Speed Dating Despite Jammers

Dominic Meier[1], Yvonne Anne Pignolet[1], Stefan Schmid[2], and Roger Wattenhofer[1]

[1] Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland
meierdo@ethz.ch, pignolet@tik.ee.ethz.ch, wattenhofer@tik.ee.ethz.ch
[2] Chair for Efficient Algorithms, Technical University of Munich, Germany
schmiste@in.tum.de

Abstract. Many wireless standards and protocols today, such as WLAN and Bluetooth, operate on similar frequency bands. While this permits an efficient usage of the limited medium capacity, transmissions of nodes running different protocols can interfere. This paper studies how to design node discovery algorithms for wireless multichannel networks which are robust against contending protocols on the shared medium. We pursue a conservative approach and consider a Byzantine adversary who prevents the communication of our protocol on t channels in a worst-case fashion. Our model also captures disruptions controlled by an adversarial *jammer*. This paper presents algorithms for scenarios where t is not known. The analytical findings are complemented by simulations providing evidence that the proposed protocols perform well in practice.

1 Introduction

Wireless networks are ubiquitous and have become indispensable for many tasks of our daily lives. Due to the limited range of frequencies available for communication between wireless nodes such as laptops, PDAs or mobile phones, many wireless standards and protocols today operate on the same frequency bands, e.g., the ISM bands. One well-known and widely discussed example is WLAN and Bluetooth (i.e., *IEEE 802.15.2*), but there are many others. Such contending access of different protocols to the shared medium leads to collisions. While ultra wide band technology may mitigate this problem and reduce interference, it is not always available or desirable.

This raises the question of how to devise protocols which are robust against transmissions of other protocols *by design*. In this paper, we seek to shed light onto this question. We adopt a conservative approach and assume that a *Byzantine adversary* can disturb our algorithms in an arbitrary manner. This model comprises scenarios where an adversarial *jammer* seeks to slow down communication or even to stop it completely. Such jamming attacks are a particularly cumbersome problem today: typically, a jamming attack does not require any special hardware and is hence simple and cheap.

This paper focuses on networks without a fixed infrastructure, such as MANETs or sensor networks, which are organized in an ad hoc manner. A fundamental operation in dynamic ad hoc networks is the search of potential communication partners. In some sense, this operation is more difficult than other communication tasks, as the nodes do not have any information about each other *a priori*. Besides the lack of information on either hardware or medium access addresses of other nodes, concurrent transmissions—either of other nodes running the same protocol or other radio transmissions—lead to

B. Krishnamachari et al. (Eds.): DCOSS 2009, LNCS 5516, pp. 1–14, 2009.

collisions and interference. In addition, by injecting a high level of noise, a jammer can slow down wireless communication significantly. Once two nodes have met, they may agree on certain communication or channel-hopping patterns (e.g., based on their medium access addresses) facilitating efficient interactions in the future. Thus, it is of utmost importance to solve this task as fast as possible.

A well-known existing protocol dealing with this problem is Bluetooth. It specifies an asymmetric way to connect and exchange information between devices such as mobile phones, cameras, GPS receivers or video game consoles. As a consequence, Bluetooth can be used to synchronize two devices as soon as they are within each other's transmission range, or to display the availability of a printer. Clearly, the device discovery time is highly relevant in these situations.

We study the problem of discovering communication partners in *multi-channel* networks, despite the presence of a Byzantine adversary. Concretely, we assume that the adversary corrupts t out of m available channels. We say that two nodes have successfully discovered each other if and only if two nodes are on the same channel, one transmitting, one receiving, there is no other node transmitting on this channel, and the channel is not jammed. In reality, nodes typically do not know *whether*, and *how many*, channels are corrupted. The goal of this paper is to devise algorithms solving the discovery problem efficiently *without knowledge of* t. We require that nodes are discovered very fast if t is small, and that the performance of the discovery algorithm *degrades gracefully* with increasing t. In other words, we want algorithms (oblivious to t) being *competitive* to a discovery algorithm knowing t.

Our main contribution are fast discovery algorithms performing well without knowledge of t and despite Byzantine disruptions. In particular, we describe a randomized algorithm which, in expectation, is at most a factor of $O(\log^2 m)$ slower than the best algorithm *knowing* t, for any t. We prove this to be optimal in the sense that this is the best ratio an algorithms that can be described by a probability distribution over the available channels can achieve. In addition, we study a scenario where the jammer chooses t according to a *probability density function* (PDF) which is known to the device discovery algorithm. Furthermore, our paper discusses how to extend our results to a multiplayer setting. In order to complement our formal analysis, we investigate the performance of our algorithms by in silico experiments.

2 Related Work

With the increasing popularity of wireless networks such as WLANs or sensor networks, security aspects and quality of service become more relevant. An important reason for disruptions are transmissions of other devices using different protocols. One widely studied example is WLAN and Bluetooth. In this case, several possible solutions have been discussed by the IEEE task force 802.15.2 [12], e.g., a non-cooperative coexistence mechanism based on adaptive frequency hopping. The model we study is quite general and comprises many types of disruptions, such as interference [21] or jamming attacks. Resilience to jamming is crucial as jamming attacks can often be performed at low costs as there is no need for special hardware [4]. For these reasons, the jamming problem in wireless networks is intensively discussed both in practice and in theory ([8,16,18,22,23]). While some researchers focus on how such attacks can be performed [24], others concentrate on countermeasures [1]. In [17], it has been shown that using methods based on signal strength and carrier sensing, detecting sophisticated jammers is difficult. Moreover, a method based on packet delivery ratios cannot decide unambiguously whether link problems are due to mobility, congestion or jamming.

The threat of jamming attacks can be mitigated by appropriate physical layer technologies. E.g., spread spectrum techniques can be used, rendering it more difficult to detect the start of a packet fast enough in order to jam it. Unfortunately, one of the most widely deployed protocols, 802.11, has only small spreading factors [4]. In fact, it has recently been shown that the MAC protocol of 802.11 can be attacked by simple and oblivious jammers [5]. Many research projects deal with jammers on the MAC layer. For instance in [7], a coding scheme for fast frequency hopping is presented. If the adversary does not know the hopping sequence it can disturb only a subset of transmissions due to energy constraints. Alternative solutions include channel surfing and spatial retreat [24], or mechanisms to hide messages [22].

The jamming problem also raises interesting algorithmic questions. Gilbert et al. [11] investigate the *efficiency* of an adversary. They find that even the uncertainty introduced by the possibility of adversarial broadcasts is sufficient to slow down many protocols. In [13] a model where the adversary has a *limited energy budget* is considered; the paper studies how to achieve global broadcasts if the adversary is allowed to spoof addresses. In [19], fault-tolerant broadcasting under *probabilistic* failures is studied. [9] presents tight bounds for the running time of the *ϵ-gossip problem* on multi-channel networks. In [10], Dolev et al. describe a randomized protocol that allows nodes to exchange authenticated messages despite a malicious adversary that can cause collisions and spoof messages. Awerbuch et al. [4] present a MAC protocol for single-hop networks that is provably robust to adaptive adversarial jamming. The jammer can block a $(1-\epsilon)$-fraction of the time steps, but it has to make decisions before knowing the actions of the nodes for this step. Several algorithms are presented which, e.g., allow to elect a leader in an energy efficient manner.

In contrast to the work discussed above, we focus on the *bootstrap problem* where a node has to find other nodes in its range. This device discovery problem has been intensively studied in literature. In [6], randomized backoff protocols are proposed for a single broadcast channel. However, their algorithms are not directly applicable in wireless networks where unlike in traditional broadcast systems such as the Ethernet, collisions may not be detectable. In [15], probabilistic protocols for Bluetooth node discovery are investigated, where the nodes seek to establish one-to-one connections. In [2] and [3], protocols for single and multi channel ad hoc networks are described. However, none of these papers attend to (adversarial) disruptions.

3 Model

Suppose we are given a shared medium consisting of m channels $c_1, ..., c_m$. There may be an adversary with access to the medium. We adopt a worst-case perspective assuming that an adversary always blocks those $t < m$ channels which minimize the discovery time of a given algorithm. We aim at devising discovery protocols that are efficient despite these circumstances. Typically, the number of jammed channels t is not known to the discovery algorithm. Consequently, our main objective is to devise algorithms which are optimal with respect to *all* t. In other words, an algorithm ALG should solve the node discovery problem efficiently if t is small, and "*degrade gracefully*" for larger t. For the analysis of the algorithms we investigate a *slotted* model where time is divided into synchronized time slots. However, note that all our results hold up to a factor of two in unslotted scenarios as well, due to the standard trick introduced in [20] for the study of slotted vs. unslotted ALOHA. In each time slot, every node can choose one channel and decide whether it wants to listen on this channel or to transmit some information (e.g., its ID or a seed for its hopping pattern sequence) on the channel. We say that two nodes v_1 and v_2 have *discovered* each other successfully if and only if the three following conditions are met:

1. v_1 and v_2 are on the same channel c
2. v_1 is in listening mode and v_2 transmits its contact information on c, or vice versa
3. channel c is not jammed

Since nodes cannot know, whether there are other nodes in their transmission area, we count the number of time slots until a successful discovery from the point in time when all of them are around (*discovery time*). In this paper, we mainly constrain ourselves to the *two node case*.

The node discovery problem turns out to be difficult to solve if we restrict ourselves to deterministic algorithms. In a scenario where all nodes are identical and do not have anything (e.g., IDs) to break the symmetry, two problems arise even in the absence of a jammer: (1) if two nodes follow a deterministic hopping pattern, they may never be on the same channel in the same slot; (2) even if the nodes meet, choosing deterministically whether to send or listen for announcements in this slot may always yield situations where both nodes send or both nodes listen. One way to break the symmetry is to allow nodes to generate random bits. Alternatively, one may assume that the two nodes which want to discover each other already share a certain number of bits which are unknown to the jammer. Due to these problems, we focus on *randomized* algorithms. We assume that every node runs the same algorithm, only decisions based on random experiments differ. We investigate the class of randomized algorithms that can be described by a probability distribution over the channels, i.e., in each round, a channel is selected for communication according to a probability distribution. We strive to find algorithms that perform well for every possible number of jammed channels. To this end, we define a measure that captures the loss of discovery speed due to the lack of knowledge of the number of channels the adversary decides to jam.

Definition 1 (Competitiveness). *In a setting with* t *jammed channels, let* T^t_{REF} *be the expected discovery time until two nodes discover each other for an optimal randomized algorithm* REF *which has complete knowledge of* t*. Let* T^t_{ALG} *be the corresponding expected discovery time of a given algorithm* ALG*. We define the*

$$\text{competitive ratio } \rho := \max_{0\leq t\leq m-1} \frac{T^t_{ALG}}{T^t_{REF}}.$$

The smaller the achieved competitive ratio ρ, the more efficient the discovery algorithm.

4 Algorithms for Device Discovery

To initiate our analysis, we first consider device discovery algorithms for the case where the total number of jammed channels t is known. Subsequently, our main results are presented together with several optimal algorithms.

4.1 Known t

In our model, a node has to select a channel c and decide whether to send or listen on c in each round. Let us determine the best strategy if t is known. As we will compare our algorithms which do not know t to this strategy, we will call this reference point algorithm REF. For two nodes which have never communicated before, it is best to send or listen with probability 0.5. The following lemma derives the optimal distribution over the channels.

Lemma 1. *Let m denote the total number of channels and assume t, the number of jammed channels, is known. If $t = 0$ the best strategy is to use one designated channel for discovery. If $0 < t \leq m/2$ then the expected discovery time is minimized for an algorithm REF choosing one of the first $2t$ channels uniformly at random. In all other cases, the best strategy for REF is to chose each channel with probability $1/m$. Thus, REF has a expected discovery time of*

$$\begin{array}{rrl} 2 & \textit{if } t = & 0, \\ 8t & \textit{if } t \leq & m/2, \\ 2m^2/(m-t) & \textit{if } t > & m/2. \end{array}$$

Proof. Let p_i denote REF's probability of choosing channel c_i. Without loss of generality, assume that the channels are ordered with decreasing probability of REF, i.e., $1 \geq p_1 \geq p_2 \geq ... \geq p_m \geq 0$. Let λ be the smallest i for which $p_i = 0$, in other words, REF uses λ channels for discovery. Clearly, if $\lambda < t+1$, the expected discovery time is infinite, and hence, we can concentrate on algorithms for which $\lambda \geq t+1$.

According to our worst-case model, the jammer blocks the channels $c_1, \ldots, c_t$. It holds that $\sum_{i=t+1}^{\lambda} p_i \leq \frac{\lambda - t}{\lambda}$, where $(\lambda - t)/\lambda$ is equal to the sum of the channel probabilities of the channels $c_{t+1}, \ldots, c_\lambda$ when the probability distribution over the first λ channels is uniform. That is, by cutting some probability from those channels with probability greater than $1/\lambda$ and distribute it over the other channels, the total probability of success will increase. Therefore, the expected discovery time is minimized for uniform probability distributions. As soon as $\sum_{i=t+1}^{\lambda} p_i = \frac{\lambda - t}{\lambda}$, we cannot further redistribute the probabilities without decreasing the overall probability of success since the jammer always blocks the t most probable channels.

It remains to show that $\lambda = \min(2t, m)$ maximizes the probability of success. As the first t channels are jammed and the probability to be chosen is $p_i = 1/\lambda$ for each channel, the probability for a successful meeting is

$$\mathbb{P}[\text{success}|t] = \tfrac{1}{2} \sum_{i=t+1}^{\lambda} \tfrac{1}{\lambda^2} = \tfrac{\lambda - t}{2\lambda^2}.$$

This probability is maximized for $\lambda = 2t$. If fewer channels are available, i.e., $2t > m$, the best decision is to pick any of the m channels with probability $1/m$.

Since the execution in one time slot is independent from the execution in all other time slots, the expected discovery time is then given by the inverse of the success probability.

4.2 Uniform Algorithm

The simplest randomized algorithm chooses one of the available m channels uniformly at random in each round. The expected discovery time of this algorithm UNI is $2m^2 /(m-t)$. Hence the competitiveness of UNI is $\rho_{UNI} = m$, reached when $t = 0$. In other words, if there are no blocked channels, the performance of this algorithm is poor.

4.3 Class Algorithms

Since we aim at being competitive to REF for any number of jammed channels t, we examine more sophisticated algorithms. Observe that for small t, selecting a channel out of a small subset of channels is beneficial, since this increases the probability that another node is using the same channel. On the other hand, for large t, using few channels is harmful, as most of them are jammed. One intuitive way to tackle the device discovery problem is to use a small number of estimators for t. In each round, we choose one of the estimators according to a probability distribution and then apply the optimal algorithm for this “known” $\hat{t}$, namely algorithm REF. In the following, we will refer to the set of channels for such a $\hat{t}$, i.e., channels $c_1, ..., c_{2\hat{t}}$, as a *class* of channels. Note that any such algorithm has to include the class $\hat{t} = m/2$, otherwise the expected discovery time is infinity. We investigate the optimal number of classes for the family of algorithms selecting the estimator for the next round uniformly at random among k guess classes $\widehat{t_1} \leq ... \leq \widehat{t_i} \leq ... \leq \widehat{t_k}$ for t for $k \leq m/2$. The algorithm chooses each such class i with a uniform probability, and subsequently selects a channel to transmit uniformly at random from a given set of $2\widehat{t_i}$ channels. We concentrate on algorithms ALG_k where the guesses grow by constant factors, i.e., whose estimations for $\widehat{t}$ are of the following magnitudes: $\widehat{t} = m^{1/k}, ..., m^{i/k}, ..., m$. We begin by deriving a bound on the expected discovery time of ALG_k.

Theorem 1. *Let m denote the number of channels and let $t < m$ be the number of jammed channels. Let $\beta_1 = \left\lfloor \frac{k \cdot \ln(t)}{\ln m} \right\rfloor$, $\beta_2 = m^{-\frac{\beta_1}{k}}$ and $\beta_3 = 2\beta_1 - 2k - 1$ for some integer value $k \leq m/2$. The expected discovery time of ALG_k is*

$$\frac{2k^2 m(m^{1/k}-1)^2}{m^{\frac{1}{k}}\beta_3 - \frac{t}{m} + \beta_3 + \beta_2\left(m^{\frac{k+1}{k}} + 2t + m - t\beta_2 m\right)}.$$

Proof. Consider a time slot where node v_1 chooses class i_1 and assume node v_2 chooses class i_2 in the same round. If a discovery is possible in this time slot, we have $i_1 \geq i_2 > \frac{k \cdot \ln(t)}{\ln m}$. The second inequality is due to our requirement that $m^{i_2/k} > t$; otherwise the devices cannot find each other since the estimator $\hat{t}$ of at least one device smaller than $t/2$. The probability that the two nodes successfully meet in this round is

$$p(i_1, i_2, t) = \frac{m^{i_2/k} - t}{m^{i_1/k} m^{i_2/k}} = \frac{1}{m^{i_1/k}} - \frac{t}{m^{(i_1+i_2)/k}}.$$

The overall success probability in some round is given by

$$\mathbb{P}[\text{success}|t] = \frac{1}{2k^2}\left(\sum_{i_1=\beta_1+1}^{k}\left(\sum_{i_2=\beta_1+1}^{i_1-1} p(i_1,i_2,t) + \sum_{i_2=i_1}^{k} p(i_2,i_1,t)\right)\right).$$

Expanding the sums leads to $\mathbb{P}[success|t] := (m^{\frac{1}{k}}(2\beta_1 - 2k - 1) - \frac{t}{m} + 2k - 2\beta_1 - 1 + m^{-\frac{\beta_1}{k}}(m^{\frac{k+1}{k}} + 2t + m - tm^{1-\frac{\beta_1}{k}}))/(2k^2 m(m^{1/k} - 1)^2)$, of which the expected discovery time can be derived.

Since we are particularly interested in an algorithm's competitiveness, we can examine the ratio achieved by this algorithm for $k = \lfloor \log m \rfloor$. We give a brief sketch of the derivation. We distinguish three cases for t. If $t = 0$, the ratio is $\Theta(\log m)$, verifiable by calculating $2/\mathbf{P}[success|t = 0]$. For $t \in [1, \ldots, m/2]$, the expected running time is $O(\log^2 m \cdot mt/(m - \log m))$, hence the ratio is $O(\log^2 m)$. It remains to consider $t > m/2$. The expected discovery time of $ALG_{\log m}$ is $2\log^2 mm^2/(m-t)$, compared to REF needing $2m^2/(m-t)$ time slots in expectation. Thus the ratio is at most $O(\log^2 m)$ for $ALG_{\log m}$. Interestingly, as we will see later, this is asymptotically optimal even for general randomized algorithms.

Corollary 1. *$ALG_{\log m}$ has a competitive ratio of at most $O(\log^2 m)$.*

4.4 Optimal Competitiveness

We have studied how to combine algorithms tailored for a certain estimated t in order to construct efficient node discovery protocols. In particular, we have derived the execution time for a general class of algorithms ALG_k. This raises two questions: What is the best competitive ratio achieved by ALG_k with the best choice of k? How much do we lose compared to *any* algorithm solving the device discovery problem by focusing on such class estimation algorithms ALG_k only?

In the following, we adopt a more direct approach, and construct an optimal algorithm using a probability distribution $\overrightarrow{p} = (p_1, \ldots, p_m)$, i.e., choosing a channel i with probability p_i, where $p_1 \geq p_2 \geq \ldots \geq p_m \geq 0$. In other words we have to find $\overrightarrow{p}$ yielding the lowest possible competitiveness. From this analysis, we can conclude that no loss incurs when using on class algorithms ALG_k , i.e., there is a class algorithm, $ALG_{\log n}$ with an asymptotically optimal competitive ratio.

Recall that the best possible expected discovery time (cf. Lemma 1) if t channels are jammed and if t is known. Thus, in order to devise an optimal algorithm OPT, we need to solve the following optimization problem.

$$\min \rho = \min_{\overrightarrow{p}} \max_{0 \leq t < m} \frac{T^t_{ALG}}{T^t_{REF}},$$

$$\text{where } \frac{T^t_{ALG}}{T^t_{REF}} = \begin{cases} \frac{1}{\sum_{i=1}^{m} p_i^2} & \text{if } t = 0 \\ \frac{1}{4t\sum_{i=t+1}^{m} p_i^2} & \text{if } t \leq m/2 \\ \frac{m-t}{m^2\sum_{i=t+1}^{m} p_i^2} & \text{if } t > m/2. \end{cases}$$

In addition, it must hold that $\sum_{i=1}^{m} p_i = 1,$ and $p_1 \geq p_2 \geq \ldots \geq p_m \geq 0.$

We simplify the $\min \max \rho$ objective function to $\min \rho$ by generating the following optimization system.

$$\begin{aligned}
&\min \rho \ \text{ such that} && \\
&\quad t=0: && \tfrac{1}{\sum_{i=1}^{m} p_i^2} \le \rho && (1)\\
&\quad 1 \le t \le m/2: && \tfrac{1}{4t\sum_{i=t+1}^{m} p_i^2} \le \rho && (2)\\
&\quad t > m/2: && \tfrac{m-t}{m^2\sum_{i=t+1}^{m} p_i^2} \le \rho && (3)\\
&\text{and } \sum_{i=1}^{m} p_i = 1,\, p_1 \ge p_2 \ge \ldots \ge p_m \ge 0. &&
\end{aligned}$$

Observe that ρ is minimal if equality holds for all inequations in $(1), (2)$ and (3). This yields an equation system allowing us to compute the values p_i. Thus the optimal channel selection probabilities are

$$p_1 = \sqrt{\frac{7}{8\rho}}, \quad p_i = \sqrt{\tfrac{1}{4i(i-1)\rho}} \ \text{ for i} \in [2, m/2],$$
$$\text{and} \quad p_j = \tfrac{1}{m} \cdot \sqrt{\tfrac{1}{\rho}} \quad \text{for } j > m/2.$$

Due to the constraint $\sum_{i=1}^{m} p_i = 1$, the competitiveness is

$$\rho = \tfrac{1}{4}\left(1 + \sqrt{7/2} + \textstyle\sum_{i=2}^{m/2} 1/\sqrt{i(i-1)}\right)^2 .$$

Since $H_{m/2-1} = \sum_{i=2}^{m/2} 1/\sqrt{(i-1)^2} > \sum_{i=2}^{m/2} 1/\sqrt{i^2} = H_{m/2} - 1$, where H_i is the i^{th} harmonic number, it holds that $\rho \in \Theta(\log^2 m)$. Thus, we have derived the following result.

Theorem 2. *Algorithm OPT solves the device discovery problem with optimal competitiveness*

$$\tfrac{1}{4}\left(1 + \sqrt{7/2} + \textstyle\sum_{i=2}^{m/2} 1/\sqrt{i(i-1)}\right)^2 \in \Theta(\log^2 m).$$

As mentioned above, the class algorithm $ALG_{\log m}$ features an asymptotically optimal competitiveness of $\Theta(\log^2 m)$ as well.

4.5 Optimality for Known Probability Distribution of *t*

In the previous section, we have described an algorithm which solves the discovery problem optimally for unknown t. In the following, we continue our investigations in a slightly different model where the algorithm has a rough estimation on the total number of jammed channels. Concretely, we assume that an algorithm has an a priori knowledge on the probability distribution of the total number of jammed channels: Let $p(0), p(1), \ldots, p(i), \ldots, p(m)$ be the probability that i channels are jammed. We know from Section 4.1 that if $t = i \leq m/2$ is known, the optimal discovery time is $8i$ in expectation. We want to devise an algorithm ALG_{PDF} which estimates t using the distribution $x_0, x_1, ..., x_{m/2}$ over the classes estimating $\widehat{t} = i$, and minimizing the expected total execution time.

Let p_i denote the success probability for $t = i \leq m/2$, i.e., i channels are jammed. For the two classes j and l used by the two nodes, we have a success probability of $\max\{\min\{2j - i, 2l - i\}, 0\}/(2 \cdot 2j \cdot 2l)$, since the nodes can only meet on unjammed channels. In order to compute p_i, we need to sum over all possible pairs of classes multiplied with the probability of selecting them.

$$\begin{aligned} p_i &= \mathbb{P}\left[\text{success}|t = i\right] \\ &= \sum_{j>i/2}^{m/2-1} \sum_{l>i/2}^{m/2} x_j x_l \frac{\min(2j-i, 2l-i)}{8jl} \\ &= \sum_{j>i/2}^{m/2-1} \sum_{l=j+1}^{m/2} 2x_j x_l \frac{2j-i}{8jl} + \sum_{j=i}^{m/2} x_j^2 \cdot \frac{2j-i}{8j^2}. \end{aligned}$$

For $t > m/2$, the expected discovery time is $\frac{m^2}{x_{m/2}^2(m-t)}$. This leaves us with the following optimization problem:

$$\min \left[\sum_{i=0}^{m/2} p(i)/p_i + \sum_{i=m/2+1}^{m} p(i) \cdot \frac{m^2}{x_{m/2}^2(m-i)} \right]$$
$$\text{subject to } \sum_{i=0}^{m/2} x_i = 1.$$

Unfortunately, this formulation is still non-linear. However, there are tools available that can compute the optimal x_i's of ALG_{PDF} numerically using this formulation.

4.6 Multi-player Settings

Scenarios with more than two nodes raise many interesting questions. One could try to minimize the time until the first two nodes have met, the time until a given node has found another given node, or the time until all nodes have had at least one direct encounter with all other nodes in the vicinity. In practice, instead of computing a complete graph where each pair of nodes has interacted directly, it might be more important to simply guarantee connectivity, i.e. ensure the existence of acquaintance paths between all pairs of nodes. In some of these models, it is beneficial to coordinate the nodes and divide the work when they meet.

We leave the study of node coordination strategies for future research. However, in the following, we want to initiate the multi-player analysis with a scenario where the total number of nodes n and the total number of jammed channels t is known, and where a node u wants to find a specific other node v while other nodes are performing similar searches concurrently. Again, we assume a symmetric situation where all nodes execute the same randomized algorithm.

Theorem 3. *Let n be the number of nodes and assume t, the number of jammed channels, is known. If there are $\Omega(n + t)$ channels available, the asymptotically best expected discovery time is $\Theta(n + t)$. The algorithm selecting one of the first $max(2t, 2n)$ channels uniformly at random and sending with probability $1/2$ achieves this bound.*

Proof. Let the i^{th} channel be selected with probability p_i, and assume a given node sends (or listens) on the channel with probability p_s (or with probability $1-p_s$). By the same argument as presented in the previous section, there exists a randomized algorithm minimizing the expected node discovery time by selecting $p_i = 1/k \ \forall i < k$ for some variable k. The discovery probability if k channels are used is given by $(k-t)k^{-2}2p_s(1 - p_s)(1 - p_s/k)^{n-2}$. Thus, it remains to compute p_s and k. Let us start with the last factor of the success probability. Using the fact that $(1 - x/n)^n > e^{-x}$, we can guarantee that the term $(1 - p_s/k)^{n-2}$ is asymptotically constant, if $p_s/k \propto n$ (Condition (1)). Clearly, we have to choose $k > t$ to ensure that a meeting can happen (Condition (2)). Asymptotically, the expected discovery time is in $\Theta(t + n)$, regardless of the precise choice of k and p_s—as long as the Conditions (1) and (2) are satisfied. Concretely, setting $p_s = 1/2$ and $k = \max(2t, 2(n-2))$ leads to an asymptotically optimal expected discovery time.

In reality, nodes typically do not know the number of nodes that are active in the same area simultaneously. What happens if we apply the optimal strategy for two nodes devised in Section 4.4, even though there might be several other nodes? Using the same arguments as in Section 4.4, we can derive that every node executing algorithm OPT is asymptotically optimal as well.

Corollary 2. *Let n be the number of nodes and t the number of jammed channels. Assume that n and t are unknown to the nodes. Algorithm OPT from Section 4.4 achieves an asymptotically optimal competitiveness.*

5 Simulations

In order to complement our formal results, we conducted several *in silico* experiments to study the behavior of our algorithms in different settings. In this section, we discuss our main simulation results. If not mentioned otherwise, we examine a system with 128 channels (Bluetooth uses 79 channels, 32 for discovery) and we discuss the average discovery time of 1,000 experiments.

5.1 Device Discovery

In a first set of experiments, we studied the average discovery time of the optimal algorithm OPT and the algorithm using a logarithmic number of estimators or classes ($ALG_{\log m}$), see also Section 4. A simple solution to the device discovery problem typically used in practice is to select the available channels uniformly at random.Therefore, we include in our plots the algorithm UNI which has a balanced distribution over the channels.

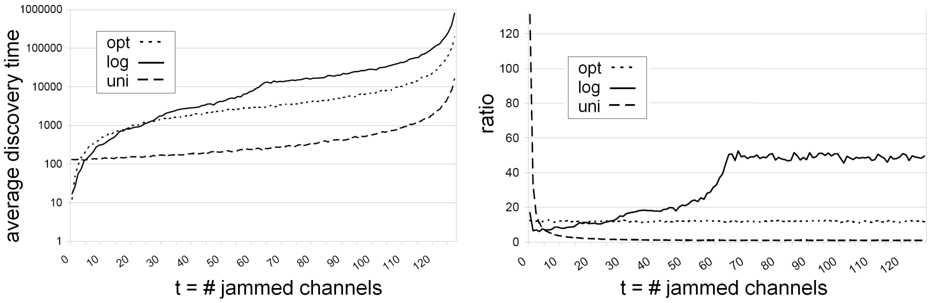

Fig. 1. *Left:* Average discovery time of OPT, $ALG_{\log m}$, and UNI as a function of the total number of jammed channels t. *Right:* Competitive ratios of OPT, $ALG_{\log m}$, and UNI as a function of the total number of jammed channels t.

Figure 1 (left) shows that in case only a small number of channels is jammed, OPT and $ALG_{\log m}$ yield much shorter discovery time (around a factor ten for $t = 0$). However, as expected, the uniform algorithm UNI is much faster if a large fraction of channels are jammed.

The study of the algorithms' competitive ratio is more interesting. Figure 1 plots the ratios of the different algorithms' discovery time divided by the optimal running time if t is known achieved by REF (cf Section 4.1). The figure shows that our optimal algorithm OPT has indeed a perfectly balanced competitiveness of around 12, independently of the number of jammed channels t. The uniform algorithm UNI is particularly inefficient for small t, but improves quickly for increasing t. However, over all possible values for t, UNI's ratio is much worse than that of OPT and $ALG_{\log m}$. Note that $ALG_{\log m}$ is never more than a constant factor off from the optimal algorithm OPT (a factor of around four in this example). The competitive ratio of $ALG_{\log m}$ reaches its maximum at $t > m/2$.

So far, we have assumed a rather pessimistic point of view in our analysis and we considered a worst case adversarial jammer only. Figure 2 (left) studies the algorithms in a setting where a random set of t channels is jammed. Clearly, OPT and $ALG_{\log m}$ perform much better than UNI even for quite a large number of jammed channels. Only if the number of jammed channels exceeds 100, the average discovery time is worse.

5.2 Microwave Case Study

Besides adversarial jamming attacks, a reason for collisions during the discovery phase is interference from other radio sources. It is well-known that microwave ovens interfere with Bluetooth channels (e.g., [14]), especially Bluetooth Channels 60-70. These channels are among the 32 channels that the Bluetooth protocol uses for discovery (called *inquiry* in Bluetooth speak). In other words, the Bluetooth protocol does not exploit the full range of available channels for discovery. The Bluetooth protocol is asymmetric, i.e., nodes either scan the inquiry channels from time to time, or they try to find nodes nearby.

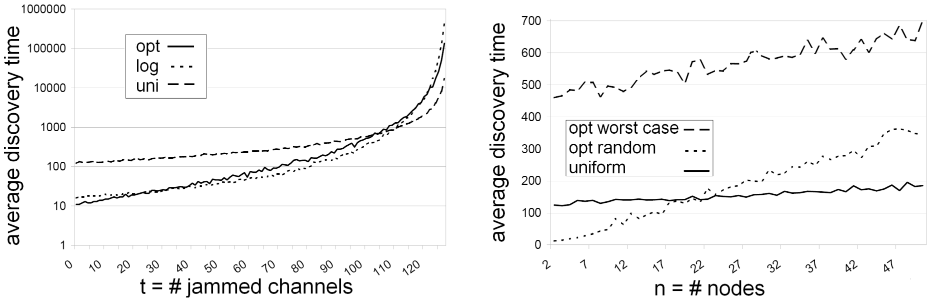

Fig. 2. *Left:* Average discovery time of OPT, $ALG_{\log m}$, and UNI as a function of the total number of jammed channels t. For this plot, the jammed channels are chosen *uniformly at random*. *Right:* Multiplayer: Comparison of the average discovery time of OPT (once with randomly and once with worst-case jammed channels) to UNI if $t = 10$ channels are jammed.

We have conducted a case study modelling the presence of other nodes and a microwave oven. To this end, we simplified the Bluetooth inquiry protocol to its core device discovery algorithm. One node scans the channels constantly and the other node performs the Bluetooth inquiry frequency hopping pattern until they meet. Since Bluetooth only uses 32 out of the 79 available channels for discovery, our optimal algorithm is clearly in advantage by exploiting the whole range of frequencies. We ignore this advantage and consider the following set up: two nodes applying the Bluetooth inquiry protocol and two nodes executing the optimal algorithm for 32 channels seek to meet the node following the same protocol. We have counted the number of time slots Bluetooth and our optimal algorithm need until this meeting happens with and without interference by a microwave oven. We obtained the following results:

Microwave	BT	OPT
off	34.49	15.16
on	45.76	15.70

There is a substantial difference between the performance of the two protocols, especially when considering that the Bluetooth protocol is asymmetric. Hence no collisions occur on the same channels in our setting with two Bluetooth nodes. In other words, our setting is punishing the optimal algorithm for being symmetric. We believe that there are many interesting scenarios where symmetry is required and protocols following a Bluetooth approach are not suitable.

5.3 Multi-player Settings

The algorithms described in Section 4 are tailored to settings where two nodes want to meet efficiently despite a adversarial jammer. However, our analysis and our experiments show (cf. Figure 2, right), that the number of time slots until two designated nodes meet increases linearly in the number of nodes in the vicinity. In large networks or times of high contentions the UNI algorithm performs much better. Thus, in these scenarios, it is beneficial to use this algorithm.

6 Conclusion

The fast and robust discovery of other devices is one of the most fundamental problems in wireless computing. Consequently, a prerequisite to efficient networking are algorithms with the twofold objective of allowing devices to find each other quickly in the absence of any interference, degrading gracefully under increasing disturbance. In other words, discovery algorithms that work well in different settings and under various conditions are necessary. This paper has presented optimal algorithms for a very general, Byzantine model of communication disruptions. This implies that our algorithms can be used in many other scenarios with stronger assumptions on the nature of such disruptions. In other words, our algorithms can cope with incidental as well as with malicious interference. Furthermore, our algorithms are ideal candidates for energy and memory constrained sensor nodes as they are simple and fully distributed.

Other approaches, e.g., based on exponential search techniques can outperform our protocols if the adversary is static, i.e. does not change the number of blocked channels. Another disadvantage of the exponential search technique is the fact that, in contrast to our algorithm, it requires the nodes to start the discovery protocol at the same time.

Our results open many directions for future research. It is important to reason about how the first successful contact between two nodes can be used for a more efficient future communications (e.g., by establishing a shared secret key), and how, subsequently, more complex tasks can be performed over the multi-channel system.

References

1. Alnifie, G., Simon, R.: A Multi-channel Defense Against Jamming Attacks in Wireless Sensor Networks. In: Proc. 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks (Q2SWinet) (2007)
2. Alonso, G., Kranakis, E., Sawchuk, C., Wattenhofer, R., Widmayer, P.: Randomized Protocols for Node Discovery in Ad-hoc Multichannel Broadcast Networks. In: Proc. 2nd Conference on Adhoc Networks and Wireless (ADHOCNOW) (2003)
3. Alonso, G., Kranakis, E., Wattenhofer, R., Widmayer, P.: Probabilistic Protocols for Node Discovery in Ad-Hoc, Single Broadcast Channel Networks. In: Proc. 17th International Symposium on Parallel and Distributed Processing (IPDPS) (2003)
4. Awerbuch, B., Richa, A., Scheideler, C.: A Jamming-Resistant MAC Protocol for Single-Hop Wireless Networks. In: Proc. 27th Symposium on Principles of Distributed Computing (PODC) (2008)
5. Bayraktaroglu, E., King, C., Liu, X., Noubir, G., Rajaraman, R., Thapa, B.: On the Performance of IEEE 802.11 under Jamming. In: Proc. 27th Joint Conference of the IEEE Computer Communication Societies (INFOCOM) (2008)
6. Bertsekas, D., Gallager, R.: Data Networks. Prentice-Hall, Englewood Cliffs (1992)
7. Chiang, J.T., Hu, Y.-C.: Cross-layer Jamming Detection and Mitigation in Wireless Broadcast Networks. In: Proc. 13th ACM Conference on Mobile Computing and Networking (MobiCom) (2007)
8. Commander, C.W., Pardalos, P.M., Ryabchenko, V., Uryasev, S., Zrazhevsky, G.: The Wireless Network Jamming Problem. In Air Force Research Laboratory, Tech. Report 07-11-06-332 (2007)

9. Dolev, S., Gilbert, S., Guerraoui, R., Newport, C.: Gossiping in a Multi-Channel Radio Network (An Oblivious Approach to Coping With Malicious Interference). In: Pelc, A. (ed.) DISC 2007. LNCS, vol. 4731, pp. 208–222. Springer, Heidelberg (2007)
10. Dolev, S., Gilbert, S., Guerraoui, R., Newport, C.: Secure Communication over Radio Channels. In: Proc. 27th ACM Symposium on Principles of Distributed Computing (PODC), pp. 105–114 (2008)
11. Gilbert, S., Guerraoui, R., Newport, C.: Of Malicious Motes and Suspicious Sensors. In: Shvartsman, M.M.A.A. (ed.) OPODIS 2006. LNCS, vol. 4305, pp. 215–229. Springer, Heidelberg (2006)
12. IEEE 802.15.2 Taskforce. Coexistence Mechanisms (2008), `http://www.ieee802.org/15/pub/TG2-Coexistence-Mechanisms.html`
13. Koo, C.-Y., Bhandari, V., Katz, J., Vaidya, N.H.: Reliable Broadcast in Radio Networks: the Bounded Collision Case. In: Proc. 25th ACM Symposium on Principles of Distributed Computing (PODC) (2006)
14. Krishnamoorthy, S., Robert, M., Srikanteswara, S., Valenti, M.C., Anderson, C.R., Reed, J.H.: Channel Frame Error Rate for Bluetooth in the Presence of Microwave Ovens. In: Proc. Vehicular Technology Conference (2002)
15. Law, C., Mehta, A., Siu, K.-Y.: Performance of a Bluetooth Scatternet Formation Protocol. In: Proc. 2nd ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHoc) (2001)
16. Law, Y.W., van Hoesel, L., Doumen, J., Hartel, P., Havinga, P.: Energy-efficient Link-layer Jamming Attacks Against Wireless Sensor Network MAC Protocols. In: Proc. 3rd ACM Workshop on Security of Ad hoc and Sensor Networks (SASN) (2005)
17. Li, M., Koutsopoulos, I., Poovendran, R.: Optimal Jamming Attacks and Network Defense Policies in Wireless Sensor Networks. In: Proc. 26th Joint Conference of the IEEE Computer Communication Societies (INFOCOM) (2007)
18. Noubir, G.: On connectivity in ad hoc networks under jamming using directional antennas and mobility. In: Langendoerfer, P., Liu, M., Matta, I., Tsaoussidis, V. (eds.) WWIC 2004. LNCS, vol. 2957, pp. 186–200. Springer, Heidelberg (2004)
19. Pelc, A., Peleg, D.: Feasibility and Complexity of Broadcasting with Random Transmission Failures. Theoretical Computer Science 370(1-3), 279–292 (2007)
20. Roberts, L.G.: ALOHA Packet System with and without Slots and Capture. SIGCOMM Computer Communication Review 5(2), 28–42 (1975)
21. Tay, Y.C., Jamieson, K., Balakrishnan, H.: Collision-Minimizing CSMA and Its Applications to Wireless Sensor Networks. IEEE Journal on Selected Areas in Communications 22(6) (2004)
22. Wood, A.D., Stankovic, J.A., Zhou, G.: DEEJAM: Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks. In: Proc. 4th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON) (2007)
23. Xu, W., Ma, K., Trappe, W., Zhang, Y.: Jamming Sensor Networks: Attack and Defense Strategies. IEEE Network (2006)
24. Xu, W., Wood, T., Trappe, W., Zhang, Y.: Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service. In: Proc. 3rd ACM Workshop on Wireless Security (WiSe) (2004)