

Rendezvous for Cognitive Radios

Nick C. Theis, *Member, IEEE*, Ryan W. Thomas, *Member, IEEE*, and
Luiz A. DaSilva, *Senior Member, IEEE*

Abstract—Cognitive radios have been touted as a solution to communicating in a Dynamic Spectrum Access environment. This paper examines how cognitive radios initially find one another among the expanse of ever-changing open spectrum, termed the rendezvous problem. Specifically, it addresses the problem of rendezvous under varying levels of system capabilities, spectrum policies, and environmental conditions. The focus is on rendezvous when there are no control channels or centralized controllers, which we term the *blind rendezvous* problem. Under these conditions, a sequence-based and modular clock blind rendezvous algorithms are proposed, and it is shown that the performance of these algorithms compares favorably to that of a random blind rendezvous algorithm. Specifically, the sequence-based algorithm provides a bounded Time To Rendezvous (TTR) and the ability to prioritize channels where rendezvous is more likely to occur; the modular clock algorithm reduces the expected TTR, requires little precoordination among radios attempting to rendezvous, and is robust to radios sensing different sets of available channels.

Index Terms—Dynamic spectrum access, cognitive radio, cognitive network, spectrum sensing, rendezvous.

1 INTRODUCTION

COGNITIVE radios, devices which alter their behavior based on perception and programming, can help solve many spectrum and waveform adaptation issues. In particular, cognitive radios have been envisioned as a chief enabler for Dynamic Spectrum Access (DSA), a mechanism for allowing secondary spectrum users (SUs) to share spectrum with primary spectrum users (PUs). For opportunistic channel access (sometimes called overlay access), SUs must dynamically sense a potentially large number of frequency channels; if a channel is not occupied by a PU, it may be available for two or more SUs to establish a communication link. However, there may be a large number of observed open channels (and at any given time, PUs may be active in some of these), making the problem of finding a common channel for SU radios to communicate on difficult.

This paper presents an investigation into the *rendezvous problem*, where two or more radios attempt to find one another in a DSA environment consisting of a set of frequency channels shared opportunistically with multiple PUs. It presents a taxonomy of the types of rendezvous, identifies a spectrum of system models under which rendezvous can occur, proposes several different rendezvous algorithms, and then provides both analytical and simulation results for their performance. In particular, we propose rendezvous algorithms and evaluate them in terms of their Time To Rendezvous (TTR), the amount of time

(typically measured in time slots) that it takes for two or more radios to rendezvous, once all radios have begun the rendezvous process. We show that these algorithms are able to achieve bounded TTR, for which expected TTR scales linearly with the number of channels, or work even when each radio senses different sets of available channels.

2 BACKGROUND AND RELATED WORK

We define frequency rendezvous as the process by which two or more cognitive radios attempt to arrive on the same frequency to begin transferring data. It is worth noting that some previous work on the topic refers to this process as “neighbor discovery” rather than rendezvous [1].

In Fig. 1, we establish a rendezvous taxonomy consisting of two branches: aided rendezvous systems and unaided rendezvous systems. Under an aided system, a centralized controller directs the cognitive radios to available channels and may even have the capability to set up links and schedule transmissions. This server operates over a well known, dedicated control channel. While this system is simple to implement and provides for greater control of the spectrum, it is not scalable, flexible, or robust. If a control channel is used, the system’s bottleneck becomes closely tied to the capacity of the channel and controller. As additional radios enter the system, the ability to coordinate them becomes strained. There are significant overhead and scalability limitations to allocating and building this required control and communications infrastructure in advance. Finally, having a centralized controller architecture can create a single point of failure.

In the other branch, unaided rendezvous, radios are left to their own devices to find common spectrum. From here, there are three broad categories of techniques: radios may have a single, dedicated control channel for use, multiple control channels, or no control channels. Under a single control channel scheme, radios which seek to rendezvous utilize a single, well known, control channel. Although this dispenses with the need for a centralized controller, it still incurs overhead and can act as a single point of failure.

• N.C. Theis and R.W. Thomas are with the Department of Electrical and Computer Engineering, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433.
E-mail: Nicholas.Theis@nellis.af.mil, ryan.thomas@afit.edu.

• L.A. DaSilva is with the Centre for Telecommunications Value-Chain Research, Department of Electronic and Electrical Engineering, Trinity College Dublin, Dublin 2, Ireland, and the Department of Electrical and Computer Engineering, Virginia Tech, Bradley. E-mail: ldsilva@vt.edu.

Manuscript received 12 Aug. 2009; revised 15 Jan. 2010; accepted 28 Feb. 2010; published online 9 Aug. 2010.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-08-0335. Digital Object Identifier no. 10.1109/TMC.2010.60.

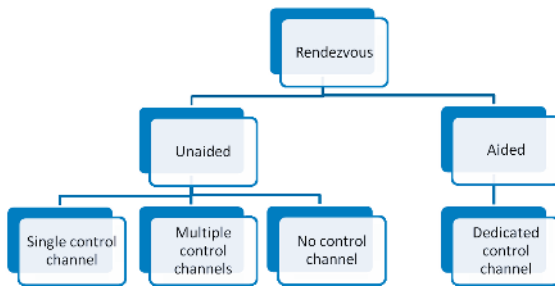


Fig. 1. Taxonomy of rendezvous techniques.

Under multiple control channels, radios must first find each other on one of the channels so they can begin negotiation. These control channels can be cluster-based, reducing the need to find a single channel that does not interfere with PUs. However, there is still overhead in creating and maintaining these clusters, and finding and identifying the control channels. If no control channel is available, the radios must figure out a way to find each other blindly, which we will refer to as the *blind rendezvous* problem.

In an infrastructure-based network, it is likely that an aided control channel approach will be used, with base stations and mobile radios agreeing on a preset control channel (or channels) to be used to identify spectrum availability and request and schedule connections. For example, Buddhikot et al. [2] have proposed an architecture in which some frequencies are set aside for use as spectrum information channels. Clients dedicate a wireless interface to scan these channels, where the base stations broadcast information regarding spectrum availability, interference conditions, etc. Clients can use those same control channels to request the use of dedicated spectrum for their traffic (or, alternatively, clients may directly proceed to the data channels that they now know to be available).

While the use of a common control channel simplifies the process of rendezvous and seems reasonable in an infrastructure-based network, some regulators worry about how to resolve the potential contention for the control channel under heavy load. If multiple base stations and associated radios, all in close vicinity of one another, wish to negotiate the use of available shared spectrum, policies must be in place to ensure that no one gets an unfair competitive advantage from being able to more efficiently access the control channel. For this reason, the IEEE 802.22 proposal does not adopt a common control channel, instead selecting a channel to use from all available spectrum holes [3].

Common control channels can also be adopted in infrastructureless networks, such as mobile ad hoc networks. Much of the work on multichannel Medium Access Control (MAC) protocols for ad hoc networks assumes that each node in the network is equipped with two radios (as described by Mo et al. [4] and references therein). The idea is that one radio constantly monitors a dedicated control channel and utilizes that channel to reserve transmissions on one of multiple potential data channels, possibly through the use of Request To Send (RTS) and Cleared To Send (CTS) frames. The second radio then tunes to the appropriate data channel for the exchange of data frames. The rendezvous process is thus simplified at the expense of

dedicating one transceiver exclusively for the monitoring and transmission of control information. Also, the control channel may still act as a bottleneck and single point of failure is still a concern.

It is possible to envision more flexible approaches that take advantage of unaided rendezvous using multiple channels for exchange of control information. A variation is to have a dynamically changing control channel: for instance, radios maybe programmed to always attempt to rendezvous in the lowest number control channel that is currently not occupied by an incumbent. The potential problem with such a method is that two radios may not correctly sense the presence of a PU in a given channel. An important feature of a rendezvous solution is robustness to dynamic spectrum occupation as well as to differing views of current spectrum occupancy, due to differences in position, range, and sensing capabilities of different radios.

In this manner, it can be argued that DSA schemes which require any control channels are less than dynamic. According to Sutton et al. [5], the concept of DSA does not include a static command and control channel to direct the agents to available spectrum white space, even though contrary implementations are suggested in [2] and [6]. Control channels of any sort can often become bottlenecks for busy networks, and make an assumption of spectrum reliability that might not always be present.

For these reasons, we are pushed toward blind rendezvous, in which all vacant channels are potentially available for the exchange of control and data. Radios are responsible for determining which channels are available and then attempting to establish a link on one of those channels. Blind rendezvous allows cognitive radios to become truly autonomous, not relying on extra infrastructure or risky channel availability. Solving the blind rendezvous problem effectively requires the ability to guarantee frequency rendezvous in a reasonable amount of time, under a variety of conditions and system requirements.

Several blind rendezvous algorithms for DSA have been proposed in the literature. Horine and Turgut [7], [8] propose a wideband transmission and sensing process, in which signal tones on multiple vacant frequency bands are simultaneously broadcast, and then the same bandwidth is sensed for a response from available radios. Sutton et al. [9] improve the process by injecting distinct cyclostationary signatures into the transmissions that can be used to detect spectrum users, identify their requirements, and acquire spectrum for a connection. Arachchige et al. [10] propose building infrastructure where there previously was none; radios are elected as leaders that are then responsible for discovering and accounting for all neighbors. Although this work does not explicitly address setting up rendezvous channels between all participating radios, once the infrastructure has been set up any aided approach can be used. Silvius et al. [11] propose having radios take on different roles to ensure radios find one another in the available spectrum.

3 RENDEZVOUS SYSTEM MODELS

In considering the blind rendezvous problem, first we must define the system limitations and assumptions. These can

be broadly divided into system capabilities, spectrum policies, and environmental conditions.

System capabilities are model assumptions related to the functionality of the cognitive radio systems attempting to rendezvous. They include:

- *Synchronization*—Radio synchronization allows radios to listen and/or transmit at the correct time, avoiding missing other radio transmissions or causing interference.
- *Heterogeneity of radio roles*—Different radios may perform different rendezvous roles (i.e., searcher, beacon, etc.) in a system. Determining how to assign these roles can be difficult.
- *Number of radios to rendezvous*—Two or more radios may attempt to rendezvous together. This value can affect whether a particular rendezvous solution is complete. While in the case of only two radios any rendezvous solution is complete, for more than two radios there exist incomplete rendezvous solutions that contain fewer than all rendezvousing radios (for instance, only two of four radios find each other).
- *Common spectrum labels*—Having common labels means that, for any unique physical frequency channel, the identification (index) of the channel is the same on each radio.
- *Master controller*—Master controllers are central agents which can control spectrum allocation for the radios in a cognitive network.

Spectrum policies are limitations on the behaviors of the cognitive radio system due to regulations. They include:

- *Wideband operation*—A single radio may sense and/or transmit in multiple channels simultaneously. Being able to do so simplifies rendezvous, but there are complications with hardware costs and regulatory policy.
- *Control channels*—Control channels are reserved channels that maybe used to facilitate rendezvous.
- *Fairness*—In the context of rendezvous, fairness requires that every observed available channel must be considered for rendezvous.

Environmental conditions limit the performance of a cognitive radio system due to external factors, such as the usage by PUs and environmental noise. They include:

- *Amount of common spectrum*—Not all radios will observe the same free spectrum at the same time. The raw amount of commonly observed and usable spectrum, as well as the ratio of this amount to the total observed spectrum affects a radio's rendezvous choices.
- *Signal detection*—The accuracy with which a cognitive radio is able to detect a PU or another SU determines the probability of false positives and negatives when determining spectrum availability and achieving successful rendezvous.
- *Malicious activity*—The presence of agents actively conspiring to degrade the rendezvous process must be considered in some environments.

Considering these, we provide a nonexhaustive categorization of rendezvous system models. A rendezvous system

TABLE 1
Comparison of Available System Properties
for Various Rendezvous System Models

	Assisted	Roles	Shared	Individual	Free-for-all
Synch available	•	•			
Heterogeneous roles	•	•			
Number radios	n	2	2	2	n
Common naming	•	•	•		
Wideband	•				
Fairness	•	•	•	•	
Control channels	•				
Master controller	•				
Common spectrum	•	•	•		
No detection errors	•	•	•	•	
No malicious radios	•	•	•	•	

model represents a grouping of assumptions about the capabilities, spectrum and environment in which rendezvous is attempted. A rendezvous solution is categorized as operating under the model that it most closely matches; this is the model that has every assumption used in the solution and the least number of unused assumptions. We present the models and their associated system properties in Table 1 in an incremental fashion, with each model representing a qualitative increase in the difficulty of performing rendezvous.

At the bottom of the list is the *assisted* model, which allows for many rendezvous resources, including a control channel and master controller to handle all connection rendezvous. Wideband sensing and beaconing is also allowed under this model. The assisted model has been proposed in several cognitive radio systems (see Section 2). This model does not require a blind rendezvous solution due to the ability to take advantage of existing infrastructure.

The *role-based* model removes the infrastructure assumptions of the assisted model, but allows the assumption that radios have agreed to distinct rendezvous roles in advance. General rendezvous literature refers to this method as *asymmetric*. One of the simpler and more well-known asymmetric strategies is known as the “wait for mommy” approach [12]. Under this approach, the “child” agent remains in its location while the “mother” agent exhaustively searches, thereby guaranteeing rendezvous. This can only be achieved if the “mother” and the “child” are identified ahead of time.

Systems that use role-based rendezvous include first responder systems [11] and Bluetooth [13]. In Silviu's first responder system, there are two roles for the radios: master or slave. The master radio has the predetermined responsibility of maintaining a fixed position within the dynamic spectrum, while the slave radio exhaustively searches for it. Bluetooth communication bootstrapping uses an asymmetric approach with a control channel. To initialize communication, the peripheral device initiates inquiry mode and broadcasts over the predefined bootstrap spectrum. Any

TABLE 2
List of Primary Notation Used in Paper

Term	Definition
\mathcal{N}	set of all cognitive radios; $n = \mathcal{N} $, $i \in \mathcal{N}$
\mathcal{C}_i	set of open channels observed by the i th radio; $m_i = \mathcal{C}_i $, $c_{ij} \in \mathcal{C}_i$
\mathcal{C}	set of all open channels observed by all radios; $\mathcal{C} = \bigcup_i \mathcal{C}_i$, $m = \mathcal{C} $
$\bar{\mathcal{C}}$	set of all common open channels between all radios; $\bar{\mathcal{C}} = \bigcap_i \mathcal{C}_i$, $\bar{m} = \bar{\mathcal{C}} $

available devices respond to the beacon messages via the control channel as well.¹

Most existing rendezvous protocols fall into some subset of the assisted and role-based models. In the taxonomy, the assisted model captures all systems (aided and unaided) save those performing blind rendezvous. Even some protocols that perform blind rendezvous, such as Horine and Turgut's work [7] and the IEEE 802.22 standard fall under these models—Horine and Turgut's work uses the assisted model as it requires wideband sensing and transmission and the IEEE 802.22 uses the role-based model with separate base station and cognitive radio roles.

Anderson and Weber wrote one of the first papers on the general rendezvous problem for discrete locations [14]. They model the problem as two players seeking to meet in one of m unique, shared locations which can be traveled between instantaneously. Alpern and Gal [12] modified the problem description without changing the model by calling it the "telephone game." Under Alpern and Gal's problem description, two players in two separate locations have m telephones directly connected to each other. At specific time intervals, both players choose a telephone, pick it up, and say "hello." The game ends when the players choose the same phone in the same time slot and can have a conversation. The basic assumptions made in these papers, namely the lack of control infrastructure or radio roles along with a set of shared locations to rendezvous in, we term the *shared model*.

The rendezvous problem under the shared model is still open, with optimality results largely relegated to special cases. In [14], Anderson and Weber² show that the optimal solution (smallest expected TTR) for two channels is for each radio to choose between the two channels at every time step randomly. They also show that the ability to leave a message for the other radio in a channel would result in an algorithm with a maximum TTR bounded by $O(\sqrt{m})$ time. Unfortunately, this is not physically possible in spectrum rendezvous.

The *individual model* further relaxes the shared model assumptions, removing the assumption that all radios observe the same set of open channels, and have the same internal labeling structure. In a sense, this means that some of phones in Alpern and Gal's description now do not connect to one another and all are randomly arranged. In a wireless environment, propagation and loss factors make it likely that not all radios will observe the same set of open channels. As the density of low-power PUs increases, the

1. This is not to be confused with the fact that Bluetooth uses Frequency Hopping Spread Spectrum (FHSS) during communication to handle noise. The handling of communication once the links are created is beyond the scope of rendezvous.

2. The general rendezvous literature is not written explicitly for the cognitive radio rendezvous problem; however, for consistency, we will use cognitive radio rendezvous terminology hereafter.

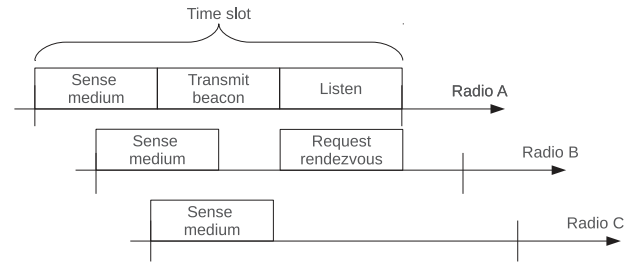


Fig. 2. Slotting for rendezvous. In this example, Radio A and B will have completed rendezvous.

individual model will likely be a better fit with reality than the shared model, since SUs will find they do not observe the same sets of vacant channels.

The individual model is relatively unexplored in the rendezvous literature. Arachchige et al.'s work [10] is one notable exception, but full rendezvous is not developed, only neighbor discovery by an elected leader. It is worth noting that while roles are developed via election in this work, they are not preassigned, so their work should not be considered as using the role-based model.

The *free-for-all model* makes the fewest assumptions about rendezvous resources and requirements. Under this model, there exists no infrastructure to assist the radio in finding other radios, nor do we know anything about the availability of spectrum or the hostility of the environment. Perhaps the only work that has investigated some of the problems of the free-for-all model is that of Ma and Shen [15], in which the rendezvous process is specifically protected against smart jammers. However, Ma and Shen assume that radios begin in a rendezvoused state and focuses on the slightly different problem of re-rendezvousing when a channel becomes taken, either by a PU or a malicious jammer.

4 PROPOSED RENDEZVOUS ALGORITHMS

In blind rendezvous, all observed open channels are potentially available for the exchange of control and data. Radios are responsible for determining which channels are available and then attempting to establish a link on one of those channels. In Table 2, a list of notation is presented to describe the set of radios and open channels they observe. Under unassisted rendezvous models, the radio must visit those available channels in random or preestablished order, alternatively transmitting beacons and listening for responses, until it is able to establish one or more links.

For most models, it is common to assume that radios operate using fixed length time slots. A radio will only change channels at time points between slots. During each time slot, a radio senses the medium for the presence of a PU or other SUs in that channel. If it does not sense others in its vicinity, the radio will transmit a beacon or heartbeat, followed by another period of silence, while it waits for a response. Note that we need not assume that time slots be synchronized between two radios for this to work. In Fig. 2, if all three radios are currently operating in the same channel, both Radio B and Radio C will be able to hear and respond to Radio A's beacon, thereby completing the handshake required for rendezvous, even though the slot boundaries are not aligned.

Of course, variations of the process depicted in Fig. 2 and of the handshake method necessary to complete rendezvous are also possible. Regardless of the method, it is necessary that at some point in time, the two radios meet in the same channel. In this section, we discuss several methods in which radios can, under various system models, accomplish this.

4.1 Random Algorithm

In random rendezvous, a radio wishing to join a network visits the potential communications channels in random order. During each time slot, the radio will select any of the m_i channels with probability $1/m_i$. For two radios following this procedure, rendezvous will be successful when two conditions occur: 1) the two nodes select the same channel and 2) one of the radios is sensing the medium while the other is transmitting a beacon in such a way that the handshake required for rendezvous is possible, as in the example in Fig. 2. Algorithm 1 describes this in more detail.

Algorithm 1. Random Algorithm

- 1: observe m_i , the number of channels available
- 2: **while** not rendezvous **do**
- 3: $j_i = \text{rand}[0, m_i)$
- 4: $c = c_{i,j_i}$
- 5: attempt rendezvous on channel c
- 6: **end while**

Expanding on point (2) from above, the success of the handshake depends on the portion of the time slot that is dedicated to sensing and beaconing and on the amount of time that the channel-sensing mechanism requires to make a correct determination that there is another radio operating in the channel. We lump all of these conditions into the probability of a successful handshake p_H . The TTR will depend on p_H as well as on the expected time until both radios select the same channel. If we let E denote the event that, in a given time slot, both radios select the same channel, we have (assuming the two events are independent)

$$P\{\text{rendezvous}\} = p_H P\{E\}. \quad (1)$$

4.1.1 Performance

The random algorithm has some probability of operating successfully under the shared, individual, and free-for-all models. Under the shared model (where $m = \bar{m} = m_i$)

$$P\{E\} = 1/m, \quad (2)$$

and the TTR is a geometrically distributed random variable representing the number of failures before the first success in a sequence of independent Bernoulli trials with probability of success equal to p_H/m . Thus, the probability mass function of the TTR under the shared model can be expressed as

$$P\{TTR = k\} = \frac{p_H}{m} \left(1 - \frac{p_H}{m}\right)^{k-1}. \quad (3)$$

The expected TTR is therefore equal to m/p_H .

Similarly, under the individual model, the probability two radios select the same channel is

$$P\{E\} = \frac{\bar{m}}{m_1 m_2}, \quad (4)$$

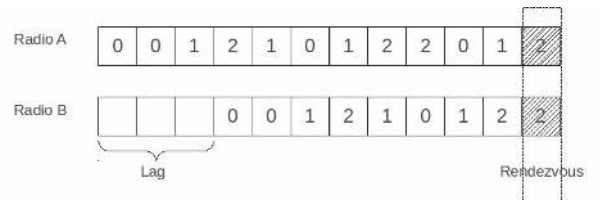


Fig. 3. SUs A and B perform generated orthogonal sequence-based rendezvous.

giving the expected TTR as

$$E[TTR] = \frac{m_1 m_2}{p_H \bar{m}}. \quad (5)$$

Results for the free-for-all model depend on both radios choosing at least one channel in \bar{C} with some nonzero probability and p_S , the probability of a successful channel transmission in the face of detection errors and malicious radios.

Without loss of generality, the subsequent discussion will ignore the constant term $1/p_H$. It is also simple to see that under the shared and individual model, rendezvous will occur in any of the \bar{m} channels with equal probability, and that the TTR for all models is unbounded (i.e., arbitrarily large TTR occurs with nonzero probability). Because of its simplicity, random rendezvous can be made to work within any of the system models.

Anderson and Weber's algorithm [14] represents a slight modification of the random algorithm for use under the shared model (with the added assumption that radios synchronize the start of their rendezvous processes, so this is technically the role-based model). In their algorithm, each radio does one of two actions for m time slots: it either follows a randomized permutation of the m channels or stays put on a random channel. The use of the random permutation ensures that all channels are visited before a channel is visited again. The fact that every channel is visited in the permutation guarantees rendezvous if one radio searches while the other remains in place. The probability of choosing to remain in place instead of following a randomized permutation is denoted by θ . Anderson concludes that the optimal θ value is 0.2475 as m gets infinitely large, and the expected TTR is approximately $0.828 \cdot m$. Note that like the random algorithm, there is no bound on the maximum TTR.

4.2 Generated Orthogonal Sequence-Based Algorithm

An alternative approach for blind rendezvous is generated orthogonal sequence-based rendezvous, first proposed by us in [16]. In this approach, radios employ predefined sequence generators to determine the order in which potential rendezvous channels are to be visited. The idea is that both transceivers follow the same sequence, albeit arbitrarily delayed with respect to each other. In Fig. 3, we illustrate the process. In the example, SUs A and B both visit channels according to the sequence $\langle 0, 0, 1, 2, 1, 0, 1, 2, 2, 0, 1, 2 \rangle$. Even though there can be a lag (in the example in Fig. 3, three time slots) between the time that radio A and radio B start to look for others with whom to

rendezvous, they eventually occupy the same channel in the same time slot and therefore rendezvous is achieved.

These sequences must be constructed in a way that minimizes the maximum and/or the expected TTR even when radios do not start their rendezvous algorithms at the same time. The properties of the TTR in this case depends on the selection of the sequence. By appropriately selecting the sequence, it is possible to: 1) establish a maximum TTR, 2) reduce the expected TTR as compared to random rendezvous, and 3) establish a priority order for potential rendezvous channels.

It should be clear that not any sequence will work. Consider again a set of m_i potentially available channels, commonly labeled 0 through $m_i - 1$. Now think of two radios visiting channels in ascending order of channel label: $0, 1, \dots, m_i - 1$. If the two radios are synchronized and start the process at the same time, they'll meet in the same channel immediately. However, in the vast majority of cases, there is no practical way to synchronize multiple radios before they join the network. Therefore, following this simple sequence will result, with high probability, in the two radios never meeting, and the expected TTR is infinite. This will obviously not do. Thankfully, there are sequences that yield a finite expected TTR and other desirable properties.

The problem of selecting sequences for sequence-based rendezvous can be thought of as the dual of the problem of selecting hopping sequences for FHSS. In FHSS, desirable sequences are those that minimize the probability that, at a given time, multiple radios occupy the same frequency channel. In sequence-based rendezvous, desirable sequences are those that maximize that same probability. The former problem has been studied extensively (see, for instance, [17]), whereas the latter is, to our knowledge, still open.

A visiting sequence $\mathcal{S} = \langle s_1, s_2, \dots \rangle$ where $s_j \in \mathcal{C}_i$ describes the order in which a radio visits channels in search of other radios with which to rendezvous. We have been investigating periodic sequences, and in our further discussion, we focus on one period of length l , i.e., $l = |\mathcal{S}|$. Further, for fairness reasons, it is desirable that each of the m_i channels appear the same number of times within one period, so l must be a multiple of m_i . Finally, it is desirable that the sequences be defined according to an algorithm that generalizes for any value of m_i .

Algorithm 2 describes the process in more detail. Sequence \mathcal{S}_i describes the order that the radio will visit the channels (with $s_{i,j}$ representing the j th element of \mathcal{S}_i), $seq(\cdot)$ is the sequence generating function that takes as input an ordered set of channels and sequence length and returns as an output an ordered sequence of length l and $perm(\cdot)$ is a function that creates an ordered permutation from a set.

Algorithm 2. Generated Orthogonal Sequence Algorithm

- 1: observe m_i , the number of channels available
- 2: $\mathcal{S}_i = seq(perm(\mathcal{C}_i), l)$
- 3: $j_i = 0$
- 4: **while** not rendezvous **do**
- 5: $c = s_{i,j_i}$
- 6: attempt rendezvous on channel c
- 7: $j_i = j_i + 1 \bmod l$
- 8: **end while**

One definition for the generator $seq(\cdot)$ is to have it use any permutation from $perm(\mathcal{C}_i)$ of the m_i channels (there are

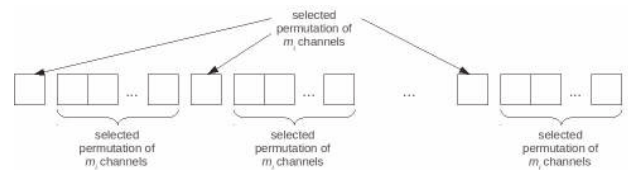


Fig. 4. One generator for creating a nonorthogonal sequence.

$m_i!$ such permutations) and build the sequence as illustrated in Fig. 4. The selected permutation appears $(m_i + 1)$ times in the sequence: m_i times the permutation appears contiguously, and once the permutation appears interspersed with the other m_i permutations. The period of the sequence is $l = m_i(m_i + 1)$.

An example may make things more clear. Take $m_i = 5$, and select at random a visiting sequence permutation of these five channels, say the permutation $\langle 3, 2, 5, 1, 4 \rangle$. The method described above to form a sequence would yield:

$$3, 3, 2, 5, 1, 4, 2, 3, 2, 5, 1, 4, 5, 3, 2, 5, 1, 4, 1, 3, 2, 5, \\ 1, 4, 4, 3, 2, 5, 1, 4.$$

More precisely, what we represented above was one period of the sequence. This ordering of channels is then repeated indefinitely.

4.2.1 Correctness

The generated orthogonal algorithm assumes both radios share the same $perm(\cdot)$ and $seq(\cdot)$ functions. This information has to be exchanged in advance, perhaps when the radio firmware is flashed. Furthermore, it only works under the shared model (where $\bar{m} = m = m_i$). It is trivial to come up with examples under the individual model where rendezvous will fail to occur.

To examine the effect of different labeling under the individual model, consider a set of five channels with universal labels A through E . The first radio will locally label channel A as 1, B as 2, C as 3, D as 4, and E as 5. The second radio will locally label channel A as 5, B as 1, C as 2, D as 3, and E as 4. The previously agreed on shared permutation function is $\langle 3, 2, 1, 4, 5 \rangle$. Therefore, the first radio's permutation under the universal labels is $\langle C, B, A, D, E \rangle$, while the second radio's permutation is $\langle D, C, B, E, A \rangle$. It is easy to see the sequences can become orthogonal. Take the case in which the sequences start at the same time (no offset): there is no point at which both radios are on the same channel in the same slot. At every point, both radios will choose the same local label, which maps to different physical channels. A similar analysis can be done when radios see different sets of available channels and one set is not a subset of the other.

Theorem 1. *Under the shared model when two radios perform the generated orthogonal sequence-based rendezvous algorithm³ with $seq(\cdot)$ as described in Fig. 4, rendezvous will occur in at most $m(m + 1)$ time slots.*

3. It is important to note that the term rendezvous is slightly abused here and in the future to refer to arriving to a common channel label. Depending on further assumptions (slotting, interference, etc.), this may or may not correspond to actual rendezvous as a function of probabilities p_H and p_S .

Proof. The outer permutation of the generated sequence appears at positions $1, 2 + m, 3 + 2m, \dots$. In mod space, this corresponds to positions $1, 2, 3, \dots$ in the inner sequence. Regardless of the lag between when radios start the rendezvous process, each entry in the outer sequence will have corresponded with each entry in the inner sequence after one complete period. Thus, rendezvous is guaranteed in l time slots, or $m(m + 1)$ time slots. \square

4.2.2 Performance

When radios visit potential rendezvous channels following such a sequence, the TTR can be shown to have some desirable properties. First of all, from Theorem 1, the TTR is bounded by $O(m^2)$. This is a valuable property when trying to meet Quality of Service (QoS) objectives for link establishment in a DSA environment. It is also an improvement over the unbounded TTR of the random algorithm.

Second, the sequence implicitly establishes a prioritization order for rendezvous channels. In other words, even though in each period of the sequence each channel appears the same number of times, rendezvous is more likely to occur in some channels than in others. This is useful whenever there is a reason to favor some channels over others, for instance, because some channels are less likely to have activity from incumbents than others, or due to propagation characteristics of a given set of channels. Further, the expected TTR when rendezvous occurs in one of those preferred channels (i.e., the conditional expectation) is lower than for random rendezvous.

Assuming all time offsets are equally likely, the expected TTR for the best channel can be shown to be [16]

$$E[TTR|_{best}] = \frac{m^3 - 2m^2 + 9m - 4}{2(3m - 1)}, \quad (6)$$

or slightly less than m^2 time slots. The expected TTR for the worst channel can be given from Theorem 1 as $m(m + 1)$. How about the overall expected TTR? For this family of sequences, we have been able to derive the expected as

$$E[TTR] = \frac{m^4 + 2m^2 + 6m - 3}{3m(m + 1)}. \quad (7)$$

So, while this sequence-based rendezvous outperforms random rendezvous in terms of *maximum* TTR and in its ability to support preferred rendezvous channels, it presents a higher expected TTR. To visualize this feature of most and least priority channels, the expected TTR for the most and least priority channels (as well as all channels in between) are plotted in Fig. 5.

There do exist nongenerated orthogonal sequences that are known to achieve rendezvous with an expected TTR of less than m slots and maximum TTR of less than m^2 slots under the same requirements [16]. However, these sequences must be created (currently through exhaustive search) for every observed channel size m , requiring a larger memory footprint. These sequences also do not have the priority feature found in the generated sequences.

4.3 Modular Clock Algorithm

The modular clock algorithm is a rendezvous algorithm that uses prime number modular arithmetic to make TTR guarantees under several different rendezvous models.

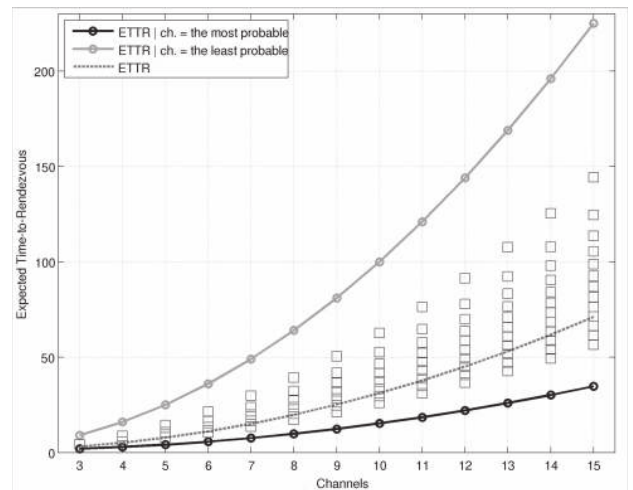


Fig. 5. $E[TTR]$ (middle curve); conditional $E[TTR]$ for most preferred (top curve), least preferred (bottom curve) and intermediary channels (boxes).

The additional notation we adopt is as follows:

- r_i is the rate with which the cognitive radio i hops between channels. Every time slot, the radio hops “forward”⁴ r_i channels in its set, wrapping around (employing modulo arithmetic) when it reaches a channel greater than a threshold.
- t indicates the time slot of the system.
- p_i is a prime number chosen by radio i .

Algorithm 3 outlines the pseudocode for the modular clock algorithm. The algorithm begins by selecting a channel randomly from the set of observed channels. After each time slot, the index (label) j_i is increased by r_i modulo p_i . The modulus p_i is lowest prime greater than or equal to the number of observed channels m_i . If the resulting channel index j_i is in $[0, m_i)$, then the radio moves to channel c_{i,j_i} , the channel with index j_i in radio i 's set of available channels. Since there will often be a “gap” between between m_i and p_i , if the resulting index is greater than $m_i - 1$, then the index is remapped via the mod function between 0 and $m_i - 1$ (inclusive) and the radio moves to that channel (note that index j_i itself is not changed in this case). If rendezvous has not been achieved after $2p_i$ time steps, a new r_i value is randomly chosen in $[0, p_i)$. The algorithm terminates at any point in which rendezvous is achieved. Note that all variables listed in Algorithm 3 are relative to the radio which is executing the algorithm, and not global values.

Algorithm 3. Modular Clock Algorithm

- 1: observe m_i , the number of channels available
- 2: calculate p_i , the next largest prime to m_i
- 3: $j_i^0 = \text{rand}[0, m_i)$
- 4: **while** not rendezvous **do**
- 5: choose r_i from $[0, p_i)$ randomly
- 6: **for** $t = 0$ to $2p_i$ **do**
- 7: $j_i^{t+1} = (j_i^t + r_i) \bmod (p_i)$
- 8: **if** $j_i^{t+1} < m_i$ **then**
- 9: $c = c_{i,j_i^{t+1}}$

4. We assume that every radio has an internal labeling for the set C_i , although, in general, we make no assumptions about this being consistent across all radios.

```

10:   else
11:      $c = c_{i, (j_i^{t+1} \bmod (m_i))}$ 
12:   end if
13:   attempt rendezvous on channel  $c$ 
14: end for
15: end while

```

We now analyze the correctness and performance of the modular clock algorithm under the shared and individual models. We observe that between Lines 6 and 14 of the modular clock algorithm, the channel index that radio i is on at a particular time step t can be expressed by the equation:

$$j_i^t = t \cdot r_i + j_i^0 \bmod (p_i), \quad (8)$$

where j_i^0 is radio i 's initial channel index and subscripts are used to distinguish between the different radios running the modular clock algorithm. Furthermore, we observe that when the channel indices are in $[m_i, p_i)$, the modular clock algorithm consistently remaps those channels to $[0, m_i)$. We will, therefore, not include this step in the analysis and refer to channels as existing in $[0, p_i)$.

4.3.1 Shared Model Correctness

We begin by analyzing the modular clock algorithm's correctness under the shared model. As described in Section 3, this model assumes that all radios observe the same set of channels and label them the same. This means that $\bar{m} = m = m_i$ and thus $p_1 = p_2$.

Theorem 2. *Under the shared model when two radios perform the modular clock algorithm with $r_1 \neq r_2$, rendezvous will occur in less than p_i time slots.*

Proof. Without any loss in generality, we start the modular clock algorithm at the point in which both radios begin attempting to rendezvous. We define $t = 0$ to be this point and note that the channel indices at this time, j_1^0 and j_2^0 , are randomly distributed within $[0, m)$. Convergence will occur when

$$t \cdot r_1 + j_1^0 \bmod (p_1) \equiv t \cdot r_2 + j_2^0 \bmod (p_2). \quad (9)$$

We define $\delta_r = r_1 - r_2$ and $\delta_j = j_1^0 - j_2^0$. We can then rewrite (9) as

$$t \cdot \delta_r \equiv -\delta_j \bmod (p_i). \quad (10)$$

We now observe two points:

1. Since $r_1 \neq r_2$ and $r_i \in [0, p_i)$, this means that $\delta_r \in (-p_i, 0) \cup (0, p_i)$.
2. The modular multiplicative inverse (i.e., a^{-1} where $a \cdot a^{-1} \bmod (p) \equiv 1$) of any number a and modulus p exists if and only if a and p are co-prime, where co-prime is defined to mean that the greatest common divisor (gcd) of a and p is 1 [18].

Combining the first point with the fact that p_i is prime, we know that $\text{gcd}(\delta_r, p_i) = 1$. From the second point, we know there exists a modular multiplicative inverse of $\delta_r \bmod (p_i)$ that we will call δ_r^{-1} . Having shown δ_r^{-1} to exist, we multiply both sides of (10) by it, resulting in

$$t \equiv -\delta_j \cdot \delta_r^{-1} \bmod (p_i). \quad (11)$$

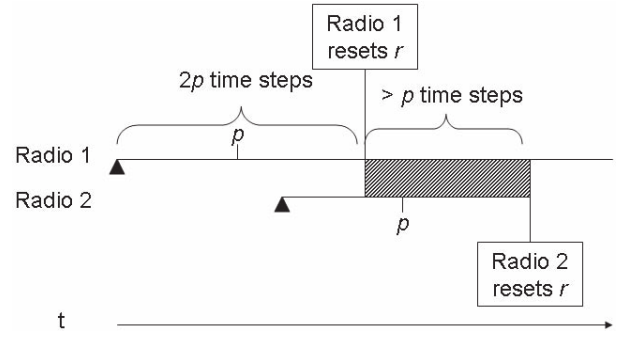


Fig. 6. Illustration of how the $2p_i$ time-out of the modular clock algorithm guarantees that under the shared model the radios have at least p_i time slots with fixed r_i .

Since the right-hand side of (11) is a value in $[0, p_i)$, our t value is in $[0, p_i)$ and we have proven convergence will occur in $t < p_i$ steps. \square

Proposition 3. *Under the shared model when two radios perform the modular clock algorithm simply following (8) with $r_1 = r_2$ and $j_1^0 \neq j_2^0$, rendezvous will not occur.*

Proof. We use the same notation from the proof of Theorem 2, proving by contradiction.

Assume that

$$t \cdot \delta_r \equiv -\delta_j \bmod (p_i). \quad (12)$$

Since $j_i^0 \in [0, p_i)$ with $j_1^0 \neq j_2^0$, this means that $\delta_j \in (-p_i, 0) \cup (0, p_i)$. Furthermore, since $r_1 = r_2$, this means that $\delta_r = 0$. Thus, (12) can be rewritten as

$$0 \equiv -\delta_j \bmod (p_i). \quad (13)$$

However, since $\delta_j \in (-p_i, 0) \cup (0, p_i)$, the right side of the equation is in $(0, p_i)$ and the equality cannot hold. We have contradicted our base assumption. \square

Another way of looking at Proposition 3 is as follows: since the δ_r value is 0, the sequences maintain the same difference in channel index (δ_j) they began with throughout the algorithm's operation.

To fix this, the modular clock algorithm has a "time-out" feature on line 6. By waiting until $2p_i$ time slots have occurred before timing out and switching r_i values, the radios can ensure that they have had at least p_i time without either radio changing their r_i . For example, assume Radio 1 begins a particular sequence at time slot 0, while Radio 2 begins its sequence at time slot $p_i - \epsilon$. If Radio 1 waits until $t = 2p_i$ to change r_i values, then they will have maintained fixed r_i values for $p_i + \epsilon$ time slots together, which guarantees from Theorem 2 that rendezvous will have enough time to occur. Fig. 6 illustrates this idea.

4.3.2 Shared Model Performance

We begin by analyzing the worst-case performance of the algorithm under the shared model. By Theorem 2, if $r_1 \neq r_2$ rendezvous is guaranteed within p_i time slots. Otherwise, the radios time-out and change their r_i values. With probability $1 - 1/p_i$, they will select different r_i values; with probability $1/p_i$ they will choose equal r_i values and again fail to rendezvous. If the radios fail to rendezvous by the p_i th change of rate r_i , the radios will have spent $2p_i \cdot p_i$ time slots.

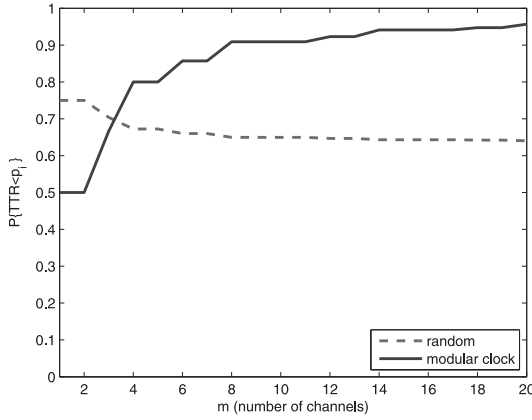


Fig. 7. Probability of rendezvous in fewer than p_i slots under the shared model for the random and modular clock algorithm, for increasing values of m ($m = \bar{m} = m_i$).

The probability of this occurring is $1/p_i^{p_i}$. Therefore, with probability $1 - 1/p_i^{p_i}$ rendezvous will occur in no worse than $O(p_i^2)$ time. In other words, with high probability (but not guaranteed), it will exceed the performance of the generated orthogonal sequence algorithm.

To determine an upper bound on the expected performance, we observe that with $1 - 1/p_i$ probability the radios will choose different r_i values and rendezvous in less than $2p_i$ time. Extending this, with probability $(1/p_i)(1 - 1/p_i)$ the radios will rendezvous in less than $4p_i$ time, and the following infinite series expresses a bound on the expected TTR:

$$E[TTR] \leq \sum_{k=1}^{\infty} (2p_i k) \left(\frac{1}{p_i}\right)^{k-1} \left(1 - \frac{1}{p_i}\right). \quad (14)$$

This reduces to the closed form expression:

$$E[TTR] \leq 2p_i + \frac{2p_i}{p_i - 1}. \quad (15)$$

Therefore, the maximum expected TTR under the shared model is only slightly larger than $2p_i$.

We now desire to compare the performance of the modular clock algorithm against the random algorithm under the shared model. For the modular clock algorithm, we know from Proposition 3 that if $r_1 \neq r_2$, we are guaranteed convergence in less than p_i time slots. Thus, the probability of converging in less than p_i time slots is the same as the probability that $r_1 \neq r_2$.

$$P_{\text{modclock}}\{TTR < p_i\} = 1 - \frac{1}{p_i}. \quad (16)$$

For the random algorithm, the probability of converging in less than p_i is the opposite of the probability of not converging in any of the previous p_i time slots, or

$$P_{\text{random}}\{TTR < p_i\} = 1 - \left(1 - \frac{1}{p_i}\right)^{p_i}. \quad (17)$$

In Fig. 7, we plot these two probabilities for increasing values of m . We see that for $m > 3$, the modular clock algorithm has a better probability of rendezvousing in less than p_i time slots than the random algorithm. Therefore,

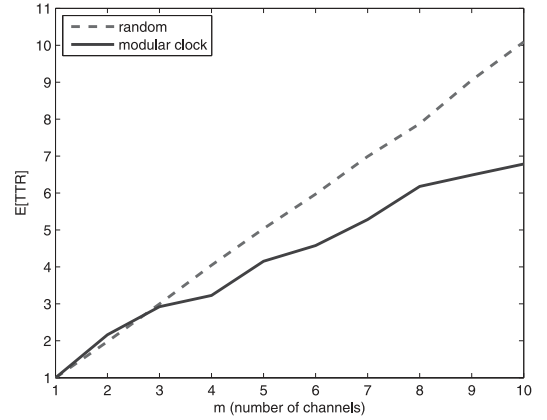


Fig. 8. Expected TTR of the modular clock algorithm under the shared model, in which $m = \bar{m}$.

under the shared model, if radios observe more than three channels, they should implement the modular clock rather than the random algorithm.

To investigate these results further, we simulated the modular clock and random algorithms and measured the expected TTR. The simulation results in Fig. 8 confirm the results in Fig. 7, showing a crossover point for the expected TTR at approximately three channels. This implies that the actual expected TTR function grows at a rate less than in (15). It also compares favorably against Anderson and Weber's expected TTR of $0.828 \cdot m$ (discussed in Section 4.1.1) and, unlike their algorithm, does not require the radios to synchronize the starting of their rendezvous processes.

4.3.3 Individual Model Correctness

We have shown in the previous section that under the shared model, the modular clock algorithm is correct and has good worst case, expected, and comparative performance characteristics. However, what about under the more demanding individual model, where observed channels and channel labeling may not be the same, and the only assumption is that $\bar{m} \geq 1$?

Theorem 4. *Under the individual model, when two radios perform the modular clock algorithm simply following (8) and $p_1 \neq p_2$, rendezvous will occur within $p_1 p_2$ time slots.*

Proof. Unlike the shared model analysis, we cannot assume that the labeling is the same for both radios and that the same indices represent the same channels. Therefore, to prove that rendezvous is guaranteed to occur, we need to be able to show there exists a t such that

$$\begin{aligned} j_1^t &\equiv t \cdot r_1 + j_1^0 \pmod{p_1}, \\ j_2^t &\equiv t \cdot r_2 + j_2^0 \pmod{p_2}, \end{aligned} \quad (18)$$

for any values of indices j_1^t and j_2^t , initial indices j_1^0 and j_2^0 , rates r_1 and r_2 , and prime numbers p_1 and p_2 (where $p_1 \neq p_2$).

We rewrite (18) (noting that r_i^{-1} exists because r_i and p_i are co-prime):

$$\begin{aligned} t &\equiv r_1^{-1}(j_1^t - j_1^0) \pmod{p_1}, \\ t &\equiv r_2^{-1}(j_2^t - j_2^0) \pmod{p_2}. \end{aligned} \quad (19)$$

The Chinese Remainder Theorem (CRT) [18] states that if p_1 and p_2 are co-prime, then for *any* a_1 and a_2 , there exists a solution x for

$$\begin{aligned} x &\equiv a_1 \pmod{p_1}, \\ x &\equiv a_2 \pmod{p_2}. \end{aligned}$$

Since p_1 and p_2 are prime and not equal, they are also co-prime. Therefore, from the CRT, there exists a t that solves (19), proving our theorem. Since there are $p_1 p_2$ combinations of a_1 and a_2 values, rendezvous will occur in at most $p_1 p_2$ time slots. \square

The following is the analysis of the degenerate case for the modular clock algorithm.

Proposition 5. *Under the individual model, when two radios perform the modular clock algorithm simply following (8) and $C_1 \neq C_2$, $\bar{m} \geq 1$, and $p_1 = p_2$, it is not possible to guarantee rendezvous.*

When $p_1 = p_2$ (which can occur whenever the next largest prime to m_1 and m_2 is the same, such as when $m_1 = m_2$) the CRT does not apply (because the moduli are not co-prime). Furthermore, Theorem 2 does not apply because $C_1 \neq C_2$. It is trivial to come up with examples where rendezvous does not occur, although through simulation, we have observed that there exist scenarios where rendezvous will still happen.

4.3.4 Individual Model Performance

When not operating under the conditions described in Proposition 5, it is possible to make a statement about how frequently rendezvous will occur. If \bar{m} is the set of common channels, from Theorem 4 within $p_1 p_2$ time slots, rendezvous will occur \bar{m} times. Since we have \bar{m} rendezvous points over the span of $p_1 p_2$ time, the expected TTR is

$$E[TTR] = \frac{p_1 p_2}{\bar{m}}. \quad (20)$$

Thus, as long as $\bar{m} \geq 1$ and $p_1 \neq p_2$, rendezvous will occur in at most $p_1 p_2$ time slots.

Of course, this assumes the radios do not time-out and change rates. Since the time-out in the modular clock algorithm is set to $2p_i$, this means that radios will always (so long as $p_i > 2$) have a time-out value that is too soon to guarantee rendezvous. To determine how often this premature time-out effects the expected TTR, we simulated a scenario in which one radio observes 10 channels and the other radio sees between \bar{m} and 20, with the two radios both seeing the same \bar{m} channels, which is varied between 1 and 9. Fig. 9 illustrates the results, which show that in all cases (including the $\bar{m} = 1$ case), the modular clock algorithm takes, on average, less than $p_1 p_2$ time slots. The gap for $m_2 \in [8, 11]$ exists because for these values of m_1 , both p_1 and p_2 are set to 11, which falls under Proposition 5.

4.4 Modified Modular Clock Algorithm

To avoid the case represented by Proposition 5, we modify the modular clock algorithm to randomize the selection of primes within a certain range. This modified modular clock algorithm is described in Algorithm 4. Under the modified modular clock algorithm, a random prime number p_i is

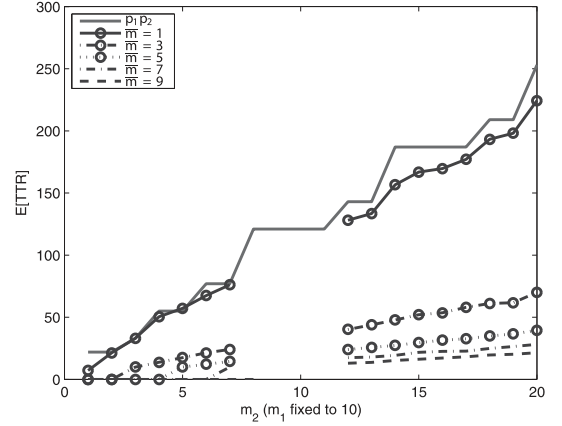


Fig. 9. Simulated expected TTR of the modular clock algorithm under the individual model (where $p_1 \neq p_2$). m_1 and \bar{m} are varied while m_2 is fixed at 10. $p_1 p_2$ is the theoretical upper bound on $E[TTR]$.

chosen from the set of primes in $[m_i, 2m_i]$. Furthermore, in the modified modular clock algorithm, instead of consistently remapping all channel indices between m_i and p_i (as was done in the modular clock algorithm), the radio chooses randomly. Since p_i is no larger than $2m_i$, a radio will spend no more than half its index selections performing this random selection.

Algorithm 4. Modified Modular Clock Algorithm

- 1: observe m_i , the number of channels available
- 2: $j_i^0 = \text{rand}[0, m_i]$
- 3: **while** not rendezvous **do**
- 4: choose r_i from $[0, m_i]$
- 5: choose p_i , a prime from $[m_i, 2m_i]$
- 6: **for** $t = 0$ to $2p_i^2$ **do**
- 7: $j_i^{t+1} = (j_i^t + r_i) \pmod{p}$
- 8: **if** $j_i^{t+1} < m_i$ **then**
- 9: $c = c_{i, j_i^{t+1}}$
- 10: **else**
- 11: $c = c_{i, \text{rand}([0, m_i])}$
- 12: **end if**
- 13: attempt rendezvous on channel c
- 14: **end for**
- 15: **end while**

4.4.1 Individual Model Correctness

If the two radios avoid selecting the same prime modulus, rendezvous is guaranteed within $p_1 p_2$ time slots by Theorem 4. Therefore, instead of timing out at $2p_i$ time slots as in the modular clock algorithm, on Line 6 of Algorithm 4, we change the time-out value to $2p_i^2$ time slots. While the correct time-out value to detect equal prime modulus would be $2p_1 p_2$, this requires knowledge of the prime number selection of both radios at both radios, something that is impossible. $2p_i^2$ is a reasonable approximation if p_1 is close to p_2 , but if there exists a large difference between p_1 and p_2 , the radio with the smaller p_i value may reset too quickly.

To determine how often the degenerate case, described in Proposition 5, occurs under the modified modular clock, we determine the probability that the same prime numbers are

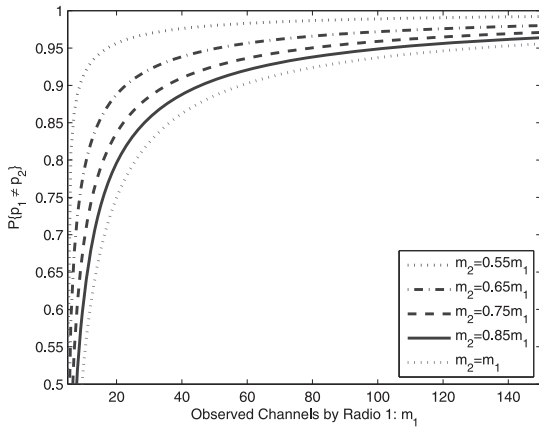


Fig. 10. Evaluating the probability of choosing different prime numbers under the modified modular clock algorithm for various ratios and values of m_1 and m_2 .

selected by the radios. By the prime number theorem [18], the number of primes from 1 to x can be approximated by $\pi(x) \approx x/\ln(x)$. Therefore, the number of primes between x and y (where $x > y$) is approximated by the function:

$$\pi(x, y) \approx \min(0, x/\ln(x) - y/\ln(y)). \quad (21)$$

If Radio 1 is arbitrarily selected to have a larger channel set size than Radio 2, and Radio 2's channel set size is expressed as a fraction of Radio 1's (i.e., $m_1 \geq m_2, m_2 = \alpha m_1, \alpha \in (0, 1]$) then, the probability of both radios not picking the same prime number can be calculated as

$$P\{p_1 \neq p_2\} = 1 - \frac{\pi(m_1, 2\alpha m_1)}{\pi(m_1, 2m_1)\pi(\alpha m_1, 2\alpha m_1)}. \quad (22)$$

Fig. 10 evaluates this probability for various values of α . For values of $\alpha \leq 0.5$, the probability is 1, since there is no overlap of the sets the prime numbers are selected from.

In general, for a fixed ratio of observed channels, increasing the numbers of observed channels by each radio increases the probability of the different prime numbers being selected. Also, decreasing the α value creates a smaller intersection of the sets of possible prime numbers, further increasing the probability of selecting different primes. Fig. 10 illustrates these trends, showing that to have a 90 percent probability of selecting different primes, it requires 90 percent fewer channels when $m_2 = 0.55m_1$ than when $m_2 = m_1$.

Fig. 11 illustrates the expected TTR for two radios that see the same number of available channels for increasing values of \bar{m} . The minimum $p_1 p_2$ line is effectively $(p_i)^2$, where p_i is the next largest prime number to m_i . The modified modular clock algorithm will select prime numbers randomly from $[m_i, 2m_i]$, so $p_1 p_2$ (and the theoretical expected maximum TTR for any \bar{m}) is quite likely to be larger than this value. The simulated data show, however, that this case must not occur frequently, as the expected TTR is close to or less than than the minimum $p_1 p_2$ line.

5 CONCLUSIONS AND FUTURE WORK

The goal of this research was to devise efficient means to achieve rendezvous for cognitive radios under a wide range

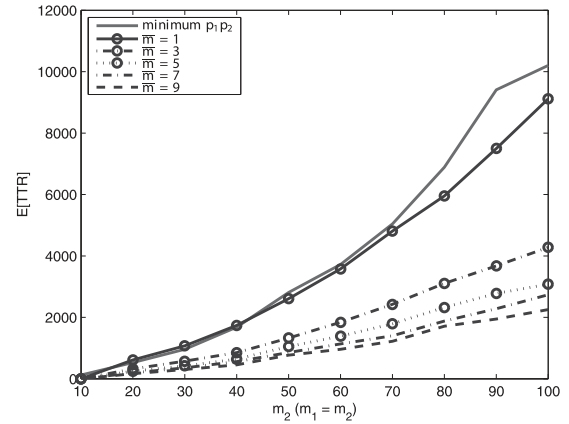


Fig. 11. Simulated expected TTR of the modified modular clock algorithm under the individual model. $m_1 = m_2$ and \bar{m} are varied. Minimum $p_1 p_2$ is the theoretical lower bound bound on maximum $E[TTR]$.

of possible system, policy, and environmental conditions. Four rendezvous algorithms were presented, providing different performance guarantees depending on the assumptions of the system model. The random algorithm provides robust operation under all condition models. The generated orthogonal sequence-based algorithm provides a bounded TTR with the ability to prioritize channels under the shared model. The modular clock algorithms, while unbounded, provides a potential m_i speedup in expected TTR, while requiring less precoordination and the flexibility to operate in both the shared and individual models.

There are several directions for future work. This paper analyzes rendezvous between just two radios. The free-for-all model requires rendezvous between an arbitrary number of radios. Any of the algorithms described in this paper could be extended to multiple radios via a combining approach: pairs of rendezvoused radios synchronize their search algorithms, searching together until they rendezvous with an additional radio or another synchronized pair of radios, repeating until all radios have rendezvoused on a common channel. Alternatively, the modular clock algorithm (under the individual model) extends to any arbitrary number of radios (because of the properties of the CRT), so long as the co-prime requirement is met for all p_i . However, both of these approaches are unlikely to be practical since as n increases, the likelihood of finding a single common channel for all radios becomes difficult. This means that different radios may have to rendezvous on different channels, and switch between these channels to bridge between them. Efficient rendezvous algorithms under these scenarios take on aspects of topology control, with a need to consider other factors during rendezvous, such as the number of rendezvous channels, diameter of the rendezvous network, and the connectivity of the radios.

The lack of a control channel or controller introduces additional possibility of unintended interference, both between SUs and between SUs and PUs. Particularly, in dense or spectrally constrained networks, there is the possibility of rendezvousing SU radios interfering with one another during the beacon portion of their handshake phase. Also, without a defined control channel, these

rendezvous algorithms may accidentally interfere with a PU when it unexpectedly returns to a channel that had been sensed as being vacant. Eliminating or minimizing these interactions is a desirable improvement.

A handshake algorithm needs to be formalized and defined. Arachchige et al. [10] developed a handshake for the leader election and neighbor discovery problem, but this is specific to their approach. A general handshaking mechanism of beaconing and listening in a channel before moving to another channel needs to be developed.

Finally, other than Ma's investigation of security for rendezvousing after disruption, the free-for-all model's allowance for malicious radios has not been investigated. Successfully rendezvousing in a hostile environment is a challenge. Rendezvous jamming models and performance metrics need to be developed and applied to existing rendezvous algorithms. These algorithms may need to be "hardened" to protect them against attacks.

ACKNOWLEDGMENTS

This research is funded in part by the US Air Force Research Lab and by the US Air Force Office of Scientific Research. The views expressed in this paper are those of the authors, and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government. This document has been approved for public release; distribution unlimited.

REFERENCES

- [1] K. Balachandran and J. Kang, "Neighbor Discovery with Dynamic Spectrum Access in Adhoc Networks," *Proc. IEEE Vehicular Technology Conf. (VTC '06)*, vol. 2, pp. 512-517, May 2006.
- [2] M.M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans, "DIMSUMNet: New Directions in Wireless Networking Using Coordinated Dynamic Spectrum Access," *Proc. Sixth IEEE Int'l Symp. World of Wireless Mobile and Multimedia Networks (WOWMOM '05)*, pp. 78-85, 2005.
- [3] Y. Xiao and F. Hu, *Cognitive Radio Networks*. CRC Press, 2008.
- [4] J. Mo, H.-S.W. So, and J. Warland, "Comparison of Multichannel MAC Protocols," *IEEE Trans. Mobile Computing*, vol. 7, no. 1, pp. 50-65, June 2008.
- [5] P. Sutton, K. Nolan, and L. Doyle, "Cyclostationary Signatures for Rendezvous in OFDM-Based Dynamic Spectrum Access Networks," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '07)*, pp. 220-231, Apr. 2007.
- [6] L. Cao and H. Zheng, "Distributed Spectrum Allocation Via Local Bargaining," *Proc. Second Ann. IEEE Comm. Soc. Conf. Sensor and Ad Hoc Comm. and Networks (SECON '05)*, pp. 475-486, Sept. 2005.
- [7] B. Horine and D. Turgut, "Link Rendezvous Protocol for Cognitive Radio Networks," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '07)*, pp. 444-447, Apr. 2007.
- [8] B. Horine and D. Turgut, "Performance Analysis of Link Rendezvous Protocol for Cognitive Radio Networks," *Proc. Second IEEE Int'l Conf. Cognitive Radio Oriented Wireless Networks and Comm. (CROWNCOM '07)*, pp. 503-507, Aug. 2007.
- [9] P. Sutton, K. Nolan, and L. Doyle, "Cyclostationary Signatures in Practical Cognitive Radio Applications," *IEEE J. Selected Areas in Comm.*, vol. 26, no. 1, pp. 13-24, Jan. 2008.
- [10] C. Arachchige, S. Venkatesan, and N. Mittal, "An Asynchronous Neighbor Discovery Algorithm for Cognitive Radio Networks," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '08)*, pp. 1-5, Oct. 2008.
- [11] M.D. Silvius, F. Ge, A. Young, A.B. MacKenzie, and C.W. Bostian, "Smart Radio: Spectrum Access for First Responders," *Proc. SPIE*, 2008.
- [12] S. Alpern and S. Gal, *The Theory of Search Game and Rendezvous*. Springer, 2003.

- [13] Bluetooth SIG, "How Bluetooth Works—Communications," 2008.
- [14] E. Anderson and R. Weber, "The Rendezvous Problem on Discrete Locations," *J. Applied Probability and Statistics*, vol. 28, pp. 839-851, 1990.
- [15] L. Ma and C.-C. Shen, "Security-Enhanced Virtual Channel Rendezvous Algorithm for Dynamic Spectrum Access Wireless Networks," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '08)*, pp. 1-9, Oct. 2008.
- [16] L. DaSilva and I. Guerreiro, "Sequence Based Rendezvous for Dynamic Spectrum Access," *Proc. IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '08)*, pp. 1-7, Oct. 2008.
- [17] D. Sarwate, "Optimum PN Sequences for CDMA Sequences," *Proc. IEEE Symp. Spread Spectrum Techniques and Applications*, pp. 27-35, 1994.
- [18] D.R. Stinson, *Cryptography: Theory and Practice*. CRC Press, 1995.



Nick C. Theis received the BS degree in computer science from the University of Michigan in 2002 and the MS degree in computer science from the Air Force Institute of Technology, Wright Patterson AFB, Ohio, in 2009. He is a senior network operator with the 57th Information Aggressor squadron at Nellis AFB, Nevada. His areas of academic interest span a wide variety of network security and network architecture disciplines. He is a member of the IEEE.



Ryan W. Thomas received the BS degree from Harvey Mudd College in 1999, the MS degree in computer engineering from the Air Force Institute of Technology in 2001, and the PhD degree in computer engineering from the Virginia Polytechnic Institute and State University in 2007. He is an assistant professor of computer engineering in the Department of Electrical and Computer Engineering at the Air Force Institute of Technology. He previously worked as a digital antenna array engineer at the Air Force Research Laboratory, Sensors Directorate. His research focuses on the design, architecture and evaluation of cognitive networks, cognitive radios, and software defined radios. He is a member of the IEEE.



Luiz A. DaSilva has been at Trinity College Dublin since May 2009. There, he holds the Stokes professorship in telecommunications in the Department of Electronic and Electrical Engineering. He is currently on leave from the faculty of the Virginia Polytechnic Institute and State University's Bradley Department of Electrical and Computer Engineering, which he first joined in 1998. His research focuses on distributed and adaptive resource management in wireless networks, and, in particular, cognitive radio networks. Other areas of current interest include dynamic spectrum access and the application of game theory to modeling distributed wireless networks. He is a senior member of the IEEE and a member of the ASEE and of the ACM.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.