Delay-Bounded Adaptive UFH-based Anti-jamming Wireless Communication

Qian Wang[†], Ping Xu[‡], Kui Ren[†], and Xiang-yang Li[‡]
[†]Department of ECE, Illinois Institute of Technology, Chicago, IL 60616. Email: {qian,kren}@ece.iit.edu
[‡]Department of CS, Illinois Institute of Technology, Chicago, IL 60616. Email: {xli,pxu3}@cs.iit.edu

***Abstract*—Anti-jamming communication without pre-shared secrets has gained increasing research interest recently and is commonly tackled by utilizing the technique of uncoordinated frequency hopping (UFH). Existing researches, however, are almost all based on ad hoc designs of frequency hopping strategies, lacking of theoretical foundations for scheme design and performance evaluation. To fill this gap, this paper introduces the online optimization theory into the solution and, for the first time, makes thorough quantitative performance characterization possible for UFH-based anti-jamming communications. Specifically, we propose an efficient online UFH algorithm achieving asymptotic optimum and analytically prove its optimality under different message coding scenarios. Extensive simulative evaluations are conducted to validate our theoretical analysis under both oblivious and adaptive jamming strategies.**

I. Introduction

The broadcast nature of wireless links makes wireless communication extremely vulnerable to denial-of-service attacks [1], [2], [3]. By mounting jamming attacks an adversary can transmit signals to interfere with normal communications and temporarily disable the network. Jamming attacks can be fatal in applications where time-critical information (*e.g.*, messages to inform the soldiers an imminent attack from the enemies) or mission-critical information (*e.g.*, messages that contain the tactical planning) should be transmitted immediately. Many mitigating protocols [4], including both frequency hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS), are proposed to cope with jamming attacks. However, the effects of these anti-jamming techniques are significantly limited by their inevitable reliance on the pre-shared secrets (*i.e.*, hopping sequences and/or spreading codes) between the communicating node pairs prior to the communication as being widely recognized in the literature [2], [5], [6]. Such reliance greatly limits their applicability in scenarios where 1) the wireless network is highly dynamic with membership changes, and thus pre-sharing secrets among node pairs is impossible and 2) a sender broadcasts messages to a large number of potentially unknown receivers [5], [7].

The problem of anti-jamming communication without pre-shared secrets was first identified in [6]. The authors proposed an UFH scheme where, in order to achieve jamming resistance, both the sender and receiver hop on randomly selected channels for message transmission without coordination. The successful reception of a packet is achieved when the two nodes reside at the same frequency (channel) during the same timeslot. [2] further studied message coding techniques for UFH-based schemes. Following the same logic of breaking the *anti-jamming/key establishment dependency*, uncoordinated direct-sequence spread spectrum (UDSSS) techniques [7], [8], [9] were proposed suiting for delay-tolerant anti-jamming communication. This is because UDSSS requires a brute-force effort on message decoding at the receiver side. The existing UFH-based anti-jamming schemes, however, are almost all based on ad hoc designs of frequency hopping strategies without being able to provide quantitative performance evaluation. This is mainly due to the lack of the theoretical foundation for scheme design and performance characterization of this type. The only work on efficiency study of UFH-based communication is [5], which gives an intuitive optimal result only for the case of random jamming attacks. To fill this gap, in this paper we introduce the online optimization theory into the solution space, which enables the receiver to perform online strategy learning and optimization in response to a potentially adaptive jammer. To our best knowledge, we, for the first time, develop a delay-bounded adaptive UFH-based anti-jamming scheme and make the thorough quantitative performance characterization possible for these type of schemes. The main contributions of this paper are:

- We propose the first online adaptive uncoordinated frequency hopping algorithm against both *oblivious* and *adaptive* jammers. We analytically show that the performance difference between our algorithm and the optimal one, called *regret* in this paper is bounded, *i.e.*, no more than $O(k_r\sqrt{Tn\ln n})$ in T timeslots, where k_r is the number of frequencies the receiver can receive on simultaneously and n is the total number of orthogonal frequencies.
- We present a thorough quantitative performance characterization of the UFH-based anti-jamming scheme under various transmission/jamming strategies of the sender, the receiver and the jammer. The performance is evaluated by analyzing the expected time for message delivery with *high* probability (w.h.p) in different scenarios (*e.g*, without message coding and with erasure coding.).
- We perform an extensive simulation study of UFH-based communication to validate our theoretical results. It is shown that the proposed algorithm is efficient and effective against both *oblivious* and *adaptive* jammers.

The rest of the paper is organized as follows: Section II describes the system model, attack model and the optimal

uncoordinated frequency hopping problem addressed in this paper. Section III discusses the related work. Section IV provides the detailed description of our proposed online optimal frequency hopping scheme. Section V and Section VI present the theoretical performance analysis and simulation results, respectively. Finally, Section VII concludes the paper.

II. Problem Formulation

A. System Model

As in [6], we consider two nodes that reside within each other's transmission range and share a common time of reference. The sender wants to transmit messages to the receiver in the presence of a communication jammer. Let M denote a message that the sender wants to transfer to the receiver. Due to the frequency hopping technique, message M that does not fit into a single transmission timeslot is partitioned into multiple fragments for transmitting in successive timeslots. The transceivers employed by the nodes enable them to hop over a set of n available orthogonal channels with the same data transmission rate to send and receive signals in parallel (in the following discussion, we do not differentiate channels and frequencies). We denote the number of channels on which a node can send and receive on by k_s and k_r ($k_s, k_r \leq n$), respectively. We assume that the sender and the receiver do not pre-share any secrets (or spreading codes) with each other, and there is no feedback channel from the receiver to the sender. We also assume that none of the three parties, *i.e.*, the sender, the jammer, and the receiver, has the knowledge regarding each other's transmission/jamming strategies before the message transmission.

We also assume that at the receiver side, efficient message verification schemes (*e.g.*, erasure coding combined with short signatures) are used for message reassembly purpose [5]. As in [6], [5], we do not consider message authentication and privacy in our model. Message authentication is orthogonal to this work and can be achieved on the application layer by making use of public cryptography, timestamps etc [5]. As for message privacy, the proposed protocol can be used to transmit messages of a key establishment protocol in order to generate a secret key.

B. Attack Model

The jammer's capability has a great impact on the transceivers' hopping strategies. Due to different attack philosophies, different attack models will have different levels of effectiveness. We assume the jammer is able to jam k_j ($k_j < n$) channels simultaneously at each timeslot. Specifically, we focus on the following two jammers:

Oblivious jammer: An oblivious jammer selects the target jamming channels independent of the past communication status he may have observed. The behaviors of the oblivious jammer can be categorized into two models: *static jamming* and *random jamming*. A *static jammer* continuously emits radio signals and keeps jamming the same set of channels for each timeslot, *i.e.*, the static jammer does not change its target jamming channels over the whole message transmission process. Note that by randomly hopping among a common set of frequencies, a successful packet reception happens when the sender sends and the receiver listens on the same channel. After a number of transmission attempts, the sender and the receiver can reconcile themselves to the unjammed channels. So it is easy to defend against the static jamming attack by only keeping using the detected unjammed channels in subsequent transmissions. Similar to a *static jammer*, a *random jammer* transmits the jamming signals over a randomly selected subset of channels in each timeslot regardless of the previous communication status. Due to the random jamming strategy, the sender and the receiver are not able to find the unjammed channels and reside on them for all timeslots.

Adaptive jammer: An *adaptive jammer* adaptively selects the targeted jamming channels utilizing his past experiences and his observation of the previous communication status. By performing channel scanning, a jammer scans a set of selected channels in each timeslot in search of the sender's signals. When signals are detected, the jammer records the indexes of the corresponding channels. We assume that the jammer cannot perform the sensing and jamming operations within the *same* timeslot under the appropriately chosen channel hopping rate. For example, consider a typical sum of channel sensing time t_s and switching time t_w being 10ms [10], for a channel with data rate $B = 10$Mbps, a successful jamming attack on the transmitted packet within the *same* timeslot requires the length of packet is at least 10^5 bits. However, for the hopping rate $f_h = 500 \sim 1500$Hz [5], the length of packets will not exceed the size $B/f_h = 7 \cdot 10^3 \sim 2 \cdot 10^4$ bits, which makes sensing then attacking impossible. Yet, we still assume a very powerful *adaptive jammer* in the sense that it not only knows the protocol and can perform jamming on a subset of the n available channels of his choice during a single timeslot, but also can monitor *all* the n available channels during the same timeslot. Furthermore, an adaptive jammer knows whether it succeeded in jamming the sender's transmitting channels for all the past timeslots and can accordingly choose the targeted jamming channels for future timelsots.

During UFH-based communication, the jammer may add his own signals to the channels, *e.g.*, he can insert self-composed or replay fragments to disrupt the communication. This data pollution attack can be addressed by using the efficient message verification techniques at the receiver side [5] and thus is not explicitly considered in this work.

C. Optimal Uncoordinated Frequency Hopping: the Problem Formulation

To achieve the full potential of the UFH-based communication, we consider a frequency hopping game among a sender, a receiver and a jammer. We assume that the sender wants to send a message (partitioned into multiple fragments/packets) to the receiver under different jamming attacks. However, the sender and the receiver do not pre-share any secrets (or spreading codes) with each other, so they cannot rely on coordinated anti-jamming techniques such as FHSS and DSSS. During each timeslot, the sender chooses k_s sending channels,

and the receiver independently chooses k_r receiving channels; the jammer chooses to jam k_j channels at his will. Now, the receiver's challenge of selecting frequency hopping strategy for minimized message reception delay lies in 1) the receiver does not know the sender's and the jammer's strategies before message transmission, thus he has no best strategy to begin with[1]; 2) the receiver's strategy is desired to be adaptive optimal regardless of which sending/jamming strategies the sender and the jammer adopt.

Therefore, in order to achieve the optimal solution, we consider the above uncoordinated frequency hopping problem as a sequential decision problem [11] in which the choice of receiving channels at each timeslot is a decision. To further formalize the problem, we consider a vector space $\{0,1\}^n$ and number the available transmitting channels from 1 to n. The strategy space for the sender is set as $S_s \subseteq \{0,1\}^n$ of size $\binom{n}{k_s}$, and the receiver's is set as $S_r \subseteq \{0,1\}^n$ of size $\binom{n}{k_r}$. If the f-th channel is chosen for sending or receiving, the value of the f-th ($f \in \{1,\ldots,n\}$) entry of a vector (or strategy) is 1; 0 otherwise. The strategy space for the jammer is set as $S_j \subseteq \{0,1\}^n$ of size $\binom{n}{k_j}$. For technical convenience, in this case, the value 0 in the f-th entry denotes that the f-th channel is jammed; the value 1 in the f-th entry denotes that the f-th channel is unjammed.

During each timeslot, the three parties choose their own respective strategies s_s, s_r, and s_j. On the sender side, to adaptively adjust the sending channels based on the encountered jamming requires the *reliable* feedback information from the receiver, which is not practical. Providing the sender with the required feedback message without being exploited by the jammer is actually the same problem as the original one to be solved [5]. From the perspective of the receiver, successful receptions are determined by both its choice of strategy and the sender's and the jammer's choices of strategies. We can look $s_s \bullet s_j$ as a joint decision made by the sender and the jammer, where $\bullet$ denotes the multiplication of corresponding entries in s_s and s_j. We say that at timeslot t the sender and jammer jointly introduce a *gain* $g_{f,t} = 1$ for channel f if the value of the f-th entry of $s_s \bullet s_j$ is 1. Note that the receiver knows the state of the channel f it has *chosen* for packet reception: i) if no packet is received on f, $g_{f,t} = 0$. ii) if jamming is detected on the received packets, then $g_{f,t} = 0$. In [12], accurate *differentiation* of packet errors due to jamming from errors due to weak links can be realized by looking at the received signal strength during bit reception, even in the case of a sophisticated jammer. iii) if the packet is successfully received without being jammed, $g_{f,t} = 1$. Therefore, after choosing a strategy s_r, the value of the gain $g_{f,t}$ is revealed to the receiver if and only if f is chosen as a receiving channel. The above dynamic frequency hopping problem can be formulated as multi-armed bandit problem (MAB) [13], where only the states of the chosen arms are revealed.

[1]Otherwise, the solution is straightforward. For example, if the receiver knows that the sender and the jammer both choose the channels randomly, then his best strategy would be randomly choosing channels to listen too as proved in [5].

In each timeslot (round) t ($t \in \{1,\ldots,T\}$), the receiver selects a strategy I_t from S_r. The gain $g_{f,t} \in \{0,1\}$ introduced by $s_s \bullet s_j$ is assigned to each channel $f \in \{1,\ldots,n\}$. We write $f \in i$ if channel f is **chosen** in strategy $i \in S_r$, *i.e.*, the value of the fth entry of i is 1. Note I_t denotes a particular strategy chosen at timeslot t from the receiver's strategy set S_r, and i denotes a strategy in S_r. The total gain of a strategy i during timeslot t is

$$g_{i,t} = \sum_{f \in i} g_{f,t},$$

and the cumulative gain up to timeslot t of each strategy i is

$$G_{i,t} \quad = \quad \sum_{s=1}^{t} g_{i,s} = \sum_{f \in i} \sum_{s=1}^{t} g_{f,s}.$$

The total gain over all chosen strategies up to timeslot t is

$$\widehat{G}_t \quad = \quad \sum_{s=1}^{t} g_{I_s,s} = \sum_{s=1}^{t} \sum_{f \in I_s} g_{f,s},$$

where the strategy I_s is chosen randomly according to some distribution over S_r. To quantify the performance, we study the **regret** over T timeslots of the game

$$\max_{i \in S_r} G_{i,T} - \widehat{G}_T,$$

where the maximum is taken over all strategies available to the receiver. The **regret** is defined as the accumulated gain *difference* over T timeslots between our strategy and the **static** optimal one in which the receiver chooses the best fixed set of channels for message reception. In other words, the **regret** is the difference between the number of successfully received packets using our proposed algorithm and that using the best fixed solution. Obviously, this metric can also be used to measure the message delivery time difference between the proposed algorithm and the static optimal one. Our goal is to develop an adaptive frequency hopping algorithm that achieves asymptotic optimum with bounded **regret**.

In this work, we introduce online optimization techniques [14], [15], [16] into the design of frequency hopping algorithm against both *oblivious* and *adaptive* jammers. We evaluate the efficiency of the proposed algorithm by analyzing the expected time to achieve message delivery with *high* probability (w.h.p) and analytically prove its optimality under different message coding scenarios. The important notation used in this paper is summarized in Table I.

III. Related Work

Anti-jamming communication without pre-shared secret. The requirement of pre-shared secrets prior to the start communication creates a *circular dependency* between anti-jamming spread spectrum communication and key establishment [6], [7], [8], [9], [5]. This problem has been recently identified by Strasser et al. [6]. To break this dependency, the authors proposed an uncoordinated frequency hopping (UFH) scheme based on which messages of Diffie-Hellman key exchange protocol can be delivered in the presence of a jammer.

TABLE I
A SUMMARY OF IMPORTANT NOTATION.

Symbol	Definition
n	# of orthogonal channels
k_s	# of channels for sending at each timeslot
k_r	# of channels for receiving at each timeslot
k_j	# of jamming channels at each timeslot
l	# of packets for transmission
N	# of strategies at the receiver side
I_t	chosen strategy at timeslot t
i	a strategy in the strategy set
f	channel entry (index) in a strategy vector
$g_{f,t}$	gain for channel f at timeslot t
$g_{i,t}$	gain for strategy i at timeslot t
$G_{i,t}$	gain for strategy i up to timeslot t
$\widehat{G}_t$	total gain over chosen strategies up to timeslot t
T	# of timeslots (rounds)
$\mathcal{C}$	covering set

Due to the sender and the receiver's random choices on the sending and receiving channels, the successful reception of fragments is achieved only when the two nodes coincidentally reside at the same channel during the same timeslot. Following the same idea, [7], [8], [9] investigated uncoordinated direct-sequence spread spectrum (UDSSS) schemes suiting for delay-tolerant anti-jamming communication (*e.g.*, delay-tolerant broadcast communication). Similar to UFH, UDSSS allows a sender to hop among a public set of spreading codes for the anti-jamming purpose. At the receiver side, the receiver adopts the "try and see" method to brute-force decode the message, which inevitably introduces additional delays. The existing UFH-based anti-jamming approaches, however, are almost all based on ad hoc designs of frequency hopping strategies, and only analyze the expected message delivery time. The first work on efficiency study of UFH-based communication is recently proposed in [5], which gives an intuitive optimal result for the case of random jamming attacks only, *i.e.*, if the sender and the jammer both choose the random strategy, the receiver's best choice would be random strategy.

Online optimization and multi-armed bandit problem. In online decision problems, a decision maker performs a sequence of actions to minimize the difference between the combined cost of the algorithm and that of the best fixed one after T rounds. In the full-feedback case where the losses (or gains) of all possible actions are revealed to the decision maker, many results are known. These results show that it is possible to construct online algorithms achieving regret $O(\sqrt{T \log N})$, almost as well as the best of N experts. Multi-armed bandit problems (MAB) are an important abstraction for decision problems that incorporates an "exploration vs. exploitation" trade-off over an online learning process [13]. In a bandit setting, the decision maker knows only the loss (or gain) corresponding to the action it has made. This adversarial MAB problem was considered in [14], where an algorithm achieving $O(\sqrt{TN \log N})$ regret for the K-armed bandit problem was proposed. The online shortest path problem, which is a special case of online optimization, has been widely studied [17], [15], [18], [16]. The decision makers has to choose a path in each round such that the weight of the chosen path be as small as possible. Because the number of possible pathes is exponentially large, the direct application of [14] to the shortest path problem results a too large bound, *i.e.*, dependence on $\sqrt{N}$. To get rid of the exponential dependence on the number of edges in the performance bound, the authors in [15], [18] designed algorithms for shortest path problem using the exponentially weighted average predictor and the follow-the-perturbed-leader algorithm. However, the dependence of number of rounds T in their algorithms is much worse than that of [14] (*i.e.*, $O(T^{\frac{2}{3}})$[15] and $O(T^{\frac{3}{4}})$[18]). In [16], the authors consider the shortest path problem under partial monitoring model and proposed an algorithm with performance bound that is polynomial in the number of edges. In this paper, we formally define the optimal uncoordinated frequency hopping problem and analyze it under partial monitoring model [16], where only the gains or losses of the chosen arms are revealed to the decision maker.

IV. The Proposed Approach

A. Solution Overview

In this section, we focus on developing the frequency hopping algorithm for the receiver. Obviously, the efficiency of such frequency hopping algorithm depends on the following setting: the message size $|M|$, message and packet coding approaches, the frequency hopping rate f_h, and the sender's and the jammer's strategies. For simplicity, we do not consider packet coding as it can be easily realized using error-correcting codes. We also follow the same message coding technique as in [5], which provides online message fragment/packet verification as elaborated below.

Message coding and verification: The message M is first partitioned into multiple fragments for transmission. Let l denote the number of resulted fragments (potentially after coding). Given a desired probability of message delivery, the sender can determine the number of timeslots/rounds T for message transmission (Parameter selection will be discussed in Section V). For each message M, the sender generates a new public/private key pair (k_{pub}, k_{pri}). Then, the sender encapsulates each fragment M_i into a packet, denoted by $p_i := k_{pub}||i||l||T||M_i||Sig_{k_{pri}}(k_{pub}||i||l||T||M_i)$. As in [5], we use short signatures [19] to generate the signature $Sig_{k_{pri}}(k_{pub}||i||l||T||M_i)$. Upon receiving a packet, the receiver uses the received public key to verify the integrity of the packet. If verification fails, the packet is dropped and the receiver concludes that the channel on which this packet is received is jammed, *i.e.*, the jammer inserts bogus packets over this channel. Note that since the public and private key pair is updated for each message, packets signed with the same private key belongs to the same message.

Discussion. Note that the receiver cannot be overwhelmed by Denial of Service (DoS) jamming attacks for the following reasons. First, since the scheme is itself a UFH-based communication, the receiver will not be able to receive all the packets (either from the jammer or the sender) in the

continuous timeslots anyways. Second, the public key and private key pair is updated for each message. When the sender transmits a message (which is divided into multiple packets), the receiver will keep the verified packets (belong to the same message) until all packets of this message are received. After this, the packets of this message are deleted. Third, when the jammer replays a legitimate packet, 1) if it interferences with the sender's packet in this timeslot, the receiver will quickly detect this jamming using techniques in [12] and discard it; 2) even if the receiver receives a legitimated packet from the jammer (in this case the sender does not transmit in this timeslot, otherwise jamming is detected [12]), the verification of this packet will not overwhelm the receiver in this timeslot. This packet is kept for future message reconstruction only if the public key of this packet is the same as the other received ones and the packet has never been received before; otherwise, it will be discarded immediately.

Frequency hopping: As stated in the system model, none of the three parties, *i.e.*, the sender, jammer and receiver, has the knowledge regarding each other's transmission/jamming strategies. The receiver, however, learns the states (or *gains*) of its previously chosen channels. Accordingly, it can dynamically adjust the receiving channels for the coming timeslot. On the jammer side, an *oblivious* jammer, which does not see the receiver's past decisions, chooses the target jamming channels upfront; an *adaptive* jammer may carefully choose the target jamming channels to outwit the receiver's strategy by utilizing his past experiences. Our algorithm design takes into consideration both types of jammers.

The main difficulty in designing any channel hopping algorithm for optimized efficiency is to appropriately balance between *exploitation* and *exploration*. Such an algorithm needs to keep *exploring* the best set of channels for transmission as jammer may dynamically adjust his strategy. The performance under any static frequency hopping strategy will be inevitably degraded by an adaptive jammer. At the same time, the algorithm also needs to *exploit* the previously chosen best strategies as too much exploration will potentially underutilize them. To meet this challenge, we propose an efficient and effective online learning algorithm that achieves a proper balance between *exploitation* and *exploration* and consequently ensures the performance optimality.

B. An MAB-based Algorithm for UFH

In this section, we describe our MAB-based algorithm for UFH as shown in **Algorithm 1**, whose performance is asymptotically optimal.

Let $N = \binom{n}{k_r}$ denote the total number of strategies at the receiver side. As shown in the algorithm, each strategy is assigned a strategy weight, and each channel is assigned a channel weight. During each timeslot, the channel weight $w_{f,t}$ is dynamically adjusted based on the channel gain revealed to the receiver. The weight of a strategy $w_{i,t}$ is determined by the product of weights of all channels of the strategy and some random factors used for *exploration*. The reason to estimate gain for each channel first instead of estimating gain for each strategy directly is that the gain of each channel can provide useful information about the other unchosen strategies containing the same channel. The parameter β is to control the bias in estimating the channel gain $g'_{f,t}$.

Algorithm 1 An MAB-based algorithm for UFH

Input: n, k_r, $\delta \in (0,1)$, T, $\beta \in (0,1]$, $\gamma \in (0, 1/2]$, $\eta > 0$.
Initialization: Set initial channel weight $w_{f,0} = 1\ \forall f \in [1,n]$, initial hopping strategy weight $w_{i,0} = 1\ \forall i \in [1,N]$, and initial total strategy weight $W_0 = N = \binom{n}{k_r}$.
For timeslot $t = 1, 2, \ldots, T$

1: The receiver selects a hopping strategy I_t at random according to the strategy's probability distribution $p_{i,t}$, $\forall i \in [1,N]$, with $p_{i,t}$ computed as follows:

$$p_{i,t} = \begin{cases} (1-\gamma)\frac{w_{i,t-1}}{W_{t-1}} + \frac{\gamma}{|\mathcal{C}|} & \text{if } i \in \mathcal{C} \\ (1-\gamma)\frac{w_{i,t-1}}{W_{t-1}} & \text{if } i \notin \mathcal{C} \end{cases}$$

2: The receiver computes the probability $q_{f,t}\ \forall f \in [1,n]$, as

$$q_{f,t} = \sum_{i:f\in i} p_{i,t} = (1-\gamma)\frac{\sum_{i:f\in i} w_{i,t-1}}{W_{t-1}} + \gamma\frac{|\{i \in \mathcal{C} : f \in i\}|}{|\mathcal{C}|}$$

3: The receiver calculates the channel gain $g_{f,t-1}\ \forall f \in I_t$ based on the outcomes of jamming detection and integrity verification. Based on the revealed gains $g_{f,t-1}$, it computes the virtual channel gains $g'_{f,t}\ \forall f \in [1,n]$ as follows:

$$g'_{f,t} = \begin{cases} \frac{g_{f,t}+\beta}{q_{f,t}} & \text{if channel } f \in I_t \\ \frac{\beta}{q_{f,t}} & \text{oththerwise.} \end{cases}$$

4: The receiver updates all the weights as $w_{f,t} = w_{f,t-1}e^{\eta g'_{f,t}}$, $w_{i,t} = \Pi_{f\in i} w_{f,t} = w_{i,t-1}e^{\eta g'_{i,t}}$, $W_t = \sum_{i=1}^{N} w_{i,t}$, where $g'_{i,t} = \sum_{f\in i} g'_{f,t}$.

End

At the beginning of each timeslot, the receiver chooses his own strategy based on certain probability distribution $p_{i,t}$, where the introduction of γ is to ensure that $p_{i,t} \geq \frac{\gamma}{|\mathcal{C}|}$ so that a mixture of exponentially weighted average distribution and uniform distribution can be used [13]. A set $\mathcal{C}$ of *covering strategy* is defined to ensure that each channel/frequency is sampled sufficiently often. It has the property that for each channel f, there is a strategy $i \in \mathcal{C}$ such that $f \in i$. Since there are totally n channels and each strategy includes k_r channels, we have $|\mathcal{C}| = \lceil \frac{n}{k_r} \rceil$. Note that we use *gains* instead of *losses* in both our notations and analysis, as we are interested in the number of successful packet reception attempts instead of delay loss in the shortest path problem. The following theorem is based on that of [16] with necessary modifications and simplifications required to accommodate for the optimal frequency hopping problem.

Theorem *1:* No matter how the status of the channels change (potentially in an adversarial manner), with probability at least $1-\delta$, the **regret** of our algorithm is at most

$$6k_r\sqrt{Tn\ln n},$$

while $\beta = \sqrt{\frac{k_r}{nT}\ln\frac{n}{\delta}}$, $\gamma = 2\eta n$, $\eta = \sqrt{\frac{\ln n}{4Tn}}$ and $T \geq \max\{\frac{k_r}{n}\ln\frac{n}{\delta}, 4n\ln n\}$.

Proof: Due to space limitations, the detailed proof is provided in the full version [20]. ■

Theorem 1 shows that in T timeslots, the difference between the number of successfully received packets using Algorithm 1 and that using the optimal solution is bounded by $6k_r\sqrt{Tn\ln n}$. It is easy to see that the normalized regret of Algorithm 1 converges to zero at an $O(1/\sqrt{T})$ rate as T goes to infinity. In the next Section, we will analyze the delay performance between our strategy and the optimal ones.

V. Performance Analysis

In this section, we analyze our algorithm in different cases. As we discussed above, the size of data packet for transmission cannot be too large. Therefore, the message for transmission should be divided into small fragments or packets. However, since the transmission process is not reliable, *e.g.*, data packets may be jammed, no algorithm can guarantee the message can be delivered in certain time with probability 100%. So we consider the expected time usage such that a message could be delivered with *high* probability. Here *high* probability means the probability tends to 1 when total number of packets tends to infinite.

We say an algorithm $\mathcal{A}$ is α-*static* (*adaptive*, respectively) approximation if and only if

1) *Static* (*adaptive*, respectively) optimal solution can transmit a message successfully with high probability $1-\frac{1}{l^\epsilon}$ in time T, where constant $\epsilon > 0$.
2) Algorithm $\mathcal{A}$ can transmit the message successfully in time αT with the same probability $1-\frac{1}{l^\epsilon}$.

A. Without Message Coding

We first analyze the performance of our algorithm in the case where no message coding methods are used. Each message M is divided into l packets $M_1, M_2, \cdots, M_l$ with the same size, *i.e.*, $|M_i| = |M|/l$ for all $1 \leq i \leq l$. All l packets of message M must be received before the message M can be reassembled. Since the sender cannot get any feedback from the receiver, he has no idea about what kinds of packets have been received. Therefore, in our protocol, every time the sender want to send a packet, he will pick up a packet with the same probability $1/l$.

Lemma *2:* Receiving $(1+\epsilon)l\ln l$ packets, the probability that reconstruct the original message is at least $1-\frac{1}{l^\epsilon}$, for any constant $\epsilon > 0$.

Proof: When receiving $(1+\epsilon)l\ln l$ packets, the probability that at least one kind of packet is not received is $p \leq \binom{l}{1}(1-\frac{1}{l})^{(1+\epsilon)l\ln l} \leq l(\frac{1}{e})^{(1+\epsilon)\ln l} = \frac{1}{l^\epsilon}$. So the probability that all l kinds of packets have been received is at least $1-\frac{1}{l^\epsilon}$. ■

Lemma *3:* Receiving $l\ln l$ packets, with probability at least $1-e^{-1/4}$, the original message cannot be reconstructed.

Proof: Here we use the result of Lemma 6 in [21]. Receiving $l\ln l$ packets, with probability at least $1-e^{-1/4}$, at least one kind of packet is not received. ■

Theorem *4:* When $l \geq 36(1+c\epsilon)k_r n/(c-1)^2\epsilon^2$, our algorithm is $(1+c\epsilon)$-static approximation for any constant $c > 1$.

Proof: According to Lemma 3, to reconstruct a message with l packets with high probability in time T, the static optimal solution need to collect at least $l\ln l$ packets. Therefore, our algorithm receives $(1+c\epsilon)l\ln l - 6k_r\sqrt{(1+c\epsilon)Tn\ln n}$ packets in $(1+c\epsilon)T$ time. When $l \geq 36(1+c\epsilon)k_r n/(c-1)^2\epsilon^2$, the number of packets is no less than $(1+\epsilon)l\ln l$. According to Lemma 2, the probability to reconstruct the message is at least $1-\frac{1}{l^\epsilon}$. ■

Theorem *5:* When the sender and jammer are using the uniformly random strategy, the static optimal solution achieves same expected gain as the adaptive optimal solution.

Proof: When the sender and jammer are using uniformly random strategy, the expected gain on each channel is $\frac{k_s}{n}\frac{n-k_j}{n}$ per round/timeslot. Therefore, both the static and adaptive optimal solutions achieve expected gain $k_r\frac{k_s}{n}\frac{n-k_j}{n}$ per round/timeslot. ■

Theorems 4 and 5 imply that our algorithm is also $(1+c\epsilon)$ adaptive approximation for any constant $c > 1$, when l is sufficiently large, and the sender/jammer are using the uniformly random strategy.

Theorem *6:* When $l \geq 36\frac{n^3\min\{k_s,k_r,n-k_j\}(1+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, our algorithm is $\frac{n^2\min\{k_s,k_r,n-k_j\}}{k_sk_r(n-k_j)}(1+c\epsilon)$-adaptive approximation for any constant $c > 1$.

Proof: The adaptive optimal solution get KT packets in T time in expectation where $K = \min\{k_r, k_s, n-k_j\}$. We know that it is necessary to collects at least $l\ln l$ packets to reconstruct the message with high probability, which implies $KT \geq l\ln l$. On the other hand, since the static optimal solution collect $k_r\frac{k_s}{n}\frac{n-k_j}{n}$ in expectation each round. Therefore, in time $\frac{n^2}{k_rk_s(n-k_j)}K(1+c\epsilon)T$, our algorithm collects at least $K(1+c\epsilon)T - 6k_r\sqrt{\frac{n^2}{k_rk_s(n-k_j)}K(1+c\epsilon)Tn\ln n}$ packets. When $l \geq 36\frac{n^3\min\{k_s,k_r,n-k_j\}(1+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, the above formula is no less than $(1+\epsilon)l\ln l$. So the probability to reconstruct the message is at least $1-\frac{1}{l^\epsilon}$. ■

B. With Erasure Codes

We also consider the case where erasure codes are used in the transmission. Erasure codes allow for schemes where a message can be reconstructed if only a subset of all packets is available. Near optimal erasure codes encode a message M into cl packets of size $|M|/(l-\epsilon)$ such that any subset of l packets can be used to reconstruct M. Example of (near) optimal erasure codes are: Reed Solomon [22] and Tornado [23] codes. In our protocol with erasure codes, every time the sender want to send a packet, he will pick up a packet with the same probability $1/cl$.

Lemma *7:* Receive $(c+\epsilon)l$ packets, the probability of reconstructing the original message is at least $1-\frac{1}{l^\epsilon}$, for any constant $\epsilon > 0$.

Proof: When receiving $(c+\epsilon)l$ packets, the probability p that at least $(c-1)l+1$ kinds of packets are not received is around $p \leq \binom{cl}{l-1}(\frac{l-1}{cl})^{(c+\epsilon)l}$. According to Stirling's approximation we have $e(\frac{n}{e})^n \leq n! \leq e(\frac{n+1}{e})^{n+1}$, we get $p \leq \frac{cl+1}{e^2}(\frac{c}{c-1})^{(c-1)l+1}c^{l-1}\frac{1}{c^{(c+\epsilon)l}} \leq l^{\epsilon}$ when $\epsilon l \geq \frac{\ln(cl+1)}{\ln c}$. Therefore, the probability that at least l different kinds of packets have been received is at least $1-\frac{1}{l^{\epsilon}}$. ■

Set $c = 1+\delta$ where δ is a small constant satisfying $\epsilon l \geq \frac{\ln((1+\delta)l+1)}{\ln(1+\delta)}$, we can reconstruct a message with probability at least $1-\frac{1}{l^{\epsilon}}$ after receiving $(1+\delta+\epsilon)l$ packets.

It is also obvious that to reconstruct a message, it is necessary to collects at least l packets.

Theorem *8:* When $l \geq 36(1+\delta+c\epsilon)k_r n \ln n/(c-1)^2\epsilon^2$, our algorithm is $(1+\delta+c\epsilon)$-static approximation for any constant $c > 1$.

Proof: The proof is similar to that of Theorem 4. To reconstruct the message with high probability, it is necessary to collect at least l packets in time T. When $l \geq 36(1+\delta+c\epsilon)k_r n \ln n/(c-1)^2\epsilon^2$, in time $(1+\delta+\epsilon)T$, our algorithm will collect at least $(1+\delta+c\epsilon)l - 6k_r\sqrt{(1+\delta+\epsilon)Tn\ln n} \geq (1+\delta+\epsilon)l$. Therefore, the probability that the message can be reconstructed successfully is at least $1-\frac{1}{l^{\epsilon}}$ which finishes the proof. ■

Similarly, Theorems 5 and 8 imply that our algorithm is also $(1+\delta+c\epsilon)$-adaptive approximation for any constant $c > 1$ if l is sufficiently large, and sender/jammer are using the uniformly random strategy. We also have following theorem. The proof is similar to that of Theorem 6.

Theorem *9:* When $l \geq 36\frac{n^3 \ln n \min\{k_s,k_r,n-k_j\}(1+\delta+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, our algorithm is $\frac{n^2\min\{k_s,k_r,n-k_j\}}{k_s k_r(n-k_j)}(1+\delta+c\epsilon)$-adaptive approximation for any constant $c > 1$.

Discussion. According to Theorem 5, we know that the expected number of packets received per round is $k_r\frac{k_s}{n}\frac{n-k_j}{n}$. To maximize the number of packets received, we can set $n = 2k_j$. As discussed in Section IV-A, the sender will determine T and **encode** it in *each* packet. After receiving the first packet, the receiver knows the parameters T and runs our algorithm. Given quality requirement P, which denotes the probability that the receiver can receive the message, the sender can decide a feasible T as follows. The sender first estimates a lower bound $\underline{k_r}$ for k_r and a upper bound $\overline{k_j}$ for k_j. Compute ϵ such that $1-\frac{1}{l^{\epsilon}} = P$. Find a feasible constant $c > 1$ such that $l = 36(1+\delta+c\epsilon)\underline{k_r}n\ln n/(c-1)^2\epsilon^2$. The total time of transmission will be $T = (1+\delta+c\epsilon)l/\underline{k_r}\frac{k_s}{n}\frac{n-\overline{k_j}}{n}$. Theorem 8 can guarantee that the receiver will obtain the message with probability at least P.

VI. Simulation Results

In this section, we conduct extensive simulations to validate our theoretical results and demonstrate the performance of our MAB-based algorithm under various jamming attacks, the sender's frequency hopping strategies and packet transmission strategies. In our simulation, the sender chooses from two strategies: static sending strategy and random frequency hopping strategy; the jammer chooses from three strategies: static, random and adaptive jamming strategies, and the receiver chooses from three strategies: static receiving strategy, random and adaptive frequency hopping strategies. Note that i) In static strategy, the chosen channels remain unchanged for all timeslots; ii) In random hopping or jamming strategies, the channels are chosen uniformly at random from a public frequency set; iii) In adaptive hopping or jamming, the channels are chosen based on the MAB-based algorithm. Also note that the *adaptive* jammer, which knows whether he succeeds in jamming the transmitting channels (where both the sender and the receiver reside on in a timeslot) for all the past timeslots, is too powerful and thus infeasible in reality. In our simulation, we also compare the performance of our proposed approach with that of the receiver's *static* optimal strategy and *adaptive* optimal strategy. The *static opt* is a fixed strategy chosen to maximize the number of received packets (total gain) over T timeslots. The *adaptive opt*, which constantly chooses the best strategy in each timeslot and obtains maximized number of received packets, is actually infeasible in reality, and hence serves as the theoretical efficiency upper bound in our simulation.

We use a three-element tuple to denote the three parties' respective strategies in a particular simulation scenario, *e.g.*, "ran sta mab" denotes that the sender chooses random hopping strategy, the jammer chooses static jamming strategy and the receiver chooses adaptive frequency hopping strategy (*i.e.*, MAB-based algorithm for UFH). For each strategy setting, we run the simulation for 1000 rounds. Without loss of generality, we assume the sender and receiver have the same number of antennas with $k_s = k_r = 3$. We vary the strategies of the three parties to study the average number of received packets when T increases and the cumulative distribution function (CDF) of the expected time to reach message delivery T^*. We also vary the jammer's jamming capability (k_j) and the total number of orthogonal frequencies n to study the impact of parameter selection on the performance of UFH-based communication. We further focus on a random sender and evaluate the effectiveness of our MAB-based frequency hopping algorithm under different packet transmission strategies (*e.g.*, without coding and with erasure coding). We show that, the MAB-based algorithm is asymptotically optimal regardless of the sending/jamming strategies.

A. Without Message Coding

We first evaluate the performance of the UFH-based communication without using message coding. The purpose of the simulation is to compare the performance of our MAB-based algorithm with that of static strategy and random hopping strategy at the receiver, under different strategies of the sender and the jammer. Fig. 1 shows (i) the average number of received packets versus the number of timeslots (T) and (ii) the CDF of the expected time to achieve message delivery under different strategy settings given $l = 20$, $k_j = 7$ and $n = 2k_j$. Since the MAB-based frequency hopping algorithm enables

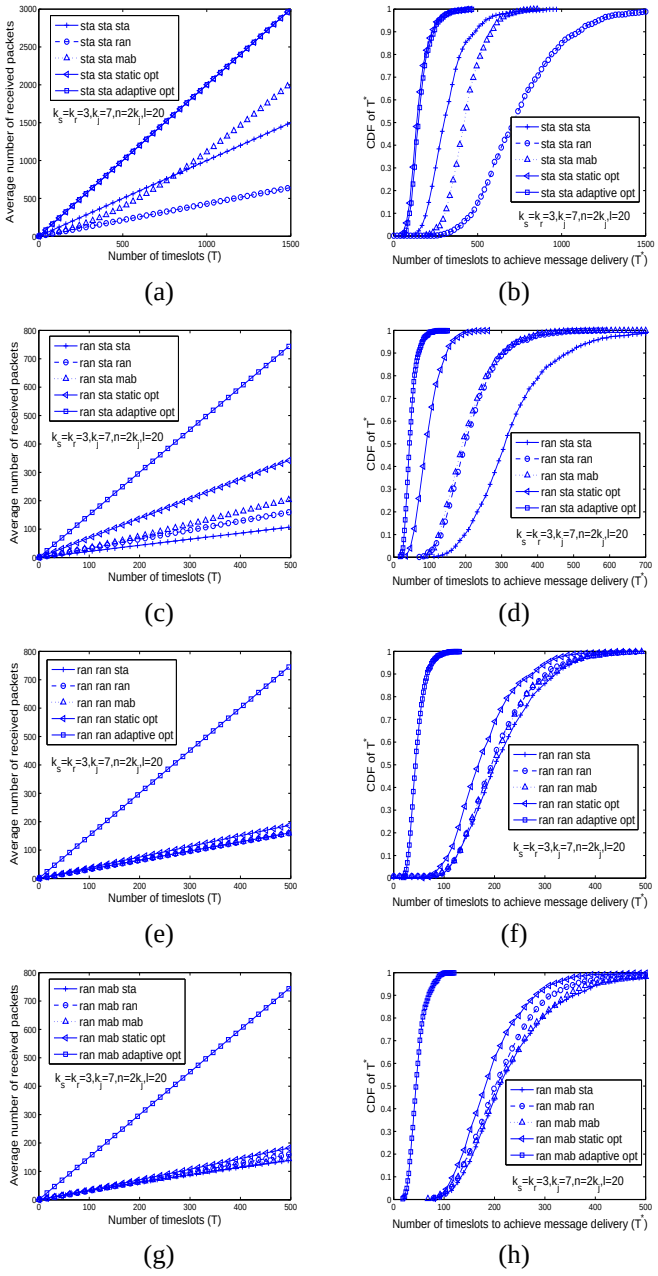

Fig. 1. Average number of received packets vs. the number of timeslots (T) and CDF of expected time to achieve message delivery under different strategy settings (without message coding)

the receiver to *explore* the best channels for transmission, it will perform better than the static strategy and random hopping strategy in a "static" environment. As shown in Fig. 1 (a) and (b), when both sender and jammer use static strategies, static receiving strategy performs the best and the random hopping performs the worst at the start of communication (In reality, by using static strategy the receiver's channels may be totally jammed or not overlap with the sender's channels. Here, we assume that the receiver chooses at least one channel that is used by the sender and not jammed.). However, as T increases, our proposed adaptive strategy outperforms the static one since the receiver has "learned" the best set of channels for transmission. In Fig. 1 (b), we find that the message is successfully received with high probability before the completion of the receiver's learning. That implies that using our MAB-based algorithm for UFH can achieve more gain when the message size is large (*i.e.*, l increases). Note that since both the sender and the jammer choose the static strategy, the static opt and the adaptive opt are the same in this case.

We next consider the case when the sender chooses random hopping strategy and the jammer chooses static jamming strategy. Here, we also assume that at least one of the receiver's chosen channels is not jammed when using static strategy. Fig. 1 (c) and (d) show that in this scenario, our adaptive hopping strategy still performs better than the static and random strategies. However, the gain difference becomes smaller between using our adaptive strategy and the random strategy due to the random hopping strategy used at the sender side. We further consider the case when both the sender and the jammer use random strategies. Fig. 1 (e) and (f) show that our adaptive strategy and the random strategy have almost the same performance. This is because, in the learning process, the receiver gradually adjust itself to a random strategy when facing a sender and a jammer both using random strategies. Note that the performance of *static opt* deteriorates much due to the random strategies used by the sender and jammer. Fixing a random sender, we explore the performance of an *adaptive* jammer in Fig. 1 (g) and (h). The results show that although being up against an adaptive jammer, the performance of our algorithm is still fairly good. In general, by using our MAB-based frequency hopping algorithm a high level of performance is achieved regardless of the sending/jamming strategies.

We next study the impact of k_j and n on the performance of UFH-based communication when our adaptive hopping strategy is used at the receiver. Assume both the sender and the receiver use random strategies, we vary k_j from 3 to 9 in our simulation. As expected, in Fig. 2 (a) and (b) the results show that the increase of k_j greatly reduces the number of received packets and delays the message delivery time especially when k_j approaches n. In Fig. 2 (c) and (d), by setting $k_j = 7$, we vary n from 8 to 18. The results show that the maximum expected number of received packets is obtained when $n = 2k_j = 14$, which matches our analytical results.

B. Message Coding Using Erasure Codes

Compared with no coding case, by using erasure codes for message coding, the message M can be reconstructed if any l distinct packets are received. Since the size of the packet pool is enlarged, the probability of picking the same packet is reduced. This results in less time in collecting l distinct packets for message recovery. Following the same parameter settings as above, we focus on a random sender and evaluate the performance of our adaptive frequency hopping strategy under different jamming attacks. Fig. 3 plots the the CDF of time to reach message delivery when different number of encoded

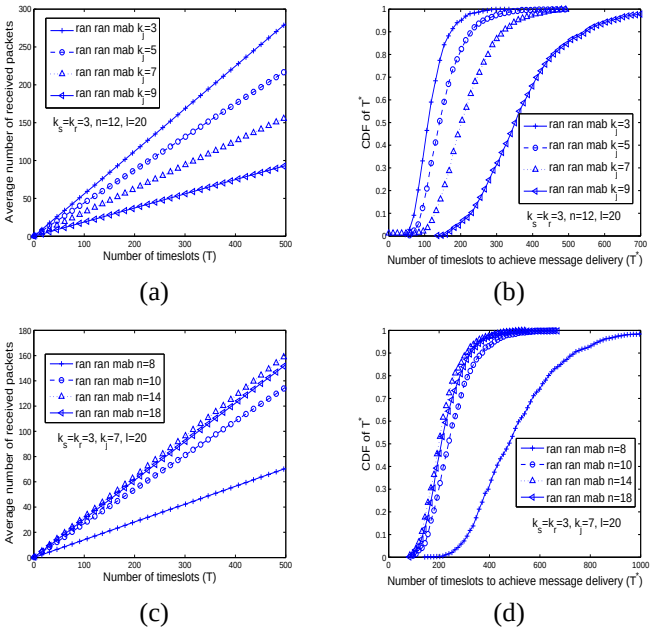

Fig. 2. Average number of received packets vs. the number of timeslots (T) and CDF of expected time to achieve message delivery under different k_j and n(without message coding)

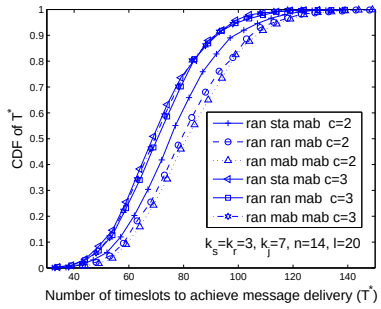

Fig. 3. CDF of expected time to reach message delivery with erasure codes.

packets are generated using erasure codes. The results show that given the probability of message delivery, the increase of c can help reduce the message delivery time. Similar to previous results, our adaptive hopping strategy performs the best when a static strategy is used by the sender or the jammer. We also note that as c becomes larger, the impact of message coding outweighs that of using different jamming attacks.

VII. Conclusion

In this paper, we introduced the online optimization theory into the frequency hopping strategy design and, for the first time, made thorough quantitative performance characterization possible for UFH-based anti-jamming communications. Specifically, we proposed an efficient online adaptive UFH algorithm achieving asymptotic optimum and analytically proved its optimality under different message coding scenarios. Extensive simulative evaluations were conducted to validate our theoretical analysis under both *oblivious* and *adaptive* jamming strategies.

Acknowledgment

This work is partially supported by the US National Science Foundation under grant CNS-0831963, CNS-0832120, NSF CNS-1035894, National Natural Science Foundation of China under Grant No. 60828003, program for Zhejiang Provincial Key Innovative Research Team, program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

References

[1] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *Proc. of ACM WISEC'09*, 2009, pp. 169–180.
[2] D. Slater, P. Tague, R. Poovendran, and B. J. Matt, "A coding-theoretic approach for efficient message verification over insecure channels," in *Proc. of ACM WISEC'09*. ACM, 2009.
[3] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. of ACM MobiHoc'05*, 2005, pp. 46–57.
[4] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Addison Wesley, 1995.
[5] M. Strasser, C. Pöpper, and S. Capkun, "Efficient uncoordinated fhss anti-jamming communication," in *Prob. of ACM MobiHoc'09*, July 2009.
[6] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in *Proc. of IEEE Security and Privacy*, May 2008.
[7] C. Pöpper, M. Strasser, and S. Capkun, "Jamming-resistant broadcast communication without shared keys," in *Proc. of the USENIX'09 Security Symposium*, 2009.
[8] T. Jin, G. Noubir, and B. Thapa, "Zero pre-shared secret key establishment in the presence of jammers," in *MobiHoc*, 2009.
[9] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized differential dsss: Jamming-resistant wireless broadcast communication," in *Proc. of IEEE INFOCOM 2010*, 2010.
[10] W. Arbaugh, "Improving the latency of the probe phase during 802.11 handoff," online at www.umiacs.umd.edu/partnerships/ltsdocs/Arbaug_talk2.pdf.
[11] A. T. Kalai and S. Vempala, "Efficient algorithms for online decision problems," *Journal of Computer System and Sciences*, vol. 71, no. 3, pp. 291–307, 2005.
[12] M. Strasser, B. Danev, and S. Čapkun, "Detection of reactive jamming in sensor networks," in *ACM Transactions on Sensor Networks (TOSN)*. ACM, 2010.
[13] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-arm bandit problem," in *Proc. of IEEE FOCS'95*, 1995, pp. 322–331.
[14] ——, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.
[15] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: distributed learning and geometric approaches," in *Proc. of ACM STOC'04*, 2004, pp. 45–53.
[16] A. György, T. Linder, G. Lugosi, and G. Ottucsák, "The on-line shortest path problem under partial monitoring," *J. Mach. Learn. Res.*, vol. 8, pp. 2369–2403, 2007.
[17] A. Kalai and S. Vempala, "Efficient algorithms for online decision problems," in *Proc. of COLT'03*, 2003, pp. 26–40.
[18] H. B. McMahan and A. Blum, "Online geometric optimization in the bandit setting against an adaptive adversary," in *COLT*, 2004.
[19] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. of ASIACRYPT'01*. Springer-Verlag, pp. 514–532.
[20] Q. Wang, P. Xu, K. Ren, and X.-Y. Li, "Delay-bounded ufh-based anti-jamming wireless communication," Illinois Institute of Technology, http://www.ece.iit.edu/ ˜ ubisec/TechReport09c.pdf, 2009.
[21] X.-Y. Li, Y. Wang, and W. Feng, "Multiple round random ball placement: Power of second chance," in *COCOON '09*, 2009, pp. 439–448.
[22] S. G. Wilson, "Digital modulation and coding," *Prentice-Hall*, 1996.
[23] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. of ACM STOC'97*, 1997, pp. 150–159.