SimpleMAC: A Jamming-Resilient MAC-Layer Protocol for Wireless Channel Coordination

Sang-Yoon Chang
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
chang6@illinois.edu

Yih-Chun Hu
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
yihchun@illinois.edu

Nicola Laurenti*
Dipartimento di Ingegneria dell Informazione
Universitá di Padova
Padova, Italy
nil@dei.unipd.it

ABSTRACT

In wireless networks, users share a transmission medium. To increase the efficiency of channel usage, wireless systems often use a Medium Access Control (MAC) protocol to perform channel coordination by having each node announce its usage intentions; other nodes avoid making conflicting transmissions minimizing interference both to the node that has announced its intentions and to a node that cooperates by avoiding transmissions during the reserved slot. Traditionally, in a multi-channel environment, such announcements are made on a common control channel. However, this control channel is vulnerable to jamming because its location is pre-assigned and known to attackers. Furthermore, the announcements themselves provide information useful for jamming. In this paper, we focus on a situation where multiple wireless transmitters share spectrum in the presence of intelligent and possibly insider jammers capable of dynamically and adaptively changing their jamming patterns.

We develop a framework for effectively countering MAC-aware jamming attacks and then propose SimpleMAC, a protocol resilient to these attacks. SimpleMAC consists of two schemes (the *Simple Transmitter Strategy* and the *Simple Signaling Scheme*) that are easily analyzed using game theory, and show the optimal adversarial behavior under these protocols. We evaluate our schemes mathematically, through Monte Carlo simulations, and by implementation on the WARP software-defined radio platform. SimpleMAC provides very rapid improvement over the alternative of not using any MAC protocol, and eventually converges to optimal performance (over six-fold improvement in SINR, 50% gains in Shannon capacity in a realistic mobile scenario).

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General

Keywords

MAC-Layer, Jamming, Wireless

1 Introduction

As wireless features are introduced into more and more electronic devices, it is becoming increasingly important to use scarce radio spectrum as efficiently as possible. An important part of efficient usage is effective coordination of user transmissions. Traditional protocols aim to avoid overlapping transmissions; typical channel access schemes separate users' usage in some combination of time, frequency, and code. As networks increasingly carry data traffic, which is characterized by bursty arrivals, fixed channelization is being replaced by dynamic Medium Access Control (MAC) protocols that change user allocations from frame to frame. In a distributed MAC, each node announces its usage intentions, both to link a transmitter-receiver pair for communication and to help other transmitters minimize interference. In this paper, we refer to this task of exchanging channel usage information as *channel coordination.* In channel coordination, a network user reserves a channel by sending one or more control packets (that contain its channel usage intentions) on a *control channel*, and then uses the reserved *data channel* to send its data traffic.

Channel coordination is only useful when other nodes respect reservations; we call such environments *collaborative environments.* In such environments, channel coordination protocols can provide substantial performance gains. In particular, if we characterize the channel capacity using *Shannon capacity* (as given by the Shannon-Hartley Theorem), we see that when two nodes with equal power levels share a band, coordinating nodes get capacity

$$\frac{B}{2}\log_2\left(1+\frac{S}{N}\right),$$

whereas when they do not cooperate, they get capacity

$$B\log_2\left(1+\frac{S}{S+N}\right),$$

since each node's transmission is interference to the every other node. Whenever the band's signal-to-noise ratio exceeds about 2 dB, coordination provides substantial gains. However, when an attacking node can receive channel coordination information, such as when the attacker is a compromised network node, coordination information can also be used to jam more effectively, since jammers know exactly on which channel to focus their jamming power to disrupt

*Part of this work was done while Prof. Laurenti was a visiting scholar at the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.

MobiCom'12, August 22–26, 2012, Istanbul, Turkey.
Copyright 2012 ACM 978-1-4503-1159-5/12/08 ...$15.00.

data communication. Also, since the location of the control channel is pre-assigned and known to network users, attackers can jam the control channel, thereby eliminating any benefit legitimate users might gain from channel coordination. In contrast to previous work, we consider attackers that are insiders, intelligent, and adversarial. In particular, such adversaries are capable of reacting to legitimate user strategy ("intelligent"), they have the keys of one or more legitimate nodes ("insider"), and their goal is to minimize the throughput of legitimate users ("adversarial"). We construct a game-theoretic framework to analyze strategies for both adversaries and legitimate users. Our adversarial model represents a worst-case scenario; an attacker with equal power that chooses any other strategy cannot result in worse legitimate user performance. (It is possible that against any particular attacker strategy, another algorithm may provide better performance; optimal performance in such environments is beyond the scope of this paper.)

Our goal is to provide a channel coordination mechanism that maximizes legitimate node throughput in the presence of intelligent jammers. The fundamental difficulty in building adversary-resistant channel coordination is that channel coordination information does good in the hands of cooperative users, because such users will avoid our transmissions, but does harm in the hands of attackers, because attackers will attempt to coincide with our transmissions. We devise SimpleMAC, a scheme in which channel coordination is shared with a specific group of users. SimpleMAC includes the *Simple Transmitter Strategy* (STS), which uses a feedback-based trial-and-error approach to select the group of users that will receive the control packet, and the *Simple Signaling Scheme* (SSS), which establishes a secure communication channel for exchanging control messages that ensures confidentiality against users other than intended recipients and ensures availability against Denial-of-Service (DoS) attacks. With the SSS, a control packet can be received only by intended recipients, and can be received even when the adversary attempts to jam the message. Against an intelligent adversary, SimpleMAC produces desirable results despite its simplicity; our protocol limits the optimal adversarial behavior, quickly outperforms schemes that do not perform channel coordination (the current state-of-the-art channel coordination strategy, which is generally adopted in most security-concerned frameworks [11, 13]), and eventually converges to optimal performance.

Our work applies to both single-channel TDMA systems (with an *energy-limited* attacker, since a power-limited attacker would jam at all times and gain no advantage from channel coordination information) and multi-channel systems (with a *power-limited* attacker). For clarity of presentation, we present our system as applied in multi-channel systems. Because we model legitimate users as cooperative and attackers as malicious, our protocol must simultaneously allow legitimate users to avoid our transmissions and yet prevent attackers from coinciding with them. For this reason, each transmission in our protocol transfer is sent using Frequency Hopping Spread Spectrum (FHSS), in which each transmission is sent while the transmitter hops from one frequency band to another according to a pseudorandom hopping pattern. Channel coordination information thus consists of the time at which a sender plans to send and the frequency hopping pattern the sender plans to use.

The rest of the paper is organized as follows. Section 2 presents security vulnerabilities in current MAC-layer protocols and a brief overview of our approach to resolve these vulnerabilities. After presenting the model to be used in our investigation in Section 3 and setting up our system's theoretical framework in Section 4, we analyze the general jammer behavior in Section 5. We then introduce our scheme in Section 6, and the jammer reaction to our scheme in Section 6.3. Next, we mathematically analyze the performance of our scheme in Section 7 and evaluate it using MATLAB simulations and WARP implementation in Section 8. Lastly, we present conclusions and open problems in Section 10.

2 Problem Statement & Our Contribution

A wide variety of MAC-layer protocols have been proposed for various environments and applications. In this section, we outline a security vulnerability of existing wireless MAC protocols, where the attacker can jam control messages and can use control messages to jam more effectively. We then give a brief overview of our protocol, which is the first to perform channel coordination in a manner that addresses these vulnerabilities.

2.1 Threat Overview & Related Work

To reduce the inefficiencies inherent in simultaneous channel usage, most wireless MAC-layer protocols (with few exceptions, such as ALOHA [22]) attempt to reserve a channel by exchanging channel coordination information. Traditionally, a common control channel is used to exchange channel coordination information among users. There are two important jamming attacks against a control channel: first, the attacker can jam the channel itself, and second, the attacker can use *jamming-relevant information* transmitted on the control channel (such as when and where data transmissions will take place) to facilitate effective jamming.

Our work focuses on *adversarial* entities. Extensive prior work (including [8, 15]) has observed that an attacker can send excessive reservation messages to prevent legitimate nodes from using the channel. Our work is orthogonal, but we suggest limiting each legitimate node to one reservation at a time, and using a central authority to prevent Sybil attacks. Another form of adversarial behavior is channel jamming. Awerbuch et al [5] propose a fair single-channel MAC protocol against a power-limited jammer that does not jam all of the time. Other papers propose mechanisms to avoid jamming [4, 14, 27] but, unlike our work, these approaches are not secure against insider attacks; that is, when jammers are compromised network participants and thus have access to some of the keys of the network nodes, jamming avoidance cannot be assured by this prior work.

2.2 Vulnerabilities in Current Protocols

The use of a common control channel, which is typical in currently available protocols, is vulnerable to jamming attacks. For example, in the IEEE 802.11 WiFi standard [2], nodes use *virtual carrier sense* in which they reserve the channel by exchanging Request to Send (RTS) and Clear to Send (CTS) messages; these messages can be jammed to reduce network performance.

Though virtual carrier sense provides an effective way for a legitimate potential transmitter to avoid collisions with another transmitter, the very mechanism that allows them to mitigate interference also allows an attacker to jam every transmission. In particular, whenever an attacker senses another user's transmission, either through carrier sense or virtual carrier sense, the attacker can jam the corresponding data packet.

In addition to carrier sense and virtual carrier sense, WiFi also uses a Collision Avoidance mechanism in which a node transmitting a frame chooses a backoff interval. The node counts down the backoff interval whenever the channel is idle; this mechanism reduces the probability that two nodes will transmit simultaneously. Several researchers have investigated the attack wherein the attacker chooses incorrect backoff intervals [9, 17, 21].

In the Out-of-Band signaling scheme [12], each receiver sends a very narrowband busy tone whenever it receives data to indicate the channel is in use. A powerful adversary may be able to jam the busy tone, and even when the jammer is unable to remove the busy tone, a jammer that hears a busy tone knows that a receiver is active within its wireless transmission range. A jammer that jams the data channel whenever it hears a busy tone can effectively deny service to receivers within its interference range. An attacker can also falsely reserve the channel by continuously sending a busy tone.

IEEE 802.16 [3], commonly called WiMAX, uses a centralized scheduling algorithm in which the base station assigns time slots to each user. Since the base station broadcasts control messages, a jammer that knows the location of the control channel can either jam the control channel to disrupt the exchange of control messages or use the received channel scheduling information to jam data transmissions at the assigned time slots (and frequency channels). In WiMAX, the control channel location is a published part of the standard, but even if it were not, an attacker that has compromised a legitimate node must know the location of the control channel. Furthermore, a node can request and be scheduled for time and frequency slots that it does not need, thus wasting time and bandwidth.

Another centralized protocol is Bluetooth [1], in which a master device sends control messages to each slave device in the network (called a *piconet*). Since an attacker who knows the frequency hopping pattern of a scheduled transmission can easily jam that transmission. In Bluetooth, these frequency hopping patterns are a public part of the standard, but even if it were not, an attacker that has compromised a legitimate node must know the location of the control channel. Furthermore, an attacker can become the master and have significant control over other legitimate users.

MAC protocols that do not perform channel coordination suffer from higher probability of collisions between simultaneous transmitters, resulting in more interference. Thus, a protocol that lacks channel coordination functionality yields lower SINR and therefore lower capacity.

In conclusion, currently implemented protocols either mitigate interference from legitimate nodes by regulating their channel usage, in which case jammers can effectively jam during legitimate node usage, or provide no channel coordination and suffer from increased interference. In this paper, we present the first protocol that mitigates interference from both legitimate users and jammers.

2.3 Our Approach

Current protocols, when faced with just one intelligent, insider jammer, will at best reach the Nash equilibrium, in which channel coordination is completely disabled and each message is spread across the entire band [11, 13]; at the Nash equilibrium, there is no reduction in collisions, which reflects a non-cooperative environment. In this paper, we construct a theoretical framework to analyze the dynamics between the adversaries and legitimate users, then propose SimpleMAC, a MAC-layer protocol that performs channel coordination while mitigating the effects of jamming.

In Section 4.1, we describe our MAC-layer framework. A MAC protocol provides reduced probability of collision by exchanging a channel usage plan with other network users; this channel usage plan is *jamming-relevant information* because the plan allows legitimate users to avoid the transmitter but also allows jammers to intentionally collide with the transmitter. We divide our scheme into two components. The *transmitter strategy* selects the set of network nodes with which to share the relevant control message; we call this set, which may vary for each packet, the *recipient list* and denote it S. The *signaling scheme* delivers a control message to each node in the recipient list, and ensures that no other nodes are able to receive the control message. When the adversary is malicious, the ideal recipient list includes all legitimate users and no attackers.

SimpleMAC consists of the *Simple Signaling Scheme* (SSS) and the *Simple Transmitter Strategy* (STS). We develop a jamming-resistant signaling scheme to deliver jamming-relevant control message to exactly the set of nodes in the recipient list S, and a transmitter strategy that decides on the recipient list based on prior receiver feedback. In our transmitter strategy, the transmitter-receiver pair measures the performance of S after sending each packet and uses this information to adapt S for future packet transmissions. The long-term goal for the transmitter is to search for a set S that provides the optimal performance. As a general rule, the choice of $S = \emptyset$ (equivalent to disabling channel coordination) represents baseline performance; after a sufficient number of independent trials, any set that performs significantly worse must contain a jammer. Therefore, the transmitter can determine whether channel coordination has been compromised by comparing the performance of the recipient list with performance when $S = \emptyset$. However, since attackers are intelligent, and thus capable of dynamically changing their jamming strategy, a recipient list S with better performance than when $S = \emptyset$ does not necessarily mean that S excludes all attackers.

SimpleMAC quickly outperforms the case where channel coordination is disabled, eventually converges to the recipient list offering optimal performance, and forces the optimal jammer strategy to be jamming at full power all the time (even though jamming alerts the user and prompts it to stop sharing information with the compromised recipient lists).

3 System Model & Assumptions

We consider an environment with $T + 1$ non-idle transmitters (each transmitter has T potential interference sources), each identified by an index $i \in \mathcal{T} = \{1, \ldots, T + 1\}$, a subset $\mathcal{N} = \{i_1, \ldots, i_N\}$ of which are N jammers. All non-jammers are protocol-compliant. We assume a shared secret key between each pair of nodes, and that all nodes operate in shared spectrum divided into C channels, each with bandwidth W Hz. No online authority governs users. We consider a repeated game with infinite horizon; either the transmission never ends or the users do not know when the transmissions will end. We index the rounds of the game $r \in \{1, 2, 3, ...\}$.

We assume that each user has technical means to transmit on the spectrum, that the attacker ignores any legal prohibitions against interference, and that there is no way to *a priori* determine which node is trustworthy. Also, because of the possibility of jamming, we make the standard assumption [20, 23] that each transmitter sends data using fast frequency hopping on randomly generated hopping patterns chosen independently for each packet. Traditionally, fast frequency hopping is characterized by a hopping time of more than one hop per symbol; here, we only require that the hopping time be faster than the jammer's reaction time.

At the physical layer, we assume there exists a known spreading gain at which any pair of neighbors can communicate with a suitably low bit error rate. Alternatively, we define a *neighbor* as a node that can be reached using a specific spreading gain. In our SimpleMAC, described in Section 6, we use Direct Sequence Spread Spectrum (DSSS) for control communications and Frequency Hopping Spread Spectrum (FHSS) for data communications, and we communicate the frequency hopping pattern in the control message.

The communication between any transmitter-receiver pair is single-hop; that is, the transmitter does not rely on a third node to relay the message to the final destination. For simplicity, in our model, each user is a neighbor of every other user. Thus, when two nodes transmit on the same frequency at the same time, those transmissions interfere with each other, resulting in reduced capacity. Our protocol can be extended to hidden-terminal environments by having both sender and receiver repeat each channel coordination transmission, although the details of this approach are beyond the scope of this paper.

Our protocol is designed for unicast data transmissions, where performance affects only the receiver. Therefore, when a sender transmits to multiple receivers, the feedback of a malicious receiver does not affect the performance of other receivers. However, a malicious receiver may be able to induce a sender to choose $S = \emptyset$ for all transmissions to that receiver. The impact of this selection depends on the transmission priority scheme, so we discuss this attack further in Section 9.

All users, including attackers, share the same power constraint P_c. The case where each attacker is more powerful than a normal user can be modeled by increasing the fraction of nodes which are attackers.

3.1 Performance Metric

When user i transmits to user j, it does so on a frequency channel that varies with time according to a frequency hopping pattern known to user i and user j. At any point in time, the user transmits on frequency channel $c \in \{1, \ldots, C\}$. Assuming a flat fading channel with additive white Gaussian noise and Gaussian signals, the Shannon capacity of the link $i \to j$ is:

$$\mathcal{R} = \int_{f_c - W/2}^{f_c + W/2} \log_2 \left[1 + \text{SINR}_{i,j}(f)\right] df \tag{1}$$

where f_c is user i's carrier frequency, and SINR is the effective signal-to-interference-and-noise ratio at the receiver

$$\text{SINR}_{i,j} = \frac{\gamma_{i,j}\widetilde{P_i}(f)}{\widetilde{N_0} + \sum_{\ell \neq i, \ell \in \mathcal{N}^{\text{c}}}[\gamma_{\ell,j}\widetilde{P_\ell}(f)] + \sum_{k \in \mathcal{N}}[\gamma_{k,j}\widetilde{J_k}(f)]} \tag{2}$$

In Equation 2, $\gamma_{a,b}$ is the channel gain between transmitter a and receiver b, $\widetilde{N_0}$ is the power spectral density of the noise, $\mathcal{N}^{\text{c}}$ are the indices of legitimate users, $\mathcal{N}$ are the indices of jammers, $\widetilde{P_\alpha}$ is transmitter α's power spectral density for some α, and $\widetilde{J_k}(f)$ is the jammer k's power spectral density.

Shannon capacity ($\mathcal{R}$) is an upper bound on communication rate performance. Shannon capacity is a mathematically simple formulation and is tight in many practical environments; existing codes very nearly achieve Shannon capacity [18]. In order to separate MAC-layer issues from physical-layer decisions such as modulation and coding, we use both SINR and Shannon capacity as representative performance metrics in our mathematical analysis. We observe that Equation 1 exhibits two properties that we use in our analysis: it is decreasing and convex with respect to jamming power and monotonically increasing with respect to the user's signal power. Though we use SINR and capacity as representative measures of performance, our approach generalizes to any utility function that is convex in interference power and monotonically increasing in SINR. (In Section 8, our implementation testbed simulation results show the effective SINR at the receiver, because achieving Shannon channel capacity involves sophisticated coding and modulation, and because the instantaneous capacity is strictly monotonic in the instantaneous SINR.)

Channel capacity $\mathcal{R}$ (Equation 1) serves as our utility function for the legitimate transmitter i. The transmitter's aim is to maximize its capacity $\mathcal{R}$. As $\mathcal{R}$ is a monotonically increasing function of P_i, the transmitter will emit full power. To aggregate capacity (which is an instantaneous metric) over time, we compute its time-average. At time t, given $\{\mathcal{R}_{t'} \mid t' < t\}$, the utility function is:

$$U = \frac{1}{N_t} \sum_{t' = t - N_t + 1}^{t} \text{E}[\mathcal{R}_{t'}] \tag{3}$$

where $\mathcal{R}_\gamma$ is the capacity measured at time γ for some γ. In an infinite-horizon game, Equation 3 is replaced by its limit as $N_t \to \infty$, where N_t represents the time duration of transmission.

3.2 Attacker Model

We consider a jammer that intends to minimize the utility function, Equation 3, subject to its power constraint:

$$\text{minimize } U \text{ subject to } \int_f \widetilde{J_k}(f)\, df \leq P_c, \ \forall k \in \mathcal{N} \tag{4}$$

We assume that jammers collude. Thus, if one jammer knows a user's frequency hopping pattern, then all jammers can make use of that information. Also, attackers know the protocol and can adaptively change their strategies according to the legitimate users' strategies.

We also consider reactive jammers that jam according to their observations on the target signal, in case they do not receive the user's channel coordination information. To counteract reactive jammers, the user can shorten the frequency hopping time so that the jammers do not have enough time to observe the spectrum and jam the used channel.

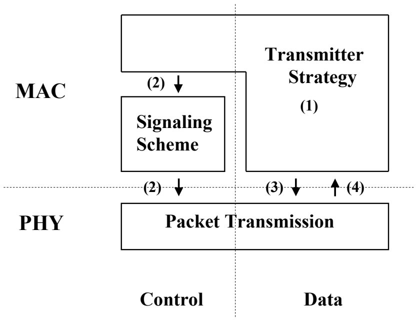


Figure 1: Our Protocol Framework

We do not consider the very strong attack of correlated jamming, where an adversary mimics the target signal with a phase offset of π at equal amplitude, canceling the target signal. Under this attack, assuming the attacker has at least as much power as the legitimate node at a target receiver, no physical layer can provide any throughput [7].

Attackers can choose between narrowband jamming (concentrating its power on one or a subset of frequency channels at a time) or wideband jamming (emitting power across the spectrum at a time) and can freely switch between these strategies. Because the jammer has so much flexibility, we do not consider legitimate user attempts to infer information about a jammer; however, our approach still converges to the optimal performance.

We also consider the possibility of non-Gaussian jamming, since Equation 1 holds only when all received signals are Gaussian signals. Given a Gaussian fading channel with Gaussian signals, where the overall signal power is much greater than the combined power of the jammer network, the optimal jammer strategy is to jam with Gaussian noise [7, 16]. Also, the transmitter can make any received jamming signal appear Gaussian by using a sufficiently long Direct Sequence Spread Spectrum (DSSS) code shared only with the receiver. When jammers do not know the code used by the transmitter, the received jamming signal looks like a Gaussian signal by the Central Limit Theorem.

4 Theoretical Framework

4.1 Overview

We design SimpleMAC from the ground up without making any MAC-layer assumptions. Our MAC-layer framework contains two parts: a transmitter strategy and a signaling scheme. For each packet, a transmitter strategy determines the set of users S that will receive the channel coordination information for that packet; this set is called the *recipient list*. The optimal transmitter strategy will prevent the attacker from gaining any advantage from its knowledge of insider network keys while minimizing interference from legitimate users. A signaling scheme delivers the control message to the recipient list and provides availability (that is, messages are not easily jammed) and confidentiality (that is, nodes not on the recipient list will not receive the control message).

Figure 1 depicts our MAC-layer framework. When sending a packet, the transmitter (1) chooses a subset S of network users, (2) transmits its frequency hopping information to S, (3) transmits the data packet using the previously reserved hopping pattern, and (4) determines the effectiveness of S based on the feedback that it receives from the receiver.

4.2 Collision Between Benign Users

Even when the spectrum is used very sparsely, a randomly selected frequency hopping pattern is likely to collide with the hopping patterns of other nodes. Channel coordination schemes are designed to reduce this inter-transmitter interference. When two nodes wish to use the same channel during the same time slot, they each determine which of the transmitters has priority, for example, based on the time at which each node claimed the channel. The node that does not have priority will not transmit at all, so the corresponding receiver will decode random data in this position. However, since the positions for these lost bits are known *a priori*, a sender mitigates the loss by using channel coding and forward error or erasure correction.

Two nodes may collide when neither node informed the other about its frequency hopping patterns, and, depending on the priority scheme, when only one of the two nodes disclosed its transmission intentions to the other. In our analysis, we assume that transmitter i has the highest priority for transmission, and is targeted by all the jammers. In other words, all nodes in the transmitter's recipient list will avoid interfering with the transmitter (we revisit this assumption in Section 9 and consider the case when all nodes have equal priority). Thus, increasing the number of benign transmitters in S reduces the number of potential interferers, increasing capacity.

4.3 Capacity Expression for Our Framework

In this section, we mathematically derive the capacity of our system when all channels have equal gains and all users emit power uniformly across their chosen channel. Since legitimate users that receive the transmitter's channel coordination information will not interfere, the transmitter's capacity depends on its selection of recipient list S. Equation 1 can be simplified to:

$$\mathcal{R}(S) = W \cdot \log_2 \left[1 + \frac{P}{N_0 + \sum_{\ell \neq i, \ell \in (\mathcal{N}^c \cap S^c)} P_\ell + \sum_{k \in \mathcal{N}} J_k(S) \cdot P_c}\right]$$

where N_0 is the noise power in the channel, P is the transmitter's signal power, P_ℓ is the amount of user ℓ's power that interferes with the transmitter's signal, J_k is the jammer k's power normalized with respect to the power constraint P_c, and

$$J_k(S) \leq \begin{cases} \frac{1}{C}, & \text{if } (S \cap \mathcal{N}) = \emptyset \\ 1, & \text{otherwise} \end{cases}$$

In the $(S \cap \mathcal{N}) = \emptyset$ case, the jammer does not receive the user's channel coordination information, and therefore can at best conduct wideband jamming across C channels, as described in Section 5.

If we further assume that legitimate users not in S emit at full power to maximize their own performance, then $E[P_\ell] = \frac{P_c}{C}, \forall \ell$, since there is a $\frac{1}{C}$ chance that any legitimate user not in S will interfere with the transmitter. Then, using Jensen's inequality, the expected capacity is bounded from below by:

$$\mathrm{E}[\mathcal{R}(S)] \geq W \cdot \log_2 \left[1 + \frac{P}{N_0 + |\mathcal{N}^c \cap S^c| \frac{P_c}{C} + \sum_{k \in \mathcal{N}} J_k(S) \cdot P_c}\right] \quad (5)$$

We use this expression in our analysis.

4.4 Transmitter Strategy

To make future recipient list decisions, each network user records historical performance data for each packet, including the recipient list used for that packet and the resulting performance. One natural choice for the recipient list is the set that has yielded the best average performance in the past. We call this the *Best so far* set and denote it with S_B: $S_B(t) = \underset{\sigma \in \{S(t'), \forall t' < t\}}{\text{argmax}} \bar{\mathcal{R}}(\sigma)$, where $\bar{\mathcal{R}}$ is the time-average performance. When jammers jam at full power, the optimal set is the set that includes all legitimate nodes and excludes all jammer nodes; we denote this optimal set S^*. However, an attacker might choose not to jam when certain nodes are in the recipient list, so S^* may not have the maximum performance for a particular jammer strategy; however, S^*'s performance is optimal in the worst case. Our scheme will converge to at least the performance of S^*, but if the attacker concedes better performance, our scheme can take advantage of the better-performing set.

In order to improve the Best so far recipient list, a sender must explore possible sets from time to time. To reach optimal performance, a transmitter strategy must eventually explore the optimal set. When we do not know the jammer's strategy or the distribution for the number of jammers, the optimal set may be any set, because the attacker may choose to cease all jamming activities when a particular recipient list is chosen. Thus, to provide optimality against arbitrary attackers, a sender must be willing to explore all possible sets. In our framework, as well as our protocol, SimpleMAC, convergence to the optimal set takes exponential time in the average case; however, we will show in Section 8 that we improve over the state-of-the-art within a single round in many cases, and fast convergence is not a goal of our design.

5 Jammer Strategy Analysis

In this section, we assume that the attacker is purely adversarial, as described in Section 3.2. Attackers are capable of using a potentially non-deterministic, time-varying strategy to meet their goal of minimizing capacity. Since an attacker's strategy depends on whether or not it receives the channel coordination information, we study both cases.

5.1 Recipient List With No Jammer

If the recipient list S contains no jammers, then jammers do not learn any jamming-relevant information, and thus do not know which channel will be used for the user's transmission. This limits jammers to a much weaker attack, since they cannot use their compromised keys and gain no advantage from collusion. The only decision to be made in this case is whether to choose narrowband jamming or wideband jamming. We assume that a legitimate user i will uniformly choose any of the C channels, and we observe that $\mathcal{R}$ is a decreasing and convex function of $\widetilde{J_k}(f)$. By Jensen's inequality, the expected capacity $\mathrm{E}[\mathcal{R}]$, under the constraint of Equation 4, is minimized by choosing $\widetilde{J_k}(f) = \frac{P_c}{C \cdot W}$ for each jammer k. Thus, to minimize capacity, jammers will conduct wideband jamming when they do not know the frequency hopping pattern, but conduct narrowband jamming when they do have the information. In our analysis, we assume the jammer uses this strategy when it does not know the frequency hopping pattern.

5.2 Compromised Recipient List

We now analyze the jammer strategy when a jammer does receive the user's channel coordination information. In this scenario, jammers know where to concentrate their power to minimize transmitter capacity. However, in an infinite-horizon repeated game, jammers must also consider how their current action will affect future capacity. Equation 1 shows that jamming with higher power causes more interference and lowers capacity. However, since a user will avoid any set S that appears to contain jammers, jammers may not wish to strongly jam the transmission, hoping to reduce the user's suspicions that S contains a jammer. If the user converges on a new S_B that contains no jammers, then jammers can no longer influence capacity except by wideband jamming. A jammer may then want the *Best so far* set to include a jammer by abstaining from excessive jamming.

In the long run, the jammer knows that the transmitter will explore S^*, and will choose the best set. If the jammer allows another set S' to have better performance than S^*, then the transmitter will pick S', otherwise it will pick S^*. If the jammer's goal is to minimize capacity, they should not concede any additional long-run performance to the sender. Thus the sender will choose $S_B = S^*$, and the optimal jammer strategy will converge to full-power jamming.

Claim 1. *Given the general transmitter strategy in Section 4.4, jammer strategy converges to full power over time:*

$$\forall k \in \mathcal{N}, J_k(t) \to 1 \text{ as } t \to \infty$$

Proof. Omitted due to space constraints. □

6 Our Protocol, SimpleMAC

Our SimpleMAC protocol has two components: the Simple Transmitter Strategy (STS) and the Simple Signaling Scheme (SSS). Despite the simplicity of the schemes, from which SimpleMAC derives its name, SimpleMAC effectively combats intelligent attackers: it quickly outperforms the case where MAC protocol is disabled (which is the standard approach for securing MAC protocols) and has an easily analyzed optimal jammer strategy.

When selecting a recipient list, we determine the effectiveness of recipient list S by comparing the capacity when S is chosen as the recipient list to the capacity when no one knows the recipient list. In the latter case (i.e., when $S = \emptyset$), there is neither gain in capacity from legitimate nodes avoiding the transmitter nor loss in capacity from the jammers using the jamming-relevant information. Whenever the capacity is less than or equal to (with some error margin) the capacity when $S = \emptyset$, the transmitter chooses a new set S before the next transmission, because the current set S provides no advantage over $S = \emptyset$.

SimpleMAC does not try to infer which nodes are jammers and which ones are not; rather, it directly uses channel feedback to determine which recipient lists result in good performance. For example, when node A shares its information with a jammer, any recipient list with node A in it will have decreased performance, so the STS will avoid such list. Similarly, if node A jams only when node B is also in the recipient list, the STS will avoid lists that contain both A and B. Because we make all of our decisions based on actual performance and not behavior, SimpleMAC is immune to collusion.

6.1 Simple Transmitter Strategy

In the STS, for each transmission t, a legitimate user has three options when choosing a recipient list S:

1. *Best so far* (S_B): the set with best average performance among explored sets, as described in Section 4.4.

2. *Randomly explore* (S_R): chosen uniformly at random among all possible sets.

3. *Empty set* ($\emptyset$): $S(t) = \emptyset$.

The transmitter always chooses one of these three strategies. The *Best so far* action, $S(t) = S_B$ corresponds to choosing the set that yielded the highest average capacity among all the recipient lists that have been tried through time $t-1$, which guarantees performance at least as good as $S = \emptyset$, since $S = \emptyset$ has been tried earlier. If jammers jam with sufficient power ($\sum_{k \in \mathcal{N}} J > \frac{T}{C}$), then the set S that yields the highest capacity is S^*. In this case, when the user explores sets occasionally (so that the user eventually visits all possible sets with probability one), the *Best so far* set S_B converges to S^*, since the probability that S^* has been previously chosen approaches one. The user chooses the *Randomly explore* action to search for a set that yields higher capacity than the previous *Best so far* set. Once such a set is found, the set S_R becomes the new *Best so far* until the node discovers another set that yields even higher capacity. The more often the user chooses to explore a random set, the more quickly S_B converges to S^*. The *Empty set* action establishes baseline performance during each time interval, so that slow time-variance in channel conditions do not bias set selection.

The STS operates in rounds. For each transmission t within round r, the user makes an independent random choice among the three options. The probabilities may vary with r, so that in expectation, round r contains $B(r)$ transmissions with the Best so far recipient list, $R(r)$ transmissions on a randomly chosen recipient list, and $E(r)$ transmissions using an empty recipient list. Round r lasts for $B(r)+R(r)+E(r)$ transmissions, and we do not rely on the secrecy of $B(r)$, $E(r)$, $R(r)$.

In order to converge to the optimal performance, we explore a user strategy where the user uses the *Best so far* set more and more often, while occasionally using *Randomly explore* and *Empty set*. One such user strategy is:

$$B(r) = r^{\delta}, R(r) = 1, E(r) = 1, \forall r \tag{6}$$

In order to converge to the optimal performance for S^*, δ needs to be positive. A higher δ corresponds to more aggressive search for a better *Best so far* set and thus quicker convergence to S^*.

6.2 Simple Signaling Scheme

In order to send a control message to exactly those nodes in a recipient list, we need a signaling scheme that provides confidentiality against malicious entities and reliability in the presence of jamming. We make no attempt to design an efficient signaling scheme; because the overhead of a control message is amortized over the data frame, and because we can choose arbitrarily long data frames, we can reach near-optimal overall protocol performance even with an extremely inefficient signaling scheme. Thus, we simply unicast the control messages to each recipient in the recipient list. We provide confidentiality by encrypting messages with a symmetric key, and availability by using direct sequence spread spectrum (DSSS) using a chip sequence known only to the sender and receiver. In a 50 node network with 20 byte reservation messages (consisting of source address, destination address, and a seed for the hopping pattern), if each reservation covers 100 kB of data (for example, 66 packets each 1500-bytes long), SSS incurs an overhead of not more than 1%, and average overhead of 0.5%. Though the data rate may be higher than the control rate due to our use of DSSS for the control message, we can continue to keep our overhead low by covering more data with each control message, or by replacing repeated-unicast with a jamming-resilient broadcast protocol, of which several have been proposed [6, 10, 24]. SSS simply requires that each node have a pairwise shared key with every other node. Such keys can be established through Diffie-Hellman exchanges over a jamming-resilient broadcast protocol.

6.3 Jammer Reaction to SimpleMAC

In Section 5, we studied the attacker strategy under our general framework and showed that the optimal attacker strategy converges to full power, even though an attacker may wish to avoid detection so that the legitimate user will use a compromised recipient list. In this section, we claim that against the STS, optimal jammers *jam at full power all the time.* The claim holds because B, R, E are independent of the jammers' strategy; unlike the user's selection of recipient lists, the user's choice of action (B, R, and E) does not adapt to jammer strategy. Intuitively, the sender forms a partial order on recipient lists based on their past performance. An attacker that does not jam at full power can jam at a higher power level and yet maintain the same partial order of recipient lists (or a functional equivalent), which shows that any strategy that does not jam at full power cannot be optimal.

Claim 2. *Against the STS, the best jammer strategy is to emit at full power all the time, i.e.,*

$$\forall t, \forall k \in \mathcal{N}, J_k(S) = \begin{cases} \frac{1}{C}, & \textit{if } (S \cap \mathcal{N}) = \emptyset \\ 1, & \textit{else} \end{cases}$$

Proof. Proof is by contradiction. Suppose that an optimal jammer strategy $J = (J_1, J_2, ..., J_N)$ does not jam at full power at some time; we let Γ be the set of times at which J does not use full power. We now show that there exists a different jammer strategy J' that yields less capacity than J while preserving the legitimate user strategy. To find such J', we assume perfect knowledge about the recipient list S. (This does *not* mean that a jammer needs perfect information; rather, it shows that even a jammer with perfect information will still choose the simple strategy of full-power jamming, and therefore *any* attacker should do the same.) J' will only diverge from J when J does not emit at full power,

At time $t' \in \Gamma$, let the two best previously measured recipient lists be A and C, where A is the best and C is the second-best. Then either $S = A$ yields higher capacity than $S = C$ or both sets $S = A$ and $S = C$ yield the same performance. We study the two cases separately:

i) If $\mathcal{R}(A) > \mathcal{R}(C)$, then pick $J'(t')$ such that $\mathcal{R}_{J'}(A) = \frac{\mathcal{R}_J(A)+\mathcal{R}_J(C)}{2}$. This choice preserves the performance order of recipient lists and thus do not change the user's choice of recipient list.

ii) If $\mathcal{R}(A) = \mathcal{R}(C)$, then pick $J'(t')$ such that its corresponding performance is ϵ smaller than that of $J(t')$ for small ϵ. This breaks the tie between A and C since $A \neq C$.

$$
\begin{aligned}
\mathrm{E}[\mathcal{R}]_{\mathrm{ss}} &= \lim_{r\to\infty} \mathrm{E}[\mathcal{R}(r)] \\
&= \frac{B}{B+R+E} \cdot W \log_2\left(1+\frac{P}{N_0+\frac{N}{C}}\right) \\
&+ \frac{R}{B+R+E} \sum_{\alpha=0}^{T-N} \left\{ \frac{\binom{T-N}{\alpha}}{2^{T-N}} \right. \\
&\quad \cdot \left[\frac{1}{2^N} \cdot W \log_2\left(1+\frac{P}{N_0+\frac{N}{C}+\frac{\alpha}{C}}\right) \right. \\
&\quad \left.\left. + \frac{2^N-1}{2^N} \cdot W \log_2\left(1+\frac{P}{N_0+N+\frac{\alpha}{C}}\right)\right]\right\} \\
&+ \frac{E}{B+R+E} \cdot W \log_2\left(1+\frac{P}{N_0+\frac{T}{C}}\right) \qquad (9)
\end{aligned}
$$

Though this changes the legitimate user's choice of recipient list (because the legitimate user will choose A over C for the *Best so far* set), the legitimate user strategy when jammer picks J' is functionally equivalent to the legitimate user strategy when jammer picks J (because when the jammer picks J, it does not matter whether the user chooses A or C). Therefore, J' yields smaller capacity than J.

Since, in both cases J' yields lower capacity than J while preserving the order of recipient lists (and thus preserving the legitimate user strategy or its equivalent), J is not optimal and there is a contradiction. □

7 Performance Analysis

Under the STS, a legitimate user chooses the recipient list from among three options: *Best so far* (B), *Randomly explore* (R), and *Empty set* (E). We use Equation 5 to determine the expected capacity. Since attackers in S jam at full power, as shown in Section 6.3, $S = S^*$ (including no jammers but all other legitimate users) yields the optimal performance. Thus, the expected capacity varies with time (in units of rounds r) until it reaches the steady state where $S_B = S^*$. The steady-state expected capacity is shown in Equation 9, where α denotes the number of legitimate users outside S (who could potentially cause interference to the transmitter) and $\binom{a}{b} = \frac{a!}{b!(b-a)!}$.

For the transient expected capacity, $\mathrm{E}[\mathcal{R}|\mathit{Random}]$ and $\mathrm{E}[\mathcal{R}|\mathit{Empty}]$ are constant in time, whereas the expected capacity for *Best so far* varies with time. The user chooses $S_B = \emptyset$ at round r if all the previously explored sets contain jammers; otherwise, he chooses the set that contains no jammer and the most nodes (minimizing α). The term $\mathrm{E}[\mathcal{R}(r)|\mathrm{Best}]$ for the r^{th} round is expressed in Equation 10 where β corresponds to the number of times that the user found a jammer-free set, and $\mathcal{B}_i$ are independent Binomial random variables with probability 0.5 ($p = 0.5$) and $T - N$ trials ($n = T - N$), since $T - N$ is the number of protocol-compliant users.

8 Evaluation

In earlier sections, we have analyzed SimpleMAC theoretically, in a manner that is general and not limited to any particular system design. In this section, we evaluate our scheme in practice both using MATLAB simulations and a testbed implementation on the WARP software radio platform [19]. As described in Section 3.1, we use SINR as our metric in this section both because capacity is strictly monotone in SINR and because we can evaluate SINR improvements without needing to make perfect modulation and coding choices that are necessary to achieve channel capacity.

8.1 Methodology and Metric

We built our implementation on the WARP (Wireless Open-Access Research) software-defined radio platform. We used four WARP boards: one acting as the source, one acting as the receiver, and the other two acting as co-existing transmitters. By using the MIMO capabilities of the boards, we built an environment consisting of one source, one receiver, and four other transmitters ($T = 4$), one of which is a jammer ($N = 1$). We divided the spectrum into five channels of equal bandwidth[1] ($C = 5$). Also, we manually calibrated the antenna locations so that the receiver observes approximately the same power from each transmitter.

For the purposes of our evaluation, we filled the queues at each node so that each transmitter transmits packets all the time. This is not a requirement of SimpleMAC; because the recipient list performance estimates will not be updated during periods without traffic, traffic is always present from the perspective of the protocol. In fact, SimpleMAC works even in dynamic environments where the jammer and competing transmitters are sometimes present and sometimes absent; in Section 8.6, we show that SimpleMAC works even better in mobile environments.

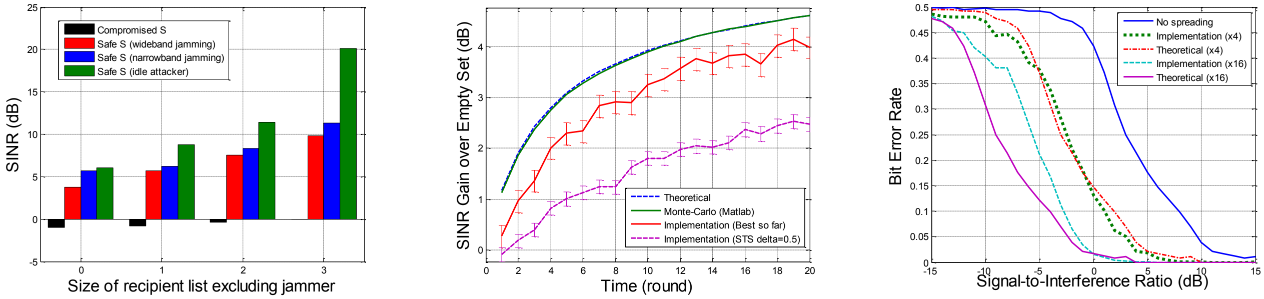
At the physical layer, we modulate data using differential quadrature phase-shift keying (DQPSK), and synchronize using a preamble which is a Barker sequence modulated using binary phase-shift keying (BPSK). We divide the entire 12 MHz-wide spectrum (centered at 2.452 GHz) into 300 OFDM subcarriers, so each channel contains 60 subcarriers. We send our control communication across the entire band (300 subcarriers). Our frequency hopping scheme is to split each data message into frames of 60 symbols, which we simultaneously send on each of 60 subcarriers in the chosen channel. We hop from channel to channel between frames.

Our transmitter sends random symbols to the receiver, and we observe the decoded symbols at the receiver. We compare these symbols to determine the error rate, and use that error rate to estimate the signal-to-interference-and-noise ratio (SINR) at the receiver. The expected bit error rate ($\overline{\mathrm{BER}}$) and the expected SINR at the receiver ($\overline{\mathrm{SINR}}$) have the following relationship for DQPSK modulation [25, 26]: $\overline{\mathrm{BER}} = \frac{1}{2}\left(1 - \frac{\sqrt{2}\cdot\overline{\mathrm{SINR}}}{\sqrt{1+4\cdot\overline{\mathrm{SINR}}+2\cdot\overline{\mathrm{SINR}}^2}}\right)$.

We also validated our results using a MATLAB-based simulation. Our simulator works on a per-packet basis: for each time slot, each transmitter chooses a recipient list according to the STS, and the channel selection according to a uniform random distribution. Our channel model is an independent, identically distributed Rayleigh fading channel with AWGN

[1] Our evaluations focus on scenarios with relatively few channels; this is not a limitation of SimpleMAC, but is a performance optimization. SimpleMAC can improve performance regardless of the number of channels, but the optimal number of channels tends to be small relative to the number of transmitters, because from a capacity perspective, it is much better to have a legitimate-to-legitimate node collision than to let spectrum go unused. We have a mathematical formulation for the optimal number of channels, but omit it due to space constraints.

$$
\begin{aligned}
\mathrm{E}[\mathcal{R}(r)|\mathrm{Best}] &= \Pr[\forall r' < r, S_R(r') \cap \mathcal{N} \neq \emptyset] \cdot \mathrm{E}[\mathcal{R}]_{S=\emptyset} + \Pr[\exists r' < r, S_R(r') \cap \mathcal{N} = \emptyset] \cdot \sum_{\alpha=0}^{T-N} \Pr[|S| = \alpha] \cdot \mathrm{E}[\mathcal{R}]_{|S|=\alpha} \\
&= (1-2^{-N})^r \cdot W \cdot \log_2\left(1 + \frac{P}{N_0 + \frac{T}{C}P}\right) \; + \; \sum_{\beta=1}^{r} \binom{r}{\beta} (1-2^{-N})^{r-\beta} (2^{-N})^{\beta} \\
&\quad \cdot \left[\sum_{\alpha=0}^{T-N} \Pr[\min_{i=1\ldots\beta} \mathcal{B}_i = \alpha] \; \cdot W \cdot \log_2\left(1 + \frac{P}{N_0 + \frac{N+\alpha}{C}P}\right)\right] \\
&= (1-2^{-N})^r \cdot W \cdot \log_2\left(1 + \frac{P}{N_0 + \frac{T}{C}P}\right) \; + \; \sum_{\beta=1}^{r} \left\{ \binom{r}{\beta} (1-2^{-N})^{(r-\beta)} 2^{-T\beta} \right. \\
&\quad \left. \cdot \sum_{\alpha=0}^{T-N} \left[\left\{ \left(\sum_{\sigma=\alpha}^{T-N} \binom{T-N}{\sigma} \right)^{\beta} - \left(\sum_{\sigma=\alpha+1}^{T-N} \binom{T-N}{\sigma} \right)^{\beta} \right\} \cdot W \cdot \log_2\left(1 + \frac{P}{N_0 + \frac{N+\alpha}{C}P}\right) \right] \right\}
\end{aligned}
\tag{10}
$$



(a) Performance for S

(b) STS performance with time

(c) The SSS performance

Figure 2: Implementation and simulation results

noise. We then compute the number of interfering users (legitimate and jammer) and calculate the resultant SINR, which we then use as feedback for the next round.

To analyze the performance of our STS in our implementation and simulation environments, we use the $S = \emptyset$ performance (corresponding to the no channel coordination) as our reference and study the performance gain over $S = \emptyset$. This gain represents the improvement over a protocol that does not reserve a channel prior to data communication. We study the *SINR gain* which is the SINR observed by the STS divided by the SINR when $S = \emptyset$. As shown in Equation 1, the instantaneous channel capacity (which we use in our theoretical analysis) is strictly monotone in instantaneous SINR, and assuming the optimal fixed strategy for jammers, this relationship extends to the time-average SINR and the time-average capacity in Equation 3.

8.2 Without attack

We first consider the performance of our protocol when all transmitters are protocol-compliant. In this scenario, our protocol minimizes unintentional interference, and the more nodes we include in our recipient list, the better our performance. Figure 2(a) shows the estimated SINR and reflects a 14.1 dB SINR increase between $|S| = 0$ (no coordination) and $|S| = 3$ (full coordination). The performance under full coordination gives us an estimate of the SNR without interference (when one channel is used): $\overline{\mathrm{SNR}} \approx 20\mathrm{dB} = 100$.

8.3 Under attack

We now consider protocol performance under jamming. If the recipient list is compromised and contains the jammer, the jammer can effectively jam the transmitter by following its hopping pattern. Otherwise, the jammer can either choose to jam across all five channels at reduced power per channel, or on a random channel at full power. Figure 2(a) displays the expected SINR at the receiver in each of the three cases. Including more legitimate users in the recipient list yields better SINR, as described in Section 8.2. We also observe a drastic drop in performance when S is compromised; whenever a set contains a jammer, its SINR is below 0 dB, since the jamming power is equal to the signal power, and other legitimate nodes may accidentally interfere. (When there is perfect coordination among legitimate nodes, the only additional noise is the receiver's thermal noise, so the SINR is very close to 0 dB in this case.) These cases with compromised recipient lists thus all perform worse than when disabling channel coordination ($S = \emptyset$), which provides 3.72 dB SINR. Furthermore, wideband jamming is more effective and yields lower SINR for the target transmitter than narrowband jamming, verifying our theoretical analysis in Section 5.1.

Despite the risk of possibly choosing the jammer, channel coordination is still potentially advantageous. Choosing a random recipient list $S = S_R$ has expected performance better than $S = \emptyset$ in expectation (our computation shows an SINR gain of about 1 dB assuming wideband jamming for uncompromised S). Furthermore, once the STS converges to the best possible set, we can reach an SINR of about 9.69 dB in spite of the wideband jamming, which reflects an SINR gain of 5.97 dB over the baseline performance of $S = \emptyset$.

8.4 Data Communication Using the STS

Now that we have established the performance of known-good and known-bad sets, we study the performance of the STS and explore its convergence behavior. For each round, the STS performs three actions (B, R, E) as described in Section 6. In our evaluation, the jammer uses the optimal strategy (full-power jamming using all available information).

Because our metric is SINR gain, and our baseline performance is the empty set, the Empty set has performance of 0 dB. The Randomly explore action chooses a recipient list at random with uniform probability. Therefore, the performance of Randomly explore is independent and has constant expectation across time. Assuming that the user randomly explores at least once per round, the Best so far performance is increasing in time, and converges to the optimal steady-state performance where $S = S^*$, as more sets are explored and the user has more sets from which to choose S_B.

In Figure 2(b), we plot the performance of the Best so far strategy under three evaluation environments: the theoretical analysis corresponding to Equation 10, our simulation, and our testbed implementation. For our implementation, we also plot 95% confidence intervals, which are not shown for our simulation results because our simulation results included enough runs that the confidence intervals would not be visible. Our results show that the performance predicted by our theoretical analysis coincides with our simulations. Our implementation performance is worse than the theoretical and simulation results because our simulation assumes a perfect measurement of SNR, whereas our implementation infers it from the bit error rate; early in the run, when the number of observed bits is small, the BER measurement can deviate from the expected BER, and the STS may as a result make suboptimal choices. However, in later rounds, this performance difference decreases as the implementation gains better information. As a result, the maximum performance difference of 23% (of implementation performance) occurs at round one and decreases to 16% at round 20 in Figure 2(b). We also show the overall (as opposed to Best so far only) performance of STS under our implementation for $\delta = 0.5$. The performance of STS overall (including Empty and Random transmission sets) is monotonically increasing in time and begins outperforming the no-channel-coordination option after round one. We will later show that the STS converges to the Best so far performance in Section 8.6.

8.5 Control Communication Using the SSS

SimpleMAC relies on a robust signaling scheme that can reach each recipient on the recipient list. We implemented our Simple Signaling Scheme in WARP using wideband communication and Direct Sequence Spread Spectrum (DSSS), as described in Section 6.2; our decoder is based on an analog correlator. In order to study the effect of spreading gain, we sent messages using three different code lengths: 1 (no spreading and thus, no redundancy), 4, and 16. We sent each chip on a separate, adjacent OFDM subcarrier. To get various signal-to-interference ratios, we fixed the signal power and varied the interference power to obtain signal-to-interference ratios ranging from -15 dB to 15 dB. Figure 2(c) shows the relationship between the BER performance and the signal-to-interference ratio.

To show the effectiveness of spreading to avoid interference in SSS, we generated theoretical curves by shifting the "no spreading" result by the expected spreading gain. As expected, the implementation result aligned well with the theoretical. When the code length is small, the performance can be slightly better than the theoretical because the x-axis considers only interference but not noise; the processing gain filters both interference and noise, which means that the implementation slightly outperforms the shifted curve. For example, when the code length is 4, the implementation has better performance than the theoretical except between -5 and -1 dB. However, with increasing code length, we use an increasing number of adjacent subcarriers, increasing intercarrier interference and degrading performance.

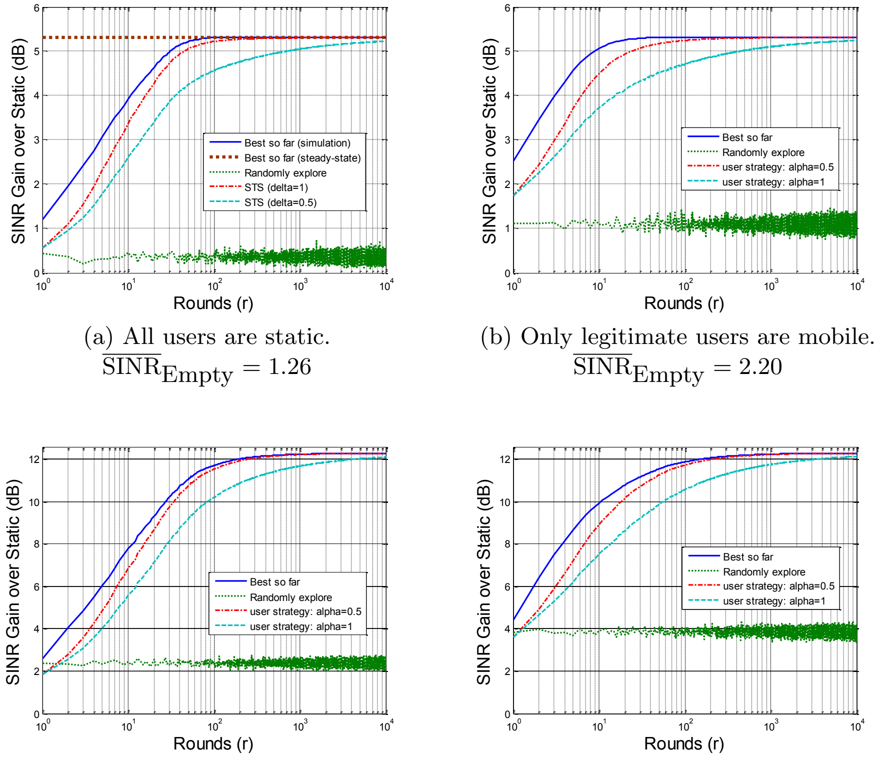
8.6 Simulations with Mobility

We have previously shown that without mobility, STS is effective. Using our simulator, we now show that introducing mobility makes STS *more effective*; that is, transmitter performance increases when it is surrounded with mobile users. In our mobile environment, at the beginning of each round, each mobile user is placed at a random position, and we compute the channel gains based on the positions and a path loss model (with path loss exponent 3) and Rayleigh fading. We fix the receiver's location and choose the position of each mobile user with a distribution such that the expected received power corresponds to unit channel gain; that is, the expected received power is the same in the mobile and static cases. We use the same parameter values as we did previously: $C = 5$, $T = 4$ (other than the source transmitter), $N = 1$, SNR (across the entire spectrum)=13 dB. To allow for comparison between scenarios, we normalize all performance to the SINR achieved using a $S = \emptyset$ in the *static case*. We also ran for an increased number of rounds (10^4). We plot performance on a semi-log scale to better show the dynamics of convergence.

In each case, we consider *convergence to improvement*, which shows how quickly the performance exceeds the no-channel-coordination case, i.e., $S = \emptyset$, and *approximate convergence*, which shows when the system reaches within 10% of steady-state performance.

For comparison, the static case, where no user moves, is shown in Figure 3(a) (and reflects the data shown in Figure 2(b)). The scheme converges to improvement instantly at round one, since the expected SINR for Random set exceeds one (as we discussed in Section 8.3), and achieves approximate convergence in about 32 rounds and 316 rounds, respectively, for $\delta = 1$ and $\delta = 0.5$, for an eventual performance gain of 5.31 dB or 3.4.

We expect that mobility will improve performance *for any set* S because of Jensen's Inequality. Using Equation 2, we observe that SINR is convex in channel gain γ of other users, so static channel gains are worse than random channel gains when the expected channel gain is equal. Adding mobility thus improves both SINR for any set S, *including* $S = \emptyset$. As more nodes become mobile (0 in Figure 3(a), 1 in Figure 3(c), 4 in Figure 3(b), and 5 in Figure 3(d)), the Empty set $\overline{\text{SINR}}$ performance monotonically increases. The values for $\overline{\text{SINR}}_{\text{Empty}}$ are shown in Figure 3.



(a) All users are static. $\overline{\text{SINR}}_{\text{Empty}} = 1.26$

(b) Only legitimate users are mobile. $\overline{\text{SINR}}_{\text{Empty}} = 2.20$

(c) Only jammers are mobile. $\overline{\text{SINR}}_{\text{Empty}} = 1.45$

(d) All users are mobile. $\overline{\text{SINR}}_{\text{Empty}} = 3.33$

Figure 3: Simulations with Mobility. Gain is relative to static-case $S = \emptyset$ performance

To show the source of the additional SINR gain, we performed two additional set of experiments, one in which only legitimate users were mobile (Figure 3(b)) and one in which only jammers were mobile (Figure 3(c)). Adding mobility for legitimate users, as shown in Figure 3(b), has minimal impact on the steady-state performance $\overline{\text{SINR}}_{\text{Best}}$, because they avoid collision and create no interference in steady-state where $S_B = S^*$. However, when legitimate users move, convergence is much faster. In particular, *approximate convergence* occurs around 16 rounds and 188 rounds, compared to the static case of 32 rounds and 316 rounds, for $\delta = 1$ and $\delta = 0.5$, respectively. This increased convergence speed is because the jammers' noise contribution is more consistent and therefore has a larger impact on performance, making jammers more easily identifiable.

On the other hand, jammer mobility, as shown in Figure 3(c), results in a substantial improvement of $\overline{\text{SINR}}_{\text{Best}}$. When $S = \emptyset$, the majority of noise comes from legitimate nodes, so jammer has limited impact. However, after many rounds, S_B approaches S^*, so jammers' interference comprises most of the noise, so $\overline{\text{SINR}}_{\text{Best}}$ improves more than $\overline{\text{SINR}}_{\text{Empty}}$.

When all nodes other than the source transmitter are mobile (Figure 3(d)), we get benefits from both user mobility and jammer mobility. leading to faster convergence and better SINR performance for all STS action choices of B,R,E as compared to the stationary scenario. The steady-state $\overline{\text{SINR}}_{\text{Best}}$ values show the mobile case having 7 dB improvement, or about five times as much improvement as, compared the static case, which yields 95% capacity improvement over the static case. Thus SimpleMAC performs even better in mobile environments than in stationary ones.

9 Alternative Transmission Priorities

In Section 4.2, we considered a single user that has the highest priority for transmission on the channel, so all other legitimate users will avoid that node's transmissions, and we calculate the performance improvement of that node. In this section, we consider nodes that have equal priority. At the steady-state where $S_B = S^*$ for all legitimate users, and where all users have equal priority, each user will defer a $1 - \sigma$ fraction of its slots, where

$$\sigma = \sum_{k=0}^{T-N} \text{Pr}_k \cdot \frac{1}{k+1}$$

where Pr_k is the probability that k other users are transmitting on the same channel and has a binomial distribution with parameters of $T - N$ trials and $\frac{1}{C}$ probability. To evaluate our scheme's performance in this egalitarian regime, we multiply the previous performance measurements by a fraction of σ. Since our simulations with mobility showed a capacity gain of 112% for a single priority node (Section 8), we observe a capacity gain of 56% when all nodes have equal priority. Similarly, without mobility, the capacity gain over baseline strategy is 51% with equal priority among all nodes.

Transmission priorities also affect how a node performs with a malicious receiver. If a malicious receiver forces a legitimate transmitter A to send with $S = \emptyset$, then the probability that any other node collides with transmitter A increases. In the worst case scenario, if transmitter A has absolute priority and is always transmitting, then A operates as a narrowband jammer. However, we can also consider the class of transmission priority schemes in which reserved channels always take priority over unreserved channels; under such schemes, in the steady-state, transmitter A will always defer to transmissions of other nodes, actually *increasing* the performance of other nodes. A more complete analysis of transmission priority schemes and malicious receivers is beyond the scope of this paper.

10 Conclusion

This paper introduces SimpleMAC, a MAC protocol that provides effective channel coordination to minimize interference among coexisting transmitters while simultaneously resisting jammers that use channel coordination information to jam more effectively. SimpleMAC avoids control channel jamming and limits jamming-relevant information to a recipient list, adjusting the recipient list to optimize performance. SimpleMAC converges to the optimal performance and forces an optimal jammer to always jam at full power. We used a game-theoretical approach to counter intelligent attackers, and analyzed the effectiveness of our scheme through theory, simulation, and implementation, and observed over 570% increases in SINR and over 50% increases in Shannon capacity gains in a realistic mobile environment.

11 Acknowledgments

This material is based upon work partially supported by USARO under Contract No. W-911-NF-0710287 and by NSF under Contract No. NSF CNS-0953600. We would also like to thank the anonymous reviewers, and our shepherd, Kun Tan, for their feedback.

12 References

[1] *IEEE Std 802.15.1-2005*, 2005.

[2] *IEEE Std 802.11-2007*, 2007.

[3] *IEEE Std 802.16-2009*, 2009.

[4] G. Alnifie and R. Simon. A multi-channel defense against jamming attacks in wireless sensor networks. In *Q2SWinet*, pages 95–104, Oct. 2007.

[5] B. Awerbuch, A. Richa, and C. Scheideler. A jamming-resistant MAC protocol for single-hop wireless networks. In *PODC*, pages 45–54, Aug. 2008.

[6] L. Baird, W. Bahn, M. Collins, M. Carlisle, and S. Butler. Keyless jam resistance. In *Information Assurance and Security Workshop*, pages 143–150, June 2007.

[7] T. Basar. The Gaussian test channel with an intelligent jammer. *IEEE Trans. Info. Theory*, 29(1):152–157, Jan. 1983.

[8] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. In *USENIX Security Symposium*, pages 15–28, Aug. 2003.

[9] A. Cardenas, S. Radosavac, and J. Baras. Performance comparison of detection schemes for MAC layer misbehavior. In *IEEE INFOCOM*, pages 1496–1504, May 2007.

[10] J. Chiang and Y. Hu. Dynamic jamming mitigation for wireless broadcast networks. In *IEEE INFOCOM*, pages 1211–1219, Apr. 2008.

[11] S. T. Chung, S. J. Kim, J. Lee, and J. Cioffi. A game-theoretic approach to power allocation in frequency-selective gaussian interference channels. In *IEEE ISIT*, pages 316–316, June 2003.

[12] L. Dryburgh and J. Hewitt. Signalling system no. 7 (SS7/C7): protocol, architecture, and services. *Cisco Press*, Aug. 2004.

[13] R. Etkin, A. Parekh, and D. Tse. Spectrum sharing for unlicensed bands. *IEEE JSAC*, 25(3):517, Apr. 2007.

[14] K. Firouzbakht, G. Noubir, and M. Salehi. On the capacity of rate-adaptive packetized wireless communication links under jamming. In *ACM WiSec*, pages 3–14, Apr. 2012.

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

[16] A. Kashyap, T. Basar, and R. Srikant. Correlated jamming on MIMO Gaussian fading channels. In *IEEE ICC*, volume 1, pages 458–462, June 2004.

[17] P. Kyasanur and N. Vaidya. Selfish MAC layer misbehavior in wireless networks. *IEEE Trans. Mobile Comput.*, 4(5):502–516, Sept. 2005.

[18] H. Li and I. Marsland. A comparison of rateless codes at short block lengths. In *IEEE ICC*, pages 4483–4488, May 2008.

[19] P. Murphy, A. Sabharwal, and B. Aazhang. Design of WARP: a flexible wireless open-access research platform. In *Proceedings of EUSIPCO*, pages 53–54, Sept. 2006.

[20] R. Pickholtz, D. Schilling, and L. Milstein. Theory of spread-spectrum communications–a tutorial. *IEEE Transactions on Communications*, pages 855–884, May 1982.

[21] M. Raya, J.-P. Hubaux, and I. Aad. Domino: a system to detect greedy behavior in ieee 802.11 hotspots. In *MobiSys*, pages 84–97, June 2004.

[22] L. G. Roberts. ALOHA packet system with and without slots and capture. *SIGCOMM CCR*, 5(2):28–42, Apr. 1975.

[23] M. Simon, J. Omura, R. Scholtz, and B. Levitt. *Spread spectrum communications handbook.* McGraw-Hill: New York, Mar. 1994.

[24] M. Strasser, S. Capkun, C. Popper, and M. Cagalj. Jamming-resistant key establishment using uncoordinated frequency hopping. In *IEEE SSP*, pages 64–78, May 2008.

[25] C. Tellambura and V. Bhargava. Unified error analysis of DQPSK in fading channels. *Electronics Letters*, 30(25):2110–2111, Dec. 1994.

[26] T. Tjhung, C. Loo, and N. Secord. BER performance of DQPSK in slow Rician fading. *Electronics Letters*, 28(18):1763 –1765, Aug. 1992.

[27] W. Xu, W. Trappe, and Y. Zhang. Channel surfing: defending wireless sensor networks from interference. In *ACM IPSN*, pages 499–508, Apr. 2007.