Multicast Rendezvous in Fast-Varying DSA Networks

Mohammad J. Abdel-Rahman, *Member, IEEE*, Hanif Rahbari, *Member, IEEE*, and Marwan Krunz, *Fellow, IEEE*

Abstract—Establishing communications between devices in a dynamic spectrum access (DSA) system requires the communicating parties to "rendezvous" before transmitting data packets. Frequency hopping (FH) is an effective rendezvous method that does not rely on a predetermined control channel. Previous FH-based rendezvous designs mainly target unicast rendezvous, and do not intrinsically support *multicast rendezvous*, where a group of nodes need to rendezvous simultaneously. Furthermore, these designs do not account for *fast* primary user (PU) dynamics, leading to long time-to-rendezvous (TTR). In this paper, we exploit the *uniform k-arbiter* and *Chinese Remainder Theorem* quorum systems to develop three FH-based multicast rendezvous algorithms, which provide different tradeoffs between rendezvous efficiency (e.g., low TTR) and security (e.g., robustness to node compromise). Our rendezvous algorithms are tailored for asynchronous and spectrum-heterogeneous DSA systems. To account for fast PU dynamics, we develop an algorithm for adapting the proposed FH designs on the fly. This adaptation is done through efficient mechanisms for channel ordering and quorum selection. Our simulations validate the effectiveness of the proposed rendezvous algorithms, their PU detection accuracy, and their robustness to insider attacks.

Index Terms—Channel sorting, dynamic frequency hopping, dynamic spectrum access, multicast rendezvous, quorum systems

1 Introduction

Motivated by the need for more efficient utilization of the licensed spectrum, and supported by recent regulatory policies (e.g., [12]), significant research has been conducted towards enabling dynamic spectrum access (DSA) networks. The communicating entities in these networks, called *secondary users* (SUs), can utilize the available spectrum in a dynamic and opportunistic fashion without interfering with co-located *primary users* (PUs). Enabling opportunistic operation requires addressing various challenges including channel access, device coordination, and various security issues. Specifically, SU devices that are built on a software-defined radio engine are particularly vulnerable to code injection and node compromise attacks.

Establishing a link between two or more SU devices requires them to rendezvous (i.e., meet on a common channel[1] at some point in time) and exchange control messages needed for connection establishment. In the absence of centralized control, the rendezvous problem is quite challenging because of the spatiotemporal variations in channel availability. Further challenges arise in the absence of node synchronization. To address the rendezvous problem, many existing MAC protocols for DSA networks rely on a dedicated control channel. While presuming a common control channel (CCC) simplifies the rendezvous process, it comes with two main drawbacks. First, a CCC can easily become a network bottleneck and a prime target for selective jamming attacks [4], [22]. Second, PU dynamics and spectrum heterogeneity make it difficult to always maintain a single dedicated CCC [23].

Frequency hopping (FH) provides an alternative method for rendezvousing without relying on a predetermined CCC. One systematic way of constructing FH sequences is to use quorum systems [13]. Quorum-based FH designs have two key advantages. First, they provide deterministic guarantees on the overlap between the FH sequences. Second, they are robust to synchronization errors [16].

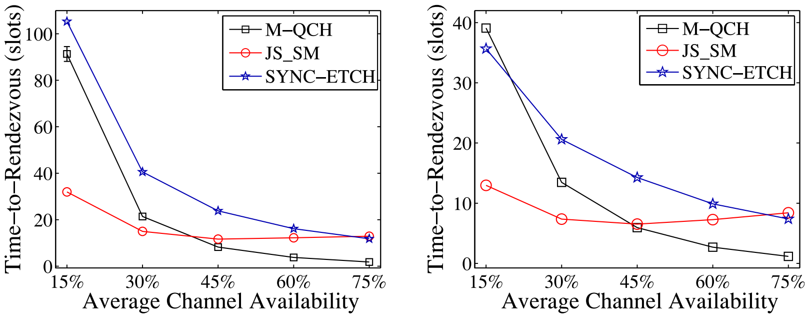
Fast PU dynamics—In DSA networks, different channels experience different patterns of PU activity, resulting in different average channel availability (a.k.a. percentage occupancy). The mean duration of the duty cycle, defined as the time between two successive idle-to-busy PU transitions, for partially occupied channels can, in general, take values from tens of seconds to several hours [14], [24].[2] Previous FH-based rendezvous designs ignore PU-related channel variations that occur during the rendezvous process. This can result in excessively long time-to-rendezvous (TTR). To account for such variations, the average channel availability time and its fluctuation level need to be considered when constructing the FH sequences. In current FH-based rendezvous designs, channel availability is often modeled as a binary variable. This simplistic approach does not capture differences in the time-averaged fraction of channel availability. By incorporating the time-averaged channel availability, our modeling approach provides an effective tool

[1] We use the terms channel and frequency interchangeably.

[2] Channel statistics, such as the PDF of idle periods, are channel-dependent. For example, the idle period of E-GSM 900 downlink (DL) channel number 23 can last up to one minute, whereas this value can exceed 15 minutes for DCS 1800 DL channel number 70. The busy periods are at most three hours and three minutes for the respective channels [24].

- *The authors are with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721.*
E-mail: {mjabdelrahman, rahbari, krunz}@email.arizona.edu.

Manuscript received 22 May 2014; revised 24 Aug. 2014; accepted 28 Aug. 2014. Date of publication 15 Sept. 2014; date of current version 1 June 2015.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2014.2356475



(a) T_{cycle} = 3 slots (b) T_{cycle} = 24 slots

Fig. 1. Fixed FH designs result in large TTR under fast PU dynamics. The frame length of M-QCH is 3 and the period is $3 \times 6 = 18$. The frame length of SYNC-ETCH is $2 \times 6 + 1 = 13$ and its period is $6 \times 13 = 78$. The frame length of JS_SM is $3 \times 7 = 21$.

for designing FH rendezvous protocols that are robust to fast PU dynamics.

To illustrate the effect of channel dynamics on the TTR, we simulate three previously proposed FH-based *unicast* rendezvous algorithms, M-QCH [8], JS_SM [20], and SYNC-ETCH [31], under different average availability times and different mean duty cycles (T_{cycle}). T_{cycle} reflects the fluctuation level of a channel; channels with higher T_{cycle} exhibit less fluctuations. The algorithms are simulated under a simplified setup, where nodes are synchronized, spectrum views are homogeneous (i.e., SUs perceive the same spectrum opportunities), and nodes start the rendezvous process at the same time. For JS_SM (M-QCH and SYNC-ETCH), sensing is performed on a per frame (slot) basis. Six channels that have the same statistics are used in the experiment. M-QCH was proposed in [17] to minimize the TTR in a synchronous environment. However, as shown in Fig. 1, its TTR is high when channel availability is low. The TTR of all considered algorithms is inversely proportional to T_{cycle}. JS_SM is less affected by the average availability time than M-QCH and SYNC-ETCH. In contrast to M-QCH and SYNC-ETCH, only "potentially available" channels are used in constructing the FH frame in JS_SM. The relatively small TTR of JS_SM comes at the cost of a high collision rate,[3] as shown in Fig. 2. In a more realistic setting with asynchronous operation and heterogeneous-spectrum opportunities, the effect of PU dynamics on the TTR is even more severe. Readers may refer to Section VIII-A of our technical report [3] for a more detailed comparison of different unicast rendezvous schemes under fast PU dynamics.

Our contributions.

- Previous rendezvous schemes are intended for unicast operation, and they do not intrinsically support multicast rendezvous. In multicast rendezvous, a subset of nodes forms a multicast group. Group members need to rendezvous simultaneously in the same time slot. We propose three FH algorithms for asynchronous multicast rendezvous in DSA networks: k-arbiter multicast quorum-based FH rendezvous (AMQFH), CRT multicast quorum-based FH rendezvous (CMQFH), and nested-CMQFH, which provide different tradeoffs between rendezvous efficiency (low TTR) and security (robustness to node compromise).
- We develop an algorithm for adapting the hopping pattern in the proposed FH designs on the fly, depending on estimated PU dynamics. To achieve this adaptation, we develop an optimal channel ordering mechanism for channel sensing and assignment, and an efficient quorum selection mechanism.

Paper organization—In Section 2, we present a literature review. We provide an overview of our proposed framework in Section 3. In Section 4, we present the system and channel models and the evaluation metrics. The 'basic' multicast rendezvous designs are explained in Section 5. These designs are augmented with channel ordering schemes, explained in Section 6, and adaptive quorum selection schemes, presented in Section 7. The complete rendezvous protocols are evaluated in Section 8. In Section 9, we provide directions for future research. Finally, Section 10 concludes the paper.

2 Related Work

To address the rendezvous problem, several MAC protocols were proposed for DSA networks assuming the availability of a CCC (e.g., [10], [15], [26], [32]). FH provides an alternative method for rendezvousing without relying on a predetermined CCC. One systematic way of constructing FH sequences is to use quorum systems. The consideration of quorum systems for FH-based rendezvous was pioneered by Bian et al. in [8]. Other FH-based rendezvous schemes were proposed in [6], [7], [11], [17], [20], [27], [31]. Readers may refer to the survey in [21] for a detailed categorization of existing rendezvous schemes.

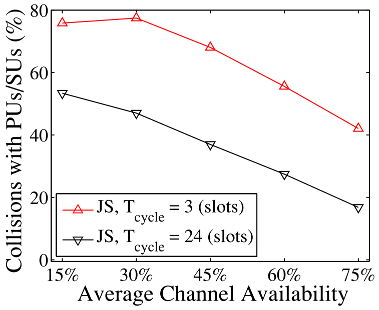


Fig. 2. Collision rate versus average channel availability for JS_SM (frame length = 21).

[3] A collision occurs if the SU attempts to access a channel in a given slot that is occupied by a PU or by another SU (outside the multicast group) at the beginning of that slot.

Previous rendezvous schemes are intended for unicast operation, and they do not intrinsically support multicast rendezvous. In [20], the authors designed an algorithm for establishing multicast communications. Instead of designing different FH sequences that overlap at common slots, multicast is supported via a series of pairwise (unicast) rendezvous operations that result in all nodes in the multicast group tuned to a common FH sequence. From a security perspective, the effectiveness of this approach cannot be maintained under node compromise, where an adversary takes control of a node and discloses its secrets. Using the approach in [20], if a node is compromised, the FH sequences of all nodes are exposed. In contrast, in our approach different nodes follow different FH sequences, so the exposure of one sequence does not jeopardize the security of other nodes.

Group-based schemes have been proposed to facilitate multicast rendezvous [23]. These schemes can be divided into two categories: (i) neighbor coordination schemes (e.g., [9]), where neighboring nodes broadcast their channel parameters to make a group-wide decision, and (ii) cluster-based schemes (e.g., [19]), where nodes are clustered according to common spectrum opportunities. One drawback of these schemes is the need for neighbor discovery prior to establishing a CCC. Furthermore, these schemes incur considerable overhead to maintain the group-based control channel. Even though these solutions establish a CCC for intra-group communications, the problem of intergroup communications is yet another challenge that have not been addressed [23].

In [22], the authors proposed an FH-based jamming-resistant broadcast communication scheme, in which the broadcast operation is implemented as a series of unicast transmissions, distributed in time and frequency. Implementing multicast as a series of unicasts can lead to multicast inconsistency. For example, a group of SUs may share a *group key* that is used to securely communicate messages between them. For security purposes, this key is updated periodically (i.e., a rotating key) [28]. However, a change in the group key has to be time-consistent among all members of the multicast group. Such consistency cannot be guaranteed if key updates are conveyed using a series of unicast transmissions.

All existing rendezvous schemes do not account for fast channel variations, where channel availability can vary during the rendezvous process itself. To the best of our knowledge, this is the first paper that addresses the multicast rendezvous problem in a fast-varying DSA environment.

3 Design Motivation

In this section, we motivate our approach for designing FH-based multicast rendezvous algorithms. In multicast rendezvous, a subset of nodes that forms a multicast group aims to rendezvous simultaneously in the same time slot. Our algorithms are based on the theory of quorum systems. To facilitate the understanding of these algorithms, we first provide a number of definitions related to quorum systems. Readers may refer to [29] for more details about quorum systems.

3.1 Quorum Systems

Definition 1. *Given a set of non-negative integers* $Z_n = \{0, 1, \ldots, n-1\}$*, a quorum system* Q *under* Z_n *is a collection of non-empty subsets of* Z_n*, each called a quorum, such that:* $\forall G \text{ and } H \in Q,\ G \cap H \neq \emptyset$.

Throughout the paper, Z_n is used to denote the set of non-negative integers less than n.

Definition 2. *Given a non-negative integer* i *and a quorum* G *in a quorum system* Q *under* Z_n*, we define* $rotate(G, i) = \{(x + i) \bmod n, x \in G\}$ *as a cyclic rotation of* G *by* i*.*

Definition 3. *A quorum system* Q *under* Z_n *satisfies the rotation* k*-closure property for some* $k \geq 2$ *if* $\forall\ G_1, G_2, \ldots, G_k \in Q$ *and* $\forall i_1, i_2, \ldots, i_k \in Z_n$*,* $\bigcap_{j=1}^{k} rotate(G_j, i_j) \neq \emptyset$*.*

3.2 Basic Idea and Key Assumptions

We consider two types of quorum systems for designing our multicast rendezvous algorithms: The uniform k-arbiter and Chinese remainder theorem (CRT) quorum systems. These quorum systems enjoy the intersection and rotation closure properties, explained in Definitions 1 and 3 above. The basic idea of our algorithms is to exploit the properties of these quorum systems by embedding one quorum in each frame of every FH sequence, such that all quorums are derived from the same quorum system. Because the quorums are guaranteed to overlap even when they are cyclically rotated, SUs that follow the constructed FH sequences are guaranteed to rendezvous even when they are time-misaligned.

The proposed algorithms have two main attractive features. First, they allow nodes to construct their FH sequences independently by knowing only the size (but not identities) of the multicast group, hence they can be executed in a distributed way. Second, these algorithms can still function in the absence of node synchronization. The size of the multicast group is conveyed through a series of unicast rendezvous operations between the multicast initiator and each of the multicast group members. Note that an SU can be a member of multiple multicast groups at the same time. In this case, the SU needs to know the size of any multicast group it belongs to. This information is obtained from the corresponding multicast initiator. The SU constructs an FH sequence for every multicast group it is associated with. It uses the same FH sequence for all multicast groups that have the same size. In our work, we assume that the multicast group memberships are not directly related to topological changes (i.e., the multicast groups do not change rapidly).

4 Models and Metrics

4.1 System Model

We consider a single-hop[4] ad hoc opportunistic DSA network, operating over L licensed channels $\mathcal{L} = \{f_1, f_2, \ldots, f_L\}$. SUs can successfully transmit over these channels if they are not occupied by PUs. Without loss of generality, we assume that FH occurs on a per-slot basis, with a slot

[4] In Section 9, we provide directions for exploiting our multicast rendezvous algorithms to achieve unicast rendezvous in multi-hop DSA networks.

duration of T seconds. A packet can be exchanged between two or more nodes if they hop onto the same channel during the same time slot. The slot duration is selected such that: (i) one half of a slot is enough for exchanging one rendezvous message, and (ii) it is short enough to capture the fast PU dynamics. We assume a slot duration in the order of 10s of milliseconds can satisfy both of the above two requirements.

Each SU j, $j=1,\dots,K$, has a unique FH sequence $\boldsymbol{w}^{(j)}$, to be designed. The channel used in the ith slot of FH sequence $\boldsymbol{w}^{(j)}$ is denoted by $w_i^{(j)}$, $w_i^{(j)} \in \mathcal{L}$. Channel f_j is called a *rendezvous frequency* for nodes $1,2,\dots,K$ if there exists a *rendezvous slot* i such that $w_i^{(m)}=f_j$, $\forall m \in \{1,\dots,K\}$. In our setup, each FH sequence is divided into several *time frames*. Each frame corresponds to a block of time-frequency pairs.

4.2 Channel Activity Model

We assume that each channel $f_m, m \in \mathcal{L}$, can be in one of three states: (i) idle (state 1), (ii) occupied by a PU (state 2), or (iii) occupied by an SU other than the SUs that form the rendezvous group, which we refer to as 'external' SUs (state 3). Transitions between these states follow a continuous-time Markov chain with state space $S=\{1,2,3\}$. For any i and j in S, $i \neq j$, $\alpha_{ij}^{(m)}$ represents the rate at which channel f_m transitions from state i to state j. Let $\rho_i^{(m)}$ denote the total rate at which channel f_m leaves state i, i.e., $\rho_i^{(m)} = \sum_{j\neq i} \alpha_{ij}^{(m)}$. Because an SU is not allowed to access channels occupied by PUs, a channel cannot directly go from state 2 to state 3, i.e., $\alpha_{23}^{(m)}=0, \forall m \in \mathcal{L}$. In contrast, when a PU becomes active on a channel occupied by an SU, the SU needs to leave that channel immediately, so $\alpha_{32}^{(m)} \neq 0$ in general. Let $A^{(m)}$ be the generator matrix for channel f_m. The (i,j) entry of $A^{(m)}$ equals to $\alpha_{ij}^{(m)}$ if $i \neq j$ and equals to $-\rho_i^{(m)}$ if $i=j$. We assume that PUs become active on channel f_m with rate $\lambda_p^{(m)}$, and terminate their activity with rate $\mu_p^{(m)}$, both according to Poisson processes. Similarly, 'external' SUs arrive on channel f_m with rate $\lambda_s^{(m)}$ and depart with rate $\mu_s^{(m)}$, both according to Poisson processes. These channel parameters form the basis of the proposed channel ordering mechanisms. They are obtained offline prior to executing the rendezvous algorithms by performing a sufficiently long feature-detection-based monitoring of various channels. $\mathbf{A}^{(m)}$ is given by:

$$\boldsymbol{A}^{(m)} = \begin{pmatrix} -(\lambda_p^{(m)}+\lambda_s^{(m)}) & \lambda_p^{(m)} & \lambda_s^{(m)} \\ \mu_p^{(m)} & -\mu_p^{(m)} & 0 \\ \mu_s^{(m)} & \lambda_p^{(m)} & -\left(\lambda_p^{(m)}+\mu_s^{(m)}\right) \end{pmatrix}.$$

Let $\boldsymbol{P}_t^{(m)}$ be a matrix whose (i,j) entry, $p_t^{(m)}(i,j)$, is the probability that channel f_m goes from state i to state j in t seconds. It is known that $P_t^{(m)} = e^{tA^{(m)}}, t \geq 0$. Let $\boldsymbol{\pi}^{(m)} = (\pi_1^{(m)}, \pi_2^{(m)}, \pi_3^{(m)})$ be the steady-state distribution for channel f_m. Then, $\boldsymbol{\pi}^{(m)}$ can be written as:

$$\begin{aligned} \pi_1^{(m)} &= \frac{\mu_p^{(m)}\left(\lambda_p^{(m)}+\mu_s^{(m)}\right)}{\left(\lambda_p^{(m)}+\mu_p^{(m)}\right)\left(\lambda_s^{(m)}+\lambda_p^{(m)}+\mu_s^{(m)}\right)} \\ \pi_2^{(m)} &= \frac{\lambda_p^{(m)}}{\left(\lambda_p^{(m)}+\mu_p^{(m)}\right)} \\ \pi_3^{(m)} &= \frac{\mu_p^{(m)}\lambda_s^{(m)}}{\left(\lambda_p^{(m)}+\mu_p^{(m)}\right)\left(\lambda_s^{(m)}+\lambda_p^{(m)}+\mu_s^{(m)}\right)}. \end{aligned}$$

4.3 Metrics

Our proposed FH algorithms will be evaluated according to the two following metrics:

4.3.1 Expected Time-to-Rendezvous

The TTR is defined as the time until the nodes that form the multicast group rendezvous. The expected TTR is considered because of two reasons: (i) the existence of a randomly assigned part in our FH sequences and (ii) the randomness in the PU dynamics. It is to be emphasized that because channel availability may change during the rendezvous process, an overlap between the FH sequences at a time slot does not necessarily lead to a successful rendezvous. Accordingly, we are not considering the maximum TTR as a performance metric because it depends on the actual channel dynamics which are unknown. Note that the maximum time until the FH sequences overlap is different from the maximum TTR.

4.3.2 Expected Hamming Distance (HD)

The expected HD for two FH sequences $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ and $\mathbf{y} = (y_1, y_2, \dots, y_n)$ is defined as $\mathbb{E}[(\sum_{i=1}^n 1_{\{x_i \neq y_i\}})/n]$, where $1_{\{\cdot\}}$ is the indicator function and n is the frame length. To calculate the HD for a set of multicast FH sequences, we take the average HD over all pairs of FH sequences. The expected HD reflects the robustness of the FH sequences to node compromise and jamming. It quantifies the amount of information that would be leaked about the sequences of other multicast group members, when the sequence of a given member is compromised by an adversary (i.e., insider attack). The larger the expected HD, the more resilient the system is to insider attacks. However, for SUs to rendezvous, their FH sequences need to retain a certain level of similarity.

5 Multicast Rendezvous Algorithms

In this section, we present the basic designs of the multicast rendezvous algorithms and evaluate them analytically.

5.1 Uniform k-*Arbiter* *Multicast* *Quorum*-Based *FH* Rendezvous

The AMQFH algorithm is based on the uniform k-arbiter quorum system, which exhibits the rotation $(k+1)$-closure property.

Definition 4. *A quorum system Q under Z_n is called k-arbiter if every set of $k+1$ quorums $\mathcal{V}_{k+1} = \{G_1, G_2, \dots, G_{k+1}\} \subset Q$ satisfies the following $(k+1)$-intersection property [25]: $\bigcap_{i=1}^{k+1} G_i \neq \emptyset$.*

One specific type of k-arbiter quorum systems that is of interest to us is the so-called uniform k-arbiter quorum system [18]. Such a system satisfies:

$$Q = \left\{ G \subseteq Z_n : |G| = \left\lfloor \frac{kn}{k+1} \right\rfloor + 1 \right\}. \tag{1}$$

For example, the quorum system $Q = \{\{0,1,2,3\}, \{0,1,2,4\}, \{0,1,3,4\}, \{0,2,3,4\}, \{1,2,3,4\}\}$ under Z_5 is a three-arbiter quorum system. The intersection among any four quorums is not empty. This system is a uniform three-arbiter because each quorum in Q contains $\lfloor 3 \times 5/(3+1) \rfloor + 1 = 4$ elements of Z_5. It has been shown in [18] that the uniform k-arbiter quorum system exhibits the rotation (k + 1)-closure property (explained in Definition 3), which enables it to work in asynchronous environments.

To generate FH sequences that satisfy the rotation $(k+1)$-closure property using a uniform k arbiter quorum system, n needs to be selected such that the number of different quorums of length $\lfloor kn/(k+1) \rfloor + 1$ that can be derived from Z_n, denoted by φ, is greater than or equal to $k+1$, i.e.,

$$\varphi \overset{\text{def}}{=} \binom{n}{\left\lfloor \frac{kn}{k+1} \right\rfloor + 1} \geq k+1. \tag{2}$$

To satisfy (2), one can easily show that n needs to be strictly greater than $k+1$.

We now explain the basic AMQFH algorithm through an example. Consider a multicast group of three nodes. Each FH sequence consists of several time frames, each containing several slots. Because the uniform two-arbiter quorum system satisfies the rotation three-closure property (i.e., any three cyclically rotated quorums overlap in at least one slot), each FH sequence is constructed using one quorum. Thus, the frame length will be n. We set n to the smallest value that satisfies (2), i.e., $n = k+2 = 4$. The following steps are used by each node to obtain its FH sequence:

1. Construct a uniform two-arbiter system Q under Z_4. $Q = \{\{0,1,2\}, \{0,1,3\}, \{0,2,3\}, \{1,2,3\}\}$.
2. For each frame $j, j = 1, 2, \ldots$, in the FH sequence:
   - Select the quorum G_j of frame j from Q (e.g., $G_1 = \{0,1,2\}$).
   - Assign the jth rendezvous frequency $h_j \in \mathcal{L}$ to the slots that correspond to G_j.
   - Assign a random frequency to the remaining unassigned slot in the frame.

The above procedure is applied independently to other FH sequences. Fig. 3 shows three frames of FH sequences $\boldsymbol{w}$, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, constructed according to AMQFH. The three nodes in the multicast group can use any three-out-of-four FH sequences from Fig. 3. Note that the sensing time is not shown in Fig. 3. The sensing time is assumed to be in the order of a few microseconds,[5] whereas the slot duration is assumed to be in the order of 10 s of milliseconds. To achieve higher accuracy, we can use the non-quorum slot for spectrum sensing. By doing this, the rendezvous performance will not change significantly.

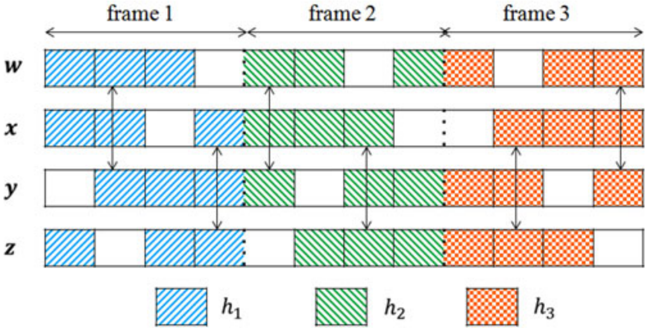


Fig. 3. AMQFH FH construction algorithm.

5.2 CRT Multicast Quorum-Based FH Rendezvous

Our second algorithm (CMQFH) uses the CRT quorum system, which also exhibits the rotation k-closure property. As will be explained in the next section, CMQFH is more resilient to insider attacks than AMQFH, however AMQFH is faster than CMQFH. The CRT is formally stated in [30]. Using CRT, one can construct quorum systems that satisfy the rotation k-closure property, as in Theorem 1 [18].[6]

Theorem 1. *Let $p_1, \ldots, p_k$ be k positive integers that are pairwise relatively prime, and let $y = \prod_{l=1}^{k} p_l$. The CRT quorum system $Q = \{G_1, \ldots, G_k\}$, where $G_i = \{p_i c_i \ , c_i = 0, 1, \ldots, \ y/p_i \ -1\}$, $i = 1, \ \ldots, k$, satisfies the rotation k-closure property.*

As an example, to construct a CRT quorum system of three quorums, we select $p_1 = 2$, $p_2 = 3$, and $p_3 = 5$. From Theorem 1, $Q = \{G_1, G_2, G_3\}$, where $G_1 = \{0, 2, 4, \ldots, 28\}$, $G_2 = \{0, 3, 6, \ldots, 27\}$, and $G_3 = \{0, 5, 10, \ldots, 25\}$ is a CRT quorum system.

The basic CMQFH algorithm for generating k FH sequences is similar to the AMQFH algorithm, with two main differences:

- The frame length, denoted by y, is given by $y = \prod_{i=1}^{k} p_i$.
- CMQFH uses the CRT quorum system instead of the uniform $(k-1)$-arbiter quorum system.

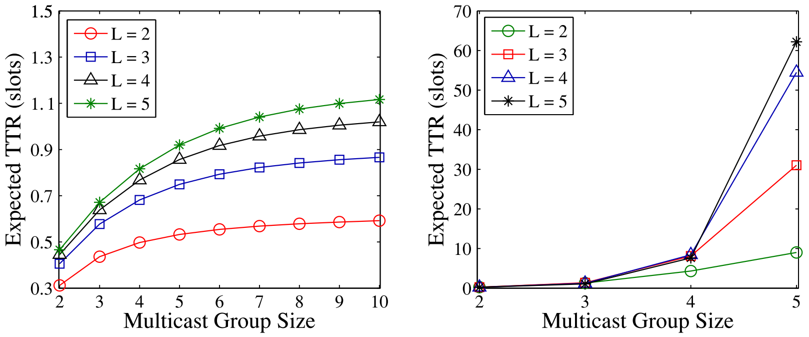
To minimize the frame length in CMQFH, the CRT quorum system is constructed using the smallest pairwise relatively prime numbers, starting with $p_1 = 2$.

5.3 AMQFH versus CMQFH (Speed versus Security)

In this section, we compare between the AMQFH and CMQFH algorithms. Both algorithms are implemented in a distributed way. Knowing the size of the multicast group, each node in the multicast group constructs its own FH sequence. The size of the multicast group is conveyed during the establishment of the multicast session. It is to be emphasized that the rendezvous schemes are intended for establishing new communication links as well as recovering

[5] Using an efficient double-threshold sensing mechanism, a sensing time in the order of a few microseconds is enough to achieve miss-detection and false-alarm probabilities as low as 10^{-3} [5].

[6] It is to be mentioned that the authors in [18] used the uniform k-arbiter and CRT quorum systems for enabling multicast communications in a power-saving ad hoc network operating on a *single licensed channel*.



(a) AMQFH (b) CMQFH

Fig. 4. Expected TTR versus the multicast group size.

disrupted communications due to the sudden appearance of a PU or a jammer. Therefore, rendezvous is not a one-time process and it might be needed any time during the network operation. Given that, the initialization phase where the group size is conveyed does not constitute a significant overhead since it is done only once, after which the network can perform multicast rendezvous at any time in a completely distributed way.

5.3.1 Expected TTR

Theorem 2. *Let $k+1$ be the size of the multicast group, n be the frame length, and L be the number of channels. The expected TTR of the AMQFH algorithm, denoted by $\mathbb{E}\left[T_{\mathrm{AMQFH}}\right]$, is given by:*

$$\mathbb{E}\left[T_{\mathrm{AMQFH}}\right] = \sum_{i=1}^{n-1}\left(\underbrace{i\,\Delta(\delta_{i+1})\prod_{j=1}^{i}(1-\Delta(\delta_j))}_{\Pr[\mathrm{T_{AMQFH}=i}]}\right), \tag{3}$$

where $\Delta(\delta_i)$ represents the probability that slot i is a rendezvous slot and δ_i represents the probability that slot i is a quorum slot (i.e., assigned a rendezvous frequency). $\Delta(\delta_i)$ and $\delta_i, i = 1, \ldots, n-1$, are given by:

$$\Delta(\delta_i) = \sum_{j=0}^{k}\left(\binom{k+1}{j}\delta_i^{k+1-j}\left(\frac{1-\delta_i}{L}\right)^j\right) + \left(\frac{1}{L}\right)^k(1-\delta_i)^{k+1}$$

$$\delta_i = \frac{\left\lfloor\frac{kn}{k+1}\right\rfloor - i + 2}{n} + \frac{i-1}{n} \times \frac{\left\lfloor\frac{kn}{k+1}\right\rfloor - i + 3}{n-i+1}.$$

Proof. See Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2014.2356475. □

Theorem 3. *Let k be the size of the multicast group. The expected TTR of the CMQFH algorithm, denoted by $\mathbb{E}[T_{\mathrm{CMQFH}}]$, is given by: where Θ is given by:*

$$\mathbb{E}[T_{\mathrm{CMQFH}}] = \sum_{i=1}^{n-1}\left(i\,\underbrace{\Theta(1-\Theta)^i}_{\Pr[\mathrm{T_{CMQFH}=i}]}\right), \tag{4}$$

where Θ is given by:

$$\Theta = \sum_{j=0}^{k-1}\left(\left(\frac{1}{L}\right)^j \sum_{\substack{\forall\{e_1,\ldots,e_{k-j}\}\\ \subseteq\{p_1,\ldots,p_k\}}} \frac{\prod_{l=k-j+1}^{k}(e_l-1)/e_l}{e_1\times\cdots\times e_{k-j}}\right) + \left(\frac{1}{L}\right)^{k-1}\prod_{m=0}^{k-1}\frac{e_m-1}{e_m}.$$

Proof. See Appendix B, available in the online supplementary material. □

$\mathbb{E}\left[T_{\mathrm{AMQFH}}\right]$ and $\mathbb{E}\left[T_{\mathrm{CMQFH}}\right]$ are plotted in Fig. 4 versus the multicast group size for different values of L. The expected TTR of CMQFH is much higher than that of AMQFH because it involves more randomly assigned slots. In both algorithms, a larger multicast group requires higher TTR. Moreover, including more channels (by increasing L) increases the average TTR due to the increased randomness in the randomly assigned slots.

5.3.2 Expected HD

In AMQFH, the expected HD is the same for all pairs of FH sequences, whereas in CMQFH it is different for different pairs. Therefore, in CMQFH, the expected value is computed over all pairs of FH sequences.

Theorem 4. *Let $\phi \overset{\mathrm{def}}{=} n - \left\{\left\lfloor\frac{kn}{k+1}\right\rfloor + 1\right\}$, and let φ be defined as in (2). The expected HD of AMQFH, denoted by $\mathbb{E}[H_{\mathrm{AMQFH}}]$, and its upper bound value, denoted by H^*_{AMQFH}, are given by:*

$$\mathbb{E}[H_{\mathrm{AMQFH}}] = \frac{L-1}{nL}\left\{\frac{(\varphi-1)(\phi+1)}{\varphi} + \frac{\phi}{\varphi}\right\}, \tag{5}$$

$$H^*_{\mathrm{AMQFH}} = \frac{\phi+1}{n}, \tag{6}$$

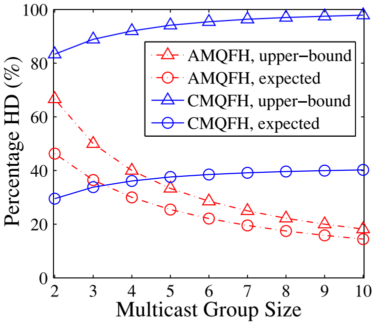


Fig. 5. Percentage HD versus the multicast group size ($L = 6$).

where H^*_{AMQFH} corresponds to the case where different SUs pick different quorums, and their randomly assigned parts are non-overlapping.

Proof. See Appendix C, available in the online supplementary material. □

Theorem 5. *The expected HD of CMQFH, denoted by $\mathbb{E}[H_{\mathrm{CMQFH}}]$, and its upper bound value, denoted by H^*_{CMQFH}, are given by:*

$$\mathbb{E}[H_{\mathrm{CMQFH}}] = \frac{L-1}{2Lk^2}\sum_{i=1}^{k}\sum_{j=1}^{k}\left(1-\frac{1}{p_ip_j}\right), \tag{7}$$

$$H^*_{\mathrm{CMQFH}} = \frac{1}{2\binom{k}{2}}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k}\left(1-\frac{1}{p_ip_j}\right), \tag{8}$$

where H^*_{CMQFH} is defined similar to H^*_{AMQFH}.

Proof. See Appendix D, available in the online supplementary material. □

$\mathbb{E}[H_{\mathrm{AMQFH}}]$, H^*_{AMQFH}, $\mathbb{E}[H_{\mathrm{CMQFH}}]$, and H^*_{CMQFH} are shown in Fig. 5. As the multicast group size increases, the HD of CMQFH increases but the HD of AMQFH decreases, and hence the gap in HD between AMQFH and CMQFH increases.

5.3.3 Nested-CMQFH

To provide a tradeoff between speed of rendezvous and robustness against node compromise, we propose a modified version of CMQFH that employs a nesting design, whereby several rendezvous channels are used within several nested quorums in each frame of the FH sequence. We call this modified CMQFH algorithm the *nested-CMQFH*.

In nested-CMQFH, each frame of the FH sequence contains a number of quorums, called the *nesting degree*. The nesting degree of an FH sequence depends on the prime number used in constructing this sequence. It provides a tradeoff between the TTR and HD. A large nesting degree results in a small TTR, but also a small HD, and vice versa. In our simulations in Section 8, we set the nesting degree of the FH sequence that uses prime number p_i to $p_i - 1$. If the prime number is selected appropriately, the nesting design can improve the speed of CMQFH significantly without drastically reducing its HD. The selection criterion of the prime number for nested-CMQFH will be explained in Section 7.

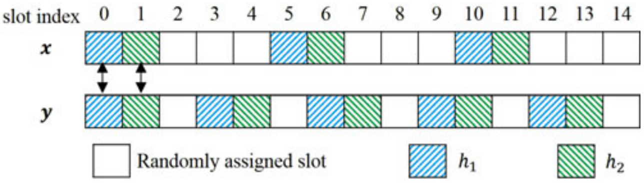


Fig. 6. Nested-CMQFH FH construction algorithm.

Fig. 6 illustrates the basic idea behind nested-CMQFH for a multicast group of two nodes. The prime numbers used in constructing FH sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ are 5 and 3, respectively. Accordingly, the frame length is $5 \times 3 = 15$ slots. In this example, we assume a common nesting degree of two for both FH sequences. $\boldsymbol{x}$ will have two nested quorums, $\{0, 5, 10\}$ and $\{1, 6, 11\}$, that are assigned channels h_1 and h_2, respectively. Similarly, $\boldsymbol{y}$ will have two nested quorums, $\{0, 3, 6, 9, 12\}$ and $\{1, 4, 7, 10, 13\}$, that are assigned channels h_1 and h_2, respectively. We refer to quorums $\{0, 5, 10\}$ and $\{0, 3, 6, 9, 12\}$ as the *outer-most* quorums of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Quorums $\{1, 6, 11\}$ and $\{1, 4, 7, 10, 13\}$ are referred to as the *next outer-most* quorums of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. In this example, quorums $\{1, 6, 11\}$ and $\{1, 4, 7, 10, 13\}$ are also called the *inner-most* quorums. h_1 and h_2 in Fig. 6 are called the *outer-most* and the *next outer-most (inner-most)* channels, respectively.

The basic AMQFH, CMQFH, and nested-CMQFH algorithms described above provide deterministic multicast rendezvous guarantees in time-synchronous and spectrum-homogeneous (i.e., symmetric) DSA networks. Next, we discuss how to use these basic rendezvous algorithms in asynchronous and heterogeneous (i.e., asymmetric) environments.

5.4 Asynchronous and Heterogeneous Rendezvous

5.4.1 Asynchronous Rendezvous

FH sequences constructed according to AMQFH, CMQFH, and nested-CMQFH support asynchronous rendezvous if each FH sequence continues to use the same (outer-most) quorum and the same (outer-most) channel in all of its frames. This result is a direct consequence of the intersection and rotation closure properties of the uniform k-arbiter and CRT quorum systems. If the (outer-most) quorum and/or the (outer-most) channel vary/varies from one frame to the next, then rendezvous is not guaranteed under any arbitrary time misalignment. However, if the misalignment can be upper-bounded by, for example, $c\,T_{\mathrm{frame}}$, where c is a positive integer and T_{frame} is the frame duration, then the rendezvous is guaranteed if the (outer-most) quorum and the (outer-most) channel remain the same for $c+1$ successive frames. After $c+1$ frames, the (outer-most) quorum and channel can change, however they need to stay on the new quorum and channel for $c+1$ frames, and so on. Although it is difficult to upper bound the misalignment between SUs before their first rendezvous time, it is easy to do so after they rendezvous. Recall that rendezvous is not a one-time process. For nested-CMQFH, the same conditions above are needed for the inner quorums and channels (i.e., quorums other than the outer-most quorum and channels other than the outer-most channel) if asynchronous rendezvous is to be ensured on the inner channels as well.

In addition to the above requirements on the selection of the (outer-most) quorum and channel, in CMQFH and nested-CMQFH nodes are required to pick different prime numbers to ensure asynchronous rendezvous. When the multicast initiator conveys the size of the multicast group through a series of unicast rendezvous operations it also assigns unique prime numbers to the multicast group members. In Section 8, we simulate nested-CMQFH assuming that different nodes may select the same prime number. In this case, nodes may still rendezvous during the randomly assigned slots.

5.4.2 Heterogeneous Rendezvous

In Figs. 3 and 6, FH sequences are constructed using the same rendezvous channels, but with different quorums. To allow nodes to construct their FH sequences in a distributed way, depending on their own views of spectrum opportunities, we consider a variant of these algorithms whereby each node assigns channels to quorum slots mainly based on the expected availability of these channels. Note that even in a heterogeneous spectrum environment, where the neighboring nodes do not necessarily share the same list of idle channels, there is still a good level of overlap in nodes' views of idle channels.

In Section 8, we evaluate AMQFH and nested-CMQFH in a heterogeneous environment with different heterogeneity levels. We define the heterogeneity level κ for a multicast group as the fraction of channels whose parameters differ between any two nodes in the multicast group. The randomly assigned slots in AMQFH and nested-CMQFH are assigned from the list of best $L\kappa_{\max}+1$ channels, where $\kappa_{\max}$ is the estimated maximum heterogeneity level that the network can have. This way, we avoid increasing the TTR by reducing the size of the set of channels that can be assigned to non-quorum (i.e., random) slots, while ensuring a non-empty intersection between every two such sets at two different nodes.

6 Optimal Channel Ordering

In Section 5, we presented the basic multicast rendezvous algorithms without explaining how the rendezvous channels are selected in each frame. We select the outer-most channel to be the "best" available channel in $\mathcal{L}$, the next outer-most channel as the next best available channel, and so on. In here, the best available channel is selected according to several factors, as will be explained in this section. Channel sorting is also exploited during assigning channels to the randomly assigned slots. Therefore, each node is required to *independently* sort the available channels (no message exchange is assumed between the nodes).

Furthermore, in Section 5, we did not specify the quorum selection procedure. One naïve approach to jointly address the channel sorting and quorum selection problems is to exhaustively examine all possible channel-quorum assignments and select the one that maximizes the number of available slots (i.e., slots during which the assigned channels are expected to be available). However, the time complexity of this exhaustive search is given by:

$$\mathcal{O}\left(\binom{n}{\left\lfloor\frac{kn}{k+1}\right\rfloor+1}\left(\left\lfloor\frac{kn}{k+1}\right\rfloor+1\right)\overline{L}\right), \quad \text{AMQFH}$$

$$\mathcal{O}\left(\sum_{i=1}^{k}\binom{\overline{L}}{\left\lceil\frac{p_i}{2}\right\rceil}\frac{p_i!\,y\left\lceil\frac{p_i}{2}\right\rceil}{\left(p_i-\left\lceil\frac{p_i}{2}\right\rceil\right)!\,p_i}\right), \quad \text{nested-CMQFH}$$

where $\overline{L}$ is the number of available channels, n is the frame length of AMQFH sequences, k is the size of the multicast group minus one for AMQFH and the size of the multicast group for nested-CMQFH, p_i is the prime number used in constructing the ith nested-CMQFH sequence, and $y=\prod_{i=1}^{k}p_i$ is the frame length for nested-CMQFH, which represents the kth primorial (given by $e^{(1+o(1))k\log k}$). This expensive exhaustive search needs to be performed by each node in each frame.

To avoid performing an expensive exhaustive search for each frame, we address the problems of quorum selection and channel assignment separately. We propose a one-time sorting algorithm that prioritizes channels, and a quorum selection mechanism that uses the order obtained by the sorting algorithm to perform the channel-quorum assignment. In this section, we present our ordering mechanism, and in Section 7 we address the quorum selection problem.

In our approach, channels are sorted primarily based on their average availability time, while providing certain probabilistic guarantees on protecting the transmissions of PUs and other SUs that do not belong to the rendezvous group. This way, less available channels are filtered out. To perform this sorting, we introduce a weight $q_m (0 \leq q_m \leq 1)$ for each channel $f_m \in \mathcal{L}$, and maximize a weighted sum of the channels average availability times with respect to these weights, while keeping the probabilities of collisions with PUs and 'external' SUs below certain thresholds. The weights will be used for two purposes. First, for the quorum slots, the weights will be used to sort channels such that the channel with the largest weight will be considered as the best channel. Second, for the randomly assigned (non-quorum) slots, these weights will be interpreted as probabilities, such that f_m will be assigned to non-quorum slots with probability q_m.

For $i \in \{1,2,3\}$ and $m \in \{1,\ldots,L\}$, let $T_i^{(m)}$ and $R_i^{(m)}$ be the sojourn time for channel f_m in state i and the first time that channel f_m returns to state i after leaving it, respectively. Let $\mathcal{T}_i^{(m)} \stackrel{\text{def}}{=} \mathbb{E}[T_i^{(m)}]$ and $\mathcal{R}_i^{(m)} \stackrel{\text{def}}{=} \mathbb{E}[R_i^{(m)}]$. Following standard Markov analysis, the fraction of time that channel f_m spends in state i (i.e., $\mathcal{T}_i^{(m)}/(\mathcal{T}_i^{(m)}+\mathcal{R}_i^{(m)})$) is $\pi_i^{(m)}$, which was given in Section 4.2. $\mathcal{T}_i^{(m)}, i \in \{1,2,3\}$ can be expressed as: $\mathcal{T}_1^{(m)} = \frac{1}{\lambda_p^{(m)}+\lambda_s^{(m)}}$, $\mathcal{T}_2^{(m)} = \frac{1}{\mu_p^{(m)}}$, and $\mathcal{T}_3^{(m)} = \frac{1}{\lambda_p^{(m)}+\mu_s^{(m)}}$.

To sort channels based on the above criteria, we propose the following optimization problem for AMQFH. This ordering mechanism starts over when the estimate of at least one of the channel parameters changes. There are different ways for monitoring and updating the channel parameters. For example, they can be updated based on observed PUs/SUs activities (by detecting collision events, successful communications, etc.). A node can also update its channel parameters by performing parallel out-of-band sensing over various channels while executing the rendezvous process.

Problem 1.

$$\underset{\mathbf{q}=(q_1,q_2,\ldots,q_L)}{\text{maximize}}\left\{\mathcal{U}(\boldsymbol{q}) \overset{\text{def}}{=} \sum_{m=1}^{L} \pi_1^{(m)} q_m\right\}.$$

Subject to.

$$\left[1-\prod_{\substack{u=0\\u\neq i}}^{n-1}\left(1-p^{(m)}_{(n+u)T}(1,s)\right)\right]q_m < \lambda^{(m)}_{s,Col}(n),$$
$$\forall s\in\{2,3\},\forall m\in\{1,\ldots,L\},\forall i\in\{1,\ldots,\varphi\} \tag{9}$$

$$\sum_{m=1}^{L} q_m = 1, \tag{10}$$

where $\lambda^{(m)}_{2,Col}(n)$ and $\lambda^{(m)}_{3,Col}(n)$ are prespecified thresholds on the probabilities of collisions with PUs and 'external' SUs, respectively. The objective function in Problem 1 represents a convex combination of the average channel availabilities. Constraint (9) restricts the collision probabilities with PUs and 'external' SUs, while considering the specific structure of the uniform k-arbiter quorum system. Each selection of i in (9) corresponds to one quorum. For each quorum, the term in the square brackets represents the probability that at least one quorum slot is in collision. Note that the collision thresholds depend on the frame length n and the channel.

A similar formulation to Problem 1 can be used for sorting in nested-CMQFH, after replacing (9) by (11).

$$\frac{1}{\psi_i}\sum_{v=1}^{\psi_i}\left[1-\prod_{u=\frac{(v-1)y}{\rho}}^{v\frac{y}{\rho}-1}\left(1-p^{(m)}_{(y+up_i)T}(1,s)\right)\right]q_m$$
$$< \lambda^{(m)}_{s,Col}(y), \tag{11}$$
$$\forall s\in\{2,3\},\forall m\in\{1,\ldots,L\},\forall i\in\{1,\ldots,k\},$$

where φ is as in (2), $\rho \overset{\text{def}}{=} \max_{1\le i\le k} p_i$, and $\psi_i \overset{\text{def}}{=} \left\lceil\frac{\rho}{p_i}\right\rceil$, $i=1,\ldots,k$. In contrast to uniform k-arbiter, different quorums in a CRT quorum system have different sizes. Because of this, the collision probabilities in (11) are computed in a slightly different way than (9). Each nested-CMQFH frame is divided into sub-frames, each with length y/ρ, and the average collision probability over the sub-frames is considered.

In addition to collision avoidance, constraints (9) and (11) are used to restrict the fluctuation level of the selected channels, such that highly fluctuating channels are excluded from the ordered list. Highly fluctuating channels will result in high collision probabilities. Because the weight of a channel is upper-bounded by $\lambda^{(m)}_{s,Col}(n)$ divided by the collision probability over this channel, highly fluctuating channels will receive very small weights. The fluctuation level of a channel affects its estimation accuracy. Less fluctuating channels can be estimated more accurately, and consequently result in a smaller TTR, as will be shown in Section 8.

In contrast to CRT, uniform k-arbiter quorum system has the unique feature that each quorum consists of several consecutive elements, therefore a big portion of the quorum slots in an AMQFH frame are consecutive. More specifically, at least $\frac{n-\phi}{\phi+1}\times 100\%$ of quorum slots are consecutive in AMQFH where $\phi \overset{\text{def}}{=} n-\left\{\left\lceil\frac{kn}{k+1}\right\rceil+1\right\}$ (note that $(n-\phi)\gg\phi$). This feature needs to be considered in the sorting mechanism of AMQFH. As explained before, the main goal of the channel-quorum assignment is to maximize the number of available quorum slots. Therefore, channels with larger mean sojourn time of state 1 (i.e., the idle state) are more preferable; because they will result in more available quorum slots, provided that all channels have similar average availability time.

For AMQFH, we account for the channel mean sojourn time of state 1 by adding a second optimization stage. The goal of this stage is to differentiate between channels with comparable average availability time based on their mean sojourn time of state 1, such that the channel with larger mean sojourn time is more preferable. Hence, the multi-objective channel sorting problem for AMQFH is formulated as a two-stage sequential optimization problem. Problem 1 above is the first stage and Problem 2 is the second stage. Let q_I^* be an optimal solution to Problem 1, and let $\mathcal{U}_I^* = \mathcal{U}(q_I^*)$. Then, Problem 2 is given by:

Problem 2.

$$\underset{\boldsymbol{q}=(q_1,q_2,\ldots,q_L)}{\text{maximize}}\left\{\mathcal{F}(\boldsymbol{q}) = \sum_{m=1}^{L}\mathcal{F}_m q_m \overset{\text{def}}{=} \sum_{m=1}^{L}\mathcal{T}_1^{(m)} q_m\right\}$$

$$\text{Subject to. } \mathcal{U}_{\text{I}}^*(1-\epsilon) < \mathcal{U}(q). \tag{12}$$

Problem 2 aims at maximizing a convex combination of the average sojourn times of state 1 subject to constraints (9)$-$(10), in addition to the new constraint in (12). Channels with larger values of $\mathcal{F}_m$ are less fluctuating between the idle and non-idle states, but $\mathcal{F}_m$ does not capture the further fluctuations between the non-idle states 2 and 3, which are captured by the constraints. Let $\mathcal{F}^*$ be the optimal value of $\mathcal{F}(\boldsymbol{q})$ in Problem 2, and let $\mathcal{U}^*$ be the corresponding value of $U(\boldsymbol{q})$. In (12), $\epsilon\in[0,1]$ restricts the reduction in the first objective function optimal value (i.e., $\mathcal{U}_I^*-\mathcal{U}^*$). Increasing ϵ increases the effect of the second objective function on channel ordering.

In heterogeneous environments, different nodes may order channels differently; because they may have different parameters for the same channel. This results in increasing the TTR. Nested-CMQFH is more robust to heterogeneity than AMQFH because of its inherent nesting design, where the rendezvous does not depend only on a single channel, but on several channels. Moreover, the nesting design of nested-CMQFH improves its robustness against an intelligent jammer, who may order the channels in a similar way and continuously jam the best channel.

Readers may refer to [3] and [1] for numerical examples on channel ordering. The ordering mechanisms are simulated in Section 8.

7 Adaptive Quorum Selection

Our quorum selection procedure relies on estimating the states of various channels in the next frame, driven by a proactive out-of-band sensing of their states in the current frame.

7.1 AMQFH

Consider the jth frame of an FH sequence. The node that follows this sequence starts sensing the channels according to the order obtained in Section 6 (some of the randomly assigned slots are used for spectrum sensing). Quorum G_j is selected from all quorums in the quorum system Q so as to maximize the number of quorum slots for which h_j is idle with probability greater than a threshold γ. Recall that h_j is the channel to be used in frame j according to the channel ordering criterion in Section 6. If more than one quorum result in the same maximum number of slots, we break the tie based on the average idle probability of h_j, averaged over all slots that belong to G_j. Let $h_j = f_{j'}, j' \in \mathcal{L}$, then the problem of selecting quorum G_j is formally described as follows:

$$\text{maximize}\left\{\mathcal{A}(z,n)=\sum_{i=0}^{n-1}1_{\{p^{(j')}_{(n-z+i)T}(1,1)\geq\gamma\}} + \frac{1}{\left\lfloor\frac{kn}{k+1}\right\rfloor+1}\sum_{i=0}^{n-1}p^{(j')}_{(n-z+i)T}(1,1)\right\}, \tag{13}$$

where $\mathbf{1}_{\{\cdot\}}$ is the indicator function, z is the index of the randomly assigned slot used for sensing channel h_j, and $k+1$ is the multicast group size. $p^{(j')}_{(n-z+i)T}(1,1)$ is the probability that h_j remains available in the ith slot of the next frame, given that it is currently available. When evaluating $\mathcal{A}(z,n)$ at a specific quorum, say $H \in Q$, $p^{(j')}_{(n-z+i)T}(1,1) = 0, \forall i \notin H$. The computation of $p^{(m)}_t(i,j)$ is explained in Section 4.2. The second term in (13) is less than one, and hence it is dominated by the first term.

7.2 Nested-CMQFH

Consider an FH sequence that uses prime number p_i (hence, the nesting degree is $p_i - 1$). After sorting the channels and obtaining the best $p_i - 1$ channels, channel-quorum assignment is performed in nested-CMQFH jointly for all nested quorums of a frame. Formally, the problem of selecting the nested quorums in the jth frame of an FH sequence with prime number p_i is described as follows:

$$\text{maximize}\left\{\mathcal{B}(p_i,y)=\sum_{j=1}^{p_i-1}\sum_{l=0}^{y-1}\mathbf{1}_{\{p^{(j')}_{(y-z_j+l)T}(1,1)\geq\gamma\}} + \frac{1}{(p_i-1)\left(\frac{y}{p_i}\right)}\sum_{j=1}^{p_i-1}\sum_{l=0}^{y-1}p^{(j')}_{(y-z_j+l)T}(1,1)\right\}, \tag{14}$$

where y is the frame length. $f_{1'}$ is the outer-most channel, $f_{2'}$ is the next outer-most channel, and so on. z_j is the index of the randomly assigned slot used for sensing channel $f_{j'}$. Recall that the nesting degree of the FH sequence that uses prime number p_i is $p_i - 1$, and each quorum of such FH sequence consists of y/p_i quorum slots.

The above maximization problem is solved by considering all combinations of $p_i - 1$ channels and p_i quorums and selecting the channels-quorums assignment that results in the maximum number of available slots. Among all prime numbers, we select the one that results in the maximum *absolute* (not fractional as in [1]) number of available slots. By considering the absolute number of available slots, we give a higher priority to large prime numbers, which have a larger fraction of quorum slots. The fraction of quorum slots in an FH sequence with prime number p_i is $\frac{p_i-1}{p_i}$. In general, for two prime numbers p_j and p_k, if $p_j > p_k$ then $\frac{p_j-1}{p_j} > \frac{p_k-1}{p_k}$. By giving a higher priority to large prime numbers, we reduce the number of randomly assigned slots, which might be assigned low quality channels that are different at different nodes. Note that a large number of quorum slots does not necessarily result in a large number of available slots. It depends on the quality of the channels used in the quorum slots.

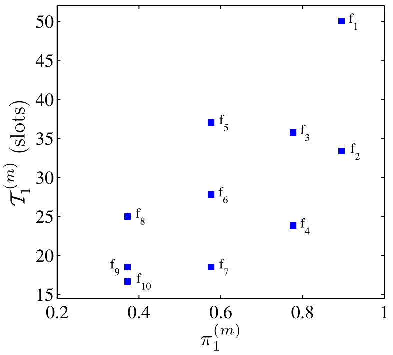


Fig. 7. Characteristics of the channels used in simulations.

8 PERFORMANCE EVALUATION

This section evaluates the performance of our multicast rendezvous algorithms.[7,8] As mentioned earlier, we consider one single-hop multicast group. Channel dynamics follow the three-state Markov model described in Section 4.2. AMQFH and nested-CMQFH are studied under different values of γ in (13) and (14), and multicast group sizes. Both multicast algorithms are studied under different heterogeneity levels κ. In [1], AMQFH and nested-CMQFH are simulated assuming that different nodes in the multicast group have the same channel parameters, but the instantaneous states of the channels are perceived differently by different nodes in the group. In this section, we simulate AMQFH and nested-CMQFH in a more realistic setup, where, for a subset of channels (κL channels), the parameters of a given channel are different at different nodes. We evaluate the multicast algorithms based on the TTR, the estimation accuracy, and the average percentage HD. The estimation accuracy is indicated by the collision rates with PUs/'external' SUs and by missed opportunities (i.e., number of actually available slots that were considered unavailable). Our algorithms are simulated under a realistic setting where nodes start rendezvous at different points in time, and in the

[7] Detailed evaluation of existing unicast rendezvous schemes under fast PU dynamics can be found in our technical report [3]. In [3], a new unicast rendezvous scheme is also proposed to account for fast PU dynamics.

[8] To the best of our knowledge, there is no other non-sequential multicast rendezvous algorithms in the literature. Hence, we only study and compare our proposed algorithms.

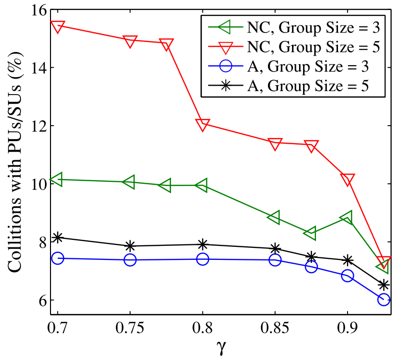


Fig. 8. Collision rate of AMQFH (labeled by 'A') and nested-CMQFH (labeled by 'NC').

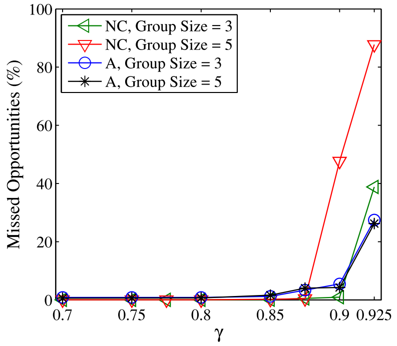


Fig. 9. Missed opportunity rate of AMQFH (A) and nested-CMQFH (NC).

absence of node synchronization. Specifically, the misalignment between the FH sequences is randomly selected in each experiment. The 95 percent confidence intervals are indicated unless they are very tight.

In our simulations, we consider ten licensed channels with various levels of availability and fluctuation. The set of channels include low, medium, and high fluctuating channels, as well as channels with low, medium, and high average availability times. The exact characteristics of these channels are shown in Fig. 7 in terms of $\mathcal{T}_1^{(m)}$ and $\pi_1^{(m)}$, which are derived from $\lambda_p^{(m)}$, $\lambda_s^{(m)}$, $\mu_p^{(m)}$, and $\mu_s^{(m)}$. These channel characteristics are inline with the "relative" scales of the average availability times and idle period durations in [14] and [24]. To avoid having the same order of channels for different runs, we slightly perturb the nominal values for the above four channel parameters within small ranges, so that the efficiency of our channel sorting and quorum selection mechanisms can be examined as well.

8.1 Collision/Missed Opportunity Rates

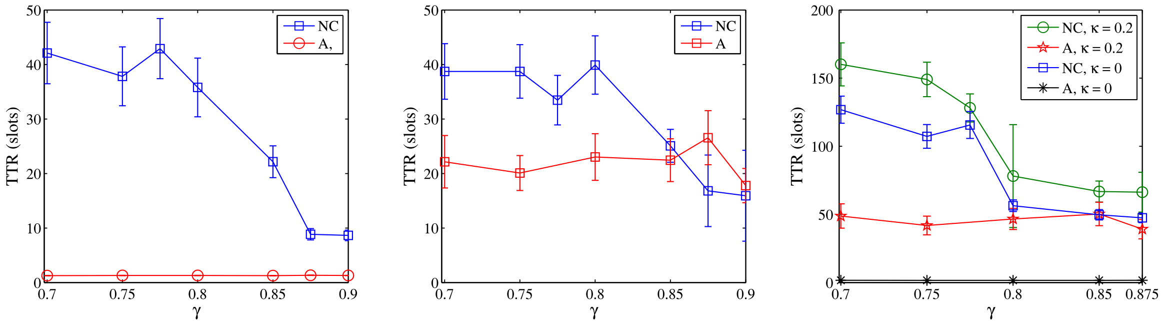
Figs. 8 and 9 depict the collision and missed opportunity rates versus γ for group sizes 3 and 5. As expected, a conservative estimation (by selecting a large value of γ) incurs low collision rate but high missed opportunity rate, and the opposite for small values of γ. AMQFH has a better estimation accuracy than nested-CMQFH because for the same group size, AMQFH has a shorter frame than nested-CMQFH. This results in higher utilization time for AMQFH compared to nested-CMQFH. Both collision and missed opportunity rates increase with the group size. When the number of SUs in the multicast group increases, the probability that one of them collides with a PU increases, and the probability that one of them misses an available rendezvous opportunity increases as well.

8.2 TTR

Fig. 10 shows the effect of γ on TTR for different values of κ and group sizes. As shown in the figure, AMQFH is faster than nested-CMQFH. As mentioned earlier, the value of γ needs to be carefully selected to avoid having large TTR, especially for nested-CMQFH. Furthermore, the selection of γ depends on the multicast algorithm, and it often depends on κ.

To examine the capability of AMQFH and nested-CMQFH in achieving rendezvous within a reasonable time, we first show in Fig. 11 the percentage of runs where the TTR of AMQFH and nested-CMQFH exceeds 400 slots. Fig. 11 shows that this percentage increases with κ. Although counterintuitive, nested-CMQFH is more likely to achieve a TTR smaller than 400 slots than AMQFH in heterogeneous environments. In AMQFH, nodes either rendezvous quickly or they rendezvous after a very long time. If nodes cannot meet quickly in AMQFH, this means that they have significantly different sets of "best channels." In contrast to AMQFH, in nested-CMQFH nodes eventually manage to rendezvous in a reasonable time because of the nesting design used in the algorithm. The curves in Fig. 11 are also increasing with the group size. As κ increases, the advantage of the nesting design (of nested-CMQFH) becomes more significant.

However, the percentage of runs with TTR exceeds 400 slots does not completely characterize the performance of the multicast algorithms. Therefore, we show in Fig. 12 the



(a) Group Size = 3, $\kappa = 0$

(b) Group Size = 3, $\kappa = 0.2$

(c) Group Size = 4

Fig. 10. TTR versus γ for AMQFH (A) and nested-CMQFH (NC).

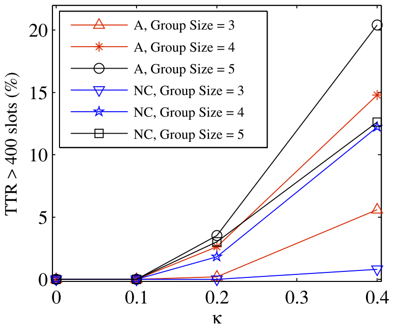


Fig. 11. TTR versus κ for AMQFH and nested-CMQFH.

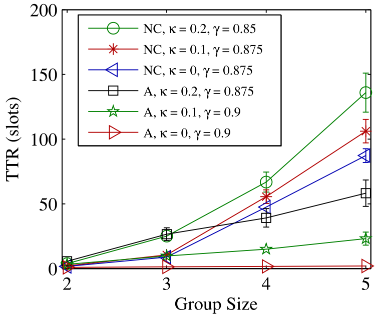


Fig. 12. TTR versus group size for AMQFH (A) and nested-CMQFH (NC).

average TTR (averaged over the runs with TTR ≤ 400) of both multicast algorithms. It can be observed that AMQFH is faster, provided that the rendezvous process does not take too long time. In general, AMQFH can accommodate large groups better than nested-CMQFH (Note that the best value of γ depends on the multicast algorithm as discussed earlier).

8.3 HD

The ability of the proposed algorithms to provide a high HD is considered in Fig. 13. Because nested-CMQFH uses several channels within a frame, and because of the sparsity of the CRT quorum systems used in nested-CMQFH, it exhibits a higher HD than AMQFH. Moreover, when the group size increases (and hence the frame length), the estimation mechanism recommends using the best channels more often (especially in AMQFH where each frame consists of a single quorum channel), which increases the similarity between the FH sequences and hence reducing the HD. For the same reason, the HD of both algorithms decreases with γ.

In Figs. 12 and 13, the values of γ for both algorithms are obtained from Fig. 10, to achieve the smallest TTR.

9 Future Research

9.1 Multicast for Multi-Hop Unicast Rendezvous

In this section, we provide directions for using our multicast rendezvous algorithms (AMQFH and nested-CMQFH) to achieve unicast rendezvous in multi-hop DSA networks.

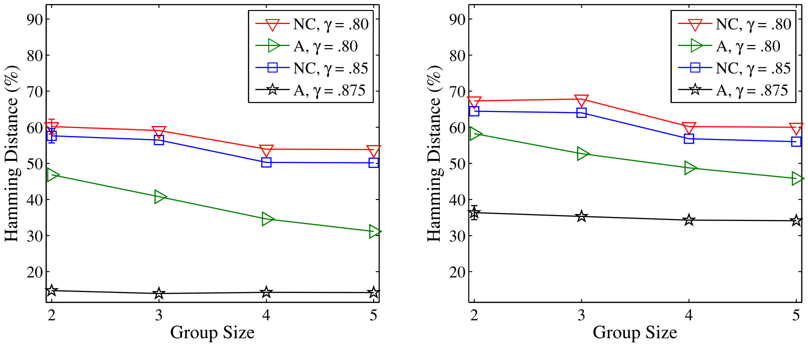
One way for a source node R to rendezvous with a sink node S that is multiple hops away is to repeatedly execute a unicast rendezvous algorithm (e.g., NGQFH in [3]) at each link in the route between R and S (see Fig. 14a). However, multiple routes can exist between R and S, as in Fig. 14b. In this case, the challenge is to find the 'best' route, which is in our case the one that results in the smallest end-to-end TTR. If the TTR of each link can be expressed as a function of the link parameters (such as the heterogeneity level). Then, a shortest path algorithm can be used to find the path with the smallest end-to-end TTR. However, it is not clear how to express the TTR as a function of the various link parameters.

Another approach to achieve unicast rendezvous in a multi-hop DSA network is to apply one of the multicast rendezvous algorithm (AMQFH or nested-CMQFH) repeatedly in each hop. This results in conveying the sender's message to all one-hop neighbors in each iteration, hence avoiding the need to find the 'optimal' next-hop neighbor. In this approach (which is similar to AODV routing), the number of one-hop neighbors needs to be known in each hop.

The first approach is superior over the second approach if the TTR of each link can be computed and the optimal route can be found; because performing unicast rendezvous is always faster than multicast rendezvous. However, as mentioned earlier, it is difficult to explicitly express the TTR of each link as a function of its parameters. Thorough investigation of this problem is left for future research.

9.2 Coexistence Rendezvous

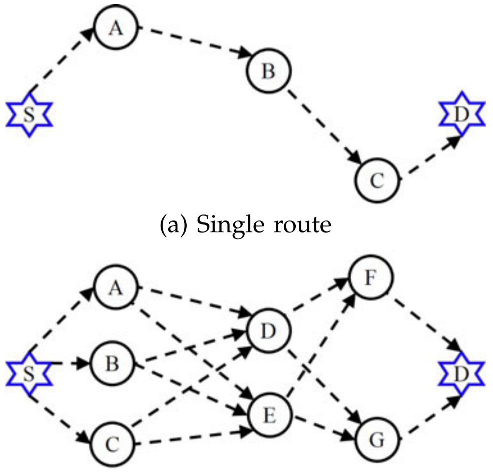
In this paper, we considered a single multicast rendezvous group. The problem of coexistence rendezvous, where multiple groups of SUs try to rendezvous simultaneously is more challenging. This is because the rendezvous process of one SU group cannot be separated from the rendezvous processes of the other coexisting groups. In particular, the



(a) $\kappa = 0$

(b) $\kappa = 0.2$

Fig. 13. HD versus group size for AMQFH (A) and nested-CMQFH (NC).



(b) Multiple routes

Fig. 14. Rendezvous in multi-hop networks.

channel ordering at one group affects the channel access of that group, which in turn affects the channel ordering of the coexisting SUs.

Game theory is a suitable tool for capturing the interactions between the rendezvous processes of coexisting SU groups. Recently, a game-theoretic FH-based rendezvous scheme was proposed in [2] to achieve *unicast* rendezvous in the presence of a single jammer. In [2], only a single pair of SUs is considered. Furthermore, the jammer in [2] has an opposite objective to the legitimate SUs, which is disrupting the rendezvous process of the SUs. In coexistence rendezvous, all SU groups try to achieve the same objective, which is rendezvous within the smallest TTR. Thorough investigation of this problem is left for future research.

10 Conclusions

In this paper, we developed asynchronous FH algorithms for multicast rendezvous in heterogeneous DSA networks. To account for fast PU dynamics, we developed optimal channel ordering and quorum selection mechanisms for adapting the proposed FH designs on the fly. Simulation results were obtained under different settings. The main learned messages from our simulations are: (i) If γ is set to a large value, all proposed algorithms incur high missed opportunity rate. On the other hand, if γ is set to a small value, they incur a high collision rate. The best value of γ depends on κ. (ii) AMQFH has a better estimation accuracy than nested-CMQFH. (iii) AMQFH is faster than nested-CMQFH, provided that the rendezvous process does not take too long. (iv) Although counterintuitive, nested-CMQFH is more likely to achieve rendezvous within a pre-specified number of slots than AMQFH in heterogeneous environments. (v) In general, AMQFH can accommodate large groups better than nested-CMQFH. Finally, (vi) nested-CMQFH exhibits a higher HD than AMQFH.

Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) (Grants IIP-1265960 and IIP-1432880) and Raytheon. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the US National Science Foundation. Preliminary results in this paper were presented at the IEEE DySPAN Conference, Oct. 2012 [1].

References

[1] M. J. Abdel-Rahman, H. Rahbari, and M. Krunz, "Adaptive frequency hopping algorithms for multicast rendezvous in DSA networks," in *Proc. IEEE Int. Symp. Dyn. Spectrum Access Netw.*, Oct. 2012, pp. 494–505.

[2] M. J. Abdel-Rahman and M. Krunz, "Game-theoretic quorum-based frequency hopping for anti-jamming rendezvous in DSA networks," in *Proc. IEEE Int. Symp. Dyn. Spectrum Access Netw.*, Apr. 2014, pp. 248–258.

[3] M. J. Abdel-Rahman, H. Rahbari, and M. Krunz, "Rendezvous in dynamic spectrum wireless networks," Dept. Electrical Comput. Eng., Univ. Arizona, Tucson, AZ, USA, Tech. Rep. TR-UA-ECE-2013-2, May 2013.

[4] M. J. Abdel-Rahman, H. Rahbari, M. Krunz, and P. Nain, "Fast and secure rendezvous protocols for mitigating control channel DoS attacks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2013, pp. 370–374.

[5] M. J. Abdel-Rahman, H. K. Shankar, and M. Krunz, "Adaptive cross-layer protocol design for opportunistic WLANs over TVWS," in *Proc. IEEE Int. Symp. Dyn. Spectrum Access Netw.*, Apr. 2014, pp. 519–530.

[6] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 216–230.

[7] K. Bian and J.-M. Park, "Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1294–1307, Jul. 2013.

[8] K. Bian, J.-M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 25–36.

[9] T. Chen, H. Zhang, M. Katz, and Z. Zhou, "Swarm intelligence based dynamic control channel assignment in CogMesh," in *Proc. IEEE Int. Conf. Commun. Workshop*, May 2008, pp. 123–128.

[10] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, "CogMesh: A cluster-based cognitive radio network," in *Proc. IEEE 2nd Int. Symp. Dyn. Spectrum Access Netw.*, Apr. 2007, pp. 168–178.

[11] L. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proc. IEEE 3rd Int. Symp. Dyn. Spectrum Access Netw.*, 2008, pp. 1–7.

[12] FCC, "Second memorandum opinion and order in the matter of unlicensed operation in the TV broadcast bands (ET Docket No. 04-186)," *Additional Spectrum for Unlicensed Devices Below 900MHz and in 3GHz Band (EC Docket No 02-380)*, Sep. 23, 2010.

[13] H. Garcia-Molina and D. Barbara, "How to assign votes in a distributed system," *J. ACM*, vol. 32, pp. 841–860, 1985.

[14] T. Harrold, R. Cepeda, and M. Beach, "Long-term measurements of spectrum occupancy characteristics," in *Proc. IEEE Int. Symp. Dyn. Spectrum Access Netw.*, 2011, pp. 83–89.

[15] J. Jia, Q. Zhang, and X. Shen, "HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 106–117, Jan. 2008.

[16] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad-hoc networks," *Mobile Netw. Appl.*, vol. 10, pp. 169–181, Feb. 2005.

[17] K. Bian, J.-M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 689–703, Apr. 2011.

[18] Y.-C. Kuo, "Quorum-based power-saving multicast protocols in the asynchronous ad-hoc network," *Comput. Netw.*, vol. 54, pp. 1911–1922, 2010.

[19] L. Lazos, S. Liu, and M. Krunz, "Spectrum opportunity-based control channel assignment in cognitive radio networks," in *Proc. 6th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh Ad Hoc Commun. Netw.*, Jun. 2009, pp. 135–143.

[20] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2011, pp. 2444–2452.

[21] H. Liu, Z. Lin, X. Chu, and Y.-W. Leung, "Taxonomy and challenges of rendezvous algorithms in cognitive radio networks," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, Jan. 2012, pp. 645–649.

[22] S. Liu, L. Lazos, and M. Krunz, "Thwarting inside jamming attacks on wireless broadcast communications," in *Proc. 4th ACM Conf. Wireless Netw. Security*, 2011, pp. 29–40.

[23] B. Lo, "A survey of common control channel design in cognitive radio networks," *Phys. Commun.*, vol. 4, pp. 26–39, 2011.

[24] M. López-Benítez and F. Casadevall, "Empirical time-dimension model of spectrum use based on a discrete-time Markov chain with deterministic and stochastic duty cycle models," *IEEE Trans. Veh. Technol.*, vol. 60, no. 6, pp. 2519–2533, Jul. 2011.

[25] Y. Manabe, R. Baldoni, M. Raynal, and S. Aoyagi, "k-arbiter: A safe and general scheme for h-out of-k mutual exclusion," *Theoretical Comput. Sci.*, vol. 193, pp. 97–112, 1998.

[26] H. Nan, T.-I. Hyon, and S.-J. Yoo, "Distributed coordinated spectrum sharing MAC protocol for cognitive radio," in *Proc. IEEE 2nd Int. Symp. Dyn. Spectrum Access Netw.*, Apr. 2007, pp. 240–249.

[27] S. Romaszko and P. Mähönen, "Quorum-based channel allocation with asymmetric channel view in cognitive radio networks," in *Proc. 6th ACM Workshop Perform. Monitoring Meas. Heterogeneous Wireless Wired Netw.*, 2011, pp. 67–74.

[28] D. R. Stinson, *Cryptography: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2006.

[29] M. Vukolić, "The origin of quorum systems," *Bull. Eur. Assoc. Theoretical Comput. Sci.*, no. 101, pp. 125–147, Jun. 2010.

[30] C.-H. Wu, J.-H. Hong, and C.-W. Wu, "RSA cryptosystem design based on the Chinese remainder theorem," in *Proc. Asia South Pac. Des. Autom. Conf.*, 2001, pp. 391–395.

[31] Y. Zhang, Q. Li, G. Yu, and B. Wang, "ETCH: Efficient channel hopping for communication rendezvous in dynamic spectrum access networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2011, pp. 2471–2479.

[32] J. Zhao, H. Zheng, and G.-H. Yang, "Distributed coordination in dynamic spectrum allocation networks," in *Proc. IEEE 1st Int. Symp. Dyn. Spectrum Access Netw.*, Nov. 2005, pp. 259–268.

Mohammad J. Abdel-Rahman is currently working toward the PhD degree in the Electrical and Computer Engineering Department at the University of Arizona. He received the MSc degree in electrical engineering from Jordan University of Science and Technology, Jordan in May 2010 and the BSc degree in telecommunications engineering from Yarmouk University, Jordan in September 2008. His research interests include the areas of wireless communications and networking, with emphasis on dynamic spectrum access networks, wireless security, resource management, adaptive protocols, satellite communications, and wireless sensor networks. He has published 14 journal articles and peer-reviewed conference papers. He is a member of the IEEE.

Hanif Rahbari received the BSc degree in information technology engineering from the Sharif University of Technology and the MSc degree in computer networks from the Amirkabir University of Technology, Tehran, Iran. He is currently working toward the PhD degree in electrical and computer engineering at the University of Arizona. His research interests include wireless communications and networking, PHY-layer security, dynamic spectrum access networks, and multimedia networs. He is a member of the IEEE.

Marwan Krunz received the PhD degree in electrical engineering from Michigan State University in 1995. He is a professor of ECE and CS at the University of Arizona. He is the site codirector at the US National Science Foundation (NSF) Broadband Wireless Access and Applications Center. He joined the University of Arizona in January 1997, after a brief postdoctoral stint at the University of Maryland. In 2010, he was a visiting chair of excellence at the University of Carlos III de Madrid. He previously held other visiting research positions at INRIA-Sophia Antipolis, HP Labs, University of Paris VI, University of Paris V, and US West Advanced Technologies. His research interests include the areas of wireless communications and networking, with emphasis on resource management, adaptive protocols, and security issues. He has published more than 215 journal articles and peer-reviewed conference papers, and is a coinventor on five US patents. He received the 2012 IEEE TCCC Outstanding Service Award. He received the US NSF CAREER Award in 1998. He currently serves on the editorial board for the *IEEE Transactions on Network and Service Management*. Previously, he served on the editorial boards for the *IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, Computer Communications Journal, and the IEEE Communications Interactive Magazine*. He was the general cochair for WiSec'12, and served as a TPC chair for INFOCOM'04, SECON'05, and WoWMoM'06. He was the keynote speaker, an invited panelist, and a tutorial presenter at numerous international conferences. He is a fellow of the IEEE, an Arizona Engineering Faculty fellow (2011-2014), and a IEEE Communications Society Distinguished lecturer (2013 and 2014).