8-19-2014

Algorithms for Blind Rendezvous in Wireless Networks

Sixia Chen
University of Connecticut - Storrs, chensixia09@gmail.com

Algorithms for Blind Rendezvous in Wireless Networks

Sixia Chen, Ph.D.

University of Connecticut, 2014

Blind rendezvous is a fundamental problem in wireless networks. Rendezvous problems involve a collection of agents, or nodes, each of which would like to discover and communicate with the other agents in the collection who are within its transmission range. Two agents are said to *rendezvous* when they become aware of one another and are able to communicate. *Blindness* refers to a set of constraints on any algorithm that is to guarantee rendezvous in a typical wireless network.

- Agents begin with no knowledge of one another and have no means of coordination other than the common radio channels available to them.
- Agents are typically not synchronized, so different agents may be deployed with their clocks offset from one another by some amount.
- Individual agents are identical. This means that two agents operating on a common rendezvous protocol can only make decisions based on their clock readings and on their experiences since deployment. They cannot act differently based on a distinction between their individual identities.

Problems of this nature take different forms in various wireless settings. This dissertation describes two different blind rendezvous problems:

1. *The multi-channel rendezvous problem* pertains to cognitive radio networks where each node has access to a potentially different subset of the radio spectrum, and can only utilize a single radio frequency at one time. The challenge here is achieving rendezvous between agents who each do not know what channel their neighbor will be using at any given time.

2. *The energy-constrained, single channel rendezvous problem* for wireless sensor networks, in which a sensor node wishes to maximize its battery life by keeping its radio powered off as much a possible, while still achieving a timely rendezvous with neighboring sensors on the network's common radio channel.

In the following, we describe the two specific settings in detail, present contributions which advance the state of the art in both settings, and discuss some lines along which we feel further investigation may yield additional progress.

Algorithms for Blind Rendezvous in Wireless Networks

Sixia Chen

M.S., Computer Science and Engineering, University of Connecticut, 2013

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2014

APPROVAL PAGE

Doctor of Philosophy Dissertation

Algorithms for Blind Rendezvous in Wireless Networks

Presented by

Sixia Chen, M.S.

Major Advisor _____
Alexander Russell

Associate Advisor _____
Sanguthevar Rajasekaran

Associate Advisor _____
Bing Wang

University of Connecticut

2014

ACKNOWLEDGEMENTS

I want to give my great gratitude to my beloved advisor, Professor Alexander Russell. I thank him for all these years' support, supervision and guidance. Without him, this dissertation would never be done. I feel honored and fortunate to be his student. His spirit and attitude toward research, his dedication to work, his enthusiastic teaching and his kindness will always have a great bearing on my life.

I would like to thank my two associate advisors, Professor Bing Wang and Professor Sanguthevar Rajasekaran. I have benefitted greatly from their courses on networks and algorithms. Professor Bing Wang also gave me plenty of advice on my work during my Ph.D. career, some of which has helped to produce the results in this dissertation. I also thank Professor Laurent Michel, Professor Donald Sheehy and Professor Vincent Huang for their advice.

I would like to express my gratitude to my collaborators, Professor Cris Moore, Professor Ravi Sundaram, Abhishek Samanta, Hieu Dinh and Ruofan Jin. It was my great honor or to work with such intelligent researchers. Thanks for all the discussions.

I also want to thank Professor Yoo-Ah Kim for her two years' advice and support. She recruited me to the graduate program in the department of Computer Science at UCONN. I will never forget that. My thanks also go to my collegues and friends, Chadi El Kari, Karpoor Shashidhar, Hang Dinh, Hari Krovi, Qiang Tang, Therese Smith, Wei Zeng and Wen Chen for all their help.

Finally, I want to thank my family, my parents, my sister and especially, my husband Matthew Coolbeth. He has given me his full support during my Ph.D. study. With his love, I have the motivation to complete the doctoral program.

TABLE OF CONTENTS

Chapter 1

Introduction

1.1 Background

When first deployed in a network environment, a wireless agent will typically be within transmission range of one or more neighbors, but communication between the agents cannot occur until they detect and identify one another. Once this happens, two agents can exchange information to synchronize for future communication, share data, and perform other cooperative tasks (which are beyond the scope of this document). Before the rendezvous is achieved, all the agent can do is blindly probe with its radio to make its presence known and listen for probes from other agents. Typically, it is in the interest of the network for nodes to rendezvous with all neighbors as early as possible after deployment, but this must often be weighed against other concerns, such as power management (radio use will drain the battery of a mobile agent) or the need to avoid congestion in public radio channels.

1.2 Environment Settings

We consider two different *blind* rendezvous problems in wireless networks.

The first problem, called the *multi-channel rendezvous problem* and concerns the rendezvous task faced by a collection of agents with cognitive radios, each of which has access to some amount of the complete radio frequency spectrum. Two agents rendezvous if they both access the same channel at the same time slot.

The second problem is called the *neighbor discovery problem* for wireless sensor networks, is a single-channel rendezvous problem with energy constraints. Each agent has a duty cycle which limits the fraction of time that is allowed for the usage of its radio. Two agents rendezvous if they both have their radio on at the same time.

The task for both of the problems is for neighboring agents to rendezvous in the shortest possible time. We will describe the two problems in detail respectively in the following chapters.

Both problems model the rendezvous task for a network of asynchronous, anonymous agents. In each case, every agent is able to measure time and has a clock that begins counting from zero at the time of the agent's deployment. We are generally concerned with *rendezvous latency*, the amount of time that it takes for two neighboring agents to rendezvous after they have both been deployed. We work on designing rendezvous protocols to minimize latency without violating other modeled constraints, and our results include rigorous proofs of the latency guarantee provided by each protocol.

1.3 Challenges

There are some challenges for both of the rendezvous problems. We will address them as follows.

1.3.1 Asynchrony

There may not be a common notion of time. Different agents may have different clocks, although the speeds of the clocks are the same. Therefore different agents may "wake-up" at different times, which may induce a time offset between their schedules. Our goal is to design protocols such that no matter what the offset is, any two agents are guaranteed to rendezvous. We also want to minimize the latency from the time they both "wake-up" until they rendezvous. Actually, for multi-channel rendezvous problem, we consider both synchronous and asynchronous scenarios.

1.3.2 Anonymity

There are no identities for the agents, therefore they have to be treated equally. For the multi-channel rendezvous problem, an agent's channel hopping schedule only depends on its available subset of channels. For the energy-constrained, single channel rendezvous problem, an agent's waking schedule only depends on its duty cycle. In other words, agents may not rely on distinct identities for creation of their schedules.

1.3.3 Asymmetry

Different agents may have different parameters. For the multi-channel rendezvous problem, we consider two settings: in the *symmetric* case, all agents have access to the same subset of the spectrum, and in the *asymmetric* case, each agent may have access to a different set of channels. For the energy constraint single channel rendezvous problem, different agents may have different duty cycles. For this problem, we only focus on the symmetric case where the agents all have the same duty cycles, which we call the *homogeneous* case. Designing protocols for the asymmetric or the heterogeneous case is worth considering in future work.

1.4 The Contributions of this Dissertation

In this dissertation, we present nontrivial results for both of the rendezvous problems.

For the multi-channel rendezvous problem, we first design a deterministic algorithm for a special case in which each agent can only access two channels. We also show that this algorithm is the best possible deterministic algorithm for the size-2 case by proving a lower bound. We then apply this size-2 algorithm to yield an algorithm for the general case, where each agent may have access to an arbitrarily large fraction of the spectrum. This algorithm is shown to be tight up to a factor of $\log \log n$, where n is the the number of channels in the complete spectrum. We remark that our algorithms are the first whose performance scales as a function of the sizes of the subsets of available channels.

Having given these main results, we go on to show some specialized results for a variant of the setting described above. We give a randomized algorithm for settings where the agents benefit from a "one bit beacon" - a single common random bit that may change at each time slot.

For the energy-constrained, single channel rendezvous problem, we give two protocols for the traditional model where time is discrete and two agents rendezvous if they are both active in one time slot. This is the model commonly considered in the previous literature on the topic. The lower bound under this model is already known. We are the first to give efficient, optimal protocols for this model. We also develop a generalized, continuous-time model with two parameters, minimum waking time and minimum contact time. We prove a lower bound for the new model and provide a reduction that can transform any schedule for the basic integer model to a schedule

for the generalized model. A nice property of this transformation is that any nearly-optimal schedule for the former model is mapped to a nearly-optimal schedule for the latter.

1.5 Dissertation Roadmap

The remainder of this dissertation is organized as follows.

In Chapter 2, we present our work on the multi-channel rendezvous problem. In Chapter 3, we describe our work on the energy-constrained, single channel rendezvous problem. Finally, we conclude this dissertation and discuss directions for future work in Chapter 4.

Chapter 2

Multi-Channel Rendezvous in Cognitive Radio Networks

2.1 Introduction

Given the ever-increasing demand for all things wireless, *spectrum* has become a scarce resource. Historically, regulators around the world have employed a command and control philosophy towards managing spectrum [35]: Some channels were statically licensed to particular users (for certain periods and in certain geographies) while others were kept aside for community use. *Cognitive radio networks* have emerged as a modern, dynamic approach to spectrum allocation [1, 40]. Exploiting recent technological developments, cognitive agents (radios) dynamically sense incumbent users and opportunistically hop to unused channels. While they can offer improved utilization, they introduce a fundamental *rendezvous* problem: the problem of discovering the existence of peers in a multichannel setting.

We work in the *blind* model where a collection of agents A_i wish to discover each other with no dedicated common control channel or other shared infrastructure. Time

is divided into discrete slots and spectrum is divided into discrete channels, $[n] = 1, 2, \ldots, n$. Each agent may access (or "hop on") a single channel in a single time slot and two agents *rendezvous* when they hop on the same channel in the same time slot. The challenge is to design a channel-hopping schedule for each agent so that they discover each other. With no further constraints, the problem has the trivial solution where all agents can hop on a specific channel, say channel 1, in the very first time slot. However, to more faithfully reflect the circumstances in practice, the standard rendezvous model has three additional requirements: *asymmetry*, *asynchrony* and *anonymity*.

Asymmetry. Different agents may have access to different subsets of channels as a result of local interference or variations in radio capabilities. Let $S_i \subseteq [n]$ be the subset of channels to which agent A_i has access. Thus the challenge is to create for each agent A_i a channel-hopping *schedule* $\sigma_i : \{0, 1, \ldots\} \to S_i$ which guarantees that $\exists\, t, \sigma_i(t) = \sigma_j(t)$ for any two agents A_i, A_j, for which $S_i \cap S_j \neq \emptyset$. (In certain cases, we analyze the *symmetric setting* in which agents have access to the identical subset of channels.)

Asynchrony. Different agents may not share a common notion of time. They may commence at different "wake-up" times inducing a relative shift in their progress through their schedules. Note that agents do possess a common understanding of slot duration. The goal, therefore, is to ensure rendezvous between a pair of agents in the shortest possible time once they have both woken up. (In certain cases, we discuss the *synchronous* setting in which all agents share a common notion of absolute time, and furthermore commence their schedules at the same time $t = 0$.)

Anonymity. In our setting an agent's schedule must depend only on the available subset of channels; i.e., σ_i must depend only on S_i. In particular, agents may not rely

on distinct identities for creation of their schedules. Note that S_j is unknown to A_i for $i \neq j$ and it is possible for two different agents to have the same set of accessible channels, i.e., $S_i = S_j$ for $i \neq j$.

The problem has a naive randomized solution, in which each agent, at each time step, selects a channel uniformly and independently at random from its subset. It is easy to see that the expected time to rendezvous is then $|S_i| \cdot |S_j|$ and, furthermore, that two agents will rendezvous in time $O(|S_i||S_j| \log n)$ with high probability (that is, probability $1 - 1/\text{poly}(n)$). However, the deterministic setting is the gold-standard in the cognitive radio networking community: it makes the weakest assumptions about the devices, which need not have an available source of randomness, and provides absolute guarantees on rendezvous time.

Here is a brief summary of our main results:

Algorithms

1. We give an $O(\log \log n)$ time algorithm for rendezvous for the special case of agents with $|S_i| = 2$.

2. We then show how to apply this algorithm to yield algorithms for arbitrary subsets of $[n]$ that guarantees rendezvous time $O(|S_i||S_j| \log \log n)$ for all pairs of sets S_i and S_j.

3. We show that a minor adaptation of this algorithm can furthermore guarantee $O(1)$ time rendezvous for the symmetric case.

4. Finally, we explore the "one bit beacon" case, where the agents have the luxury of a single common random bit during each time slot. In this model, we show that $O\left(\log^2 n \cdot |S_i \cup S_j|/|S_i \cap S_j|\right)$ time is sufficient, with high probability, to rendezvous.

Lower Bounds

1. We prove an $\Omega(\log \log n)$ lower bound on the rendezvous time, even for *synchronous* agents with the promise that the channel sets S_i have constant size. This shows that some dependence on n, the size of the channel universe, is always necessary. In particular, this shows that the algorithm of 1 above is tight up to a constant.

2. For channel subsets of size k we prove a k^2 lower bound on the synchronous rendezvous time, under the promise that $k = O(\log n / \log \log n)$. For larger values of k, we obtain a weaker family of results.

3. In the asynchronous time model, we prove that $|S_i||S_j|$ steps are necessary to rendezvous, so long as $|S_i| + |S_j| \leq n + 1$.

4. All the above lower bounds are for deterministic algorithms. We also prove that, in the asynchronous time model, the lower bound for randomized algorithms is $|S_i||S_j|$ in the case when $|S_i| + |S_j| \leq n + 1$.

2.2 Related Work

Rendezvous problems have a long history in mathematics and computer science—an early example is Rado's famous "Lion and Man" problem [6]. Over time a variety of problems and techniques have evolved in both adversarial [18] and cooperative settings [2]. Rendezvous in networks has been extensively studied in the computer science community [27]. Though the study of rendezvous in cognitive radio networks is relatively recent there already exists a comprehensive survey [25] that contains a detailed taxonomy of the different models including the specific one relevant to this work. The

problem of guaranteed blind rendezvous in the asymmetric, asynchronous and anonymous case was first considered by [8] and subsequently by [36] and [24]. After further progress by [32], the general case of the problem withstood attack until work of [29] and [23]. The current state of the art is the algorithm of [14] which achieves an $O(n^2)$ algorithm for the asymmetric case and $O(n)$ for the symmetric case.

Finally, we mention work of [12]: his *globally synchronous* and *locally synchronous* models correspond to our asynchronous and synchronous models, respectively. However, [12] works in a model that requires explicit control of conjestion by demanding that exactly one node transmit on a single fixed channel for a successful broadcast; this assumption significantly changes the underlying combinatorics of the problem. We remark that in typical practical settings, "chirp and listen" techniques [39]—where an agent sends a short chirp at the beginning and end of each interval of activity on a channel, while listening for other chirps during the interval—can avoid the necessity of explicitly modeling collisions.

2.3 Our Contributions

A crucial difference between previous constructions and ours is that we explicitly exploit the fact that the schedule σ_i can depend arbitrarily on S_i, whereas the earlier constructions [14, 23, 29] derive the schedule for a channel subset by (essentially) projecting onto the desired subset from a single, uniformly generated schedule for the full set of channels. In particular, we provide a general framework that yields significantly more efficient schedules with guaranteed rendezvous in time $O(|S_i||S_j| \log\log n)$. We remark that our schedules are the first whose performance scales as a function of the sizes of the sets S_i.

Real-world cognitive networks [38] operate in a pooled hyperspace occupied by signals with dimensions of frequency, time, space, angle of arrival, etc., comprising spectrum that may range from radio frequencies and TV-band white spaces to lasers. In these networks the total number of channels (n in our parlance) is large, while the channel subsets accessible to any given device may be small. A similar situation prevails in military situations where different members of a (dynamic) coalition operate in a small portion of the available spectrum which guarantees overlap with allies. In such situations (where available sets of channels are small) our scheme achieves a near-quadratic factor gain over the previous results. For the symmetric setting, discussed in detail later, our construction achieves $O(1)$ rendezvous time, which clearly cannot be bettered. Table 1 presents a summary of our upper bounds in the context of prior work. Full details appear in Section 2.5.

Table 1: Upper bounds for deteministic rendezvous.

Paper	Asymmetric	Symmetric
Shin-Yang-Kim [29]	$O(n^2)$	$O(n^2)$
Lin-Liu-Chu-Leun [23]	$O(n^3)$	$O(n)$
Gu-Hua-Wang-Lau [14]	$O(n^2)$	$O(n)$
Our results	$O(\|S_i\|\|S_j\| \log\log n)$	$O(1)$

We are also the first to provide nontrivial lower bounds for the problem; a notable feature of our lower bounds is a connection between the rendezvous problem and Ramsey theory. See Section 2.6.

We are also the first to consider the "one-bit beacon" setting in which the agents have a single common random bit in each time slot. We give an efficient randomized algorithm for the problem variant.

2.4 Definitions and Notation

Let $\mathcal{S}$ be a collection of subsets of $[n]$. An *$\mathcal{S}$-schedule* is a family of schedules $\sigma_S : \mathbb{N} \to S$, one for each $S \in \mathcal{S}$. In fact, we focus solely on two special cases:

- An *n-schedule* is a $2^{[n]}$-schedule, one that supplies a schedule for every subset of $[n]$.
- An *(n, k)-schedule* is a $\mathcal{S}$-schedule, where $\mathcal{S}$ consists of all subsets of $[n]$ of size k.

We will typically reserve the notation $\Sigma = (\sigma_A)_{A \in \mathcal{S}}$ to denote an $\mathcal{S}$-schedule; departing from the notation used in the introduction, the schedule associated with the set A is simply denoted σ_A.

Let $\sigma_A : \mathbb{N} \to A$ and $\sigma_B : \mathbb{N} \to B$ be two schedules for overlapping subsets A and B of $[n]$. We say that σ_A and σ_B *rendezvous synchronously in time T* if there is a time $t \leq T$ so that $\sigma_A(t) = \sigma_B(t)$. Recall that the asynchronous model introduces arbitrary "wake-up" times t_A and t_B into each of the two schedules, after which they proceed with their schedules. Of course, in this case they cannot possibly rendezvous before time $\max(t_A, t_B)$, when they are finally both "awake." Thus, we say that these two schedules *rendezvous asynchronously in time T* if, for all $t_A, t_B \geq 0$, there is a time

$$\max(t_A, t_B) \leq t \leq \max(t_A, t_B) + T$$

so that $\sigma_A(t - t_A) = \sigma_B(t - t_B)$.

For a fixed (n, k)-schedule Σ, we define $R_s(\Sigma)$ to be the minimum T for which σ_A and σ_B synchronously rendezvous in time T for all $A, B \in \mathcal{S}$. We likewise define $R_a(\Sigma)$ for asynchronous rendezvous. Finally, we define:

$$R_s(n, k) \triangleq \min_{\Sigma} R_s(\Sigma) \quad \text{and} \quad R_a(n, k) \triangleq \min_{\Sigma} R_a(\Sigma)\,,$$

where these are minimized over all (n, k)-schedules Σ. Of course, $R_s(n, k) \leq R_a(n, k)$. The simple randomized algorithm described in the introduction suggests that one might be able to achieve

$$R_a(n, k) \lessapprox k^2 \log n \,.$$

Finally, we remark that even a precise understanding of $R_a(n, k)$ does not necessarily yield n-schedules that guarantee satisfactory bounds on pairwise rendezvous because it is not, in general, clear how to stitch together (n, k)-schedules for different values of k to provide guarantees for pairs of sets of different sizes.

Notation We use $[n] = \{1, \ldots, n\}$ and invent the shorthand notation $\log^\sharp n \triangleq \lceil \log_2 n \rceil$. Whenever a variable, x, represents a natural number, we use x_2 to denote the canonical base-two encoding of x. When the variable x is drawn from a set $\{0, \ldots, m\}$, we further assume that x_2 is zero-padded on the left out to length $\log^\sharp m$.

2.5 Schedules for Efficient Rendezvous

2.5.1 Sets of Size Two

We begin with a construction of a family of schedules for channel sets of size 2 that achieves rendezvous in time $O(\log \log n)$; these will be used as a subroutine for the general construction. We shall see in Section 2.6 that these schedules are within a constant of optimal. Thus, the goal of this section is to prove the following theorem.

Theorem 1. *For all $n > 0$,*

$$R_a(n, 2) = O(\log \log n) \,.$$

Specifically, for any $n > 0$, there is an $(n, 2)$-schedule so that for any two sets A and B of size two, σ_A and σ_B rendezvous asynchronously in time no more than $O(\log \log n)$.

The size-2 construction is based on the remarkable fact that there is an edge coloring of the linear poset, using only $\log^\sharp n$ colors, for which no path of length two is monochromatic. Specifically, consider the directed graph $L_n = (V_n, E_n)$, with vertex set $V_n = [n]$ and directed edges $E_n = \{(a, b) \mid a < b\}$. A *edge coloring* of L_n is a mapping $\chi : E_n \to P$ with the property that $\chi(a, b) \neq \chi(b, c)$ for any pair of directed edges (a, b) and (b, c) that form a directed path of length 2.

Lemma 2. *The graph L_n has an edge coloring with a palette of size $\log^\sharp n$.*

Proof. With hindsight, associate with each vertex $k \in V_n$ the set

$$X_k = \{i \mid \text{the } i\text{th bit of } k_2 \text{ is a } 1\} \subset \{1, \ldots, \log^\sharp n\}\,.$$

Observe that if $a < b$, there is an element in $X_b \setminus X_a$. In this case, we may safely color the edge (a, b) with any element of $X_b \setminus X_a$, as it follows immediately that any pair of edges forming a directed path must have distinct colors. The scheme uses no more than $\log^\sharp n$ colors. □

Proof of Theorem 1. We begin with a construction for the simpler synchronous model, and then show how to reduce the asynchronous model to this case.

The synchronous model. In the synchronous model, we will simplify the presentation by discussing finite length schedules with the understanding that rendezvous is guaranteed by the time the schedule has been exhausted. Consider now a subset of two channels $A = \{a_0, a_1\}$, where $a_0 < a_1$. We will treat such size-2 subsets as directed edges of the linear poset (directed from the smaller element to the larger element). In this size-2 case, we may express a schedule as a binary string $s_0 s_1 s_2 \ldots \in \{0, 1\}^*$ with the convention that at time t, the schedule calls for a_{s_t}: thus, when $s_t = 0$ the schedule calls for the smaller of the two channels; when $s_t = 1$, the schedule calls for the larger of the two channels.

Consider now a pair of overlapping subsets $A = \{a_0, a_1\}$ and $B = \{b_0, b_1\}$ with $a_0 < a_1$ and $b_0 < b_1$. When these two edges form a directed path (so that their common element is the larger of one set and the smaller of the other), a sufficient condition for two schedules $r_0 r_1 \dots r_{\ell-1}$ and $s_0 s_1 \dots s_{\ell-1}$ to rendezvous is that each of the two tuples $\{(0,1),(1,0)\}$ can be realized as (r_t, s_t) for some t, which is to say that

$$\{(0,1),(1,0)\} \subset \{(r_t, s_t) \mid 0 \le t < \ell\}\,. \tag{1}$$

We reserve the notation $r \diamondsuit_1 s$ to denote the statement that the strings r and s satisfy condition (1). Likewise, when $\{a_0, a_1\}$ and $\{b_0, b_1\}$ do not form a path of length two (that is, share a common largest or smallest element), a sufficient condition for rendezvous is that

$$\{(0,0),(1,1)\} \subset \{(r_t, s_t) \mid 0 \le t < \ell\}\,. \tag{2}$$

We reserve the notation $r \diamondsuit_0 s$ to denote the statement that r and s satisfy (2).

In the remainder of the proof we identify a map $x \mapsto \mathsf{C}(x)$ with the property that

$$x = y \quad \Rightarrow \quad \mathsf{C}(x) \diamondsuit_0 \mathsf{C}(y)\,, \tag{3}$$

$$x \neq y \quad \Rightarrow \quad \mathsf{C}(x) \diamondsuit_1 \mathsf{C}(y)\,. \tag{4}$$

With such a map in hand, we adopt the schedule $\mathsf{C}(\chi(\alpha,\beta)_2)$ for the set $\{\alpha, \beta\}$, where χ is the edge coloring of Lemma 2. Observe that if $A = \{a_0, a_1\}$ and $B = \{b_0, b_1\}$ form a path of length two, $\chi(a_0, a_1) \neq \chi(b_0, b_1)$ and this schedule guarantees rendezvous by dint of property (4). Otherwise, these schedules guarantee rendezvous by dint of property (3).

We return to the problem of constructing the map $\mathsf{C}(\cdot)$. By adopting the convention that all schedules start with the prefix 01, we can immediately guarantee property (3): $(0,0)$ and $(1,1)$ appear in $\{(r_t, s_t) \mid 0 \le t < \ell\}$. It is easy to check that the map

$x \mapsto 01 \circ x \circ \overline{x}$, where $\circ$ denotes concatenation and $\overline{x}$ the coordinatewise negation of x, has the desired properties.

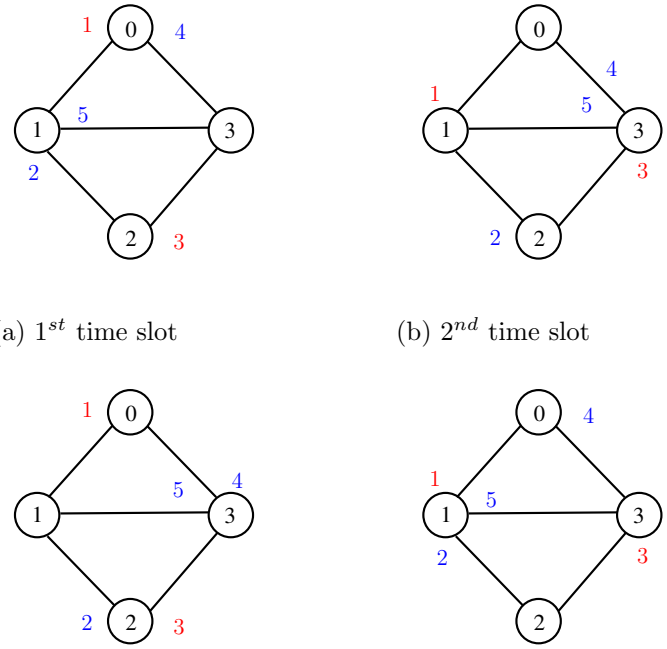
A leaner mapping can be obtained by the rule

$$\mathsf{C}(x) \triangleq 01 \circ x \circ \overline{\mathrm{wt}(x)_2}, \tag{5}$$

where $\mathrm{wt}(x)$ denotes the weight (number of 1s) of the string x. To see that this encoding has property (4), observe that when $\mathrm{wt}(x) = \mathrm{wt}(y)$, both $(0,1)$ and $(1,0)$ must appear in the set $\{(x_i, y_i) \mid 1 \leq i \leq |x|\}$ (where x_i is the i^{th} bit of x) as $x \neq y$ and they have common weight. When $\mathrm{wt}(x) < \mathrm{wt}(y)$, it follows immediately that $(0,1) \in \{(x_i, y_i) \mid 1 \leq i \leq |x|\}$; as for the tuple $(1,0)$, this must be realized by one of the coordinates of $\overline{\mathrm{wt}(x)_2}$ and $\overline{\mathrm{wt}(y)_2}$ as the canonical encoding of integers in binary ensures that when $n < m$, there is a coordinate in which n_2 contains a 0 and m_2 contains a 1. The case when $\mathrm{wt}(x) > \mathrm{wt}(y)$ is handled similarly.

Finally, we remark that when x has length ℓ, $\mathsf{C}(x)$ has length $\ell + \log^\sharp \ell + 2$. As L_n can be edge colored with a palette of size $\log^\sharp n$, this yields a family of schedules for sets of size 2 that guarantees rendezvous in time no more than $\log^\sharp \log^\sharp n + \log^\sharp \log^\sharp \log^\sharp n + 2$.

An example. Consider an example network with 5 agents having access to a total of 4 channels. Agents, $1, 2, 3, 4, 5$ have access to sets of channels $\{0,1\}, \{1,2\}, \{2,3\}, \{0,3\}, \{1,2\}$, respectively. Figure 1 reflects this setting: the nodes in the graphs represent corresponding channels; the edge connecting nodes x and y represents the agent having access to channels x and y. The coloring χ then induces a schedule as described by Lemma 2: $\chi(0,1) = 0, \chi(0,3) = 1, \chi(1,3) = 1, \chi(1,2) = 1, \chi(2,3) = 0$. Equation (5) then yields the binary hopping sequences followed by the agents: $\mathsf{C}(\chi(0,1)) = 0101, \mathsf{C}(\chi(0,3)) = 0110, \mathsf{C}(\chi(1,3)) = 0110, \mathsf{C}(\chi(1,2)) = 0110, \mathsf{C}(\chi(2,3)) = 0101$. Using these sequences, the 5 agents rendezvous with each other in 4 time slots.



(a) 1^{st} time slot

(b) 2^{nd} time slot

(c) 3^{rd} time slot

(d) 4^{th} time slot

Figure 1: Example of channel hopping in synchronous setting. An active agent is colored with $\chi(x, y)$ if it has access to channels x and y.

The asynchronous model. We return now to the asynchronous model described in the introduction, in which the two agents' schedules are subjected to an unknown shift due to potentially distinct start-up times. In this model, we are obligated to define schedules for all nonnegtive times (that is, our schedules have the form $\sigma : \mathbb{N} \to S \subset [n]$); one straightforward method for describing such schedules is to adopt *cyclic schedules*, which cyclicly repeat the same finite sequence of channels. In particular, if $\sigma : \{0, \ldots, \ell-1\} \to S \subset [n]$, we let $\sigma^\circ : \mathbb{N} \to S$ denote the schedule $\sigma^\circ : t \mapsto \sigma(t \bmod \ell)$.

Continuing in the spirit of the previous discussion, we observe that if $r = r_0 \ldots r_{\ell-1}$ and $s = s_0 \ldots s_{\ell-1}$ are two schedules for a pair of sets $A = \{a_0, a_1\}$ and $B = \{b_0, b_1\}$ forming a path, the cyclic schedules they induce will guarantee rendezvous (in time ℓ)

if, for all i and j,

$$\mathsf{S}^i r \diamondsuit_1 \mathsf{S}^j s\,, \tag{6}$$

where $\mathsf{S}^i x$ denotes the result of cyclicly shifting x forward i symbols. To save ink, we define $r \blacklozenge_1 s$ to denote the condition (6): $\mathsf{S}^i r \diamondsuit_1 \mathsf{S}^j s$ for all i and j. Likewise, we define $r \blacklozenge_0 s$ when $S^i r \diamondsuit_0 S^j s$ for all i and j. As above, when these two sets do not form a path, $r \blacklozenge_0 s$ is a sufficient condition for rendezvous.

Thus our strategy shall be to define a map $x \mapsto \mathsf{R}(x)$ with the property that for two strings x, y,

$$x = y \Rightarrow \mathsf{R}(x) \blacklozenge_0 \mathsf{R}(y) \qquad \text{and} \qquad x \neq y \Rightarrow \mathsf{R}(x) \blacklozenge_1 \mathsf{R}(y)\,. \tag{7}$$

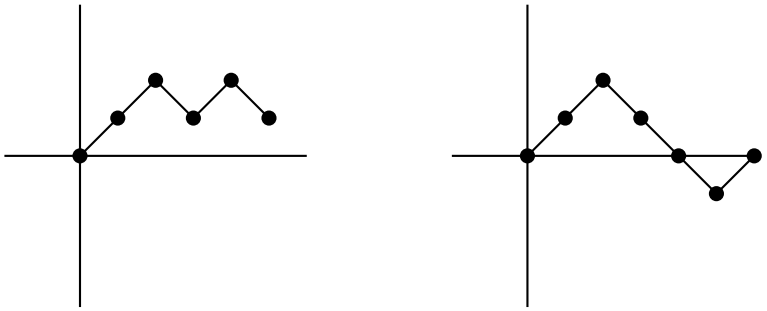
With such a map defined, the construction follows that of the previous construction: the cyclic schedule adopted by the pair (α, β) is given by $\mathsf{R}(\chi(\alpha, \beta)_2)$ where χ is an edge coloring of L_n.

Anticipating the construction, we set down some terminology. For a string z, we define the "graph" of z to be the function $G_z : \{0, \ldots, |z|\} \to \mathbb{Z}$ given by

$$G_z(0) = 0, \quad G_z(k) = \sum_{i=1}^{k} (2z_i - 1)$$

so that G_z traces out the "walk" prescribed by z in which each 1 corresponds to a step northeast and each 0 corresponds to a step southeast as in Figure 2a. We say that a binary string z is *balanced* if $\mathrm{wt}(z) = |z|/2$ (so that $|z|$ is necessarily even); equivalently $G_z(|z|) = 0$, see Figure 2b. A balanced string z is *Catalan* if G_z is never negative. If G_z is positive, which is to say that $G_z(i) > 0$ for all $0 < i < |z|$, we say that z is *strictly Catalan*; see Figure 3. We remark that if z is Catalan, $1 \circ z \circ 0$ is strictly Catalan. Finally, we say that z is *t-maximal* if the set $\{i \mid G_z(i) = \max_j G_z(j)\}$ has size exactly t; the notion *t-minimal* is defined analogously. Note that a strictly Catalan sequence

z is 1-minimal and this single minimum appears at $i = 0$. We remark that if the string z is t-maximal (or t-minimal), the same can be said of all shifts of z.



(a) The graph of the sequence 11010.

(b) The graph of the balanced sequence 110001.

Figure 2: Graphs and balanced strings.

Our strategy is to work with an injective map $\mathsf{R}(\cdot)$ with the property that $\mathsf{R}(x)$ is balanced, strictly Catalan, and 2-maximal. Before describing a construction, we observe that such a map has the properties outlined in (7) above.

Observe, first of all, that if two distinct strings $\mathsf{R}(x)$ and $\mathsf{R}(y)$ are balanced, it follows immediately that $\mathsf{R}(x) \diamondsuit_1 \mathsf{R}(y)$, indeed, the number of appearances of $(0, 1)$ is the same as the number of appearances of $(1, 0)$ and cannot be zero because the strings are distinct. Thus, when $\mathsf{R}(x)$ and $\mathsf{R}(y)$ are balanced, the condition that

$$\mathsf{R}(x) \notin \{\mathsf{S}^i \mathsf{R}(y) \mid i \in [\ell]\}$$

is enough to guarantee that $\mathsf{R}(x) \blacklozenge_1 \mathsf{R}(y)$. Note that if a string z is strictly Catalan, no nontrivial shift of z can be strictly Catalan. In particular, all nontrivial shifts

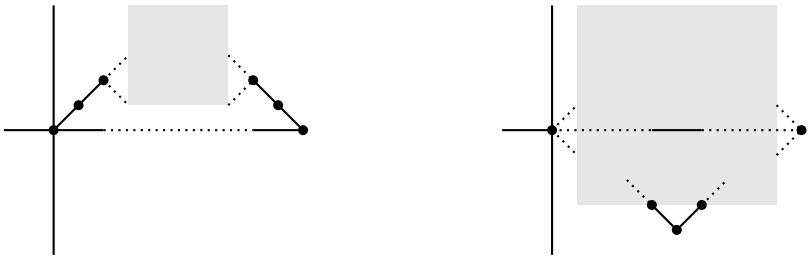
of a strictly Catalan string are 1-minimal (as this is a property enjoyed by strictly Catalan strings) with a different unique point of minimality. It follows that $x \neq y \Rightarrow \mathsf{R}(x) \blacklozenge_1 \mathsf{R}(y)$, as desired.

To ensure that $\mathsf{R}(x) \lozenge_0 \mathsf{R}(y)$, when $\mathsf{R}(x)$ and $\mathsf{R}(y)$ are balanced it suffices to exclude the possibility that $\mathsf{R}(x) = \overline{\mathsf{R}(y)}$; similarly, the number of appearances of $(0,0)$ is the same as the number of appearances of $(1,1)$, and cannot be zero unless the strings are complements. We conclude that, for two balanced strings $\mathsf{R}(x)$ and $\mathsf{R}(y)$, the condition

$$\mathsf{R}(x) \notin \{\mathsf{S}^i\overline{\mathsf{R}(y)} \mid i \in [n]\}$$

implies that $\mathsf{R}(x) \blacklozenge_0 \mathsf{R}(y)$. Observe that as string z is k-maximal if and only if $\overline{z}$ is k-minimal. Thus if $\mathsf{R}(x)$ and $\mathsf{R}(y)$ are 1-minimal (as they must be if they strictly Catalan), and 2-maximal, then $\mathsf{R}(x) \neq \overline{\mathsf{R}(y)}$. Thus $\mathsf{R}(x) \blacklozenge_0 \mathsf{R}(x)$ for all x, as desired.

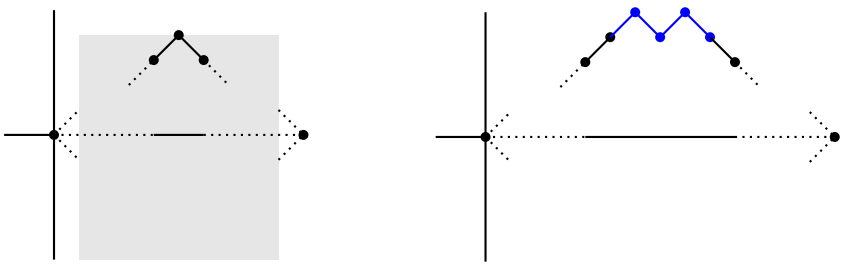
It remains to show that we can efficiently construct such a function.



(a) The graph of a strictly Catalan sequence z; remaining G_z values must lie in the shaded area.

(b) The graph of a shifted strictly Catalan sequence z. Remaining G_z values lie in the shaded region.

Figure 3: Catalan sequences.



(a) The graph of a sequence, showing a maximum value.

(b) The sequence after the transformation to 2-maximality.

Figure 4: The transformation to 2-maximality.

Our starting point shall be the "Knuth mapping" $x \mapsto \mathsf{K}(x)$ on all the binary strings; this is an efficient, injective mapping with the property that $\mathsf{K}(x)$ is balanced; moreover,

$$|\mathsf{K}(x)| \leq |x| + \log^\sharp |x| + 4\,.$$

(See [20] for further discussion.) Observe that if z is balanced, there is at least one shift $\mathsf{S}^c z$ which is Catalan. To yield an invertible process, we consider the map

$$\mathsf{U}(z) = (\mathsf{S}^c z) \circ \underbrace{1^{m/2} \circ \mathsf{K}(c_2) \circ 0^{m/2}}_{(*)},$$

where $m = |\mathsf{K}(c_2)|$. Note that the string $(*)$ is Catalan, as $\mathsf{K}(c_2)$ is balanced and hence has no more than $m/2$ zeros. It follows that $\mathsf{U}(z)$ is Catalan (as the concatenation of two Catalan strings is Catalan). Since the shift c is encoded into $\mathsf{U}(\cdot)$, the function is clearly injective. It follows that the map $z \mapsto 1 \circ \mathsf{U}(\mathsf{K}(z)) \circ 0$ is invertible, and carries z to a *strictly* Catalan image. Finally, we observe that inserting the string 1010 at any maximal point in a string z transforms it into a 2-maximal string in an invertible fashion (and preserves the other properties we care about). We let $\mathsf{M}(z)$ denote this

transformation; see Figure 4. To complete the story, we define

$$\mathsf{R}(z) \triangleq \mathsf{M}(1 \circ \mathsf{U}(\mathsf{K}(z)) \circ 0) \tag{8}$$

and observe that $|\mathsf{R}(z)| \leq |z| + 3\log^\sharp |z| + 2\log^\sharp \log^\sharp |z| + O(1)$, where the constant term is no more than 24. Since z is an edge color with length $\log^\sharp \log^\sharp n$, the theorem is proved. □

An example. Consider an example with 5 agents having access to a total of 4 channels. Similar to the example in synchronous setting, agents $1, 2, 3, 4, 5$ have access to sets of channels $\{0, 1\}, \{1, 2\}, \{2, 3\}, \{0, 3\}, \{1, 2\}$, respectively. The directed graphs in Figure 5 capture this scenario. Every node in a graph (Figure 5) represents a channel in the network and the edge connecting nodes x and y represents the agent having access to channels x and y. Edges are directed from smaller to larger nodes.

Agents are colored according to Lemma 2: $\chi(0,1) = 0, \chi(0,3) = 1, \chi(1,3) = 1, \chi(1,2) = 1, \chi(2,3) = 0$. Encoding these colors yields hopping sequences for the agents. For simplicity, in this example instead of "Knuth mapping"([20]) we use the following function to compute balanced string.

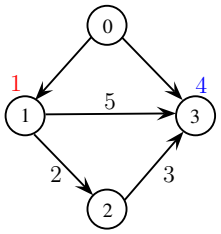
$$I(x_2) = x_2 \circ \overline{x_2}\,.$$

(Use of mapping I over "Knuth mapping" increases the length of a sequence by a constant factor.) Using I in Equation (8) yields the following encodings (Table 2). The sequence $\mathsf{R}(\chi(x, y))$ is followed by an agent which has access to channels x and y.

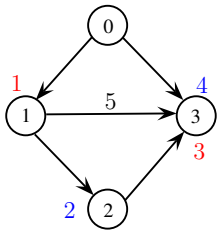
Table 2: Encodings of the 5 agents in the example

$\mathsf{R}(\chi(0,1))$	110111010000
$\mathsf{R}(\chi(0,3))$	111010010100
$\mathsf{R}(\chi(1,3))$	111010010100
$\mathsf{R}(\chi(1,2))$	111010010100
$\mathsf{R}(\chi(2,3))$	110111010000

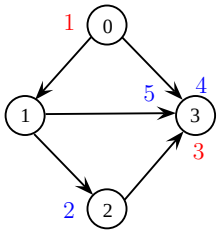
As for delays, let us assume that agents $1, 4$ start together first. Both agents 2 and 3 start 1 time slot after agent 1. Agent 5 starts 1 time slot after agent 2. Then the agents rendezvous within 8 time slots after 1 starts.



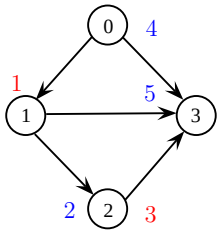
(a) 1^{st} time slot



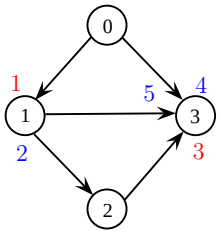
(b) 2^{nd} time slot



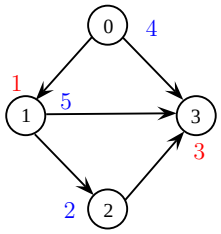
(c) 3^{rd} time slot



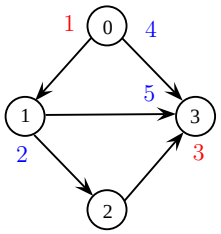
(d) 4^{th} time slot



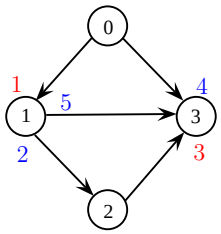
(e) 5^{th} time slot



(f) 6^{th} time slot



(g) 7^{th} time slot



(h) 8^{th} time slot

Figure 5: Example of rendezvous process in asynchronous setting. An active agent is colored depending on the 2 channels it has access to.

2.5.2 A General n-Schedule

In this section we show how to apply the previous result to yield n-schedules that provide rendezvous in time $O(|A||B| \log\log n)$. Specifically, we prove the following theorem.

Theorem 3. *There is an n-schedule so that for all overlapping $A, B \subseteq [n]$, the schedules σ_A and σ_B rendezvous asynchronously in time $O(|A||B| \log\log n)$.*

Proof. Consider a set $A = \{a_0, \ldots, a_{k-1}\}$. The schedule for A depends on a pair of primes p, p' in the range $[k, 3k]$ (there always exist two primes in this range). We then construct a schedule consisting of a sequence of *epochs*, where the rth epoch calls for

the size-2 schedule of Theorem 1 involving the two channels a_i and a_j, where $i \equiv r \mod p$ and $j \equiv r \mod p'$. (If either i or j do not fall in the range $\{0, \ldots, k-1\}$, then we choose an arbitrary element of A to fill its place.)

In the following, we will say a pair of prime numbers (p, q) is *helpful* for the rendezvous of two agents A and B if: *(i.)* p is one of the primes selected by the first agent as described above, *(ii.)* q is one of the primes selected by the second agent as described above, and *(iii.)* $p \neq q$. The construction above specifies that each agent must choose two primes to ensure that any two agents are guaranteed to have a *helpful* pair between them.

Now, suppose $A \cap B = \{c\}$, and that $c = a_x = b_y$ (so that c is the x^{th} channel in A and the y^{th} channel in B). In the synchronous model, we use the construction described in the proof of Theorem 1 to get a schedule for (a_i, a_j) in each epoch. In this case, it suffices to show that there is an epoch r satisfying $r \equiv x \pmod{p}$ and $r \equiv y \pmod{q}$, where p and q are a helpful pair as described above. According to the Chinese Remainder Theorem, there exists a solution for r that is no more than pq. Therefore, in the worst case, the two agents will both access the common channel at one time, no later than $pq(\log\log n + \log\log\log n + 2) = O(|A||B| \log\log n)$ steps after their schedules commence.

The asynchronous model requires only a slight modification. Suppose that, for a given epoch, r, an agent using the scheme described immediately above with subset A executes schedule σ_A^r of length R (all epochs have the same length). Then, we can handle asynchronous rendevous by doubling the length of each epoch and executing $\sigma_A^r \sigma_A^r$. Assume the commencement time for the σ_A is to be t_a and the commencement time for σ_B is to be t_b where, without loss of generality, $t_a \leq t_b$. Let μ denote the closest integer to $(t_b - t_a)/(2R)$. Then for any r, the r^{th} epoch of σ_A will overlap

with the $(r-\mu)^{th}$ epoch of σ_B by at least R timesteps. For any r such that $r \equiv x \pmod{p}$ and $(r-\mu) \equiv y \pmod{q}$, where the pair (p,q) is helpful, then the r^{th} epoch of σ_A will overlap with the $(r-\mu)^{th}$ epoch of σ_B no less than R timeslots. Since Theorem 1 guarantees rendezvous between σ_A^r and any cyclic shift of $\sigma_B^{r-\mu}$, this overlap must contain such a rendezvous point.

Again by the Chinese Remainder Theorem, there exists an epoch r such that $r-\mu$ is no more than pq. Therefore, in the worst case, the two agents will access the same channel in time $2pqR = O(|A||B| \log\log n)$ after t_b. □

2.5.3 A General Reduction that Guarantees Fast Symmetric Rendezvous

The rendezvous literature has given special attention to the *symmetric* case, where $A = B$. For a general schedule that guarantees rendezvous for all (perhaps distinct) pairs of sets, one specifically examines the rendezvous time in this symmetric case. In this section, we observe that any schedule that guarantees rendezvous for all pairs of sets can be transformed into one that additionally guarantees $O(1)$ rendezvous time in the symmetric case, at the expense of a constant blow-up in the rendezvous time for all other pairs of sets.

Specifically, for a family of schedules $\Sigma = (\sigma_A)_{A \subset [n]}$, for each $A \subset [n]$, we define a new schedule $\hat{\sigma}_A$ as follows: when σ_A calls for the channel c_1, $\hat{\sigma}_A$ carries out a short sequence of accesses, consisting of the channel c_1 and the channel $c_0 = \min\{A\}$ (the smallest element of A) in the pattern $c_0c_1c_0c_0c_1c_1$ repeated twice. The significance of this pattern is that

$$010011 \blacklozenge_0 010011\,;$$

thus any pair of rotations of $c_0c_1c_0c_0c_1c_1$, will yield simultaneous accesses to both (c_0, c_0) and (c_1, c_1). To ensure that there is sufficient overlap in these short sequences

of accesses, we repeat them twice: as in the proof of Theorem 3, this guarantees that a full rotation of the sequence overlaps. By a similar argument, it follows that the time to rendezvous, for any pair of sets, is no more than a constant factor (12, by this construction) larger than in Σ. However, when $A = B$, such a pair will rendezvous (at their smallest element) in constant time.

2.6 Lower Bounds

In this section, we establish that

1. $R_s(n,k) = \Omega(\log\log n)$ for any $k \leq n/2$. (Theorem 4 and Corollary 5.)
2. $R_s(n,k) \geq k^2$ for all $k = O(\log n/\log\log n)$ and, in general, $R_s(n,k) \geq \alpha k$ for all $k \leq n^{1/2\alpha}$ (so long as $\alpha \leq k$). (Theorem 6.)
3. $R_a(n,k) \geq k^2$ for all $2 \leq k \leq n/2$. (Theorem 8.)

The lower bounds provided by items 2 and 3 exhibit an enormous gap for large k and, indeed, the behavior of $R_s(n,k)$ and $R_a(n,k)$ must diverge for $k \approx \sqrt{n}$. In particular, $R_a(n,k) = \Omega(k^2)$ while there is a simple algorithm that shows that $R_s(n,k) \leq n$ for all k: each agent hops on channel t at time t when t is in the channel set, and remains silent otherwise.

Beside the above three lower bounds for deterministic algorithms, we also prove a lower bound for randomized algorithms:

4. $\mathcal{R}(k,\ell;n) = \Omega(k\ell)$ (Theorem 8.)

In the above, $\mathcal{R}(k,\ell;n)$ denotes the smallest expected rendezvous time of any randomized algorithm for a pair of agents where one has access to k channels and the other has access to ℓ.

The following sections will establish the four lower bounds individually.

2.6.1 The Dependence of Rendezvous Time on n.

We begin with two lower bounds that establish that $R_s(n,k) \to \infty$ as $n \to \infty$.

Theorem 4. *For all $n \geq 2$, $R_s(n,2) = \Omega(\log\log n)$. Rendezvous requires at least $\Omega(\log\log n)$ time, even in the synchronous model when agents are promised to have sets of size 2.*

Proof. Consider the complete graph K_n, with the interpretation that each vertex represents a channel and each edge represents a set of size two. In this case where agents correspond to two channels, we represent schedules as binary sequences, $s \in \{0,1\}^{\mathbb{N}}$, with the convention that a 0 calls for hopping on the smaller channel and 1 calls for hopping on the larger channel.

Let Σ be an $(n,2)$-schedule which guarantees rendezvous synchronously in T. In this case, we may treat each $\sigma_{(i,j)}$ as a finite length string in $\{0,1\}^T$, with the understanding that rendezvous is guaranteed before any schedule is exhausted. Treat the schedules $\sigma_{(i,j)} \in \{0,1\}^T$ as a coloring of the edges of K_n. According to a variant of Ramsey's theorem, any m-coloring of the edges of the complete graph must have a monochromatic triangle when $n \geq em!$. (See, e.g., [13].) Note, however, that a monochromatic triangle yields, in particular, an ordered triple $i < j < k$ for which the schedules associated with (i,j) and (j,k) are identical; such schedules never rendezvous . It follows that $e(2^T)! \geq n$ and, by Sterling's estimate $x! \sim \sqrt{2\pi x}(x/e)^x$ that $T = \Omega(\log\log n)$. □

Corollary 5. *For any $k \leq n/2$, $R_s(n,k) = \Omega(\log\log n)$.*

Proof. Write $[n]$ as the disjoint union of two sets $A = \{1,\ldots,m\}$ and $B = \{m+1,\ldots,n\}$, where $|B| \geq |A|(k-2) = m(k-2)$; our strategy will be to extend the sets

of size two in A to a family of subsets of $[n]$ of size k in such a way that schedules for these extended sets can be "pulled back" to schedules for the sets of size two (for which the previous lower bound applies). To proceed with this idea, we express B as a disjoint union $B = (B_1 \cup \cdots \cup B_m) \cup B_{\text{rest}}$, where each B_i has size exactly $k-2$. Now, we consider the $\binom{|A|}{2}$ sets of the form

$$X_{\{i,j\}} \triangleq \{i,j\} \cup B_{i+j \bmod m}\,,$$

where $i,j \in A$. Let Σ be an (n,k)-schedule. Observe that a schedule $\sigma_{X_{\{i,j\}}}$ for the set $X_{\{i,j\}}$ can be treated as schedule $\check{\sigma}_{\{i,j\}}$ (for $\{i,j\}$) by *restriction*, simply replacing all references to elements outside $\{i,j\}$ with, say, the smaller of i and j. In general, restriction of an (n,k)-schedule to an (n,ℓ)-schedule (for $\ell < k$) does not provide any guarantee on rendezvous, even when the original (n,k)-schedule does. However, the intersection pattern of the sets $X_{\{i,j\}}$ above is chosen in such a way that the $(m,2)$-schedule $\check{\Sigma}$ obtained by defining $\check{\sigma}_{i,j}$ to be the restriction of the schedule $\sigma_{X_{\{i,j\}}}$ will guarantee rendezvous.

Consider two subsets $\{i,j\}$ and $\{i',j'\}$ of A, each of size two. If these two sets are not identical but share a common element, it follows that $i+j \bmod m \neq i'+j' \bmod m$. Thus,

$$B_{i+j \bmod m} \cap B_{i'+j' \bmod m} = \emptyset$$

and

$$X_{\{i,j\}} \cap X_{\{i',j'\}} = \{i,j\} \cap \{i',j'\}$$

If $\sigma_{X_{i,j}}$ and $\sigma_{X_{i',j'}}$ rendezvous, this must occur at a channel in $\{i,j\} \cap \{i',j'\}$, and it follows that the rendezvous time of the schedule Σ is at least that of the schedule $\check{\Sigma}$;

we conclude that $R(n,k) \geq R(m,2)$ so long as $n \geq m + m(k-2) = m(k-1)$. Thus

$$R(n,k) \geq R(\lfloor (n/(k-1) \rfloor, 2) = \Omega(\log \log n/k)\,.$$

However, it is clear that $R_s(n,k) \geq k$ for all $k \leq n/2$, so the bound above is only relevant when $k = \Omega(\log \log n)$ which yields a $\Omega(\log \log n)$ lower bound for all k. $\square$

2.6.2 The Dependence of Rendezvous Time on k in the Synchronous Setting

Theorem 6. *Let $1 \leq \alpha \leq k$ and $k \leq n^{1/(2\alpha)}$. Then $R_s(n,k) \geq k\alpha$. In particular, for $k = O(\log n/\log \log n)$,*

$$R_s(n,k) \geq k^2\,.$$

Proof. Let Σ be an (n,k)-schedule. Partition the n channels into n/k disjoint subsets, $S_1, \ldots, S_{n/k}$, each of size k. Suppose, for the sake of contradiction, that Σ guarantees rendezvous synchronously in less than αk. In this case, we focus only on the first $\alpha k - 1$ time slots of the schedules and treat each σ_A as a function defined on $\{1, \ldots, \alpha k - 1\}$.

For each $i \in \{1, \ldots, n/k\}$, let σ_i denote the schedule of subset S_i and observe that some $a_i \in S_i$ must appear fewer than $\alpha \leq k$ times in the schedule. Letting $\sigma_i^{-1}(a_i) \subseteq \{1, 2, \cdots, \alpha k - 1\}$ denote the set of time indices at which a_i appears in σ_i, we then have $|\sigma_i^{-1}(a_i)| < \alpha$. By possibly adding some elements to the set $\sigma_i^{-1}(a_i)$, we may construct a set A_i, containing $\sigma_i^{-1}(a_i)$, of size exactly $\alpha - 1$. Observe that there are $\binom{\alpha k - 1}{\alpha - 1}$ possible values (subsets) that these A_i can assume.

If n/k, the number of disjoint subsets in our original partition, exceeds $(k-1) \cdot \binom{\alpha k - 1}{\alpha - 1}$, then there must be at least k of these subsets, say $S_{i_1}, \ldots, S_{i_k}$, for which

$$A_{i_1} = \cdots = A_{i_k} = Z\,,$$

for a set Z of size $\alpha - 1 < k$; it follows that $\sigma_{i_j}^{-1}(a_{i_j}) \subset Z$ for each i.

Finally, let

$$\hat{S} = \{a_{i_1}, \ldots, a_{i_k}\}$$

and let $\hat{\sigma}$ be its schedule in Σ. For any $j \in \{1, \ldots, k\}$, $\hat{\sigma}$ must rendezvous with σ_{i_j}, which requires that $\hat{\sigma}^{-1}(a_{i_j}) \cap Z \neq \emptyset$. As the $\hat{\sigma}^{-1}(a_{i_j})$ are disjoint, this implies that $|Z| \geq k$, a contradiction. To satisfy the condition that

$$n/k > (k-1)\binom{(\alpha-1)k}{\alpha-1},$$

it suffices for $n \geq k^{2\alpha}$, where we have applied the coarse bound

$$\binom{\alpha k - 1}{(\alpha - 1)} \leq \binom{k^2}{\alpha - 1} \leq k^{2(\alpha-1)}.$$

□

2.6.3 A Stronger Lower Bound in the Asynchronous Model

Finally, we show that in the asynchronous model, it is possible to extend the k^2 lower bound to all k less than $n/2$. In fact, we show that in any n-schedule, for any k and ℓ with $k + \ell \leq n + 1$ there are sets of size k and ℓ that cannot rendezvous asynchronously in time less than $k\ell$.

Theorem 7. *For all $k \leq n/2$, $R_a(n, k) \geq k^2$. Moreover, for any n-schedule and any k and ℓ for which $k + \ell \leq n + 1$, there are sets of size k and ℓ that require at least $k\ell$ steps to rendezvous in the asynchronous model.*

Proof. Let Σ be an n-schedule. We will show that there exist two subsets, A and B, such that $|A| = k$, $|B| = \ell$, $|A \cap B| = 1$, and σ_A and σ_B require at least $k\ell$ time steps to rendezvous in the asynchronous model. First, consider uniformly random selection

of $A, B \subset [n]$ according to the following process: *(i.)* select A uniformly among all the sets of size k, *(ii.)* select a channel h uniformly from A, and *(iii.)* select B' uniformly at random from all subsets of $[n] \setminus A$ of size $\ell - 1$ and define $B = B' \cup \{h\}$. We remark that reversing the roles of A and B in the above process (initially selecting B uniformly among all sets of size ℓ, selecting h from B, and selecting A by adding $k-1$ random elements of $[n] \setminus A$ to $\{h\}$) yields the same probability distribution on (A, B).

We let $\Delta(h, \sigma; T)$ denote the density of occurrences of h during the first T time steps in schedule σ:

$$\Delta(h, \sigma; T) \triangleq \frac{|\{t \in [0, T) \mid \sigma(t) = h\}|}{T}\,.$$

(Here the notation $[0, T)$ denotes $\{0, 1, \ldots, T-1\}$) . For any length-T prefix of the schedule σ_A for A, note that

$$\begin{aligned}\mathop{\mathbb{E}}_{A, h \in A}[\Delta(h, \sigma_A; T)] &= \mathop{\mathbb{E}}_{A}\left[\sum_{x \in A} \Pr(h = x)\Delta(x, \sigma_A; T)\right] \\ &= \mathop{\mathbb{E}}_{A}\left[\frac{1}{k}\sum_{x \in A}\Delta(x, \sigma_A; T)\right] = \frac{1}{k}\,.\end{aligned}$$

(Here $\mathbb{E}[\cdot]$ denotes expectation). Likewise, considering the reversed procedure for selecting A and B, for any T' we have $\mathbb{E}[\Delta(h, \sigma_B; T')] = 1/\ell$. By linearity of expectation, for any T, T',

$$\mathop{\mathbb{E}}_{A, B, h}[k \cdot \Delta(h, \sigma_A; T) + \ell \cdot \Delta(h, \sigma_B; T')] = 2\,. \tag{9}$$

Let r be the minimum integer so that all intersecting subsets, A and B of sizes $|A| = k$ and $|B| = \ell$, intersect in time r; let $R \gg r$. From the expectation calculation (12), it follows that there exist two sets, A and B, intersecting at an unique element h, for which $k\Delta(h, \sigma_A; R) + \ell\Delta(h, \sigma_B; r) \leq 2$. Observe then that the product

$$k\Delta(h, \sigma_A; R) \cdot \ell\Delta(h, \sigma_B; r) \leq 1$$

and hence $\Delta(h, \sigma_A; R) \cdot \Delta(h, \sigma_B; r) \leq 1/k\ell$.

Consider, finally, the circumstances when the schedule σ_A starts at time 0 and the schedule σ_B starts at some time $t \in [0, R-r]$. Let

$$P = \{(x, y) \in [0, R) \times [0, r) \mid \sigma_A(x) = \sigma_B(y) = h, x \geq y\}.$$

Each such pair (x, y) is a possible rendezvous point which can occur only if σ_B starts at time $x - y$. We have

$$|P| \leq R \cdot \Delta(h, \sigma_A; R) \cdot r \cdot \Delta(h, \sigma_B; r) \leq \frac{R \cdot r}{k\ell}.$$

As rendezvous is guaranteed in the range $[t, t + r)$ for any $t \in [0, R - r]$, we must have $|P| \geq R - r$ (otherwise, there is a time that is not covered by any rendezvous pair of P), which implies that $R \cdot r/k\ell \geq R - r$ and, therefore,

$$r \geq \frac{R - r}{R} \cdot k\ell.$$

As $R \to \infty$, this quantity approaches $k\ell$. □

2.6.4 A tight lower bound for randomized asynchronous rendezvous

We consider a randomized algorithm R for rendezvous: given a subset $S \subset [n]$, the algorithm R must generate a schedule $\mathsf{R}_S \triangleq \mathsf{R}_S(1), \mathsf{R}_S(2), \ldots \in S$. The sequence R_S is a random variable, which we treat as a function of an infinite random string $r \in \{0, 1\}^{\mathbb{N}}$ of independent, uniform bits. When we wish to emphasize the dependence on r, we write R_S^r.

Our goal is to lower bound the expected time to rendezvous for such randomized algorithms. More precisely, we wish to lower bound the following:

$$\mathcal{R}(k, \ell; n) \triangleq \min_{\mathsf{R}} \max_{\substack{A,B: \\ |A|=k, \\ |B|=\ell}} \max_{w} \mathop{\mathbb{E}}_{r,r'} [\min_{t} \mathsf{R}_A^r(t + w) = \mathsf{R}_B^{r'}(t)].$$

Theorem 8. *For any k, ℓ for which $k + \ell \leq n + 1$, $\mathcal{R}(k, \ell; n) = \Omega(k\ell)$.*

Proof. Fixing k, ℓ, and n as in the statement of the theorem, we proceed via Yao's principle [37]: specifically, we identify a particular distribution on subsets A and B (of appropriate size) and the "shift" $w \in \mathbb{N}$ for which any *deterministic* algorithm must take $\Omega(k\ell)$ time to rendezvous (in expectation over A, B, and w). By exchanging the expectation operators, this yields a lower bound for the expected rendezvous time of randomized algorithms.

The random variables A and B are defined to be uniformly drawn from all pairs of subsets of $[n]$ for which $|A| = k$, $|B| = \ell$, and $|A \cap B| = 1$. It will be convenient in the proof to consider a specific procedure for selecting A and B: *(i.)* A is selected uniformly at random among all subsets of size k, *(ii.)* a single channel h is selected uniformly from A, *(iii.)* a set B' is selected uniformly at random from all subsets of $[n] \setminus A$ of size $\ell - 1$ and, finally, B is defined to be $B' \cup \{h\}$. We remark that the reversing roles of A and B in the above process (that is, initially selecting B uniformly among all sets of size ℓ, selecting h from B, etc.) yields the same probability distribution on (A, B). The random variable w is selected uniformly in the set $[0, R) \triangleq \{0, 1, \ldots, R - 1\}$ for a large value R discussed later in the proof.

By Yao's principle,

$$\begin{aligned}
\mathcal{R}(k, \ell; n) &\geq \min_{\mathsf{R}} \mathop{\mathbb{E}}_{A,B,w} \left[\mathop{\mathbb{E}}_{r,r'} \left[\min_t \mathsf{R}_A^r(t + w) = \mathsf{R}_B^{r'}(t) \right] \right] \\
&= \min_{\mathsf{R}} \mathop{\mathbb{E}}_{r,r'} \mathop{\mathbb{E}}_{A,B,w} \left[\min_t \mathsf{R}_A^r(t + w) = \mathsf{R}_B^{r'}(t)] \right] \\
&\geq \min_{r,r'} \mathop{\mathbb{E}}_{A,B,w} \left[\min_t \mathsf{R}_A^r(t + w) = \mathsf{R}_B^{r'}(t) \right] \\
&= \min_{\sigma,\tau} \underbrace{\mathop{\mathbb{E}}_{A,B,w} \left[\min_t \sigma_A(t + w) = \tau_B(t) \right]}_{(*)},
\end{aligned}$$

where σ and τ, in the last line above, denote a pair of deterministic rendevous schedules. The remainder of the proof fixes σ and τ, and establishes a lower bound for the quantity $(*)$ above.

Recall that for any nonnegative, integer-valued random variable X we may expand $\mathbb{E}[X] = \sum_{r \geq 0} \Pr[X > r]$ and hence

$$\mathop{\mathbb{E}}_{A,B,w}\left[\min_t \sigma_A(t+w) = \tau_B(t)\right] = \sum_{r=0}^{\infty} \Pr_{A,B,w}\underbrace{\left[\forall t \leq r, \sigma_A(t+w) \neq \tau_B(t)\right]}_{E_r}. \tag{10}$$

In light of this, we focus on the probability of the event E_r for $r \geq 0$. Define

$$P_{A,B}^{(r)} = \{(x,y) \in [0, R+r) \times [0,r) \mid \sigma_A(x) = \tau_B(y) = h, x \geq y\}.$$

Conditioned on a particular value for A and B, each pair $(x,y) \in P_{A,B}^{(r)}$ is a possible rendezvous point which can occur only if σ_A starts at time $w = x - y$. In particular, conditioned on a particular value of A and B,

$$\Pr_w\left[\overline{E_r}\right] = \Pr_w\left[\exists t \leq r, \sigma_A(t+w) = \tau_B(t)\right] \leq \left|P_{A,B}^{(r)}\right| / R. \tag{11}$$

To control $\left|P_{A,B}^{(r)}\right|$, we introduce some further notation: For a (deterministic) schedule ρ, we let $\Delta(c, \rho; T)$ denote the density of occurrences of c during the first T time steps in schedule ρ:

$$\Delta(c, \rho; T) \triangleq \frac{|\{t \in [0,T) \mid \rho(t) = c\}|}{T}.$$

Then it is immediate that

$$\begin{aligned}\frac{|P_{A,B}^{(r)}|}{R} &\leq \frac{(R+r) \cdot \Delta(h, \sigma_A; R+r) \cdot r \cdot \Delta(h, \tau_B; r)}{R} \\ &= \left(\frac{R+r}{R}\right) \cdot \Delta(h, \sigma_A; R+r) \cdot r \cdot \Delta(h, \tau_B; r).\end{aligned}$$

For any length-T prefix of the schedule σ_A, note that

$$\mathop{\mathbb{E}}_{h \in A}[\Delta(h, \sigma_A; T)] = \sum_{a \in A} \Pr(h = a)\Delta(a, \sigma_A; T) = \frac{1}{k}\sum_{a \in A} \Delta(a, \sigma_A; T) = \frac{1}{k}$$

and hence $\mathbb{E}_{A,h}[\Delta(h,\sigma_A;T) = 1/k$. Likewise, considering the reversed procedure for selecting A and B, for any T' we have $\mathbb{E}_{B,h}[\Delta(h,\tau_B;T')] = 1/\ell$. By linearity of expectation, for any T, T',

$$\mathop{\mathbb{E}}_{A,B}[k \cdot \Delta(h,\sigma_A;T) + \ell \cdot \Delta(h,\tau_B;T')] = 2 \tag{12}$$

and, by Markov's inequality, $\Pr_{A,B}[k \cdot \Delta(h,\sigma_A;T) + \ell \cdot \Delta(h,\tau_B;T') > \alpha] \leq 2/\alpha$ for any $\alpha > 0$. When $k \cdot \Delta(h,\sigma_A;T) + \ell \cdot \Delta(h,\tau_B;T') \leq \alpha$ it follows that $\Delta(h,\sigma_A;T) \cdot \Delta(h,\tau_B;T') \leq \alpha^2/(4k\ell)$, and we conclude that for any $\alpha > 0$,

$$\Pr_{A,B}\left[\Delta(h,\sigma_A;T) \cdot \Delta(h,\tau_B;T') > \frac{\alpha^2}{4k\ell}\right] \leq \frac{2}{\alpha} \tag{13}$$

and hence

$$\Pr_{A,B}\left[\frac{\left|P^{(r)}_{A,B}\right|}{R} > \frac{r\alpha^2}{4k\ell} \cdot \left(\frac{R+r}{R}\right)\right] \leq \frac{2}{\alpha}\,. \tag{14}$$

In the event that $|P^{(r)}_{A,B}|/R$ is large (in selection of A and B), we may assume that $\Pr[\overline{E_r}] = 1$ conditioned on this choice of A and B. In light of (11), this results in the bound

$$\begin{aligned}\Pr_{A,B,w}\left[\overline{E_r}\right] &\leq \frac{2}{\alpha} + \left(1 - \frac{2}{\alpha}\right)\frac{r\alpha^2}{4k\ell} \cdot \left(\frac{R+r}{R}\right) \leq \left[\frac{2}{\alpha} + \left(1 - \frac{2}{\alpha}\right)\frac{r\alpha^2}{4k\ell}\right] \cdot \left(\frac{R+r}{R}\right) \\ &\leq \sqrt[3]{\frac{r}{k\ell}}\left(2 - \sqrt[3]{\frac{r}{k\ell}}\right) \cdot \left(\frac{R+r}{R}\right),\end{aligned}$$

where the last inequality follows by assuming that $r \leq k\ell$ and choosing $2/\alpha = \sqrt[3]{r/k\ell}$. Finally, returning to (10),

$$\begin{aligned}
\mathcal{R}(k,\ell;n) &\geq \lim_{R\to\infty}\left[\mathop{\mathbb{E}}_{A,B,w}\left[\min_t \sigma_A(t+w)=\tau_B(t)\right]\right] \\
&\geq \lim_{R\to\infty}\left[\sum_{r=0}^{k\ell}\left(1-\sqrt[3]{\frac{r}{k\ell}}\left(2-\sqrt[3]{\frac{r}{k\ell}}\right)\cdot\left(\frac{R+r}{R}\right)\right)\right] \\
&= \sum_{r=0}^{k\ell}\left(1-\sqrt[3]{\frac{r}{k\ell}}\left(2-\sqrt[3]{\frac{r}{k\ell}}\right)\right) \\
&\geq k\ell\int_0^1 1-\sqrt[3]{r}\left(2-\sqrt[3]{r}\right)\,dr = \frac{k\ell}{10}\,. \qquad \square
\end{aligned}$$

Remark.

By somewhat complicating the proof, one can optimize the selection of $\delta \triangleq 2/\alpha$ further by computing an analytic expression for the positive root of the polynomial $\delta^3 + (1-\delta)\delta^{-2}(r/k\ell) - 2(r/k\ell) = 0$. The final integral then yields the lower bound $(.105922...)k\ell$.

2.7 Rendezvous with a One-Bit Beacon

In this section we consider the rendezvous problem when the agents are supplied with a "one-bit random beacon." Specifically, we work under the assumption that the agents exist in an environment that supplies them with a (common) uniformly random bit $c_t \in \{0,1\}$ during each time step t; we assume that the c_t are independent (for different t) and available to all agents. We remark that random beacons have been studied in a number of related models [11, 28] and–in practice–beacons are available, e.g., for GPS receivers in close proximity [22, 34].

We shall see that augmenting the basic model with a one-bit beacon can dramatically reduce the rendezvous time: in particular, with a one-bit beacon, (asynchronous)

rendezvous is possible with high probability in time

$$O\left(\log^2 n \cdot \frac{|S_i \cup S_j|}{|S_i \cap S_j|}\right) .$$

. (In contrast, asynchronous rendezvous, without such a beacon, requires time $\Omega(|S_i||S_j|)$.)

For a number n, we let $\mathfrak{S}_n$ denote the set of all permutations of the elements $\{1, \ldots, n\}$, the set of channels. The schedule for an agent i with available channels S_i is constructed as follows:

- At time t, the sequence $c_1, \ldots, c_t$ is used to determine a permutation $\pi_t \in \mathfrak{S}_n$. (We write $\pi_t = \Pi(c_1 \ldots c_t)$, and discuss below various choices for the function Π.)
- The agent hops on the channel $\arg\min_{a \in S_i} \pi_t(a)$, which is to say that the agent hops on the channel that maps to the smallest element of $\{1, \ldots, n\}$ under the permutation π_t.

It remains to describe Π, the rule that determines the permutation π_t from the sequence $c_1, \ldots, c_t$. For this purpose, we recall the notion of a *min-wise family of permutations.*

Definition 1. *We say that a subset $R \subset \mathfrak{S}_n$ is* ϵ-min-wise independent *if, for every subset $A \subset \{1, \ldots, n\}$ and every element $a \in A$,*

$$\Pr_{\pi \in R}[\pi(a) = \min\{\pi(a') \mid a' \in A\}] \geq \frac{1}{|A|}(1 - \epsilon) .$$

(Here π is given the uniform distribution in R.)

For any n and ϵ, Indyk [17] gave an efficient construction of a family of ϵ-minwise independent permutations that can be represented with $O(\log n \cdot \log 1/\epsilon)$ bits. In our setting, it suffices to set $\epsilon = 1/2$; for the remainder of this section, we let R_n denote a family of 1/2-minwise independent permutations in $\mathfrak{S}_n$. Note that $d \log n$ bits are required to represent an element in R_n, for a fixed constant d.

Consider now two sets of channels S_i and S_j and an element $\alpha \in S_i \cap S_j$. If π is a permutation drawn at random from R_n, then

$$\begin{aligned}&\Pr\big[\alpha = \arg\min_{a \in S_i} \pi(a) = \arg\min_{a' \in S_j} \pi(a')\big] \\ =&\Pr\big[\alpha = \arg\min_{a \in S_i \cup S_j} \pi(a)\big] \geq \frac{1}{2|S_i \cup S_j|}\,. \end{aligned} \tag{15}$$

A $O(\log n \cdot |S_i \cup S_j|/|S_j \cap S_j|)$ ***rendezvous protocol.*** Let us consider the protocol induced by defining $\Pi(c_1 \ldots c_t)$ to be the permutation from R_n determined by the last $d \log n$ bits of $c_1 \ldots c_t$. At times $d \log n, 2d \log n, \ldots, Td \log n$, these selections from R_n are independent. In light of (15), the probability that each of these permutations failed to induce rendezvous is no more than

$$\left(1 - \frac{|S_i \cap S_j|}{2|S_i \cup S_j|}\right)^T \leq e^{-T|S_i \cap S_j|/(2|S_i \cup S_j|)}\,.$$

It follows that for $T = 2\alpha|S_i \cup S_j|/|S_i \cap S_j|$, the probability that this protocol fails to rendezvous is no more than $e^{-\alpha}$, as desired. For this T, no more than $Td \log n = O(\alpha \log n \cdot |S_i \cup S_j|/|S_i \cap S_j|)$ time has passed. Note that by choosing $\alpha = r \ln n$ for constant r, we achieve rendezvous with probability $1 - 1/n^r$ in time $O(\log^2 n \cdot |S_i \cup S_j|/|S_i \cap S_j|)$.

An $O(|S_i \cup S_j|/|S_i \cap S_j| + \log n)$ ***synchronous rendezvous protocol.*** In the synchronous setting, where the nodes arrive at the same time and have synchronized clocks, we can improve upon this protocol by applying *deterministic amplification*. The protocol described above uses $O(\log n)$ independent random bits to produce each independent element of R_n. By "walking on an expander graph," one can achieve the same performance guarantees with only $O(|S_i| + |S_j| + \log n)$ random bits. Specifically, one associates the elements of the set R_n with the vertices of a constant-degree expander graph and generates a collection of elements of R_n by the following process: the first

$d \log n$ bits of c_i are used to generate a random element of the expander graph (and, hence, an element of R_n); each subsequent element of R_n is generated by using $O(1)$ bits of the string $c_1, c_2, \ldots$ to take one step in the natural random walk on the graph. This yields the following bound on the failure probability (that is, the probability that no rendezvous has taken place) after k steps:

$$\left(1 - (1-\lambda)\frac{|S_i \cap S_j|}{|S_i \cup S_j|}\right)^k \leq e^{-k(1-\lambda)|S_i \cap S_j|/|S_i \cup S_j|},$$

where λ is the second eigenvalue of the expander graph. Applying a constant degree expander, λ is a constant less than 1 and we find that for $k = \alpha|S_i \cup S_j|/(|S_i \cap S_j|(1-\lambda))$, the probability of failure is no more then $e^{-\alpha}$. This requires $O(k + \log n) = O(\log n + \alpha|S_i \cup S_j|/|S_i \cap S_j|)$ bits. See [15] for a survey of these techniques and, in particular, a description of this particular form of deterministic amplification.

Chapter 3

Energy Constrained Single Channel Rendezvous in Wireless Networks

3.1 Introduction

In this chapter, we consider a single channel rendezvous problem, in which each node's radio only has two statuses: on and off. It may sound easier than multi-channel rendezvous problem, however, we have an energy constraint in this problem. In this chapter, we focus on the *duty-cycled* case, where each node is equipped with a (single-channel) transceiver which may only be active for a small fraction (which we denote $d > 0$) of the time; for our purposes, a network node is characterized by its *schedule*, which determines when its transceiver is active. Two nodes are said to *discover* each other when their transceivers are both active. This energy constrained single channel

rendezvous problem is usually called the neighbor discovery problem, which is a fundamental issue in many wireless networking environments, and has been the subject of intense study in a variety of theoretical and practical settings.

While duty-cycled discovery can be studied in a variety of natural models, results that agree with practice can be obtained even in extremely weak models, which has led existing work to focus on the *deterministic, anonymous, and asynchronous* setting. Such a model is quite weak (and therefore widely applicable), yet at the same time permits rapid discovery consistent with practice. Specifically, *nodes are deterministic*: the schedules must be determined in advance, and may not rely on any randomness; *nodes are anonymous*: nodes do not have the luxury of individual identities; and *nodes' arrivals and clocks are asynchronous*: nodes may arrive at any time, and do not have the luxury of synchronized clocks. Finally, the model is characterized by a hardware-dependent *switching-interval* $\Delta > 0$: once a node has switched from an active to passive state, or vice versa, it must remain in this new state for at least a Δ time period. The goal is to design schedules that guarantee discovery as quickly as possible.

A few remarks about these assumptions are in order. Without the switching-interval constraint, it is possible to design schedules that rendezvous arbitrarily quickly; in the literature, the quantity Δ is often set to 1 by convention; see below. In typical settings, nodes actually use a "chirp-and-listen" protocol which sends a short chirp at the beginning and end of any interval of activity, and simply listens for the remainder of the interval. This can provide the simple notion of discovery we describe above (where nodes discover each other if their intervals of activity overlap). Finally, randomness appears to be quite powerful in this setting, especially considering that nodes are not permitted to have identities; however, as we see below, it is possible to achieve—deterministically—discovery times that beat the natural randomized protocols. Furthermore, deterministic

schedules have other significant advantages for network maintenance—they allow nodes that have discovered each other to predict when each other will be active in the future.

Roadmap

The rest of the chapter is organized as follows. Section 3.2 introduces the basic model. Section 3.3 summarizes our contribution. Section 3.4 briefly describes related work. Section 3.5 presents the two optimal constructions for the basic integer model. Section 3.6 presents the generalized model, a lower bound on discovery time, and a generalized reduction that converts optimal schedules in the basic integer models to optimal schedules in the generalized model.

3.2 The Basic Integer Model

Most previous literature on neighbor discovery considers schedules obtained by *discretizing* time into blocks of width Δ; a schedule can then be constructed by calling for a node to be *awake* during some of these blocks and *asleep* during the others. If the blocks are given the names $0, 1, 2, \ldots$, a schedule is simply determined by a subset of $\mathbb{N}$, the natural numbers; we call such a schedule an *integer schedule*. To further simply matters, previous work focuses exclusively on *periodic* integer schedules which repeat with some positive period n and so can be represented as a subset $S \subset \mathbb{Z}_n$, the integers modulo n. For such a periodic schedule, we say that a schedule has duty cycle d when $|S|/n \leq d$. With very few exceptions, we simply treat S as a subset of $\mathbb{Z}_n$; in the rare case that we wish to explicitly refer to the associated subset of $\mathbb{R}$, we shall write $S_{\mathbb{R}}$ to denote the union of Δ-width intervals

$$\bigcup_{\substack{z \in \mathbb{Z} \\ z \bmod n \in S}} [z\Delta,\ (z+1)\Delta)\,.$$

Our goal, given a duty cycle d, is to design a schedule $S \subset \mathbb{Z}_n$, for some appropriate n, that minimizes the *discovery latency*. To be more precise, consider a pair of nodes using the same schedule $S \subset \mathbb{Z}_n$, the second of which arrives t units of time after the first has begun its schedule. While the schedules treat time in blocks of width Δ, time is otherwise treated as continuous, so that t may be any positive real value. The discovery time for this pair is the least time during which both are active:

$$L(S;t) = \min_{\ell \geq 0} \quad \text{such that} \quad \ell \bmod n \in S_{\mathbb{R}} \quad \text{and} \quad \ell + t \bmod n \in S_{\mathbb{R}}\,.$$

Here, the notation $a \bmod n$, for a real value a, denotes the non-negative (real) remainder upon division by n (that is to say that if $a = qn + r$, where $q \in \mathbb{Z}$ and $0 \leq r < n$, we have $r = a \bmod n$). It is easy to see that—as S has period n—we may assume without loss of generality that $t, \ell < n$ as well. To restate our goal: we wish design S so as to minimize this quantity (for the worst case t). Specifically, we wish to design S to minimize

$$L(S) = \max_{t \in [0,n)} L(S;t)\,,$$

and call this the *discovery latency* or *discovery time* of the schedule S. Of course, the quantity $L(S)$ is undefined for many sets S, as there are shifts that do not give rise to any overlap at all; in this case, we write $L(S) = \infty$. On the other hand, when $L(S)$ is finite, it is never more than n (the period of S), and the basic strategy in the literature (which we adopt as well) simply establishes that $L(S) < \infty$ and applies the bound $L(S) \leq n$.

We make one final remark about such schedules, which further simplifies reasoning about them.

Principle 9 (The Integer Intersection Principle)**.** *Let S be an integer schedule with period n. Then $L(S)$ is finite if and only if $L(S;t)$ is finite for all* integer *times $t \in \{0, \ldots, n-1\}$.*

Thus, it suffices to ensure, for a schedule $S \subset \mathbb{Z}_n$, that for any *integer* t there is an integer ℓ so that $\ell \in S$ and $\ell + t \in S$. Equivalently, it suffices to show that for all $t \in \mathbb{Z}_n$, $S \cap (S - t) \neq \emptyset$, where $S - t = \{s - t \bmod n \mid s \in S\}$. In this case, we can immediately conclude that $L(S) \leq n$ by the integer intersection principle. We discuss an amplification of the principle in Section 3.6, where we discuss non-integer schedules in detail. For Sections 3.4 and 3.5, however, we will focus entirely on integer schedules (and integer times t, which suffices by the principle above). Finally, we remark that in this chapter we focus exclusively on this *homogeneous* setting, where nodes have the same duty cycle d, and hence—as they do not have identities—use the same schedules.

The state-of-the-art

With $\Delta = 1$ as above, prior work [41] has established a $1/d^2$ lower bound on discovery time, and a sequence of studies have established schedules that achieve c/d^2 discovery time for various constants c. The best existing result [19] established schedules that achieve discovery time $2.25/d^2$, a factor of 2.25 over the the lower bound. While it is known [21] that schedules *exist* that achieve the lower bound, the only known construction takes time exponential in $1/d$, and so is impractical except for very large duty-cycles.

Recent work [4] has begun to explore non-integer schedules, demonstrating that these can provide significant improvement. For this case, neither a tight lower bound on discovery time nor optimal schedule have been developed in the literature.

3.3 Our Contributions

In this chapter we give the first efficient, optimal schedules for discovery in the basic integer model, discussed above. That is, we give schedules S which achieve $L(S) = (1 + o(1)) \cdot \Delta/d^2$, for a quite dense family of d. We also develop a generalized model that allows us to place existing work with various assumptions into a single setting, and establish a new family of lower bounds for the model and a connection to the integer model.

Specifically, in the basic integer model, we give two constructions that lead to optimal schedules. Both constructions are based on the generation, given the parameter d, of an appropriate *difference set* which is guaranteed to intersect with any rotation of ifself. One is based on a novel application of the theory of *Sidon sets* and the other is based on the *Singer construction*. Both lead to practical, polynomial-time optimal algorithms for neighbor discovery.

Our generalized model is motivated by the observation that the basic integer model needlessly constrains switching to occur on integer boundaries and, additionally, does not reflect an important constraint that arises in practice. In our proposed, generalized model, time is continuous: nodes may become active or inactive at any point in time, subject to a few constraints. In addition to the switching latency Δ, we also introduce a *required meeting time* δ: to discover one another, two nodes must be simultaneously awake for a continuous interval of duration at least δ. This generalized model permits unified treatment of the assumptions in existing studies: the basic integer model arises as when $\delta = \Delta/2$; the model of [4] arises when $\delta \ll \Delta$. We provide a lower bound on the best achievable latency guarantee under this new model (which reduces to the lower bound for the basic integer model when $\delta = \Delta/2$). In addition, we provide a

reduction that can transform any schedule in the basic integer model to a schedule in the generalized model. Applying this reduction, our optimal schedules under the basic integer model become optimal schedules under the generalized model.

3.4 Related Work

Most existing schemes for neighbor discovery adopt the basic integer model described above. Earlier studies (e.g., [26]) developed probabilistic solutions, which establish discovery in expected time $1/d^2$, and hence discovery with probability $1-\epsilon$ in time $\log(\epsilon^{-1})/d^2$. As mentioned in the introduction, deterministic schedules can likewise achieve such bounds—with certainty—and provide other benefits, which has prompted recent studies to focus on deterministic solutions. Zheng et al. [41] have given (tight) lower bounds on the duty cycle required for discovery in time T equivalent to the $L(S) \geq 1/d^2$ bound described above. Dutta and Culler [10] provide a simple schedule where, for duty cycle d, each node selects two primes p_1, p_2 so that $1/p_1 + 1/p_2 \approx d$ and wakes up in time slots t satisfying $t \equiv 0 \mod p_1$ or $t \equiv 0 \mod p_2$. By virtue of the Chinese Remainder Theorem, this guarantees two nodes discover each other within one cycle of length approximately $4/d^2$ (i.e., $L(S) \leq 4/d^2$). Kandhalu et al. [19] improves the above scheme to achieve worst-case discovery time $L(S) \leq 2.25/d^2$. Lai et al. [21] expands on the idea of *cyclic quorum systems* that has been used in earlier studies [16, 33]. They present a construction based on *difference pairs*, which allows optimal discovery latency of $1/d^2$. Their construction, however, takes exponential time in $1/d$, and so is impractical except for very large d. Our study, for the first time, proposes optimal and practical optimal schedules under the basic integer model. We remark that our schedules, like those of previous work, actually only yield constructions for certain values of d (with certain number-theoretic properties); as these are fairly dense, they

are satisfactory for practice. They can be applied to general d by padding, in which case analysis of the worst case leading constant requires detailed analysis of the density of the relevant d.

Recently, Bakht et al. [4] break away from integer schedules, and show that non-integer schedules can significantly reduce the discovery latency. Our work further improves upon their results: we propose a generalized model that encompasses their assumptions, derive a lower bound in this model, and provide a reduction that transforms optimal schedules for the basic integer model into optimal schedules for the generalized model.

Dolev et al. [9], Bradonjic et al. [7], and Barenboim et al. [5] consider the related problem of "clock synchronization" in wireless networks. In this setting, the network contains m nodes and the objective is for *all* nodes to rendezvous at the same time with a minimum of network activity. Bradonjic et al. [7] and Barenboim et al. [5] deal with a network model very similar to the one considered in this paper. Dolev et al. [9] considers a very general model, in which each node can utilize any of a set of radio frequencies, and may need to deal with radio interference on each channel. Neighbor discovery is a crucial component of clock synchronization protocols, so the results given in these papers implicitly incorporate neighbor discovery protocols. There is no duty cycle, but the protocols strive to minimize the number of active time slots while guaranteeing discovery. Protocols from both [7] and [5] guarantee neighbor discovery using no more than $4\sqrt{n}$ active time steps , where n is the discrepancy (in steps) between the starting times of the two nodes.

3.5 Optimal Schedules in the Basic Integer Model

In this section, we present two constructions that lead to optimal schedules in the basic integer model. For ease of exposition, we regard the step length, Δ, as one unit of time and approach the dual problem: fixing a period n, how can one schedule a node's wake-up slots so that the schedule intersects with every rotation of the schedule and, furthermore, minimizes the duty cycle, d (determined by the number of active slots)? A solution to the dual problem can be immediately converted to a schedule for the original problem, i.e., for a given duty cycle, scheduling the wake-up times so that the discovery time is minimized.

We next present the two constructions. The first construction is based on a novel application of the theory of *Sidon sets* (see, e.g., [31] for a detailed description of the theory); the second construction is based on the *Singer construction*. Both constructions provides cyclic integer schedules achieving $L(S) = (1 + o(1))1/d^2$ for particular values of d we describe below.

3.5.1 The Sidon Set Construction

We let $\mathbb{Z}_n$ denote the residue classes of the integers modulo n. For a given integer n, we wish to construct a set $S \subseteq \mathbb{Z}_n$ of size $|S| \approx \sqrt{n}$ such that for any $\alpha, \beta \in \mathbb{Z}_n$,

$$|(\alpha + S) \cap (\beta + S)| \neq 0\,,$$

where $\alpha + S$ denotes the set $\{\alpha + s \mid s \in S\}$. Note that this immediately implies that the set S is a schedule with $L(S) \leq n$ (and $d = |S|/n$). To simplify our requirements on the set S, we observe the following:

Lemma 10. *$|(\alpha+S)\cap(\beta+S)| \neq 0$ for any $\alpha, \beta \in \mathbb{Z}_n$ if and only if $\mathbb{Z}_n \subseteq S - S$, where $S - S$ denotes the set of differences $\{s_1 - s_2 \mid s_1, s_2 \in S\}$.*

Proof. It is easy to see the statement that $|(\alpha + S) \cap (\beta + S)| \neq 0$ for any $\alpha, \beta \in \mathbb{Z}_n$ is equivalent with $|(\alpha + S) \cap S| \neq 0$ for any $\alpha \in \mathbb{Z}_n$. If $|(\alpha + S) \cap S| \neq 0$ for any $\alpha \in \mathbb{Z}_n$, then there exist two elements $s_1, s_2 \in S$ such that $\alpha + s_1 = s_2$; thus $\alpha = s_2 - s_1 \in S$, which proves one direction of the lemma. For the other direction, as $\mathbb{Z}_n \subseteq S - S$ for any $\alpha \in \mathbb{Z}_n$, we can find two elements $s_1, s_2 \in S$ such that $\alpha = s_2 - s_1$. Therefore $\alpha + s_1 = s_2 \in (\alpha + S) \cap S$, as desired. □

This allows us to focus our efforts on construction of *saturated difference sets* in $\mathbb{Z}_n$, that is, subsets A for which $S - S$ contains all elements of $\mathbb{Z}_n$.

3.5.1.1 Constructions of Saturated Difference Sets

We say that a subset $S \subset \mathbb{Z}_n$ is a *saturated difference set* if

$$S - S \triangleq \{s_1 - s_2 \mid s_1, s_2 \in S\}$$

contains every element of $\mathbb{Z}_n$. Observe that when $|S| = k$, $|S - S| \leq k^2 - k + 1$ as $s - s = 0$ for any $s \in S$; in particular, if S is a saturated difference set we must have $k(k-1) \geq n - 1$. It follows, in particular, that $|S| \geq \sqrt{n}$ and our goal shall be to construct saturated difference sets with size very close to this lower bound.

An elementary construction

For a given $k > 0$, consider the set $A_k = \{k, 2k, \ldots, k^2\} \cup \{0, 1, 2, \ldots, k-1\} \subset \mathbb{Z}$. Observe that $\{-k^2, \ldots, k^2\} \subset A_k - A_k$ and hence that $A_k - A_k$ contains an element from every equivalence class of the integers modulo $2k^2 + 1$. To be precise, defining $A_{k,n} = \{a \bmod n \mid a \in A_k\}$ it follows immediately that $\mathbb{Z}_n \subset A_{k,n} - A_{k,n}$ so long as $n \leq 2k^2 + 1$.

To summarize, for the positive integer n define $k_n = \lceil \sqrt{(n-1)/2} \rceil$ and $D_n = A_{k_n,n}$. By the discussion above, D_n is a saturated difference set in $\mathbb{Z}_n$ of size $2k_n$. As a function of n, this yields a saturated difference set of size $2\lceil \sqrt{(n-1)/2} \rceil \leq \sqrt{2n} + 2 = \sqrt{2n} + O(1)$. While this does not achieve our desired bound of $\sqrt{n} + O(1)$, it does have the virtue of simplicity. We will use it as an ingredient in the following construction.

The Sidon set construction

Consider a prime number p and let ϑ be a the generator of the *multiplicative* group $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$ of units modulo p. Then the map $t \mapsto \vartheta^t$ is an isomorphism between $\mathbb{Z}_{p-1}$, the integers modulo $p-1$ under addition, and $\mathbb{Z}_p^*$, the multiplicative group of units modulo p. Consider the set

$$S_0 = \{(t, \vartheta^t) \mid t \in \mathbb{Z}_{p-1}\} \subset \mathbb{Z}_{p-1} \oplus \mathbb{Z}_p \,.$$

While the set S_0 is defined using the multiplicative structure of the ring of integers modulo p, our goal is to establish additive properties of the set (as a subset of the additive group $\mathbb{Z}_{p-1} \oplus \mathbb{Z}_p$).

Lemma 11. $\{(a, b) \mid a, b \neq 0\} \cup \{(0, 0)\} \subset S_0 - S_0$.

Proof. It is easy to see that $(0, 0) \in S_0 - S_0$. For a pair $(a, b) \in \mathbb{Z}_{p-1} \times \mathbb{Z}_p$ for which $a, b \neq 0$, consider the differences

$$(a + \ell, \vartheta^{a+\ell}) - (\ell, \vartheta^\ell) = (a, \vartheta^\ell(\vartheta^a - 1))$$

for $\ell \in \mathbb{Z}_{p-1}$. As $a \neq 0$, the sum $\vartheta^a - 1 \neq 0$ and it follows that

$$b \in \{\vartheta^\ell(\vartheta^a - 1) \mid \ell \in \mathbb{Z}_{p-1}\} = \mathbb{Z}_p^*$$

as ϑ is a multiplicative generator. In particular, by choosing ℓ to be $\log_\vartheta(b/(\vartheta^a - 1))$ we find that $(a, b) = (a + \ell, \vartheta^{a+\ell}) - (a, \vartheta^a)$, as desired. (Here the notation $\log_\vartheta x$ denotes the unique exponent in the set $\{1, \ldots, p-1\}$ for which $\vartheta^\alpha = x$.) $\square$

Observe that since p and $p-1$ are relatively prime, $\mathbb{Z}_{p-1} \oplus \mathbb{Z}_p \cong \mathbb{Z}_{p(p-1)}$ by the Chinese remainder theorem, and a saturated difference set in $\mathbb{Z}_{p-1} \oplus \mathbb{Z}_p$ immediately yields a saturated difference set in the cyclic group $\mathbb{Z}_{p(p-1)}$. However, the set S_0 above is *not* saturated: differences of elements of the set miss a few evasive "slices" of $\mathbb{Z}_{p-1} \oplus \mathbb{Z}_p \cong \mathbb{Z}_{p(p-1)}$: the elements (a, b) where $a = 0$ or $b = 0$. To rectify the construction, define $S_1 = \{(0, b) \mid b \in D_p\}$ and $S_2 = \{(a, 0) \mid a \in D_{p-1}\}$ (where D_n is the elementary construction discussed above), and observe now that the set $S \triangleq S_0 \cup S_1 \cup S_2$ clearly has the property that $Z_{p-1} \oplus Z_p \subset S - S$. To summarize, this construction yields a saturated difference set in $Z_{p(p-1)} \cong Z_{p-1} \oplus Z_p$ of size no more than

$$p - 1 + \sqrt{2p} + \sqrt{2(p-1)} \leq \sqrt{p(p-1)} + \sqrt{2p} + \sqrt{2(p-1)} \leq \sqrt{p(p-1)} + 3\sqrt[4]{p(p-1)}\,,$$

as $p > 2$. (The fact that $\sqrt{2p} + \sqrt{2(p-1)} \leq 3\sqrt[4]{p(p-1)}$ for $p \geq 2$ follows immediately by expanding the fourth power of both sides.)

We record the results of this construction in the following theorem.

Theorem 12. *For any n of the form $p(p-1)$ with $p > 2$ there is an explicit saturated difference set in $\mathbb{Z}_n$ of size no more than $\sqrt{n} + 3\sqrt[4]{n}$. For such a set, defining $d = |S|/n$, we have $L(S) \leq |S| = (1 + o(1))/d^2$.*

We use the word *explicit* in the above theorem to indicate that one can efficiently compute, given p, the elements of the difference set. The construction requires generating the elements of S_0, S_1, and S_2, which is straightforward given a generator ϑ of the cyclic group $\mathbb{Z}_p^*$. Such a generator can be found very quickly with randomization;

deterministically, one can simply check the first $\sqrt{p}$ elements of $\mathbb{Z}_p$ to find one with the property that $x^{p-1} \equiv 1$ but $x^{(p-1)/r} \not\equiv 1$ for each prime r dividing $p-1$. See [3] for details. It remains, of course, to represent the set S inside $\mathbb{Z}_{p(p-1)}$ via the isomorphism promised by the Chinese remainder theorem. For this purpose, note that

$$p \equiv 1 \mod p-1 \qquad (p-1)^2 \equiv 0 \mod p-1$$

$$p \equiv 0 \mod p \qquad (p-1)^2 \equiv 1 \mod p\,.$$

It follows immediately that the element $(a,b) \in \mathbb{Z}_{p-1} \oplus \mathbb{Z}_p$ is carried to the element $a \cdot p + b(p-1)^2 \mod p(p-1)$ in $\mathbb{Z}_{p(p-1)}$. As arithmetic in $\mathbb{Z}_p$, $\mathbb{Z}_{p-1}$ and $\mathbb{Z}_{p(p-1)}$ can be carried out in time $\text{poly}(\log p)$, we conclude that the entire set S can be computed in time $p \cdot \text{poly}(\log p)$.

3.5.2 The Singer Difference Set Construction

Prior work [21] in this area cites the existence of difference sets which achieve properties appropriate for our discovery problem, but the authors were not aware of explicit constructions that achieve these properties. Instead, they propose a exponential-time (in $1/d$) algorithm that is only practical when $1/d$ is small. We describe an efficient construction below, refering to [30] for proofs of the main intersection results.

Definition 2. *A subset $D \subseteq \mathbb{Z}_v$ is called a* $(v,k,1)$-difference set *if $|D| = k$ and, for any $g \in \mathbb{Z}_v \backslash \{0\}$, there exists exactly one pair $(x,y) \in D \times D$ such that $g \equiv x - y \mod v$. (Note that a difference set is a saturated difference set with the stronger condition that each non-identity element of $\mathbb{Z}_v$ is yielded by precisely one difference.)*

Theorem 13. *Let q be a prime power. Then there exists an explicit $(q^2+q+1, q+1, 1)$-difference set in $\mathbb{Z}_{q^2+q+1}$. For such a set, letting $n = q^2+q+1$ and defining $d = |S|/n$, we have $L(S) \leq |S| = (1+o(1))/d^2$.*

Proof sketch. The finite field $\mathbb{F}_{q^3}$ is a three-dimensional vector space over $\mathbb{F}_q$ for any prime power q. Let $\mathcal{V}_1$ denote the collection of all one-dimensional subspaces of $\mathbb{F}_{q^3}$ and $\mathcal{V}_2$ denote the collection of all two-dimensional subspaces of $\mathbb{F}_{q^3}$. For each $B \in \mathcal{V}_2$, let $A_B \triangleq \{C \in \mathcal{V}_1 \mid C \subset B\}$. It is easy to check that $|\mathcal{V}_1| = q^2 + q + 1$, a quantity we call ℓ throughout the construction; furthermore $|\mathcal{V}_2| = |\mathcal{V}_1|$, as every two-dimensional space is dual to a unique one-dimension space under the bilinear map $\langle x, y\rangle = \mathrm{Tr}_{\mathbb{F}_{q^3}/\mathbb{F}_q}(xy) = (xy)^{q^2} + (xy)^q + xy$. It is also easy to check that $|A_B| = q + 1$ for each $B \in \mathcal{V}_2$ and that any distinct pair $(C_1, C_2) \in \mathcal{V}_1 \times \mathcal{V}_1$ is contained exactly in one two-dimensional space $B \in \mathcal{V}_2$.

Let ω denote a primitive element of $\mathbb{F}_{q^3}$ (that is, a generator for the multiplicative group $\mathbb{F}_{q^3}^*$). Define a mapping $\alpha : \mathbb{F}_{q^3} \to \mathbb{F}_{q^3}$ by $\alpha(x) = \omega x$. Note that α is $\mathbb{F}_q$-linear (and clearly bijective): it carries subspaces to subspaces. It is not difficult to show that α permutes the elements in $\mathcal{V}_1$ in a single cycle of length ℓ. Fixing a particular one-dimensional space C_0, we write $\mathcal{V}_1 = \{C_0, C_1, \cdots C_{\ell-1}\}$ with the convention that $\alpha(C_i) = C_{i+1 \mod \ell}$ and, hence, $C_i = \omega^i C_0$. (Note that this is well-defined because α^ℓ, corresponding to left-multiplication by ω^ℓ, has the identity action on the C_i.) Rather remarkably, it can be proved that α also permutes the elements in $\mathcal{V}_2$ in a single cycle of length ℓ.

To define the difference set, let B_0 denote a fixed two-dimensional subspace and define $D \triangleq \{i \mid C_i \subset B_0\} \subset \mathbb{Z}_\ell$. To see that D is a difference set, consider an element $g \in \mathbb{Z}_\ell$, $g \neq 0$. Now, the pair (C_0, C_g) is contained in exactly one two-dimensional subspace B; let i have the property that $\alpha^i B_0 = B$. Now it follows that $\alpha^{-i} B = B_0$ and both C_{-i} and C_{g-i} are contained in B_0; hence $-i$ and $g - i$ are both in D, with $g - i - (-i) \equiv g \mod \ell$. Uniqueness follows similarly. Thus D is a $(q^2 + q + 1, q + 1, 1)$-difference set in $\mathbb{Z}_{q^2+q+1}$. □

Construction: For a prime number p, $\mathbb{F}_{p^3}$ can be realized as $\mathbb{F}_p[x]/(g(x))$ where $g(x) \in \mathbb{F}_p[x]$ is an irreducible polynomial with degree 3. Then the elements of $\mathbb{F}_{p^3}$ can be represented as polynomials in $\mathbb{F}_p[x]$ having degree at most 2. We may choose $C_0 = \mathbb{F}_q$, the constant polynomials; we then choose $B_0 \in \mathcal{V}_2$ to be $\text{span}(1, x) = \{i+jx \mid i, j \in \mathbb{F}_p\}$. This yields the difference set

$$D = \{y \in \mathbb{Z}_{p^2+p+1} \mid \omega^y = ax + b \text{ for some } a, b \in \mathbb{F}_q\},$$

where ω is a primitive element of $\mathbb{F}_{p^3}$.

The remaining work is to find a irreducible polynomial $g(x)$ and a primitive element ω. It turns out that we can do both at once. Initially, factor $p^3 - 1$ such that $p^3 - 1 = p_1^{n_1} \cdots p_k^{n_k}$, where each p_i is a prime number. It suffices to identify a polynomial $g(x) = x^3 + bx^2 + cx + d$ (where $b, c, d \in \mathbb{F}_p$) so that: (i.) $x^{p^3-1} \equiv 1 \mod g(x)$, and (ii.) $x^{(p^3-1)/p_i} \not\equiv 1 \mod g(x)$ for each i. Such $g(x)$ is irreducible in $\mathbb{F}_p[x]$ and guarantees that x is a primitive element of $\mathbb{F}_{q^3}$. As such g are known to be sufficiently dense (in fact, they have density $\phi(p^3-1)/(3p^3)$, where ϕ is the Euler totient function), this can be carried out very quickly with a randomized algorithm; see [3]. As construction of the set S will take time polynomial in p anyway, it is also possible to carry out this search over all triples a, b and c. In either case, so long as fast modular exponentiation is used to compute the powers x^v, the construction can be carried out in time $p^3 \,\text{poly}(\log p)$, as desired.

3.6 Generalized Model and Optimal Schedules

In this section, we first present a property that holds for any valid schedule in the basic integer model, and use it to motivate the generalized model. We then describe the generalized model, followed by lower bound on neighbor discovery in this model.

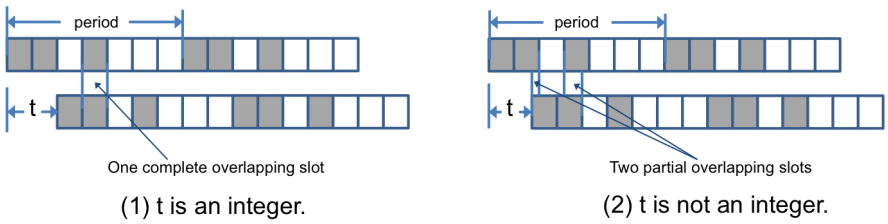


Figure 6: Illustration of the property in the basic integer model.

After that, we present a general reduction that transforms an integer schedule in the basic integer model to a non-integer schedule in the generalized model.

3.6.1 Motivation for the Generalized Model

Our generalized model is motivated by the observation that a valid schedule in the basic integer model has the following property:

Property 14. *Consider two nodes using a schedule in the basic integer model for neighbor discovery. Suppose the schedule guarantees discovery in time Δn. Then, for any offset t, we find one of two possible cases:*

- *The shift t is an integer, and there exists an entire time slot during which both nodes are awake.*
- *The shift t is non-integral, and there exist two time periods (among the first n) during which both agents are awake; furthermore, the durations of these time periods, r and r', sum to 1. Hence either r or r' is longer than $\Delta/2$.*

This property is illustrated in Fig. 6. The first case in the above property is straightforward. To see why the second case holds, consider t as a fractional number between two integers z and $z+1$; we shall show that the schedules of the two nodes, S_1 and S_2, in any n time slots have (at least) two overlaps, one of which has length $t - z$

and the other has length $z+1-t$. We know that if S_2 started at z, then there would be a complete time slot (say s_1 in the schedule of S_2) so that S_1 and S_2 overlap. Since now S_2 starts at t, they will just overlap for part of s_1 which has length $(1-(t-z) = z+1-t$. Similarly, if S_2 started at $z+1$, there would be a complete time slot (say s_2 in the schedule of S_2 which is different from s_1) that S_1 and S_2 overlapped. Since the real case is to shift S_2 to left with for $z+1-t$ length, the overlap is actually part of s_2 that is with length $1-(z+1-t) = t-z$. It is clear then that one of the overlapping portions has length at least half of one slot length (Δ), as desired.

The above property prompts us to define a further hardware-dependent parameter, the *required meeting time* $\delta > 0$: This is the minimum amount of time required by two devices to discover each other when they are both active. In the "chirp-and-listen" framework, this is essentially the length of the "chirp." The basic integer model implicitly assumes that $\delta = \Delta/2$, which is quite pessimistic for certain wireless devices. For instance, the study in [4] shows that on current smartphones, $\delta \ll \Delta/2$, and use this to construct improved discovery protocols. We therefore propose a generalized model that considers both δ and Δ. This model provides a unified setting that accommodates the characteristics of a diverse range of wireless devices. In addition, this model meaningfully considers continuous time, which is much less restrictive than the basic integer model. When only considering unit slot length (of Δ), the generalized model reduces to the basic integer model.

3.6.2 Generalized Model

Let $S \subset \mathbb{R}$ denote the wake-up schedule of a node. Specifically, S consists of a set of disjoint intervals, $S = \bigcup_i [a_i, b_i)$. We again consider periodic schedules, which repeat in time period $n \in \mathbb{R}$, and hence can be represented as a subset $S \subset [0, n)$.

For such a periodic schedule, we say that a schedule has duty cycle d when $|S|/n \leq d$, where $|S| = \sum_i (b_i - a_i)$. Again the goal is, given a duty cycle d, to design a schedule $S \subset [0, n)$, for some appropriate n, that minimizes the discovery latency. In addition, we impose the requirement on meeting time δ, i.e., two nodes must be simultaneously awake for a continuous interval of duration at least δ to discover each other. To be more precise, consider a pair of nodes using the same schedule S, the second of which arrives $t \in \mathbb{R}$ after the first has begun its schedule. The two nodes discover each other when $|S \cap S + t| \geq \delta$. Specifically, the discovery time for this pair is

$$L(S;t) = \min_{\ell \geq 0} \quad \text{so that} \quad [\ell, \ell+\delta) \bmod n \in S \quad \text{and} \quad [\ell+t, \ell+\delta+t) \bmod n \in S\,;$$

the goal is to design S so as to minimize discovery time, i.e, $L(S) = \max_t L(S;t)$.

3.6.3 Lower Bounds

We now give a lower bound on the duty cycle, d, for schedules which guarantee discovery by time n and prove the following theorem.

Theorem 15. *To guarantee discovery of two nodes within time n, for parameters $\delta \leq \Delta/2$, the duty cycle d has to be at least $\Delta/\sqrt{2n(\Delta-\delta)}$. Conversely, for a given duty cycle d, the discovery latency is at least $\Delta^2/(2d^2(\Delta-\delta))$.*

Proof. Suppose there are k maximal continuous intervals in S: $B_1, B_2, \cdots, B_k$, each with length $\ell_1, \ell_2, \cdots, \ell_k$. An overlap of S and $S+t$ could be from a pair of intervals B_i, B_j for $i, j \in \{1, 2, \cdots k\}$ (see Fig. 7).

For each pair of intervals, to satisfy the condition, the offset of initiation time can be during some time period with length $\ell_i + \ell_j - 2\delta$. We must have

$$\sum_{i,j} (\ell_i + \ell_j - 2\delta) = 2k \sum_i \ell_i - 2\delta k^2 = 2k|S| - 2\delta k^2 \geq n\,.$$

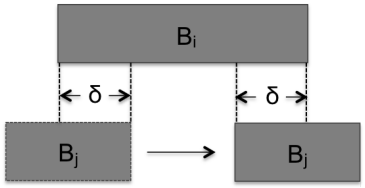


Figure 7: Overlap occurrences from a pair of intervals

Therefore, $|S| \geq (n + 2\delta k^2)/2k$. We also have $|S| = \sum_{i=1}^{k} \ell_i \geq k\Delta$.

Combining the two inequalities together, we have

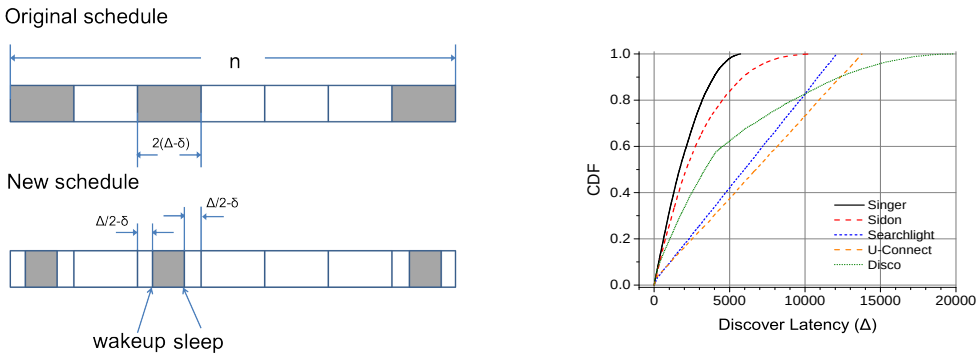
$$|S| \geq \min_k \max \left\{ k\Delta, \frac{n + 2\delta k^2}{2k} \right\}.$$

When $k = \sqrt{n/(2\Delta - 2\delta)}$, $k\Delta = (n + 2\delta k^2)/2k$, and when $k = \sqrt{n/2\delta}$, $(n + 2\delta k^2)/2k$ is the minimum. Since $\delta \leq \Delta/2$, $\sqrt{n/(2\Delta - 2\delta)} \leq \sqrt{n/(2\delta)}$. Therefore, $|S| \geq \sqrt{n/(2\Delta - 2\delta)} \cdot \Delta$, and hence the duty cycle is at least $\Delta/\sqrt{2n(\Delta - \delta)}$. In addition, for a given d, since $d \geq \Delta/\sqrt{2n(\Delta - \delta)}$, it follows immediately that the discovery latency, $n \geq \Delta^2/(2d^2(\Delta - \delta))$. □

Notice that when $\delta = \Delta/2$, the lower bound on discovery latency is Δ/d^2, which coincides with the lower bound for integer schedules [41]. When $\delta \ll \Delta$, the lower bound is approximately $\Delta/(2d^2)$, which is a factor 2 less than the lower bound for integer schedules, showing that δ is indeed an important parameter to consider.

3.6.4 General Reduction from Integer Schedules to Non-Integer Schedules

We next provide a reduction that can transform an integer schedule in the basic integer model to a schedule in the generalized model. Specifically, consider an integer schedule that aims to minimize the duty cycle for a given discovery latency of n. The reduction proceeds as follows (see Fig. 8(a)). It first divides the period, n, into



(a) (b)

Figure 8: (a) Illustration of the reduction that converts an integer schedule into a non-integer schedule; and (b) an example that compares our optimal schedules for the generalized model and several existing schemes ($d = 0.01$, $\delta = 0.1\Delta$, and the distribution of discovery latency is obtained from $10,000$ simulation runs).

$n/(2\Delta - 2\delta)$ slots, each of length $2(\Delta - \delta)$. After the schedule has been constructed (following the integer schedule that we consider), it "trims" the active slots by removing $\Delta/2 - \delta$ from each end and leaving an active interval of length Δ in the middle. By Property 14, the original integer schedule guarantees that two agents will meet for an interval of at least $\Delta - \delta$ in each slot. Since the trimming removes $\Delta - 2\delta$ from each such overlap, it leaves a meeting interval of exactly δ, satisfying the minimum required time interval for neighbor discovery.

The above reduction can be applied to any integer schedule. When applying it to an optimal schedule in the basic integer model, the resultant schedule is an optimal schedule in the generalized model. This is because, when the period, n, is divided into $n/(2\Delta - 2\delta)$ equal-length slots, the duty cycle of an optimal integer schedule is $\sqrt{(2\Delta - 2\delta)/n}$. The trimming reduces the amount of awake time in each active slot to a fraction, $\Delta/(2\Delta - 2\delta)$, of that in the original schedule, and hence reduces the duty cycle to $\sqrt{1/n(2\Delta - 2\delta)} \cdot \Delta$, equal to the lower bound for the generalized model that we established in Theorem 15. Specifically, applying the reduction to the two optimal

schedules (from Sidon set and Singer constructions, respectively) that we establish for the basic integer model, we obtain two optimal schedules for the generalized model. Fig. 8(b) compares the optimal schedules thus established and several existing schemes, where Searchlight [4] is a non-integer schedule, and U-Connect and Disco correspond to non-integer schedules reduced from the integer schedules in [19] and [10], respectively, where $d = 0.01$, $\delta = 0.1\Delta$. When $\delta \ll \Delta$ (the case in Fig. 8(b)), the state-of-the-art non-integer schedule, Searchlight, has discovery latency of $(1 + o(1))\Delta/d^2$, while our non-integer optimal schedules have discovery latency of $(1/2 + o(1))\Delta/d^2$, approximately half of the discovery latency of Searchlight.

Chapter 4

Conclusions and Future Works

In this chapter, we first summarize the work described in previous chapters of this dissertation and then discuss several questions which remain open and whose answers would make natural additions to the above work.

4.1 Conclusions

Rendezvous is a fundamental problem in wireless networks. Depending on the specific settings, the problems can be very different. We consider two kinds of rendezvous problems in the dissertation. One is the multi-channel rendezvous, in which each node has access to a subset of the entire spectrum of channels. The goal is for any pair of nodes to rendezvous in the shortest possible time. This is a problem originating from cognitive radio network community. The other one is the problem of energy-constrained single channel rendezvous, often called neighbor discovery. There is only one channel in this problem, but each node has a duty cycle which restricts the fraction of time that it can be awake. The objective is also to minimize the rendezvous time.

For the multi-channel rendezvous problem, we first give an $O(\log \log n)$ time algorithm for the special case when each agent can only access two channels. We then apply this algorithm to yield an $O(|S_i||S_j| \log \log n)$ for the general case, where $|S_i|$ and $|S_j|$ are the sizes of the subsets of available channels for the two nodes. Both of the bounds work for both synchronous and asynchronous scenarios. We also consider the "one bit beacon" case, where the agents have a single common random bit during each time slot. We give a randomized algorithm that guarantees rendezvous time $O\left(\log^2 n \cdot |S_i \cup S_j|/|S_i \cap S_j|\right)$ with high probability. We also provide lower bounds for the problem. We prove an $\Omega(\log \log n)$ lower bound on the rendezvous time, even for synchronous scenario. This shows that our size-2 algorithm is the best possible deterministic algorithm. In the asynchronous scenario, we prove an $\Omega(|S_i||S_j|)$ lower bound when $|S_i| + |S_j| \leq n + 1$. Our general case algorithm is tight up to a factor of $\log \log n$. Finally, we prove a lower bound for randomized algorithms, which is also $\Omega(|S_i||S_j|)$ when $|S_i| + |S_j| \leq n + 1$

For the energy-constrained single channel problem, we first gave two efficient constructions yielding optimal schedules for neighbor discovery in the basic integer model. These schedules are centered around two constructions of difference sets, one based on a novel application of the theory of Sidon sets and the other based on Singer's construction. This is the first time when such optimal schedules are proposed. We then proposed a generalized model that allows us to place existing work with various assumptions into a single setting, and established a new family of lower bounds for the model. Last, we provided a reduction that can transform any schedule in the basic integer model to the generalized model. Specifically, applying this reduction, our optimal schedules under the basic integer model became optimal schedules under the generalized model.

4.2 Future Work

In the future of my research, I would like to pursue results for the following directions.

Multi-channel rendezvous

- The upper bound of our algorithm and the lower bound for the asynchronous case still has a gap. We would like to pursue a tighter upper and lower bounds.

- In the synchronous case, so far we only have the natural algorithm which guarantees a latency of at most n. Whether there is room for improvement over this algorithm is an interesting question. We also don't know have any interesting lower bound on the best achievable latency guarantee when k, the subsets' size, is larger than $n^{1/2\alpha}$ (for any $\alpha \leq k$). Producing a lower bound that is substantially tighter than $\Omega(k)$ would be worthwhile.

- For the asynchronous case, a construction was given in [14, 29] which guarantees a latency at most $3n^2$, regardless of which specific subsets of the spectrum each agent can access. A question is whether it is possible to reduce the complexity to $2n^2$ or even n^2.

- Our deterministic protocols show superior performance in the case where each agent has access to only a small fraction of the frequency spectrum. There may be room for improvement when agents have access to large fractions of the spectrum and their subsets overlap by a substantial amount. The natural randomized algorithm can be applied to guarantee rendezvous in $|S_1||S_2| \log n/m$ steps with high probability, where m is the number of shared channels. We want to find a

deterministic algorithm that achieves a similar bound, or show that none exists if we cannot.

Energy constrained single-channel rendezvous

- Our contributions restrict the attention to the *homogeneous* setting, where each node is to follow the same activation schedule. It will be of value to have protocols with similar performance guarantees for a *heterogeneous* setting in which different duty cycles are appropriate for different nodes.

- A worthwhile future direction for this project is to adapt our protocols to function in the face of radio interference. In real life, two nodes may not discover one another if a third node is transmitting at the same time. Our model can be augmented to reflect this constraint by considering more than two agents, and considering two agents to rendezvous when they are the only two transmitting on the same channel. In this modified setting, we will need to consider not only the latency guarantee of our protocol, but the maximum number of agents for which it can guarantee that every two of them will rendezvous during the specified interval.

Bibliography

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks Journal*, 50:2127–2159, 2006.

[2] S. Alpern and S. Gal. *The theory of search games and rendezvous.* Springer, 2003.

[3] E. Bach and J. Shallit. *Algorithmic Number Theory, Volume 1: Efficient Algorithms.* Foundations of Computing. MIT Press, 1996.

[4] M. Bakht, M. Trower, and R. H. Kravets. Searchlight: Won't you be my neighbor? In *Proc. of ACM MobiCom*, 2012.

[5] L. Barenboim, S. Dolev, and R. Ostrovsky. Deterministic and energy-optimal wireless synchronization. In *DISC*, 2011.

[6] B. Bollobas and J. E. Littlewood. *Littlewood's Miscellany.* Cambridge University Press, 1986.

[7] M. Bradonjic, E. Kohler, and R. Ostrovsky. Near-optimal radio use for wireless network synchronization. In *ALGOSENSORS*, 2009.

[8] L. A. DaSilva and I. Guerreiro. Sequence-based rendezvous for dynamic spectrum access. In *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 1–7. IEEE, 2008.

[9] S. Dolev, S. Gilbert, R. Guerraoui, F. Kuhn, and C. Newport. The wireless synchronization problem. In *PODC*, 2009.

[10] P. Dutta and D. Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *SenSys*, 2008.

[11] S. Dziembowski and U. M. Maurer. Tight security proofs for the bounded-storage model. In *STOC*, pages 341–350, 2002.

[12] L. Gasieniec, A. Pelc, and D. Peleg. The wakeup problem in synchronous broadcast systems. *SIAM J. Discrete Math.*, 14(2):207–222, 2001.

[13] R. Graham, B. Rothschild, and J. Spencer. *Ramsey Theory*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 1990.

[14] Z. Gu, Q.-S. Hua, Y. Wang, and F. Lau. Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks. In *The 10th IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2013.

[15] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[16] C. S. Hsu, J. R. Jiang, Y. C. Tseng, and T. H. Lai. Quorum-based asynchronous power-saving protocols for ieee 802.11 and hoc networks. *ACM Journal on Mobile Networks and Applications (MONET)*, 10(1-2):169–181, 2005.

[17] P. Indyk. A small approximately min-wise independent family of hash functions. *Journal of Algorithms*, 38:84–90, 2001.

[18] R. Isaacs. *Differential Games*. John Wiley and Sons, 1965.

[19] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar. U-Connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proc. of IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, 2010.

[20] D. E. Knuth. Efficient balanced codes. *IEEE Transactions on Information Theory*, IT-32(1):51–53, 1986.

[21] S. Lai, B. Zhang, B. Ravindran, and H. Cho. CQS-Pair: Cyclic quorum system pair for wakeup scheduling in wireless sensor networks. In *Proc. of Principles of Distributed Systems*, 2008.

[22] H. H. Lee, E.-C. Chang, and M. C. Chan. Pervasive random beacon in the internet for covert coordination. In *Information Hiding*, pages 53–61, 2005.

[23] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung. Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks. In *Proceedings of INFOCOM 2011*, pages 2444–2452, April 2011.

[24] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Ring-walk based channel-hopping algorithms with guranteed rendezvous for congnitive radio networks. In *IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing (GREENCOM-CPSCOM)*, pages 755–760. IEEE, 2010.

[25] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung. Taxonomy and challenges of rendezvous algorithms in cognitive radio networks. In *2012 International Conference on Computing, Networking and Communications (ICNC)*, pages 645–649, 2012.

[26] M. J. McGlynn and S. A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proc. of ACM MobiHoc*, October 2001.

[27] A. Pelc. Deterministic rendezvous in networks: A comprehensive survey. *Networks*, 59(3):331–347, May 2012.

[28] M. O. Rabin. Transaction protection by beacons. *J. Comput. Syst. Sci.*, 27(2):256–267, 1983.

[29] J. Shin, D. Yang, and C. Kim. A channel rendezvous scheme for cognitive radio networks. *Communications Letters*, 14(10):954–956, 2010.

[30] D. R. Stinson. Combinatorial designs: Constructions and analysis. *Springer Verlag*, 2003.

[31] T. Tao and V. Vu. *Additive Combinatorics*. Number 105 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.

[32] C. N. Theis, R. W. Thomas, and L. A. DaSilva. Rendezvous for cognitive radios. *IEEE Transactions on Mobile Computing*, 10(2):216–227, 2010.

[33] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. In *Proc. of IEEE INFOCOM*, 2002.

[34] Wikipedia. Gps signals.

[35] Wikipedia. Spectrum management.

[36] D. Yang, J. Shin, and C. Kim. Deterministic rendezvous scheme in multichannel access networks. *Electronics Letters*, 2010.

[37] A. Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, pages 222–227, 1977.

[38] T. Yücek and H. Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. *IEEE Communications Surveys and Tutorials*, 11(1):116–130, 2009.

[39] L. Zhang, J. Luo, and D. Guo. Neighbor discovery for wireless networks via compressed sensing. *Performance Evaluation*, 70:457–471, 2013.

[40] Q. Zhao. A survey of dynamic spectrum access: Signal processing, networking, and regulatory policy. In *IEEE Signal Processing Magazine*, pages 79–89, 2007.

[41] R. Zheng, J. C. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. In *Proc. of ACM MobiHoc*, 2003.