δ and period $\Delta \times \Gamma$, applied to codewords of length Γ symbols.

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 packet. 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. 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 [17]. 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



(a) (b) (c)

Fig. 9: (a) The *EVM* CDF at the RO, CO, and TO regions, (b) average *RSS* at different positions, (c) normalized correlation values for 10 BCN packets.

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 Fig. 8. Under cryptographic interleaving, the required jamming period for corrupting y symbols of the same codeword is given by the following proposition.

Proposition 5: 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: The proof is provided in Appendix 6. □

By combining Propositions 3 and 5, the jammer can determine the appropriate jamming period τ_j that leads to the irrecoverable corruption of a frame *P*.

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 or locally shared secret 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 $\rho(gk, t_e)$ 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 spans many frame transmissions.

9 Testbed Experiments and Simulations

We performed testbed experimentations to validate the PHY-layer techniques used by FD-MMAC. Moreover, we used packet level simulations to measure the FD-MMAC performance under various topologies.

9.1 Validation of the PHY-layer Techniques

Testbed: We performed our experiments on NI USRPs devices [16], over the 2.4 GHz band. The signal processing blocks were implemented in Labview [16]. 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, we replicated the topology of Fig. 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 Fig. 3 and measured the EVM, RSS, and the decodability of BCNs. Fig. 9(a) 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).

For position C_2, $EVM < \gamma_{EVM}$ due to the capture effect. To avoid the classification of a terminal located at C_2 as an exposed terminal, we use the mean *RSS* value. Fig. 9(b) shows the mean RSS value for different locations, averaged over the experiment duration. For the *RSS* classification rule, we set γ_{RSS} to 1dBm. We observe that for 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.

Signal correlation: We experimentally evaluated the signal correlation technique for the exposed terminal topology of Fig. 4(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. Fig. 9(c) 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 2.

Table 2: Fraction of Detected BCN Packets

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

Table 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.

9.2 Performance Evaluation of FD-MMAC

We evaluated the performance of FD-MMAC via packet-level simulations using OPNET^{TM} [19]. We simulated three topologies: (a) a single-hop network with n contending flows, where n varied from three to twelve, (b) n flows operating in the presence of one terminal exposed to all other senders and one terminal hidden from all other senders, and (c) a topology of 48 terminals randomly deployed in a 160m×160m square area (multiple partially overlapping collision domains). The channel capacity was set to 2Mbps. The frame arrival process at each sender followed the Poisson distribution with an average arrival rate of λ frames per second. 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 split-phase MMAC (SP-MMAC) in [27] and the DCC MMAC in [31]. The control and data phase of SP-MMAC were set to 20ms and 80ms, respectively. Figures 10(a), 10(b), and 10(c) 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 in high-load (total capacity of the three channels is 6Mbps). Figures 10(d) and 10(e) show the average per-flow throughput of FD-MMAC and SP-MMAC. FD-MMAC significantly outperforms SP-MMAC in high-load conditions.

In the second set of experiments, we placed five flows $S_1 \rightarrow D_1, \ldots, S_5 \rightarrow D_5$ in the main collision domain, while S_E operated as an exposed terminal to S_1-S_5 and S_H operated as a hidden 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. Fig. 10(f) shows the aggregate throughput for varying λ. We observe that in high-load, FD-MMAC increases the aggregate throughput of SP-MMAC and DCC MMAC by 62% under ideal operating conditions. The throughput improvement drops to 49% when 5% of BCNs and ACKs are lost. The superior performance of FD-MMAC is attributed to the parallel operation of S_E with any of the S_1-S_5 and the elimination of the control channel overhead. In fact, the individual throughput of S_E was about 63% higher than the throughput of S_1-S_5 because S_E did not contend with other senders. The addition of S_H increases the contention within the main collision domain. This is because S_1 - S_5 cannot transmit concurrently with S_H on the same channel.

In the last set of experiments, we evaluated the throughput improvements of FD-MMAC in a random topology of 48 terminals, deployed in a 160m×160m square area. This topology simulated multiple partially overlapping collision domains. Fig. 10(g) shows the average per-flow throughput for varying λ. We observe that FD-MMAC offers scalable performance, when extended to larger topologies. Each sender achieves on average 53% higher throughput compared to SP-MMAC and 67% higher throughput compared to DCC-MMAC. The throughput gains are attributed to the higher spectral efficiency of FD-MMAC due to the elimination of signaling for channel negotiation and virtual carrier sensing and the parallel operation of exposed terminals.

Validation of the theoretical throughput analysis: To validate the Markov model proposed in Section 7, we compared the saturation throughput computed via Proposition 1 with the throughput measured in simulations. 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 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 Fig. 10(h) 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 7. Nevertheless, the two theoretical scenarios yield useful best-case and worst-case performance indicators.

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 used Jain's *Fairness Index* (FI). 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

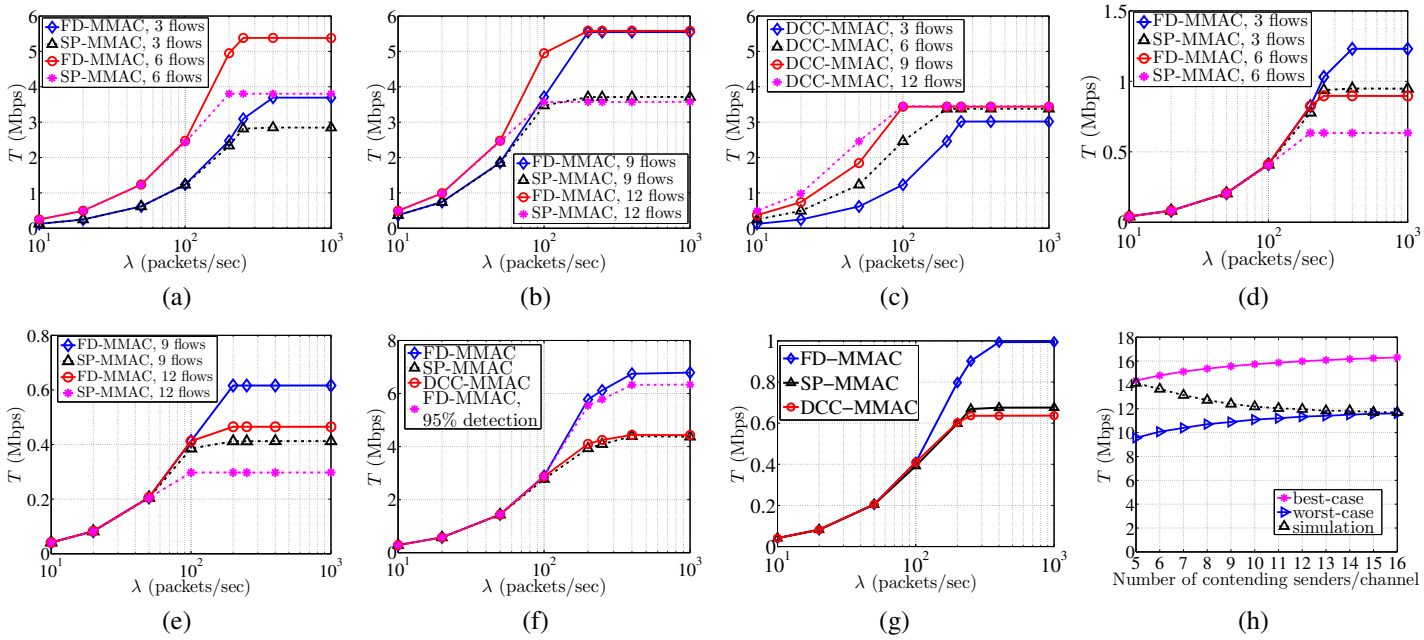


Fig. 10: (a),(b),(c) Aggregate T of FD-MMAC, SP-MMAC, and DCC-MMAC when 3, 6, 9, and 12 flows are within same collision domain, (d),(e) per-flow average T for FD-MMAC and SP-MMAC when 3, 6, 9, 12 flows are within same collision domain, (f) aggregate T in the presence of one exposed and one hidden terminal, (g) per-flow average T for a random topology of 48 terminals spanning 6 partially overlapping collision domains, (h) comparison of the analytical aggregate throughput with the simulated throughput.

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 0.96 for the topology of 48 terminals randomly deployed in the square area, 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 [4]. This is because collisions are rare due to the use of BCNs for virtual carrier sensing. 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*) in high-load:

$$LBI = \frac{(\sum_{i=1}^{N} T_{f_i})^2}{N \times \sum_{i=1}^{N} (T_{f_i})^2}, \tag{14}$$

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 0.99 for the random topology of 48 terminals.

9.3 Evaluation of Jamming Impact on FD-MMAC

We studied the relationship between the jamming effort, the jammer's hopping strategy, the adopted PHY-layer parameters, and the achievable throughput/goodput. We did not evaluate split-phase and DCC MMACs, as these protocols have 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. We placed senders and destinations in the same collision domain. Senders were always backlogged with data frames. We varied the probability of corrupting a jammed frame by varying the jamming period τ_j (see Section 8.1 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 [20] to translate the achieved throughput to goodput, for varying ECC thresholds.

Jamming effort and effective hopping rate: We evaluated the jamming effort $\mathcal{A}$ and effective hopping rate $\mathcal{R}$ for varying τ_j. Fig. 11(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

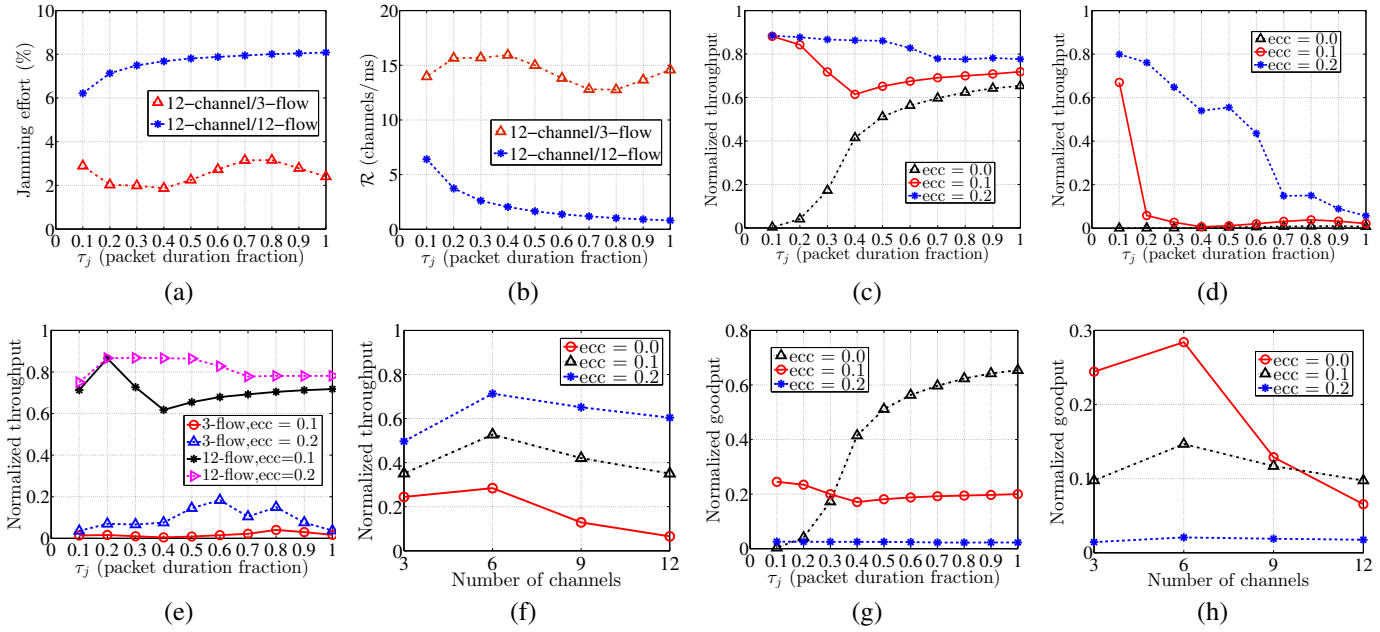


Fig. 11: (a) Jamming effort (%) when 3 and 12 flows contend over 12 channels, (b) effective hopping rate (channels/ms) when 3 and 12 flows contend over 12 channels, normalized throughput as a function of τ_j for varying ECC for (c) 12 flows contending over 12 channels with secret channel priority list, (d) 3 flows contending over 12 channels with secret channel priority list, and (e) 12 and 3 flows contending over 12 channels with public channel priority list, (f) normalized throughput as a function of the number of available channels for a 6-flow scenario, for varying ECC when $\tau_j = 0.4$, (g) goodput as a function of τ_j for varying ECC capability, (h) goodput as a function of the number of available channels for varying ECC capability.

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). Fig. 11(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 Fig. 11(a), because the effective hopping rate is roughly the inverse of the jamming effort, for a short sensing period τ_s.

Impact of error correction capability: We evaluated the impact of the ECC on the jammer's effectiveness under the cryptographically protected channel priority list. Fig. 11(c) shows the normalized 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 packets beyond recovery, but reduces the effective hopping rate, thus reducing the number of packets 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. Fig. 11(d) 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 packet corruption probability and hence, reduce throughput.

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 strategy, it is expected that the jammer would discover occupied channels faster. Fig. 11(e) 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 (Fig. 11(c)). 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.

Impact of the number of available channels: We evaluated the impact of the number of available channels. Fig. 11(f) 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 low.

Goodput evaluation: We also evaluated the normalized goodput under the reactive jamming strategy for varying ECC capability. We used the GV lower bound [20] to convert throughput to goodput, by finding an achievable code rate for a given relative distance. Fig. 11(g) shows the normalized goodput for the scenario simulated in Fig. 11(c) (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 Fig. 11(c), 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.

Fig. 11(h) shows the normalized goodput for the scenario

simulated in Fig. 11(f). Although ECC=0 yields the lowest throughput in the experiments of Fig. 11(f), 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 goodput performance. As in the case of Fig. 11(g), 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 11(g) and 11(h) indicate that the anti-jamming properties of FD-MMAC are primarily due to avoiding the jammer rather than correcting jammed packets. For jammers with low effective hopping rate, eliminating coding overall yields better performance. This strategy maximizes the per-packet goodput for jamming-free packets. For more aggressive jammers with faster effective hopping rates, offering moderate jamming protection using coding yields better goodput results.

10 Conclusion

We proposed FD-MMAC, a distributed MMAC protocol that exploits FD communications to coordinate multi-channel access at low control overhead. FD-MMAC eliminates control signaling over a common control channel to mitigate the impact of jamming attacks and improve spectral efficiency. 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. We further analyzed the impact of possible jamming attacks on FD-MMAC. Finally, we experimentally validated the PHY layer techniques on the NI USRP testbed and measured its performance via simulations. Our simulations showed that FD-MMAC achieves scalable performance at high spectral efficiency.

Acknowledgments

This research was supported in part by the NSF under grant CNS-1409172 and ARO grant W911NF-13-1-0302. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] 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.

[2] 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 MOBICOM*, pages 216–230, 2004.

[3] D. Bharadia, E. McMilin, and S. Katti. Full duplex radios. In *Proceedings of ACM SIGCOMM*, pages 375–386, 2013.

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

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

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

[7] W. Choi and H. Lim. Immediate acknowledgement for single-channel full-duplex wireless networks. In *Proc. of IEEE MASS*, pages 477–478, 2012.

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

[9] F. Hou, L. Cai, X. Shen, and J. Huang. Asynchronous multichannel MAC design with Difference-Set-Based hopping sequences. *IEEE Transactions on Vehicular Technology*, 60(4):1728–1739, 2011.

[10] 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 MobiCom*, pages 301–312, 2011.

[11] S. Keranidis, V. Passas, K. Chounos, W. Liu, T. Korakis, I. Koutsopoulos, I. Moerman, and L. Tassiulas. Online assessment of sensing performance in experimental spectrum sensing platforms. In *Proc. of WiNTECH*, pages 33–40, 2014.

[12] K. Kim, J. Park, Y. Bae, and B. Choi. Performance analysis of a slotted multi-channel MAC protocols for cognitive radio networks. In *Proc. of the QTNA*, pages 148–155, 2010.

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

[14] 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.

[15] S. Liu, L. Lazos, and M. Krunz. Time-delayed broadcasting for defeating inside jammers. *to appear in IEEE Transactions on Dependable and Secure Computing*, 2014.

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

[17] 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.

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

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

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

[21] 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.

[22] S. Sen, R. R. Choudhury, and S. Nelakuditi. CSMA/CN: carrier sense multiple access with collision notification. *IEEE/ACM Transactions on Networking (ToN)*, 20(2):544–556, 2012.

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

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

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

[26] 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 WiOpt*, pages 94–101, 2011.

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

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

[29] L. Wang, K. Wu, and M. Hamdi. Combating hidden and exposed terminal problems in wireless networks. *IEEE Transactions on Wireless Communications*, 11(11):4204–4213, 2012.

[30] Q. Wang, S. Leng, H. Fu, and Y. Zhang. An IEEE 802.11p-based multichannel MAC scheme with channel coordination for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):449–458, 2012.

[31] 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.

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

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

[34] X. Zhang and H. Su. CREAM-MAC: cognitive radio-enabled multi-channel MAC protocol over dynamic spectrum access networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(1):110–123, 2011.

[35] Y. Zhang, L. Lazos, K. Chen, B. Hu, and S. Shivaramaiah. FD-MMAC: Combating multi-channel hidden and exposed terminals using a single transceiver. In *Proc. of INFOCOM*, pages 2742–2750, 2014.

[36] W. Zhou, K. Srinivasan, and P. Sinha. RCTC: Rapid concurrent transmission coordination in full duplex wireless networks. In *Proc. of IEEE ICNP*, pages 1–10, 2013.