Towards Optimal Adaptive UFH-Based Anti-Jamming Wireless Communication

Qian Wang, *Student Member, IEEE,* Ping Xu, *Student Member, IEEE,* Kui Ren, *Senior Member, IEEE,* and Xiang-Yang Li, *Senior Member, IEEE*

***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, mainly due to lack of theoretical foundations for scheme performance evaluation. To fill this gap, in this paper we introduce the online optimization theory into our solution and, for the first time, make the thorough quantitative performance characterization possible for UFH-based anti-jamming communications. Specifically, we formulate the UFH-based anti-jamming communication as a non-stochastic multi-armed bandit (MAB) problem and propose an online learning-based UFH algorithm achieving asymptotic optimum. To reduce the time and space complexity, we further develop an enhanced algorithm exploiting the internal structure of strategy selection process. We analytically prove the optimality of the proposed algorithms under various message coding scenarios. An extensive simulation study is conducted to validate our theoretical analysis and show that the learning-based UFH algorithms are resilient against both *oblivious* and *adaptive* jamming attacks.**

***Index Terms*—Anti-jamming, uncoordinated frequency hopping, multi-armed bandit problem, wireless communication.**

I. Introduction

THE BROADCAST nature of wireless links makes wireless communication extremely vulnerable to denial-of-service attacks [2], [3], [4]. 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 [5], 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 [3], [6], [7]. Such reliance greatly limits their applicability in scenarios where 1) the wireless network is highly dynamic with membership changes, and thus the pre-sharing of secrets among node pairs is impossible; and 2) a sender broadcasts messages to a large number of potentially unknown receivers [6], [8].

The problem of anti-jamming communication without pre-shared secrets was first identified in [7]. The authors proposed an uncoordinated frequency hopping (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. [3] 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 [8], [9], [10] were proposed suiting for delay-tolerant anti-jamming communication, where a brute-force effort on message decoding is required 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 the efficiency study of UFH-based communication is [6], which gave an intuitive optimal result only for the case of random jamming attacks. In practice, however, the sender and the receiver do not know the attacker's strategy in the first place when facing the jamming attack. Obviously, instead of hurriedly going to random hopping, learning first will help the receiver to get most out of the situation. To fill this gap, in this paper we introduce the online optimization theory into the solution space, which enables the receiver to perform online learning and optimization in response to a potentially adaptive jammer. To the best of our knowledge, we, for the

Manuscript received 15 January 2011; revised 15 July 2011. The research of Kui Ren is partially supported by the US National Science Foundation under grants CNS-0831963 and CNS-1117811. The research of Xiang-Yang Li is partially supported by the US National Science Foundation under grants CNS-0832120, CNS-1035894, program for Zhejiang Provincial Key Innovative Research Team, and program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program). A preliminary version [1] of this paper was presented at the 30th IEEE International Conference on Computer Communications (INFOCOM'11), April 10-15, Shanghai, China.

Q. Wang and K. Ren are with Department of Electrical and Computer Engineering, Illinois Institute of Technology, 3301 South Dearborn Street, Suite 103 Siegel Hall, Chicago, IL, 60616 USA (e-mail: {qian, kren}@ece.iit.edu).

P. Xu is with Department of Computer Science, Illinois Institute of Technology, 10 West 31st Street, Stuart Building 235, Chicago, IL 60616 USA (e-mail: pxu3@cs.iit.edu).

X.-Y. Li is with Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China and Department of Computer Science, Illinois Institute of Technology, 10 West 31st Street, Stuart Building 235, Chicago, IL 60616 USA (e-mail: xli@cs.iit.edu).

Digital Object Identifier 10.1109/JSAC.2012.120103.

first time, develop an almost optimal and adaptive UFH-based anti-jamming scheme and make the thorough quantitative performance characterization possible for this type of schemes. The main contributions of this paper are:

1. We formulate the UFH-based anti-jamming communication as a non-stochastic MAB problem and propose the first online adaptive UFH 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 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 simultaneously and n is the total number of orthogonal frequencies. We also show that the proposed algorithm can be implemented efficiently with time complexity $O(k_r nT)$ and space complexity $O(k_r n)$.

2. We present a thorough quantitative performance characterization of UFH-based anti-jamming schemes 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, with (rateless) erasure coding). We also discuss the parameter selection (the number of transmitting packets l, the total transmission time T, and the optimal number of orthogonal frequencies n) for achieving performance optimality. 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.

Organization. The rest of the paper is organized as follows: Section II describes the system model, the attack model, the multi-armed bandit problem and the optimal UFH problem addressed in this paper. Section III provides the detailed description of our proposed adaptive UFH schemes. Section IV and Section V present the theoretical performance analysis and simulation results, respectively. Section VI discusses the related work. Finally, Section VII concludes the paper.

II. Network Models and Problem Formulation

A. System Model

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 the message the sender wants to transfer to the receiver. Due to the use of frequency hopping, 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 to send and receive signals in parallel, with the same data transmission rate. 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 with each other, and there is no feedback channel from the receiver to the sender (see Fig. 1). We also assume that none of the three parties, *i.e.*, the sender, the jammer, and the receiver, has knowledge of each other's transmission/jamming strategies before the message transmission.

We assume that at the receiver side, efficient message verification schemes (*e.g.*, erasure coding combined with short signatures) are used for message reassembly purpose [6]. As in [7], [6], 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. 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 in each timeslot. Specifically, we focus on the following two types of jammers:

1) Oblivious jammer: An *oblivious jammer* selects the target jamming channels independent of the past communication status he may have observed. The behaviors of the jammer can be categorized into *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.*, it 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. On the other hand, a *random jammer* transmits the jamming signals over a randomly selected subset of channels in each timeslot. Due to the use of random jamming strategy, the sender and the receiver are not able to find the unjammed channels and reside on them for all timeslots.

2) Adaptive jammer: An *adaptive jammer* adaptively selects the target 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 [11], 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 [6], 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

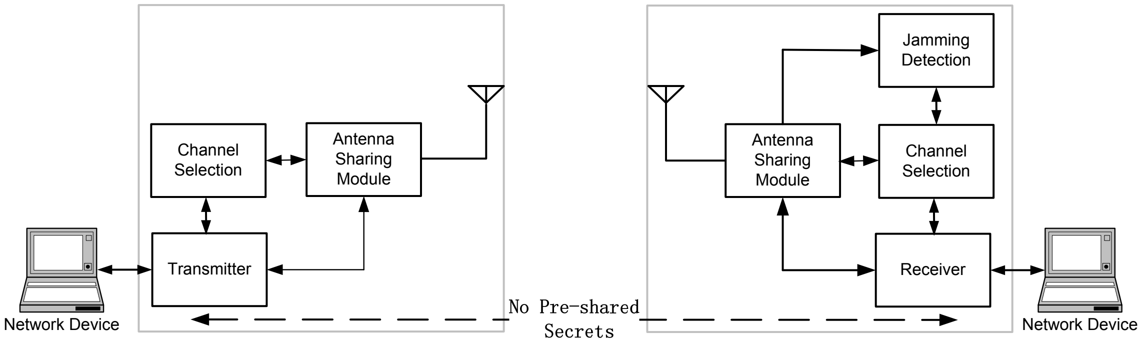


Fig. 1. Anti-jamming Communication without Pre-shared Secrets.

the protocol and can perform jamming on k_j channels of his choice during a single timeslot, but also knows whether it succeeded in jamming the sender's transmitting channels for all the past timeslots and can accordingly choose the target jamming channels for future timeslots.

Discussion. Note that the assumption that sensing and attacking within the same timeslot is impossible is made by most of research works in this area [11], [12]. The empirical data in [11] clearly shows that sensing a channel alone takes tens of ms and probing a new one also takes at least tens of ms, and the lower bound is chosen for the purpose of exposition in the example. However, it does not mean that the proposed anti-jamming scheme is constrained by this bound. Actually, even if the adversary has more powerful capability, *e.g.*, sensing in less than 10ms, such attack can be defended by reducing the packet length so that the attacker cannot have enough time to perform "sensing and jamming" in each timeslot. Also note that, 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 [6] and thus is not explicitly considered in this work.

C. Multi-armed Bandit Problem

In classic multi-armed (k-armed) bandit (MAB) problems, a gambler operates *exactly* one machine at each timeslot; all other machines remain frozen. Each operated machine provides a reward drawn from a known distribution associated with that specific machine. The objective of the gambler is to maximize the sum of rewards earned through a sequence of machine operations. Gittins *et al.* [13] proved that an optimal solution for the this problem is of *index type*. When $m (m < k)$ machines are operated each time and each machine evolves over time even not being operated, the problem becomes a restless multi-armed bandit problem (RMBP). Whittle [14] showed that an optimal solution of the *index type* can also be established in some cases. In this version of the bandit problem, the generation of rewards is assumed to be subject to certain distributions that are known to the gambler. Non-stochastic multi-armed bandit problems are another important version of MAB problems that incorporate an "exploration vs. exploitation" trade-off over an online learning process [15], [16]. The non-stochastic MAB is widely used in solving online shortest path problems, where the decision makers has to choose a path in each round such that the weight of the chosen path is as small as possible [17], [18], [19], [20].

In this paper, we formulate the anti-jamming spectrum sensing and access problem as a non-stochastic MAB problem and analyze it under partial monitoring model, where only the rewards (gains) of the chosen arms are revealed to the decision maker.

D. 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 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 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 UFH problem as a sequential decision problem [21] 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

[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 jam as proved in [6].

$S_j \subseteq \{0,1\}^n$ 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 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 [6]. 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 (not a dot product). We say that at timeslot t the sender and jammer jointly introduce a *gain* $g_{f,t} = 1$ to channel f if the value of the f-th entry of $s_s \bullet s_j$ is 1. Whether the receiver can obtain the reward or not depends on the state of the channel f it has *chosen* for packet reception:

1) No packet is received on f, $g_{f,t} = 0$;
2) A packet is received on f. If the packet fails to pass the verification (*i.e.*, jamming based DoS attack), no *gain* is obtained, $g_{f,t} = 0$. For packet verification and message reassembly purpose, we use efficient message verification schemes in [6] (*e.g.*, erasure coding combined with short signatures);
3) A packet is received on f. If jamming/collision is detected on the received packet, no *gain* is obtained, $g_{f,t} = 0$. As for jamming detection, real experiments have shown in [22] that accurate *differentiation* of packet errors due to jamming and errors due to weak links can be realized by looking at the received signal strength during bit reception. Here, we do not differentiate packet jamming and collision as they both cause interference to the legitimate packets. For simplicity, we do not consider packet coding, so the jammed or collided packets are discarded, resulting in no reward;
4) A packet is received on f. If no jamming is detected, a *gain* 1 is obtained, $g_{f,t} = 1$.

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) [15], where only the states of the chosen arms are revealed.

In each timeslot (round) t ($t \in \{1,\ldots,T\}$), the receiver selects a strategy I_t from S_r. The gain $g_{f,t} \in \{0,1\}$ 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 f-th 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

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

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

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

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.

In this work, we introduce online optimization techniques [16], [18], [20] 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.

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

III. The Proposed Approach: Optimal Adaptive Uncoordinated Frequency Hopping

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 factors: the message size $|M|$, the 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 follow the same message coding technique as in [6], 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 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 IV). 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 [6], we use short signatures [23] to generate the signature $Sig_{k_{pri}}(k_{pub}||i||l||T||M_i)$ (The reason to use short signatures instead of conventional signatures is to reduce the signature length and the public key length). 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. Note that since the public and private key pair is updated for each message, packets signed with the same private key belong to the same message. In our protocol, since the receiver will not send an ACK for each received packet, the sender does not know whether an individual packet is received. However, after message M is reconstructed, the receiver will transmit an acknowledgment to notify the sender that the whole message is delivered.

Discussion. Before the transmission of messages, the sender and receiver will first authenticate each other to prevent the insertion of fake messages generated by the jammer. Specifically, in our protocol, the sender and receiver can exchange of public key certificates issued by the CA using the proposed adaptive UFH protocol. Since the sender may generate different public/private key pairs for different messages, these public key certificates can be pre-loaded by the CA prior to the protocol execution to reduce the involvement of CA in signing public keys.

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 [22] 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 [22]), the verification of this packet will not overwhelm the receiver in this timeslot. This is because we can use timestamps to preclude replay attacks and 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 knowledge of 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 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. In our algorithm, each strategy is assigned a strategy weight, and each channel is assigned a channel weight. During each timeslot, the channel weight is dynamically adjusted based on the channel gains revealed to the receiver. The weight of a strategy is determined by the product of weights of all channels of that strategy and some random factor used for *exploration*. The reason to estimate gain for each channel first instead of estimating gains for each strategy directly is that the gains of each channel can provide useful information about the other unchosen strategies containing the same channels.

Let N denote the total number of strategies at the receiver side. The parameter β is to control the bias in estimating the channel gain $g'_{f,t}$. 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 [15]. 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 [20] with necessary modifications and simplifications required to accommodate for the optimal UFH problem.

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 $t = 1, 2, \ldots, T$
    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:
        $p_{i,t} = \begin{cases} (1-\gamma)\frac{w_{i,t-1}}{W_{t-1}} + \frac{\gamma}{|\mathcal{C}|} & \text{if } i \in \mathcal{C} \\ (1-\gamma)\frac{w_{i,t-1}}{W_{t-1}} & \text{if } i \notin \mathcal{C} \end{cases}$
    The receiver computes the probability $q_{f,t} \, \forall f \in [1,n]$, as
        $q_{f,t} = \sum_{i: f \in i} p_{i,t} = (1-\gamma)\frac{\sum_{i: f \in i} w_{i,t-1}}{W_{t-1}} + \gamma \frac{|\{i \in \mathcal{C} : f \in i\}|}{|\mathcal{C}|}$
    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:
        $g'_{f,t} = \begin{cases} \frac{g_{f,t}+\beta}{q_{f,t}} & \text{if channel } f \in I_t \\ \frac{\beta}{q_{f,t}} & \text{oththerwise.} \end{cases}$
    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
```

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

Proof: See Appendix A. ■

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 Section IV, we will analyze the delay performance between our strategy and the optimal ones.

C. An Enhanced Algorithm

It is obvious that the implementation of Algorithm 1 has time and space complexity $O(n^{k_r})$. As the number of channels increases, the strategy space will become exponentially large, which will result in low efficiency. To address this problem, we propose an enhanced algorithm utilizing dynamic programming techniques, as shown in Algorithm 2. The basic idea of the enhanced algorithm is to choose the receiving channels one by one until k_r channels are chosen, instead of choosing strategy from the large strategy space in each round (timeslot). Let $S(\overline{f}, \overline{k})$ denote the strategy set in which each strategy chooses $\overline{k}$ channels from channel $\overline{f}, \overline{f}+1, \cdots, n$. We also use $\overline{S}(\overline{f}, \overline{k})$ to denote the strategy set in which each strategy chooses $\overline{k}$ channels from channel $1, 2, \cdots, \overline{f}$. We define

$$\begin{aligned} W_t(\overline{f}, \overline{k}) &= \sum_{i \in S(\overline{f}, \overline{k})} \prod_{f \in i} w_{f,t} \\ \overline{W}_t(\overline{f}, \overline{k}) &= \sum_{i \in \overline{S}(\overline{f}, \overline{k})} \prod_{f \in i} w_{f,t}, \end{aligned}$$

and they have the following properties:

$$W_t(\overline{f}, \overline{k}) = W_t(\overline{f}+1, \overline{k}) + w_{\overline{f},t} W_t(\overline{f}+1, \overline{k}-1) \quad (1)$$

$$\overline{W}_t(\overline{f}, \overline{k}) = \overline{W}_t(\overline{f}-1, \overline{k}) + w_{\overline{f},t} \overline{W}_t(\overline{f}-1, \overline{k}-1). (2)$$

By letting $W_t(\overline{f}, 0) = 1$, $W_t(n+1, \overline{k}) = \overline{W}_t(0, \overline{k}) = 0$, both $W_t(\overline{f}, \overline{k})$ and $\overline{W}_t(\overline{f}, \overline{k})$ can be computed in time $O(k_r n)$ by using dynamic programming for all $1 \leq \overline{f} \leq n$ and $1 \leq \overline{k} \leq k_r$.

Instead of drawing a strategy as in Algorithm 1, we now choose channel one by one until a strategy is found. Assume we make decision on each channel one by one in increasing order of their indices, *i.e.*, we first decide whether channel 1 should be chosen or not, and then channel 2, and so on. For any channel f, if $k \leq k_r$ channels have been chosen in channel $1, \cdots, f-1$, we choose channel f with probability

$$\frac{w_{f,t-1} W_{t-1}(f+1, k_r - k - 1)}{W_{t-1}(f, k_r - k)}. \quad (3)$$

Correctness: Let $w(f) = w_{f,t-1}$ if channel f is chosen in the strategy i; 0 otherwise. $w(f)$ is the weight of f in the total weight of the strategy. In our algorithm, $w_{i,t-1} = \prod_{f=1}^{n} w(f)$. Let $c(f) = 1$ if channel f is chosen in the strategy i; 0 otherwise. $\sum_{f=1}^{\overline{f}} c(f)$ denotes the number of channels chosen among channels $1, 2, \cdots, \overline{f}$ in strategy i. In this implementation, the probability that a strategy i is chosen is

$$\begin{aligned} \prod_{\overline{f}=1}^{n} \frac{w(\overline{f}) W_{t-1}(\overline{f}+1, k_r - \sum_{f=1}^{\overline{f}} c(f))}{W_{t-1}(\overline{f}, k_r - \sum_{f=1}^{\overline{f}-1} c(f))} &= \frac{\prod_{\overline{f}=1}^{n} w(\overline{f})}{W_{t-1}(1, k_r)} \\ &= \frac{w_{i,t-1}}{W_{t-1}}. \end{aligned}$$

The probability is exactly same as that in Algorithm 1, which implies the correctness of this implementation.

In Algorithm 2, we do not maintain the total weight of each strategy $w_{i,t}$. Thus, different from Algorithm 1, the probability $q_{f,t}$ can be computed within $O(nk_r)$ as in Eq. (4) for each round. It is easy to see that the time and space complexity of Algorithm 2 are $O(k_r nT)$ and $O(k_r n)$, respectively.

IV. 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 that the message will be delivered in certain time with probability 100%. So we

$$(1-\gamma)\frac{\sum_{k=0}^{k_r-1}\overline{W}_{t-1}(f-1,k)w_{f,t-1}W_{t-1}(f+1,k_r-k-1)}{W_{t-1}(1,k_r)}+\gamma\frac{|\{i\in\mathcal{C}:f\in i\}|}{|\mathcal{C}|} \tag{4}$$

Algorithm 2 An enhanced 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 $w_{f,0} = 1 \; \forall f \in [1,n]$,
Let $W_t(f,0) = 1$ and $W(n+1,k) = \overline{W}(0,k) = 0$ and
compute $W_0(f,k)$ and $\overline{W}_0(f,k)$ following Eqs. (1) and (2),
respectively.
For timeslot $t = 1, 2, \ldots, T$
    The receiver selects channel $f \; \forall f \in [1,n]$ one by
    one according to the channel's probability distribution
    computed following Eq. (3) until a strategy with $k_r$ chosen
    channels is selected.
    The receiver computes the probability $q_{f,t} \; \forall f \in [1,n]$
    following Eq. (4).
    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 com-
    putes the virtual channel gains $g'_{f,t} \; \forall f \in [1,n]$ as follows:
        $g'_{f,t} = \begin{cases} \frac{g_{f,t}+\beta}{q_{f,t}} & \text{if channel } f \in I_t \\ \frac{\beta}{q_{f,t}} & \text{oththerwise.} \end{cases}$
    The receiver updates the channel weight as $w_{f,t} = w_{f,t-1}e^{\eta g'_{f,t}} \; \forall f \in [1,n]$, and computes $W_t(f,k)$ and
    $\overline{W}_t(f,k)$ following Eqs. (1) and (2), respectively.
End
```

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.

Definition 1: The *static* optimal solution is the best fixed strategy, *i.e.*, the best fixed k_r receiving channels over T timeslots. The *adaptive* optimal solution is a sequence of strategies that always maximize the gains at each timeslot, *i.e.*, a sequence of k_r receiving channels that adaptively change. An algorithm $\mathcal{A}$ is α-*static* (*adaptive*, respectively) approximation if and only if

(1) The *Static* (*adaptive*, respectively) optimal solution can transmit a message successfully with high probability (w.h.p) $1-\frac{1}{l^\epsilon}$ in time T, where constant $\epsilon > 0$.

(2) Algorithm $\mathcal{A}$ can transmit the message successfully in time αT with the same probability $1-\frac{1}{l^\epsilon}$.

Theorems derived in the following sections clearly identify the approximation ratio of the proposed adaptive UFH algorithm under different coding scenarios.

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 fregaments/packets $M_1, M_2, \cdots, M_l$ with the same size, *i.e.*, $|M_i| = |M|/l$ for all $1 \le i \le l$. 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 wants 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 to 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 $p \le \binom{l}{1}(1-\frac{1}{l})^{(1+\epsilon)l\ln l} \le l(\frac{1}{e})^{(1+\epsilon)\ln l} = \frac{1}{l^\epsilon}$. 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 [24]. 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 $l \ge 36(1+c\epsilon)k_r n/(c-1)^2\epsilon^2$, 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 needs 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 $l \ge 36(1+c\epsilon)k_r n/(c-1)^2\epsilon^2$, 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 the 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 $l \ge 36\frac{n^3\min\{k_s,k_r,n-k_j\}(1+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, our algorithm is $\frac{n^2\min\{k_s,k_r,n-k_j\}}{k_s k_r(n-k_j)}(1+c\epsilon)$-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 collect at least $l\ln l$ packets to reconstruct the message with high probability, which implies $KT \ge l\ln l$. On the other hand, the static optimal solution collect $k_r\frac{k_s}{n}\frac{n-k_j}{n}$ packets in expectation each round. Therefore, in time $\frac{n^2}{k_r k_s(n-k_j)}K(1+c\epsilon)T$, 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 [25] and Tornado [26] 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 collect at least l packets.

Theorem 8: When $l \geq 36(1+\delta+c\epsilon)k_rn\ln n/(c-1)^2\epsilon^2$, 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 $l \geq 36(1+\delta+c\epsilon)k_rn\ln n/(c-1)^2\epsilon^2$, in time $(1+\delta+c\epsilon)T$, our algorithm will collect at least $(1+\delta+c\epsilon)l-6k_r\sqrt{(1+\delta+c\epsilon)Tn\ln n} \geq (1+\delta+\epsilon)l$. 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 $\frac{n^2\min\{k_s,k_r,n-k_j\}}{k_sk_r(n-k_j)}(1+\delta+c\epsilon)$-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 collect at least l packets to reconstruct the message with high probability, which implies $KT \geq l$. On the other hand, since the static optimal solution collect $k_r\frac{k_s}{n}\frac{n-k_j}{n}$ packets in expectation each round. Therefore, in time $\frac{n^2}{k_rk_s(n-k_j)}K(1+\delta+c\epsilon)T$, our algorithm collects at least $K(1+\delta+c\epsilon)T-6k_r\sqrt{\frac{n^2}{k_rk_s(n-k_j)}K(1+\delta+c\epsilon)Tn\ln n}$ packets. When $l \geq 36\frac{n^3\ln n\min\{k_s,k_r,n-k_j\}(1+\delta+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, the above formula is no less than $(1+\delta+\epsilon)l$. So the probability to reconstruct the message is at least $1-\frac{1}{l^\epsilon}$. ■

C. With Fountain Codes

We also consider the case where fountain codes are used in the transmission. Fountain codes (also called rateless erasure codes) do not generate a finite set of packets but a potentially infinite packet sequence. When the fregament/packet size of a message M is $|M|/(l-\epsilon)$, the encoded message can be reconstructed from any set of l different packets. $\epsilon = 0$ for optimal fountain codes. Example of efficient near optimal fountain codes are: Online [27], LT [28], and Raptor [29] codes.

Similar to previous subsection, we can obtain following theorems. Briefly speaking, compared with erasure codes, the approximation ratios are reduced by a factor $\frac{1+c\epsilon}{1+\delta+c\epsilon}$ when fountain codes are used in our protocol. Notice that the improvement could be big when l is not sufficiently large for a small δ.

Theorem 10: When $l \geq 36(1+c\epsilon)k_rn\ln n/(c-1)^2\epsilon^2$, our algorithm is $(1+c\epsilon)$-static approximation for any constant $c > 1$.

Proof: To reconstruct the message with high probability, it is necessary to collect at least l packets in time T. In time $(1+c\epsilon)T$, our algorithm will collect at least $(1+c\epsilon)l-6k_r\sqrt{(1+c\epsilon)Tn\ln n}$ packets. When $l \geq 36(1+c\epsilon)k_rn\ln n/(c-1)^2\epsilon^2$, the number of packets is no less than $(c+\epsilon)l$. Therefore, the probability that the message can be reconstructed successfully is at least $1-\frac{1}{l^\epsilon}$ which finishes the proof. ■

Theorem 11: When $l \geq 36\frac{n^3\ln n\min\{k_s,k_r,n-k_j\}(1+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, 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: We know that it is necessary to collect at least l packets to reconstruct the message with high probability, which implies $KT \geq l$. On the other hand, since the static optimal solution collect $k_r\frac{k_s}{n}\frac{n-k_j}{n}$ packets 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 $l \geq 36\frac{n^3\ln n\min\{k_s,k_r,n-k_j\}(1+c\epsilon)}{k_s(n-k_j)(c-1)^2\epsilon^2}$, the above formula is no less than $(c+\epsilon)l$. So the probability to reconstruct the message is at least $1-\frac{1}{l^\epsilon}$. ■

D. Parameter Analysis

1) Impact of Number of Total Channels n: Previous analysis implies that a large n does not achieve a good performance. Essentially, if n is too large, even there is no jammer, it will be very difficult for sender and receiver to meet in a common channel without preknowledge. So here we discuss how to choose the number of total channels n. 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 $n = 2k_j$. Based on this setting, our algorithm is constant adaptive approximation for various cases since the values of k_s, k_r, k_j are generally not very large. For example, when $k_s = k_r = k_j$, with erasure coding, our algorithm is $4(1 + \delta + c\epsilon)$-adaptive approximation for any constant $c > 1$.

2) Impact of Number of Total packets l: Next we discuss how to choose a good l from the perspective of the sender. We know that the size of each packet cannot be too large, otherwise, the jammer can sense then jam the transmission in the same timeslot. This introduces a lower bound of the number of total packets l. Our previous analysis *seemly* implies that larger l leads to better performance for our algorithm. Essentially, larger l induces more rounds for transmission, which gives our algorithm more opportunities to learn and achieve static optimal solution. However, our analysis is based on the performance difference between our algorithm and the static optimal solution. In practice, erasure coding induces constant size of overhead in each packet. A too large l will lead to a large number of overhead which decreases the performance for both our algorithm and the static optimal solution. For example, assume the size of overhead is C. With erasure coding, to reconstruct a message of size M with high probability, we need to collect at least l packets. The total size of data transmission is $l(\frac{|M|}{l} + C) = |M| + Cl$. When the transmission rate is fixed, the time spent in transmission is linear to the size of data. To minimize the total time spent in transmission, we should choose l as small as possible. The smallest possible is $l = \frac{|M|}{S-C}$, where $S = B/f_h$. Therefore, there is a trade-off between using relatively large l for facilitating learning and using small l for reducing the size of the transmission data.

3) Impact of Total Transmission Time T: Notice that the parameters β, η and γ are determined by the total transmission time T. Here we discuss how to choose a feasible T for our algorithm. In our protocol, the sender will decide T and encode it in each packet. After receiving the first packet, the receiver knows the parameter 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. Here we use the case where erasure codes are used as an example. The sender first computes n and $l = \frac{|M|}{S-C}$ as we previously discussed. Then the sender needs to estimate a lower bound $\underline{k_r}$ for k_r and a upper bound $\overline{k_j}$ for k_j. It computes ϵ such that $1-\frac{1}{l^\epsilon} = P$ and finds 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. Similarly, we can compute a feasible T when no message codes are used, or fountain codes are used.

V. Simulation Results

In this section, we conduct extensive simulations to validate our theoretical results and demonstrate the performance of our MAB-based algorithm for UFH under various jamming attacks, sender's sending strategies and packet transmission strategies.

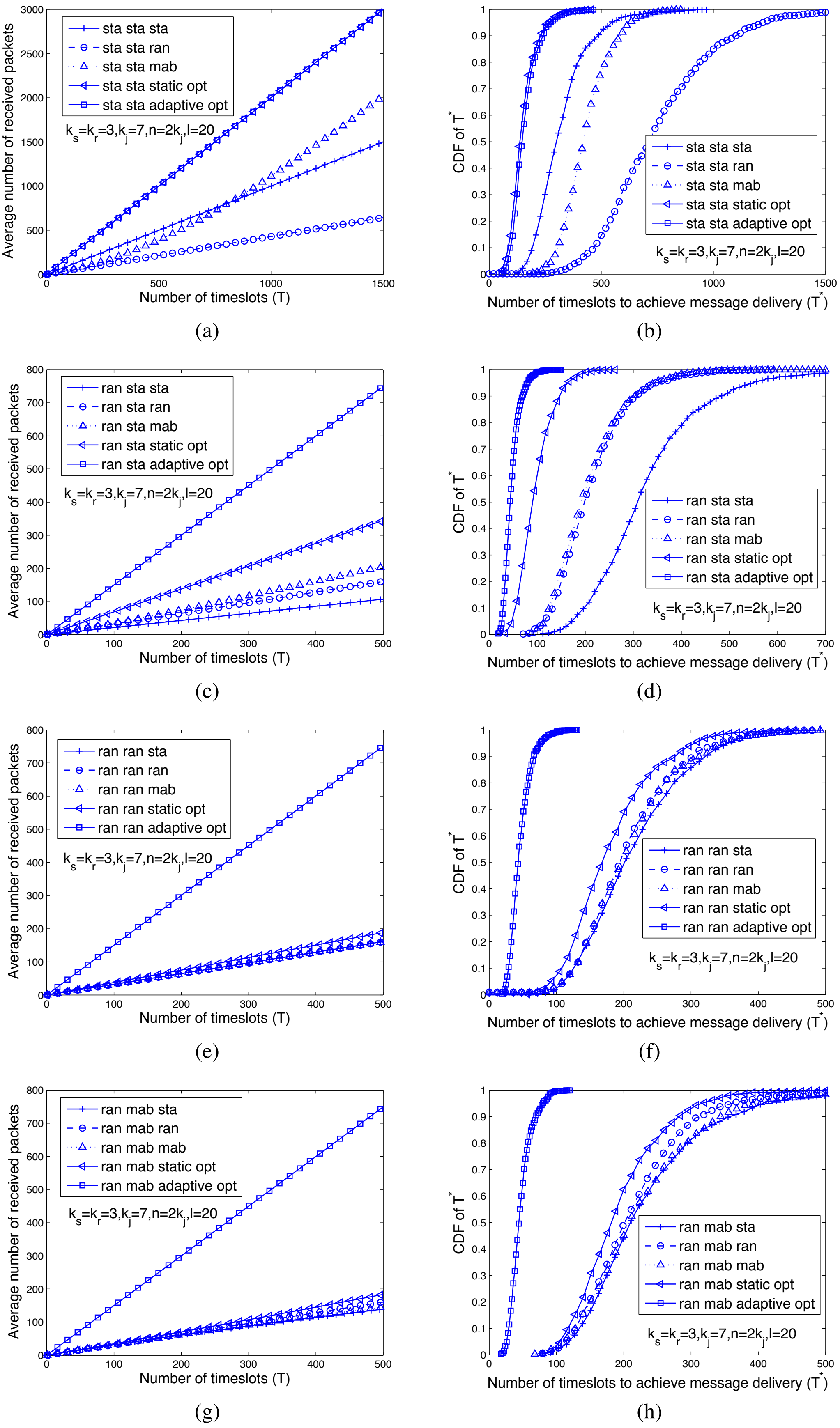
In our simulation, the sender chooses from one of two strategies: static sending strategy and random frequency hopping strategy; the jammer chooses from one of three strategies: random, static and adaptive jamming strategies, and the receiver chooses from one of three strategies: static receiving strategy, random and adaptive frequency hopping strategies. Note that i) In static strategies, the chosen channels remain unchanged for all timeslots; ii) In random strategies, the channels are chosen uniformly at random from a public frequency set; iii) In adaptive strategies, the channels are chosen using the MAB-based algorithm. Also note that an *adaptive* jammer, which knows whether it succeeded in jamming the transmitting channels (*i.e.*, both the sender and the receiver reside on in a timeslot) for all the past timeslots, is too powerful and thus infeasible in reality. However, it can be used to demonstrate the scheme performance in the worst case. In our simulation, we also compare the performance of our proposed approach with that of the receiver's *static* and *adaptive* optimal strategies. The *static opt* is a fixed strategy chosen to maximize the number of received packets (total gains) 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 it 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). Without loss of generality, we assume the sender and receiver have the same number of antennas with $k_s = k_r = 3$. In the simulation, we choose $\delta = 0.1$. After a feasible T is chosen (as discussed in the Section IV), we can determine the other inputs of the algorithm as follows $\beta = \sqrt{\frac{k_r}{nT}\ln\frac{n}{\delta}}$, $\gamma = 2\eta n$ and $\eta = \sqrt{\frac{\ln n}{4Tn}}$, where n is the total number of available channels. Note that a reasonable T should be chosen to ensure $\beta \in (0, 1]$ and $\gamma \in (0, 1/2]$.

We vary the strategies of the three parties to study i) the average number of received packets when T increases and ii) 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 (*i.e.*, without coding and with (rateless) 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 methods. The pur-



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 2. 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)

pose of the simulation is to compare the performance of our MAB-based algorithm with that of static receiving strategy and random hopping strategy at the receiver under different strategies of the sender and the jammer. Fig. 2 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 T^* under different strategy settings given $l = 20$, $k_j = 7$ and $n = 2k_j$. Since the MAB-based frequency hopping algorithm enables the receiver to *explore* the best channels for transmission, it will perform better than the static and random hopping in a "static" environment. As shown in Fig. 2 (a) and (b), when both sender and jammer use static strategies, static receiving strategy performs the best, and the random hopping strategy 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. 2 (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 much 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 the static strategy. Fig. 2 (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 strategy used at the sender side. We further consider the case when both the sender and the jammer use random strategies. Fig. 2 (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. 2 (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. 3 (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. 3 (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 (Rateless) Erasure Codes

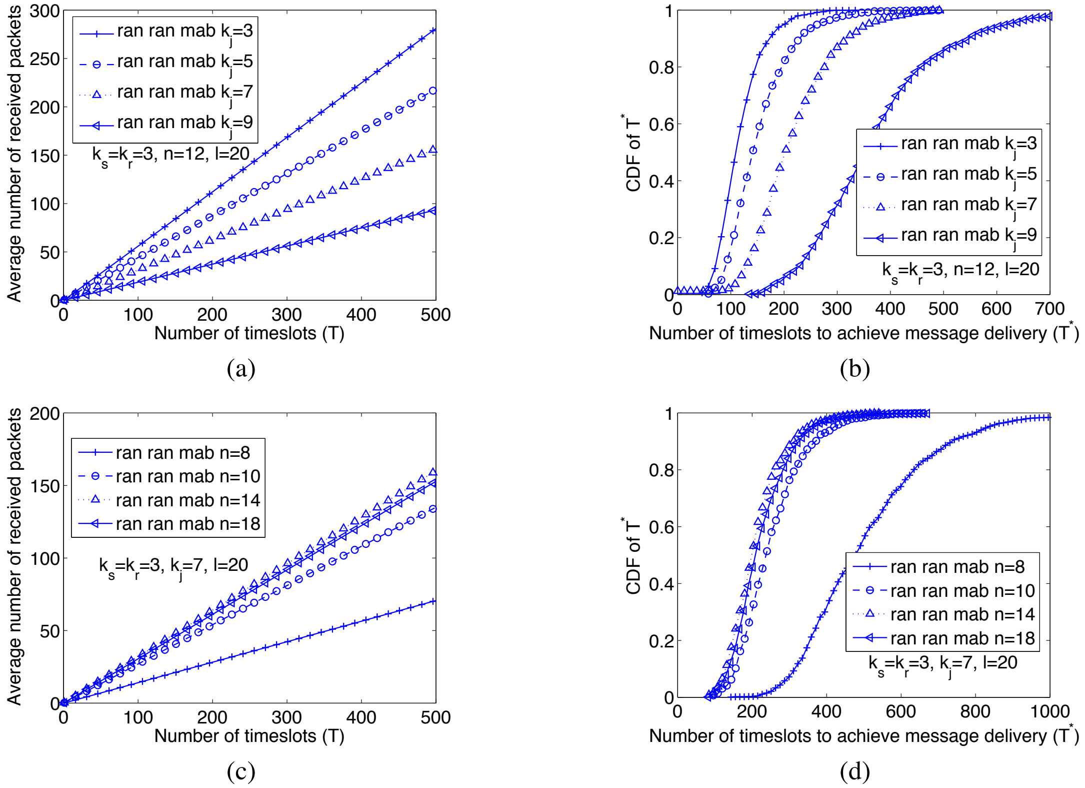
Compared with no coding case, by using erasure codes for message coding, the message M can be reconstructed if any l distinct packets are received. Since the size of the packet pool is enlarged, the probability of picking the same packet is reduced. This results in less time in collecting l distinct 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. 4 (a) plots the the CDF of time to reach message delivery when different number of encoded 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. In Fig. 4 (b), we show that by using rateless erasure codes, the time to achieve message delivery can be further reduced, *e.g.*, the probability of reaching message delivery is almost 1 when $T^* = 90$. In a practical system design, there is a trade-off between the time efficiency of message delivery and the encoding and decoding complexity.

C. Round-robin Packet Selection and Time Complexity

We also consider a different packet transmission strategy using round-robin. Different from the random strategy where a packet is picked from the packet pool of size l with probability $1/l$, in the round-robin method, the packets are picked one by one from 1 to l until the next round begins. Fig. 5 compares the performance of the system using two packet selection strategies: random selection and round-robin selection, both under the without coding case. Assume adaptive frequency hopping strategy is used at the receiver, it is shown that the round-robin method has better performance than random selection method when no static strategy is used at the sender and the jammer.

Next we compare the time complexity of Algorithm 1 and the enhanced Algorithm 2. Fig. 6 plots the time cost ratio r of running Algorithm 2 to Algorithm 1 as k_r increases. The results show that r decreases fast as k_r increases. This is due to the fact that when k_r increases, the time cost of Algorithm 1 with $O(n^{k_r})$ increases exponentially, and that of the enhanced Algorithm 2 with $O(nk_r)$ remains almost constant.

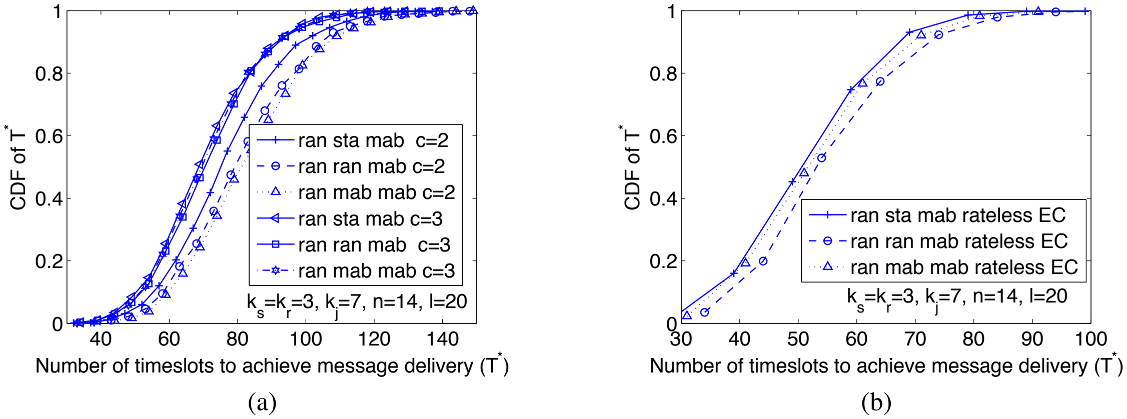
***Discussion*.** The proposed MAB-based anti-jamming scheme can work under both the worse-case random jammer and the non-worst-case jammers. Whatever the jamming strategies and the sending strategies are, the receiver's strategy will converge to the best one by using the MAB-based anti-jamming scheme. In practice, the key point is that the sender and the receiver do not know the attacker strategy in the first place when facing the jamming attack. Obviously, instead of



(a) (b)

(c) (d)

Fig. 3. 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)



(a) (b)

Fig. 4. CDF of expected time to reach message delivery (with erasure codes and rateless erasure codes)

hurriedly going to random hopping, learning first will help the receiver to get most out of the situation and there is nothing to lose. As indicated in the attack model, we do consider various types of jammers, *i.e.*, static, random and adaptive jammers. The proposed learning based scheme is a unified approach and can deal with all these types of jammers and maximize the benefit of the system.

VI. Related Work

A. 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 [7], [8], [9], [10], [6]. This problem has been recently identified by Strasser et al. [7]. 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. 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, [8], [9], [10] 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,

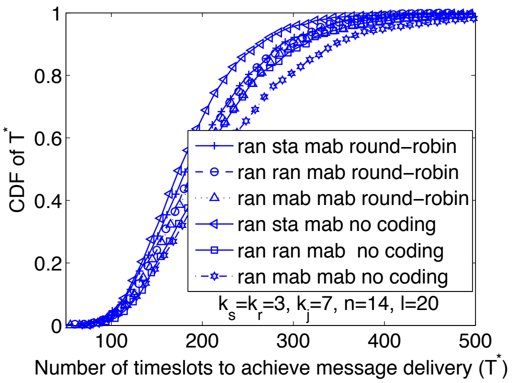


Fig. 5. CDF of expected time to reach message delivery (with round-robin).

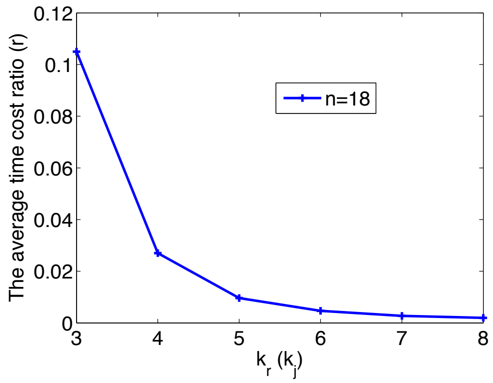


Fig. 6. The average time cost ratio r between Algorithm 2 and Algorithm 1.

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 was recently proposed in [6], which gave 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.

B. 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 [15]. 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 [16], 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], [18], [19], [20]. The decision makers has to choose a path in each round such that the weight of the chosen path is as small as possible. Because the number of possible pathes is exponentially large, the direct application of [16] 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 [18], [19] 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 [16] (*i.e.*, $O(T^{\frac{2}{3}})$[18] and $O(T^{\frac{3}{4}})$[19]). In [20], the authors considered 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 [20], where only the gains or losses of the chosen arms are revealed to the decision maker.

VII. Conclusion

In this paper, we formulated the UFH-based anti-jamming communication as a non-stochastic MAB problem and introduced the online optimization theory into the frequency hopping strategy design. We for the first time made the thorough quantitative performance characterization possible for UFH-based anti-jamming communications. Specifically, we formulated the UFH-based anti-jamming communication as a non-stochastic multi-armed bandit (MAB) problem and proposed an online learning-based UFH algorithm achieving asymptotic optimum. To reduce the time and space complexity, we further developed an enhanced algorithm exploiting the internal structure of strategy selection process. We analytically proved the optimality of the proposed algorithms under various message coding scenarios. An extensive simulation study was conducted to validate our theoretical analysis and show that the learning-based UFH algorithms are resilient against both *oblivious* and *adaptive* jamming attacks.

Appendix A
Proof of Theorem 2

Proof: We introduce some notations for performance analysis:

$$G_{i,T} = \sum_{t=1}^{T} g_{i,t} \quad \text{and} \quad G'_{i,T} = \sum_{t=1}^{T} g'_{i,t}$$

for all $1 \le i \le N$, where $G_{i,T}$ ($G'_{i,T}$) denotes the total gain (virtual gain, respectively) of strategy i in T timeslots, and

$$G_{f,T} = \sum_{t=1}^{T} g_{f,t} \quad \text{and} \quad G'_{f,T} = \sum_{t=1}^{T} g'_{f,t}$$

for all $1 \le f \le n$, where $G_{f,T}$ ($G'_{f,T}$) denotes the total gain (virtual gain, respectively) on channel f in T timeslots.

We prove the bound of regret by using the quantity $\ln \frac{W_T}{W_0}$ as following. First of all, we have the lower bound by definitions

$$\ln \frac{W_T}{W_0} = \ln \sum_{i=1}^{N} e^{\eta G'_{i,T}} - \ln N \geq \eta \max_{1 \leq i \leq N} G'_{i,T} - \ln N.$$

Then we derive the upper bound as follows:

$$\eta g'_{i,t} = \eta \sum_{f \in i} g'_{f,t} \leq \eta \sum_{f \in i} \frac{1+\beta}{q_{f,t}} \leq \frac{\eta k_r (1+\beta)|\mathcal{C}|}{\gamma} \leq 1,$$

where the second inequality follows because $q_{f,t} \geq \frac{\gamma}{|\mathcal{C}|}$ for all f by definition.

Using the fact that $e^x \leq 1 + x + x^2$ for all $x \leq 1$, for all $t = 1, 2, \cdots, T$, we derive the bound for $\ln \frac{W_t}{W_{t-1}}$ as follows

$$\begin{aligned}
\ln \sum_{i=1}^{N} \frac{w_{i,t-1}}{W_{t-1}} e^{\eta g'_{i,t}} &\leq \ln(\sum_{i=1}^{N} \frac{w_{i,t-1}}{W_{t-1}} (1 + \eta g'_{i,t} + \eta^2 {g'}^2_{i,t})) \\
&\leq \ln(1 + \sum_{i=1}^{N} \frac{p_{i,t}}{1-\gamma} (\eta g'_{i,t} + \eta^2 {g'}^2_{i,t})) \\
&\leq \frac{\eta}{1-\gamma} \sum_{i=1}^{N} p_{i,t} g'_{i,t} + \frac{\eta^2}{1-\gamma} \sum_{i=1}^{N} p_{i,t} {g'}^2_{i,t}.
\end{aligned}$$

The above inequalities hold using the fact that $\sum_{i=1}^{N} p_{i,t} \leq 1 - \gamma$ and inequality $\ln(1+x) \leq x$ for all $x > -1$.

Let $\mathcal{N}$ denote the strategy set $\{1, \ldots, N\}$. On the one hand, we have

$$\begin{aligned}
\sum_{i=1}^{N} p_{i,t} g'_{i,t} = \sum_{i=1}^{N} p_{i,t} \sum_{f \in i} g'_{f,t} &= \sum_{f=1}^{n} g'_{f,t} \sum_{i \in \mathcal{N}: f \in i} p_{i,t} \\
&= \sum_{f=1}^{n} g'_{f,t} q_{f,t} = g_{I_t,t} + n\beta.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\sum_{i=1}^{N} p_{i,t} {g'}^2_{i,t} = \sum_{i=1}^{N} p_{i,t} (\sum_{f \in i} g'_{f,t})^2 &\leq \sum_{i=1}^{N} p_{i,t} k_r \sum_{f \in i} {g'}^2_{f,t} \\
&= k_r \sum_{f=1}^{n} {g'}^2_{f,t} \sum_{i \in \mathcal{N}: f \in i} p_{i,t} \\
&= k_r \sum_{f=1}^{n} {g'}^2_{f,t} q_{f,t} \\
&\leq k_r (1+\beta) \sum_{f=1}^{n} g'_{f,t},
\end{aligned}$$

which holds the fact that $g'_{f,t} \leq \frac{1+\beta}{q_{f,t}}$. Therefore,

$$\ln \frac{W_t}{W_{t-1}} \leq \frac{\eta}{1-\gamma} (g_{I_t,t} + n\beta) + \frac{\eta^2 k_r (1+\beta)}{1-\gamma} \sum_{f=1}^{n} g'_{f,t}.$$

Summing for $t = 1, \cdots, T$, we have the following inequality

$$\begin{aligned}
\ln \frac{W_T}{W_0} &\leq \frac{\eta}{1-\gamma} (\widehat{G}_T + n\beta T) + \frac{\eta^2 k_r (1+\beta)}{1-\gamma} \sum_{f=1}^{n} G'_{f,T} \\
&\leq \frac{\eta}{1-\gamma} (\widehat{G}_T + n\beta T) + \frac{\eta^2 k_r (1+\beta)}{1-\gamma} |\mathcal{C}| \max_{1 \leq i \leq N} G'_{i,T}.
\end{aligned}$$

Note that $\widehat{G}_T$ is the expected total gain of our algorithm in T time slots. Combining the upper bound with the lower bound, we have

$$\begin{aligned}
\widehat{G}_T \geq (1 - \gamma - \eta k_r (1+\beta)|\mathcal{C}|) &\max_{1 \leq i \leq N} G'_{i,T} \\
&- \frac{1-\gamma}{\eta} \ln N - n\beta T.
\end{aligned}$$

For any fixed f, $u > 0$ and $c > 0$, by the Chernoff bound, we have $\mathbb{P}[G_{f,T} > G'_{f,T} + u] \leq e^{-cu} \mathbb{E}[e^{c(G_{f,T} - G'_{f,T})}]$. Let $u = \ln \frac{n}{\delta} / \beta$ and $c = \beta$, we get $e^{-cu} \mathbb{E}[e^{c(G_{f,T} - G'_{f,T})}] = \frac{\delta}{n} \mathbb{E}[e^{\beta(G_{f,T} - G'_{f,T})}]$. So it suffices to prove that $e^{\beta(G_{f,T} - G'_{f,T})} \leq 1$ for all T. Let $Z_t = e^{\beta(G_{f,t} - G'_{f,t})}$. By showing that $\mathbb{E}[Z_t] \leq Z_{t-1}$ for all $t \geq 2$ and $\mathbb{E}[Z_1] \leq 1$, it suffices to prove that for any $\delta \in (0,1)$, $0 \leq \beta < 1$ and $1 \leq f \leq n$,

$$\mathbb{P}[G_{f,T} > G'_{f,T} + \frac{1}{\beta} \ln \frac{n}{\delta}] \leq \frac{\delta}{n}.$$

Applying the above bound, we can have that, with probability at least $1 - \delta$,

$$\begin{aligned}
\widehat{G}_T \geq (1 - \gamma - \eta k_r (1+\beta)|\mathcal{C}|) &(\max_{1 \leq i \leq N} G_{i,T} - \frac{k_r}{\beta} \ln \frac{n}{\delta}) \\
&- \frac{1-\gamma}{\eta} \ln N - n\beta T.
\end{aligned}$$

Here, we use the fact $1 - \gamma - \eta k_r (1+\beta)|\mathcal{C}| > 0$ which follows the assumptions of the theorem.

By doing some transpositions and using the following fact $\max_{1 \leq i \leq N} G_{i,T} \leq T k_r$, we have

$$\begin{aligned}
&\max_{1 \leq i \leq N} G_{i,T} - \widehat{G}_T \leq (\gamma + \eta(1+\beta) k_r |\mathcal{C}|) T k_r + \\
&(1 - \gamma - \eta(1+\beta) k_r |\mathcal{C}|) \frac{k_r}{\beta} \ln \frac{n}{\delta} + \frac{1-\gamma}{\eta} \ln N + n\beta T
\end{aligned}$$

with probability at least $1 - \delta$. Let $K = \min\{k_s, n - k_j, k_r\}$. Since $\hat{G_T} = KT - \hat{L_T}$ and $\max_{1 \leq i \leq N} G_{i,T} = KT - \min_{1 \leq i \leq N} L_{i,T}$, we have

$$\begin{aligned}
\hat{L_T} \leq KT(&\gamma + \eta(1+\beta) k_r |\mathcal{C}|) + \\
&(1 - \gamma - \eta(1+\beta) k_r |\mathcal{C}|) \min_{1 \leq i \leq N} L_{i,T} + \\
(1 - \gamma &- \eta(1+\beta) k_r |\mathcal{C}|) \frac{k_r}{\beta} \ln \frac{n}{\delta} + \frac{1-\gamma}{\eta} \ln N + n\beta T
\end{aligned}$$

with probability $1 - \delta$. Simplify above inequality, we can get

$$\begin{aligned}
\hat{L_T} - \min_{1 \leq i \leq N} L_{i,T} \leq k_r T \gamma + 2\eta T k_r n + \frac{k_r}{\beta} \ln \frac{n}{\delta} + \frac{1-\gamma}{\eta} k_r \ln n \\
+ n\beta T
\end{aligned}$$

with probability $1 - \delta$.

Setting $\beta = \sqrt{\frac{k_r}{nT} \ln \frac{n}{\delta}}$ and $\gamma = 2\eta k_r |\mathcal{C}|$, we can get

$$\max_{1 \leq i \leq N} G_{i,T} - \widehat{G}_T \leq 4\eta T {k_r}^2 |\mathcal{C}| + \frac{\ln N}{\eta} + 2\sqrt{k_r n T \ln \frac{n}{\delta}},$$

which holds with probability $1 - \delta$ if $T \geq \frac{k_r}{n} \ln(\frac{n}{\delta})$. Finally, using the facts $|\mathcal{C}| = \lceil \frac{n}{k_r} \rceil$ and $N \leq n^{k_r}$. and setting $\eta = \sqrt{\frac{\ln N}{4k_r^2 T |\mathcal{C}|}}$, we prove that

$$\max_{1 \leq i \leq N} G_{i,T} - \widehat{G}_T \leq 6 k_r \sqrt{T n \ln n}$$

with probability $1 - \delta$. ■

References

[1] Q. Wang, P. Xu, K. Ren, and X.-Y. Li, “Delay-bounded adaptive ufh-based anti-jamming wireless communication,” in *Proc. IEEE INFOCOM’11*, 2011, pp. 1413–1421.

[2] L. Lazos, S. Liu, and M. Krunz, “Mitigating control-channel jamming attacks in multi-channel ad hoc networks,” in *Proc. ACM WISEC’09*, 2009, pp. 169–180.

[3] D. Slater, P. Tague, R. Poovendran, and B. J. Matt, “A coding-theoretic approach for efficient message verification over insecure channels,” in *Proc. ACM WISEC’09*, 2009, pp. 151–160.

[4] W. Xu, W. Trappe, Y. Zhang, and T. Wood, “The feasibility of launching and detecting jamming attacks in wireless networks,” in *Proc. ACM MobiHoc’05*, 2005, pp. 46–57.

[5] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication.* Addison Wesley, 1995.

[6] M. Strasser, C. Pöpper, and S. Capkun, “Efficient uncoordinated fhss anti-jamming communication,” in *Proc. ACM MobiHoc’09*, 2009, pp. 207–218.

[7] M. Strasser, C. Pöpper, S. Capkun, and M. Cagalj, “Jamming-resistant key establishment using uncoordinated frequency hopping,” in *Proc. IEEE S&P’08*, 2008, pp. 64–78.

[8] C. Pöpper, M. Strasser, and S. Capkun, “Jamming-resistant broadcast communication without shared keys,” in *Proc. USENIX’09 Security Symposium*, 2009, pp. 231–248.

[9] T. Jin, G. Noubir, and B. Thapa, “Zero pre-shared secret key establishment in the presence of jammers,” in *Proc. MobiHoc’09*, 2009, pp. 219–228.

[10] Y. Liu, P. Ning, H. Dai, and A. Liu, “Randomized differential dsss: Jamming-resistant wireless broadcast communication,” in *Proc. IEEE INFOCOM’10*, 2010, pp. 695–703.

[11] W. Arbaugh, “Improving the latency of the probe phase during 802.11 handoff,” online at www.umiacs.umd.edu/partnerships/ lts-docs/Arbaug_talk2.pdf.

[12] T. Shu and M. Krunz, “Throughput-efficient sequential channel sensing and probing in cognitive radio networks under sensing errors,” in *Proc. MobiCom’09*, 2009, pp. 37–48.

[13] J. C. Gittins, “Bandit processes and dynamic allocation indices,” *Journal of the Royal Statistical Society Series B Methodological*, vol. 41, pp. 148–177, 1979.

[14] P. Whittle, “Restless bandits: activity allocation in a changing world,” *Journal of Applied Probability*, vol. 25A, pp. 287–298, 1988.

[15] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “Gambling in a rigged casino: The adversarial multi-arm bandit problem,” in *Proc. IEEE FOCS’95*, 1995, pp. 322–331.

[16] ——, “The nonstochastic multiarmed bandit problem,” *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.

[17] A. Kalai and S. Vempala, “Efficient algorithms for online decision problems,” in *Proc. COLT’03*, 2003, pp. 26–40.

[18] B. Awerbuch and R. D. Kleinberg, “Adaptive routing with end-to-end feedback: distributed learning and geometric approaches,” in *Proc. ACM STOC’04*, 2004, pp. 45–53.

[19] H. B. McMahan and A. Blum, “Online geometric optimization in the bandit setting against an adaptive adversary,” in *Proc. COLT’04*, 2004, pp. 109–123.

[20] 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.

[21] 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.

[22] M. Strasser, B. Danev, and S. Čapkun, “Detection of reactive jamming in sensor networks,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 7, pp. 16:1–16:29, September 2010.

[23] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” in *Proc. ASIACRYPT’01*. Springer-Verlag, pp. 514–532.

[24] X.-Y. Li, Y. Wang, and W. Feng, “Multiple round random ball placement: Power of second chance,” in *Proc. COCOON ’09*, 2009, pp. 439–448.

[25] S. G. Wilson, “Digital modulation and coding,” *Prentice-Hall*, 1996.

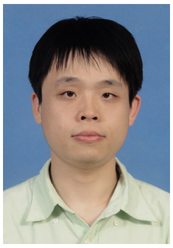
[26] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *Proc. ACM STOC’97*, 1997, pp. 150–159.

[27] P. Maymounkov, “Online codes,” *Technical Report TR2002-833, New York University*, 2002.

[28] M. Luby, “Lt codes,” in *Proc. IEEE FOCS’02*, 2002, pp. 271–280.

[29] A. Shokrollahi, “Raptor codes,” *IEEE/ACM Trans. Netw.*, vol. 14, no. SI, pp. 2551–2567, 2006.

Qian Wang received the B.S. degree from Wuhan University, China, in 2003 and the M.S. degree from Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, China, in 2006, both in Electrical Engineering. He is currently working towards the Ph.D. degree in the Electrical and Computer Engineering Department at Illinois Institute of Technology. His research interests includewireless network security and privacy, cloud computing security, and applied cryptography. He is a co-recipient of the Best Paper Award from IEEE ICNP 2011.



Ping Xu has been a PhD student of Computer Science Department at the Illinois Institute of Technology since 2006. He received BS and MS degrees in Electronic Engineering from Shanghai Jiaotong University, China, in 1999 and 2003 respectively. His current research interests include algorithm design and analysis for wireless ad hoc networks, wireless sensor networks, online algorithms, and algorithmic game theory.

Kui Ren is an assistant professor in the Electrical and Computer Engineering Department at Illinois Institute of Technology. He obtained his Ph.D. degree in electrical and computer engineering from Worcester Polytechnic Institute in 2007. His research interests include security and privacy in cloud computing, wireless security, smart grid security, and sensor network security. His research is supported by NSF, DoE, AFRL, and Amazon. He is a recipient of NSF Faculty Early Career Development (CAREER) Award in 2011. He is a co-recipient of the Best Paper Award from IEEE ICNP 2011. He is a Senior Member of IEEE and a Member of ACM.

Xiang-Yang Li has been an Associate Professor (since 2006) and Assistant Professor (from 2000 to 2006) of Computer Science at the Illinois Institute of Technology. He is recipient of China NSF Outstanding Overseas Young Researcher (B). Dr. Li received M.S. (2000) and Ph.D. (2001) degree at Department of Computer Science from University of Illinois at Urbana-Champaign. He received a Bachelor degree at Department of Computer Science and a Bachelor degree at Department of Business Management from Tsinghua University, P.R. China, both in 1995. He published a monograph “Wireless Ad Hoc and Sensor Networks: Theory and Applications”. He also co-edited the book “Encyclopedia of Algorithms”. The research of Dr. Li has been supported by USA NSF, HongKong RGC, and China NSF. His research interests span the wireless sensor networks, game theory, computational geometry, and cryptography and network security. Dr. Li is an editor of several journals, including IEEE Transaction on Parallel and Distributed Systems (2009 to present). He is a senior member of the IEEE.