Rendezvous in Highly-dynamic CRNs: A Frequency Hopping Approach

Mohammad J. Abdel Rahman, Hanif Rahbari, and Marwan Krunz
Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721
{mjabdelrahman, rahbari, krunz}@email.arizona.edu
Technical Report
TR-UA-ECE-2012-3
Last update: November 18, 2012

Abstract
Establishing communications in a dynamic spectrum access (DSA) network requires communicating nodes to “rendezvous” before transmitting their data packets. Frequency hopping (FH) provides an effective method for rendezvousing without relying on a predetermined control channel. Current FH rendezvous designs have two main limitations. First, they do not account for fast primary user (PU) dynamics. Second, they mainly target pairwise rendezvous, and do not intrinsically support multicast rendezvous. In this paper, we first design a grid-quorum-based FH algorithm, denoted by NGQFH, for pairwise rendezvous. NGQFH can achieve efficient rendezvous under fast PU dynamics. It is also robust against node compromise. Using the uniform k-arbiter and Chinese Remainder Theorem quorum systems, we then propose three multicast rendezvous algorithms: AMQFH, CMQFH, and nested-CMQFH, which provide different tradeoffs between the time-to-rendezvous and robustness to node compromise. Our pairwise and multicast rendezvous algorithms are tailored for asynchronous and spectrum-heterogeneous DSA networks. To account for fast PU dynamics, we develop an algorithm for adapting the proposed FH designs on the fly. This adaptation is done through an optimal mechanism for channel sensing and assignment, and a quorum selection mechanism. Extensive simulations are used to evaluate our algorithms.

Index Terms
Control channel, dynamic frequency hopping, dynamic spectrum access, quorum systems.

I. INTRODUCTION
Motivated by the need for more efficient utilization of the licensed spectrum and facilitated by recent regulatory policies [7], significant research has been conducted towards developing cognitive radio (CR) technologies for dynamic spectrum access (DSA) networks. CR devices utilize the available spectrum in a dynamic and opportunistic fashion without interfering with co-located primary users (PUs). The communicating entities of an opportunistic CR network are called secondary users (SUs).

Establishing a communication link in a DSA network requires nodes to rendezvous. In the absence of centralized control, the rendezvous process needs to be carried out in a distributed manner. To address

Some of the results in this paper were presented at the IEEE DySPAN Conference, Oct. 2012 [1].
this problem, many existing MAC protocols for CR networks rely on a dedicated global or group control channel (e.g., [6]). Presuming a common control channel (CCC) surely simplifies the rendezvous process, but it comes with two main drawbacks. First, a CCC can become a network bottleneck, creating a single point of failure. Second, PU dynamics and spectrum heterogeneity make it very difficult to always maintain a single CCC [15].

Frequency hopping (FH) provides an alternative method for rendezvousing without relying on a prede-terminated CCC. Most existing work on FH designs (e.g., [2]) is based on ad hoc approaches that do not provide any performance guarantees. One way to construct FH sequences in a systematic manner is to use quorum systems [8]. Quorums have been widely used in distributed systems to solve the mutual exclusion problem, the agreement problem, and the replica control problem. Systematic quorum-based approaches for designing and analyzing FH protocols for control channel establishment have been proposed in [3], [4]. An advantage of quorum-based FH designs is their robustness to synchronization errors [9]. As explained later, some quorum systems, such as grid, uniform k-arbiter [10], and Chinese Remainder Theorem (CRT) [19] quorum systems, enjoy certain properties that allow them to be used for asynchronous communications. The approaches in [3], [4] do not intrinsically support multicast rendezvous, where all the nodes in a multicast group are required to rendezvous in the same time slot. Furthermore, these approaches do not account for fast channel variations, where channel availability can vary during the rendezvous process. Other systematic FH approaches were proposed in [13], [20]. Similar to [3], [4], these approaches do not account for fast PU dynamics and do not support multicast rendezvous. In [13], an algorithm for establishing multicast communications was proposed. Instead of designing different FH sequences that overlap at common slots, the multicast communication in [13] is established after a series of pairwise rendezvous operations. These operations result in all nodes in the multicast group being tuned to a common FH sequence. The effectiveness of this approach cannot be maintained under node compromise (if a node is compromised, then the FH sequences of all nodes are exposed). By requiring every pair of nodes to utilize all rendezvous channels for control, the FH designs in [20] improve the capacity of DSA networks
during the rendezvous phase.

Group-based schemes have been proposed in [15] to facilitate multicast rendezvous. These schemes can be divided into two categories: (i) neighbor coordination schemes (e.g., [5]), where neighboring nodes broadcast their channel parameters to make a group-wide decision, and (ii) cluster-based schemes (e.g., [12]), where nodes group around a cluster head according to their channel availabilities. One drawback of these schemes is the need for the initial step of neighbor discovery prior to establishing a CCC. Furthermore, these schemes incur considerable overhead for maintaining the group-based control channel. Even though these solutions establish a CCC for intra-group communications, the problem of inter-group communications is yet another challenge that remains to be addressed [15].

In [14], 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. This scheme does not account for PU dynamics during the rendezvous process. Moreover, implementing the multicast operation 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 encode/decode common secure communication messages. For security purposes, this key may have to be updated periodically [18]. 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 changes are conveyed using a series of unicast transmissions.

Previous efforts on FH-based rendezvous ignore channel variations (e.g., due to PU dynamics) during the rendezvous process. This can result in excessively long time-to-rendezvous (TTR). To account for such channel variations, the average channel availability time and its fluctuation level (i.e., rate of transitions between idle and busy states) need to be considered when constructing the FH sequences. Previous FH designs model channel availability as a binary variable, where ‘1’ means the channel is available and ‘0’ means it is not. This modeling approach does not differentiate between different levels of channel availability. By considering the average availability time, we model channel availability in this paper as a continuous variable, taking values in the interval \((0, 1]\). This modeling approach gives us an effective
tool in designing FH rendezvous protocols that are robust against fast PU dynamics, as explained later in the paper.

To illustrate the effect of channel dynamics on the TTR, we simulated three previously proposed FH algorithms, M-QCH [4], JS_SM [13], and SYNC-ETCH [20], under different average availability times and average duty cycles (DCs). The average DC is defined as the average length (in slots) of consecutive busy and idle periods. It reflects the fluctuation level of the channel (channels with higher DCs exhibit less fluctuations). The algorithms are simulated under a simplified setup, where nodes are synchronized, spectrum is homogeneous (i.e., SUs are assumed to perceive the same spectrum opportunities), and nodes are assumed to start the rendezvous process at the same time. Six channels are used in the experiment. M-QCH was proposed in [4] to minimize the maximum TTR in a synchronous environment. As shown in Figure 1, the TTR increases significantly with a smaller average availability time. It also increases with the decrease in the average DC. JS_SM is less affected by 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 high collisions, as shown in Figure 2. In a more realistic setting with asynchronous operation and heterogeneous-spectrum environment, the effect of PU dynamics on the TTR is even more severe, as will be shown in Section VII.

**Our Contributions**—The main contributions of this paper are summarized as follows:

- We design a grid quorum-based FH algorithm called NGQFH for asynchronous pairwise rendezvous in heterogeneous DSA networks. NGQFH employs a nested design, whereby the rendezvous is achieved using several nested channels instead of a single channel. When integrated with the optimal channel ordering and adaptive quorum selection, NGQFH operates efficiently in the presence of fast PU dynamics. In addition to the improved TTR, the nesting approach of NGQFH improves its robustness to node compromise.

- We propose three algorithms, namely AMQFH, CMQFH, and nested-CMQFH, that achieve multicast
Fig. 1: Fixed FH designs result in large TTR under fast PU dynamics. For JS_SM (M-QCH and SYNC-ETCH), sensing is performed on a per frame (slot) basis. The frame length of M-QCH is 3 and the period is $3 	imes 6 = 18$. The frame length for SYNC-ETCH is $2 	imes 6 + 1 = 13$ and its period is $6 	imes 13 = 78$. The frame length for JS_SM is $3 	imes 7 = 21$.

Fig. 2: Collision rate vs. average channel availability for JS_SM.

rendezvous while maintaining multicast consistency. These algorithms are tailored for asynchronous and heterogeneous DSA networks. They provide different tradeoffs between the TTR and robustness to node compromise.

- We develop an algorithm for adapting the proposed FH designs on the fly, depending on estimated PU dynamics. To achieve this adaptation, an optimal channel ordering mechanism for channel sensing and assignment, and a quorum selection mechanism are developed.

**Paper Organization**—The remainder of this paper is organized as follows. In Section II, we present the system and channel activity models. In Section III, we discuss the proposed nested quorum-based FH algorithm for pairwise rendezvous. In Section IV, we present our proposed AMQFH, CMQFH, and nested-CMQFH multicast rendezvous algorithms. We introduce our optimal channel ordering algorithm in Section V, followed by our adaptive FH and quorum selection algorithm in Section VI. We evaluate the protocol in Section VII. Finally, Section VIII concludes the paper.

II. System and Channel Models

A. System Model

We consider an opportunistic DSA network, operating over $L$ licensed channels, $f_1, f_2, \ldots, f_L$. SUs can hop 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 duration of $T$ seconds. A packet can be exchanged between two or more nodes if they hop onto the same channel in the same time slot. We assume that one time slot is sufficient to exchange one packet. If multiple SU pairs happen to rendezvous on the same channel in the same time slot, they use a CSMA/CA-like procedure to resolve channel contention.

Each SU $j, j = 1, \ldots, k$, has a unique FH sequence $w^{(j)}$, to be designed. The channel used in the $i$th slot of FH sequence $w^{(j)}$ is denoted by $w^{(j)}_i$, $w^{(j)}_i \in \{f_1, \ldots, f_L\}$. Channel $f_j$ is called a rendezvous frequency for $w^{(1)}, w^{(2)}, \ldots, w^{(k)}$ if there exists a rendezvous slot $i$ such that $w^{(m)}_i = f_j, \forall m \in \{1, \ldots, k\}$.

As in previous quorum-based FH designs, in our setup each FH sequence is divided into several time frames. Each frame consists of a block of time-frequency pairs.

**B. Channel Activity Model**

We assume that each channel $f_m, m = 1, \ldots, L$, can be in one of three states: idle (state 1), occupied by a PU (state 2), or occupied by an SU (state 3). Transitions between these states are assumed to follow a continuous-time Markov chain (CTMC) with state space $S = \{1, 2, 3\}$, as shown in Figure 3. For any $i$ and $j$ in $S$, $i \neq j$, we assign a nonnegative number $\alpha_{ij}^{(m)}$ that 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., $\alpha_{23}^{(m)} = 0, \forall m \in \{1, \ldots, L\}$. In contrast, when a PU becomes active on a channel occupied by an SU, the SU must leave that channel immediately, so $\alpha_{32}^{(m)} \neq 0$ in general. Let $A^{(m)} = [\alpha_{ij}^{(m)}]_{i,j}$ be the infinitesimal generator matrix for channel $m$, and let $P_t^{(m)}$ be a matrix whose $(i,j)$ entry, $p_t^{(m)}(i,j)$, is the probability that channel $m$ goes from state $i$ to state $j$ in $t$ seconds. It is known that [11]:

$$P_t^{(m)} = e^{tA^{(m)}}. \quad (1)$$

Without loss of generality, we assume that PUs become active on channel $m$ with rate $\lambda_p^{(m)}$, and terminate their activity with rate $\mu_p^{(m)}$, both according to Poisson processes. Similarly, SUs arrive on channel $m$ with
Fig. 3: State transition diagram for channel $f_m$. 

rate $\lambda_s^{(m)}$ and depart with rate $\mu_s^{(m)}$, both according to Poisson processes. Let $\pi^{(m)} = (\pi_1^{(m)}, \pi_2^{(m)}, \pi_3^{(m)})$ be the steady-state distribution for channel $m$. Then, $\pi^{(m)}$ can be written as:

$$\pi_1^{(m)} = \frac{\mu_p^{(m)}(\lambda_p^{(m)} + \mu_s^{(m)})}{(\lambda_p^{(m)} + \mu_p^{(m)})(\lambda_s^{(m)} + \lambda_p^{(m)} + \mu_s^{(m)})}, \quad \pi_2^{(m)} = \frac{\lambda_p^{(m)}}{(\lambda_p^{(m)} + \mu_p^{(m)})(\lambda_s^{(m)} + \lambda_p^{(m)} + \mu_s^{(m)})}, \quad \text{and} \quad \pi_3^{(m)} = \frac{\mu_p^{(m)}\lambda_s^{(m)}}{(\lambda_p^{(m)} + \mu_p^{(m)})(\lambda_s^{(m)} + \lambda_p^{(m)} + \mu_s^{(m)})}.$$ 

C. Metrics

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

**Expected Time-to-Rendezvous (TTR):** The TTR is defined as the time required for two or more nodes to rendezvous. The expectation is considered because of the existence of a randomly assigned part in our FH sequences.

**Expected Hamming Distance (HD):** The expected HD for two FH sequences $x = (x_1 \ldots x_n)$ and $y = (y_1 \ldots y_n)$ is defined as $E[\sum_{i=1}^n 1_{\{x_i \neq y_i\}} / n]$, where $1_{\{\cdot\}}$ is the indicator function and $n$ is the frame length. In addition to robustness to node compromise, FH sequences with higher HD result in a lower collision probability. A collision occurs when two or more neighboring multicast groups meet on the same channel in the same time slot.

III. NESTED QUORUM-BASED FH ALGORITHM FOR PAIRWISE RENDEZVOUS

Before describing our nested quorum-based FH design, we start with a few basic definitions that will facilitate further understanding of subsequent concepts in the paper.

A. Preliminaries

**Definition 1.** Given a set $Z_n = \{0, 1, \ldots, n - 1\}$, 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 $\text{rotate}(G, i) = \{(x + i) \mod n, x \in G\}$ to denote a cyclic rotation of quorum $G$ 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 $\forall i_1, i_2, \ldots, i_k \in Z_n$, $\bigcap_{j=1}^{k} \text{rotate}(G_j, i_j) \neq \emptyset$.

Quorum systems that enjoy the rotation $k$-closure property can be exploited to achieve asynchronous unicast and multicast communications, as will be explained later. An example of a quorum system that satisfies the rotation 2-closure property is the grid quorum system [9].

**Definition 4.** A grid quorum system arranges the elements of $Z_n$ as a $\sqrt{n} \times \sqrt{n}$ array. In this case, a quorum is formed from the elements of one column and one row of the grid.

Figure 4 illustrates the rotation closure property for two quorums $G$ and $H$, each with 7 elements, in a grid quorum system $Q$ under $Z_{16}$. One quorum’s column must intersect with the other quorum’s row, and vice versa. Hence, the two quorums have at least two intersections (labeled $I$ in Figure 4). If a grid quorum $G$ contains the elements of column $c$, then $G' = \text{rotate}(G, i)$ must contain all the elements of column $(c + i) \mod \sqrt{n}$. Furthermore, $G'$ must contain at least one element of every column of the grid quorum system $Q$. Hence, $G'$ intersects with all the quorums of $Q$ and all of its cyclically rotated quorums in at least two elements. In Figure 4, $G' = \text{rotate}(G, 1)$ and $H' = \text{rotate}(H, 2)$ intersect at the two elements labeled as $I'$.

**B. Nested Grid Quorum-Based FH Algorithm (NGQFH)**

We now introduce the NGQFH algorithm for pairwise rendezvous. In NGQFH, every frame of every FH sequence uses $\sqrt{n} - 1$ rendezvous frequencies (as before, $n$ denotes the frame length in slots). We
call the number of rendezvous frequencies in the frame of a FH sequence the *nesting degree* of this sequence. To explain the operation of the NGQFH algorithm, we take $n = 16$ (hence, each frame contains $\sqrt{n} - 1 = 3$ rendezvous frequencies). Generalization to other values of $n$ (with $n$ being a square of an integer) is straightforward.

1) Construct a grid quorum system $Q$ under $\mathbb{Z}_{16}$. $Q$ has 16 different quorums, each containing $2\sqrt{16}−1 = 7$ elements that comprise one row and one column of the $4 \times 4$ grid.

2) Construct an FH sequence $w$ using the following procedure:
   - Select the outer-most quorum $G_{1}^{(1)}$ from the quorum system $Q$ (e.g., $G_{1}^{(1)} = \{0, 1, 2, 3, 4, 8, 12\}$, where each entry represents the index of a time slot in a 16-slot frame). The quorum selection procedure will be explained in Section VI.
   - Assign the first rendezvous channel $h_{1}^{(1)} \in \{f_1, f_2, \ldots, f_L\}$ to the slots that correspond to $G_{1}^{(1)}$. The selection of rendezvous channels $h_{i}^{(1)}, i \in \{1, \ldots, \sqrt{n} − 1\}$, will be discussed in Section V.
   - Delete quorum $G_{1}^{(1)}$ from the original $4 \times 4$ grid and select the next outer-most quorum $G_{2}^{(1)}$ from the resulting $3 \times 3$ grid (e.g., $G_{2}^{(1)} = \{6, 9, 10, 11, 14\}$). Then, assign another rendezvous frequency $h_{2}^{(1)}$ to the slots that correspond to $G_{2}^{(1)}$.
   - Delete quorum $G_{2}^{(1)}$ from the $3 \times 3$ grid, and select the next outer-most quorum $G_{3}^{(1)}$ from the resulting $2 \times 2$ grid (e.g., $G_{3}^{(1)} = \{7, 13, 15\}$). Then, assign a third rendezvous channel $h_{3}^{(1)}$ to the FH slots that correspond to $G_{3}^{(1)}$.
   - Assign a random frequency $h_{x}^{(1)} \in \{f_1, f_2, \ldots, f_L\} \setminus \{h_{1}^{(1)}, h_{2}^{(1)}, h_{3}^{(1)}\}$ to each of the remaining unassigned slots in the frame, according to the procedure in Section V.
   - Repeat the above procedure for the other frames in sequence $w$.

3) Repeat Step 2 for other FH sequences.

Throughout this paper, $h_{i}^{(j)}, i \in \{1, \ldots, \sqrt{n} − 1\}$, denotes the $i$th quorum channel that is assigned to the $(\sqrt{n} − i + 1) \times (\sqrt{n} − i + 1)$ quorum $G_{i}^{(j)}$ in the $j$th frame. $h_{1}^{(j)}$ and $G_{1}^{(j)}$ are called the outer-most channel and the outer-most quorum of frame $j$. To simplify the notation, when the nesting degree is 1, we use $h_{i}^{(j)}$
Fig. 5: Generation of the nested quorums for two FH sequences, \( w \) and \( x \), according to Algorithm 1 (\( n = 16 \)). Since only one frame is considered, the superscript in \( h^{(j)}_i \) is dropped. The circled numbers refer to the sequence of steps.

to denote the rendezvous channel that is assigned to quorum \( G_i \) in the \( i \)th frame. A pseudo-code of the NGQFH algorithm for constructing one frame of the FH sequence \( w \) is shown in Algorithm 1. Figure 5 shows the resulting frames of two FH sequences \( w \) and \( x \), constructed according to Algorithm 1.

Algorithm 1 NGQFH Algorithm

```
Input: \( f = \{f_1, \ldots, f_L\} \), \( h = \{h_1^{(i)}, \ldots, h_{\sqrt{n}-1}^{(i)}\} \), \( U = \mathbb{Z}_n \), and a grid quorum system \( Q \) under \( U \)
Output: \( i \)th frame of \( w \)
1: for \( j = 1 : \sqrt{n} - 1 \) do
2: Select a \( (\sqrt{n} - j + 1) \times (\sqrt{n} - j + 1) \) grid quorum \( G_{j}^{(i)} \) from \( Q \)
3: for \( k = (i-1)n : in - 1 \) do
4: if \( k \in G_{j}^{(i)} \) then
5: \( w_k = h_j^{(i)} \)
6: end if
7: end for
8: if \( j \neq \sqrt{n} - 1 \) then
9: \( U = U \setminus \{G_{j}^{(i)}\} \). \( Q \) is a grid quorum system under \( U \)
10: end if
11: end for
12: for \( l = (i-1)n : in - 1 \) do
13: if \( l \notin \bigcup_{j=1}^{\sqrt{n}-1} G_{j}^{(i)} \) then
14: \( w_l = h_{\frac{l}{\sqrt{n}}}^{(i)} \), randomly chosen from \( f \setminus h \)
15: end if
16: end for
```

C. Features of the NGQFH Algorithm

NGQFH has two main attractive features. First, because of the nested generation of quorums, the overlap ratio between two FH sequences (fraction of rendezvous slots in a frame) is significantly higher than the overlap ratio for a non-nested design, herein referred to as GQFH. In GQFH, an FH sequence consists of only one rendezvous channel, assigned to a \( \sqrt{n} \times \sqrt{n} \) quorum. FH systems with a higher overlap ratio are more appropriate for DSA networks, given that PUs may suddenly become active on a rendezvous
channel. Besides having a higher overlap ratio, NGQFH involves multiple rendezvous frequencies per frame, which increases the likelihood of a successful rendezvous.

The advantage of a nested grid quorum with multiple rendezvous frequencies can be formalized by deriving the expected overlap ratio for GQFH and NGQFH, denoted by $O_{GQFH}$ and $O_{NGQFH}$, respectively. $O_{GQFH}$ is composed of the sum of two parts; the expected overlap ratio between the quorum-based assigned parts of the FH sequences, denoted by $O^Q_{GQFH}$, and the expected overlap ratio between the randomly assigned parts, denoted by $O^R_{GQFH}$. Similarly, $O_{NGQFH}$ is composed of $O^Q_{NGQFH}$ and $O^R_{NGQFH}$.

For a given $n$, $O^Q_{GQFH}$ and $O^Q_{NGQFH}$ can be determined numerically. $O^R_{GQFH}$ and $O^R_{NGQFH}$ can be expressed as in the following result.

**Result 1.** $O^R_{GQFH}$ and $O^R_{NGQFH}$ can be expressed as functions of $L$ and $n$ as follow:

\[
O^R_{GQFH} = \frac{(\sqrt{n} - 1)^2}{L} \left\{ 2 - \frac{(\sqrt{n} - 1)^2}{n} \right\} \quad (2)
\]

\[
O^R_{NGQFH} = \frac{1}{L} \left\{ 2 - \frac{1}{n^2} \right\}. \quad (3)
\]

**Proof.** Consider two FH sequences $x$ and $y$, each with frame length $n$. If $x$ and $y$ are constructed according to the GQFH algorithm using grid quorums $H_1$ and $H_2$, respectively, then one of the following three cases can occur:

Case 1: $H_1 = H_2$, which occurs with probability $\frac{1}{n}$.

Case 2: $H_1$ and $H_2$ have the same column or the same row, but not both, which occurs with probability $\frac{2(\sqrt{n} - 1)}{n}$.

Case 3: $H_1$ and $H_2$ have different columns and rows, which occurs with probability $\frac{((\sqrt{n} - 1)^2)}{n}$.

We know that the number of randomly assigned slots in $x$ and $y$ is $n - (2\sqrt{n} - 1) = (\sqrt{n} - 1)^2$. The numbers of randomly assigned slots that are common to $x$ and $y$ in cases 1, 2, and 3 are $(\sqrt{n} - 1)^2$, $(\sqrt{n} - 2)(\sqrt{n} - 1)$, and $(\sqrt{n} - 2)^2$, respectively.

To compute $O^R_{GQFH}$, we initially assume that the randomly assigned portions of $x$ and $y$ are disjoint. Then, we subtract the common slots that are counted twice. After some manipulations, the $O^R_{GQFH}$
expression in (2) can be easily obtained. The randomly assigned portions of \( x \) and \( y \) in the NGQFH algorithm consist only of one slot, which can be common to \( x \) and \( y \) with probability \( \frac{1}{n^2} \). The \( O_{NGQFH}^R \) expression in (3) can be obtained by following the same approach used in deriving \( O_{GQFH}^R \).

Figure 6 depicts \( O_{GQFH} \) and \( O_{NGQFH} \) vs. \( n \) for different values of \( L \). It can be observed that \( O_{NGQFH} \) is larger than \( O_{GQFH} \), and both decrease with \( n \).

The second attractive feature of NGQFH is its robustness to node compromise. Because the quorum-based assigned part of the FH sequence is the part that is intended to support rendezvous, if this part is compromised, the rendezvous capability may be eliminated or at least reduced significantly. NGQFH sequences are composed of \( \sqrt{n} - 1 \) nested quorums that are generally different for different frames of a given FH sequence, and also different for different FH sequences. Hence, if a node is compromised and its FH sequence is exposed, less information will be leaked about other FH sequences, compared with GQFH sequences. The number of different channel assignments for a given \( n \) (\( K_n \)) is given by:

\[
K_n = \prod_{j=0}^{\sqrt{n}-2} (\sqrt{n} - j)^2.
\]

(4)

Note that \( K_n \) increases with \( n \). A higher \( K_n \) value implies more robustness to node compromise. Detailed investigation of the resilience of FH sequences to node compromise is left for future work.

D. Asynchronous and Heterogeneous NGQFH
1) Asynchronous NGQFH: For NGQFH to function in the absence of node synchronization, we impose a constraint on Algorithm 1, as stated in the following result.

**Result 2.** FH sequences constructed according to Algorithm 1 can support asynchronous pairwise rendezvous if each FH sequence continues to select the same outer-most quorum and outer-most frequency in all frames of the FH sequence, i.e., for all FH sequences, $h^{(j)}_1$ and $G^{(j)}_1$ are the same for all $j$.

**Proof.** Result 2 is a direct consequence of the intersection and rotation 2-closure properties of the grid quorum system, and the fact that each frame in an FH sequence consists of $\sqrt{n} − 1$ nested grid quorums, each being assigned one frequency. ■

If the rendezvous channel varies from one frame to the next, then there is no guarantee that two misaligned FH sequences will be able to rendezvous. The condition in Result 2 is sufficient but not necessary. Thus, FH sequences may still rendezvous even if the outer-most quorum is changed in some frames, provided that this change does not occur very frequently. This is illustrated in Figure 7, where the outer-most quorum of sequence $w$ changes from $H_1$ to $H_2$, and the outer-most quorum of sequence $x$ changes from $H_3$ to $H_4$. The left shaded part of sequence $x$ in Figure 7 represents a cyclic rotation of $H_3$, and hence, by the rotation closure property of grid quorum systems, this part overlaps with quorum $H_1$ of sequence $w$. The right shaded part of sequence $x$ does not generally overlap with $H_1$ in $w$.

Because the condition in Result 2 is sufficient but not necessary, we require the outer-most rendezvous channel to be available for a certain number of slots in the current outer-most quorum in order to keep assigning this channel to the same outer-most quorum in the next frame. Otherwise, the outer-most channel is assigned to the quorum for which this channel is maximally available (i.e., the quorum that has the maximum number of available slots during which this frequency is predicted to be idle). Quorum selection will be discussed in detail in Section VI.

2) Heterogeneous NGQFH: In Figure 5, the two FH sequences $w$ and $x$ are constructed using the same rendezvous channels ($h_1$, $h_2$, and $h_3$), but with different quorums. To allow nodes to construct their FH sequences in a fully distributed way, depending on their own views of spectrum opportunities, we
consider a variant of NGQFH whereby each node assigns channels to quorum slots based on the forecasted availability of these channels. Note that even in a heterogeneous spectrum environment, neighboring nodes are still likely to partially overlap in their views of idle channels. Hence, when neighboring nodes construct their FH sequences independently based on the forecasted channels availability, they will likely end up having similar channel assignments. The algorithm for forecasting the channel state and accordingly assigning rendezvous channels is discussed in Section VI.

IV. QUORUM-BASED FH ALGORITHMS FOR MULTICAST RENDEZVOUS

In this section, we present two algorithms for constructing a set of FH sequences \( \geq 2 \) for multicast rendezvous. These algorithms have two main attractive features. First, they allow a node to construct its sequence by only knowing the size (but not identities) of the multicast group. Hence, these algorithms can be executed in a distributed way. Second, these algorithms can still function in the absence of node synchronization.

A. Uniform \( k \)-arbiter Multicast FH Algorithm (AMQFH)

The AMQFH algorithm uses the uniform \( k \)-arbiter quorum system, which exhibits the rotation \((k + 1)\)-closure property.

**Definition 5.** A \( k \)-arbiter quorum system \( Q \) under \( \mathbb{Z}_n \) is a collection of \( k + 1 \) quorums that satisfy the following \((k + 1)\)-intersection property [17]:

\[
\bigcap_{i=1}^{k+1} G_i \neq \emptyset, \quad \forall G_1, G_2, \ldots, G_{k+1} \in Q.
\]
One specific type of $k$-arbiter quorum systems that is of interest to us is the so-called uniform $k$-arbiter quorum system. Such a system satisfies [10]:

$$Q = \left\{ G \subseteq Z_n : |G| = \left( \left\lceil \frac{kn}{k+1} \right\rceil + 1 \right) \right\}. \tag{6}$$

For example, the quorum system $Q = \{\{0, 1, 2\}, \{0, 1, 3\}, \{0, 2, 3\}, \{1, 2, 3\}\}$ under $Z_4$ is a 2-arbiter quorum system. The intersection among any three quorums is not empty. This system is a uniform 2-arbiter because each quorum in $Q$ contains $\lfloor 2 \times 4/(2 + 1) \rfloor + 1 = 3$ elements of $Z_4$. It is known [10] that the uniform $k$-arbiter quorum system exhibits the rotation $(k + 1)$-closure property.

To create 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 $\varphi$, is greater than or equal to $k + 1$, i.e.,

$$\varphi \overset{\text{def}}{=} \left( \begin{array}{c} n \\ \left\lceil \frac{kn}{k+1} \right\rceil + 1 \end{array} \right) \geq k + 1. \tag{7}$$

To satisfy (7), $\left\lfloor \frac{kn}{k+1} \right\rfloor$ should be less than $n - 1$, which requires $n$ to be greater than $k + 1$ ($k$ and $n$ are positive integers, and $\frac{k}{k+1}$ is monotonically increasing in $k$).

We now explain AMQFH through an example. Consider a multicast group of 3 nodes. In AMQFH, each FH sequence consists of several time frames, each containing several slots. Because the uniform 2-arbiter quorum system satisfies the rotation 3-closure property (i.e., any three cyclically rotated quorums overlap in at least one slot), each frame is constructed using one quorum. Thus, the frame length will be $n$. We set $n$ to the smallest value that satisfies (7), i.e., $n = k + 2 = 4$. The following steps are used to obtain the various FH sequences:

1) Construct a universal set $Z_4 = \{0, 1, 2, 3\}$.

2) Construct a uniform 2-arbiter system $Q$ under $Z_4$.

3) Construct an FH sequence $w$ as follows:

- Select a quorum from $Q$ and assign it to $G_1$ (e.g., $G_1 = \{0, 1, 2\}$).
• Assign a frequency $h_1 \in \{f_1, f_2, \ldots, f_L\}$ to the FH slots in the given frame that correspond to $G_1$, and assign a random frequency $h_x$ to the other slots.

• Repeat the above procedure for the other frames in sequence $w$.

4) Repeat step 3 to construct the other FH sequences.

The channel and quorum selection procedures will be explained in Sections V and VI, respectively. Figure 8 shows three frames of FH sequences $w, x, y,$ and $z$, constructed according to AMQFH.

B. CRT Multicast FH Algorithm (CMQFH)

The CMQFH algorithm uses the CRT quorum system, which also exhibits the rotation $k$-closure property. The CRT is formally described as follows [19].

**Theorem 1.** Let $p_1, p_2, \ldots, p_k$ be $k$ positive integers that are pairwise relatively prime, i.e., $\gcd(p_i, p_j) = 1, \forall i, j \in \{1, \ldots, k\}$, where $\gcd(p_i, p_j)$ is the greatest common divisor of $p_i$ and $p_j$. Let $y = \prod_{l=1}^{k} p_l$ and let $z_1, z_2, \ldots, z_k$ be $k$ integers, where $z_i < p_i, \forall i \in \{1, \ldots, k\}$. Then, there exists a solution $I$ for the following system of simultaneous congruences:

$$z_1 \pmod{p_1} \equiv z_2 \pmod{p_2} \equiv \ldots \equiv z_k \pmod{p_k}.$$

Furthermore, any two solutions $I$ and $I'$ to the above system are congruent modulo $y$, i.e., $I' \equiv I \pmod{y}$. That is, there exists exactly one solution $I$ between 0 and $y - 1$.

Using Theorem 1, we can construct quorum systems that satisfy the rotation $k$-closure property, as in Theorem 2 [10].
Theorem 2. Let \( p_1, \ldots, p_k \) be \( k \) positive integers that are pairwise relatively prime, and let \( y = \prod_{i=1}^{k} p_i \).

The CRT quorum system \( Q = \{ G_1, \ldots, G_k \} \), where \( G_i = \{ p_i c_i, c_i = 0, \ldots, y/p_i - 1 \} \), satisfies the rotation \( k \)-closure property.

As an example of the CRT quorum system, consider three pairwise relatively prime numbers \( p_1 = 2, p_2 = 3, \) and \( p_3 = 5 \). Then, \( y = p_1 p_2 p_3 = 30 \). We can construct three quorums \( G_1 = \{ 0, 2, 4, \ldots, 28 \} \), \( G_2 = \{ 0, 3, 6, \ldots, 27 \} \), and \( G_3 = \{ 0, 5, 10, \ldots, 25 \} \) according to \( p_1, p_2, \) and \( p_3 \), respectively, under \( Z_{29} \). When \( z_1 = 0, z_2 = 1, \) and \( z_3 = 0 \), \( \bigcap_{j=1}^{3} \text{rotate}(G_j, z_j) = 10 \). It is not difficult to verify that \( \forall z_1, z_2, z_3 \in Z_{29} \), the three quorums \( G_1, G_2, \) and \( G_3 \) have an intersection. Thus, the CRT quorum system \( Q = \{ G_1, G_2, G_3 \} \) satisfies the rotation 3-closure property.

The CMQFH algorithm for generating \( k \) asynchronous multicast FH sequences is similar to the AMQFH algorithm, with two main differences. First, The frame length, denoted by \( g \), is equal to \( g = \prod_{i=1}^{\left\lceil \frac{k+1}{2} \right\rceil} p_i \). Second, CMQFH uses the CRT quorum system instead of the uniform \((k - 1)\)-arbiter quorum system.

Remark 1. AMQFH and CMQFH are implemented in a distributed way as follows. The source node uses a series of pairwise rendezvous to communicate only the number of nodes in the multicast group to the target multicast group. Then, each receiving node constructs its own multicast FH sequence.

C. AMQFH vs. CMQFH: Speed vs. Security

This section compares AMQFH and CMQFH. Both algorithms are implemented in a distributed way as follows. First, the source node uses a series of pairwise rendezvous to communicate the number of nodes in the multicast group to the target multicast group. Then, each receiving node constructs its own multicast FH sequence. Note that for AMQFH and CMQFH, knowing the number of nodes in the multicast group is enough to construct the multicast FH sequences.

In Appendix A, analytical results for the expected TTR and expected HD of AMQFH and CMQFH algorithms are provided. These results are plotted in Figures 9, 10, and 11. Figures 9 and 10 show the expected TTR of AMQFH and CMQFH vs. the multicast group size for different values of \( L \). The expected TTR of CMQFH is much higher than AMQFH. For both algorithms, the expected TTR increases with
the multicast group size. Figure 11 depicts the expected HD vs. the multicast group size for AMQFH and CMQFH. As the multicast group size increases, the HD of CMQFH (denoted by $H_c$) increases but the HD of AMQFH (denoted by $H_a$) decreases, and hence the gap between $H_c$ and $H_a$ increases with the increase in the size of the multicast group.

As shown in Figures 9, 10, and 11, the TTR of CMQFH is much larger than that of AMQFH, but its average HD is also much higher. To provide a tradeoff between speed of rendezvous and robustness against node compromise, we propose a modified version of CMQFH that borrows the nesting design of NGQFH, proposed in Section III-B. We call the modified CMQFH algorithm the nested-CMQFH. As will be shown in Section VII, nested-CMQFH is faster than CMQFH, but not as fast as AMQFH. At the same time, the HD of nested-CMQFH is larger than that of AMQFH, but not as large as CMQFH.

Similar to NGQFH, in nested-CMQFH each frame of an FH sequence contains a certain number of quorums, which is called the nesting degree. In contrast to NGQFH, the nesting degree in nested-CMQFH is different for different FH sequences and it depends on the prime number that is used in constructing the FH sequence. The nesting degree constitutes a tradeoff between TTR and HD. Large values of the nesting degree result in a small TTR, but also a small HD. In our design, the FH sequence that uses a prime number $p_i$ will have a nesting degree of $p_i - 1$. Using this nesting degree, if $p_j > p_k$ then $\frac{p_j - 1}{p_j} > \frac{p_k - 1}{p_k}$. This property is needed for our quorum selection mechanism, discussed in Section VI.

Figure 12 illustrates the nested-CMQFH design when the multicast group size is 3. The prime numbers used in constructing FH sequences $x$, $y$, and $z$ are 5, 3, and 2, respectively, and the corresponding nesting
degrees are 3, 2, and 1, respectively. Hence, sequence $x$ will have three nested quorums, each of 5 slots, and each quorum is assigned a different channel (the same treatment is done for sequences $y$ and $z$).

![Diagram](image_url)

**Fig. 12:** Nested-CMQFH FH construction algorithm.

**Result 3.** FH sequences constructed according to AMQFH (nested-CMQFH) can support asynchronous multicast rendezvous if each FH sequence continues to select the same quorum (outer-most quorum) and frequency (outer-most frequency) in all frames of the FH sequence.

**Proof.** Result 3 is a direct consequence of the intersection and rotation $(k + 1)$-closure properties of the uniform $k$-arbiter and CRT quorum systems, and the fact that each frame in an FH sequence is constructed using one quorum. ■

**Remark 2.** Similar to NGQFH, AMQFH and nested-CMQFH will be evaluated in a heterogeneous environment with different heterogeneity levels. We define the heterogeneity level, denoted by $\kappa$, as the fraction of channels that have different parameters ($\lambda_p^{(i)}$, $\lambda_s^{(i)}$, $\mu_p^{(i)}$, and $\mu_s^{(i)}$ for channel $i$) between every pair of nodes in a multicast group. The randomly assigned slots in AMQFH and nested-CMQFH are assigned among the top $L\kappa_{\max} + 1$ channels, where $\kappa_{\max}$ is the maximum heterogeneity level that the network can have. Channel ordering will be discussed in Section V.

**Remark 3.** As will be shown in Section VII, AMQFH is faster than nested-CMQFH in homogeneous environments (i.e., $\kappa = 0$) as well as environments with limited heterogeneity (i.e., $\kappa$ is small), but it is less robust than nested-CMQFH. However, in environments with high heterogeneity (i.e., $\kappa$ is large) nested-CMQFH is faster as well as more robust than AMQFH.
V. Optimal Channel Ordering

In the previous sections, we presented unicast and multicast rendezvous algorithms without explaining how channels $h_1^{(j)}, \ldots, h_{\sqrt{\pi-1}}^{(j)}$ are selected in the $j$th frame. Each $h_i^{(j)}$ channel is assigned to $2(\sqrt{n-i+1}) - 1$ slots in the $j$th frame. Thus, $h_1^{(j)}$ is assigned to more slots than $h_2^{(j)}$, which in turn is assigned to more slots than $h_3^{(j)}$, and so on. Accordingly, we select $h_1^{(j)}$ to be the “best” available channel in $\{f_1, \ldots, f_L\}$, $h_2^{(j)}$ as the next best available channel, and so on. This requires each node to independently sort available channels (no message exchange is assumed between the nodes). In here, the best available channel is selected according to several factors, as will be explained in this section. With this approach, better channels are assigned to outer quorums (with more rendezvous slots in a frame). In addition to channel assignment, the sorted list of channels is also used to sequentially sense channels, such that the best channel will be sensed first, followed by the next best channel, and so on.

Furthermore, in the previous sections, 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 available). However, the time complexity of this exhaustive search is $O\left(\left(\frac{L}{\sqrt{\pi-1}}\right)\left(\frac{1}{n!}\right)^2\right)$, where $\bar{L}$ is the number of available channels. To avoid performing an expensive exhaustive search for each frame and also delaying making the decision until all channels are sensed, we address the problems of quorum selection and channel assignment separately, and propose a one-time sorting algorithm that prioritizes channels. In this section, we present a channel ordering algorithm, and in Section VI we address the quorum selection problem.

In our approach, channels are sorted based on their probabilistic availability, which is determined by the channel average availability time and its fluctuation level. The fluctuation level of a channel affects its prediction accuracy. Channel activity prediction is more conservative when a channel exhibits higher fluctuations for a given mean channel availability time. Conservative prediction of channel availability results in less exploitation of actually available slots. In addition to probabilistic channel availability,
A sorting mechanism also considers the probabilities of collision with PUs/SUs over these channels, which we require to be below a specific threshold. To sort channels based on these criteria, we propose an optimization problem for NGQFH. This ordering mechanism starts over when the estimate of at least one channel parameter changes.

We formulate the channel sorting problem as a multi-objective linear programming problem. Let \( q_m \) be a weight associated with channel \( f_m, m \in \{1, \ldots, L\} \). The weights will be used for two different purposes. First, in the quorum-based assigned 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, in the randomly assigned (non-quorum) slots, these weights will be interpreted as probabilities, such that channel \( f_m \) will be assigned to non-quorum slots with probability \( q_m \). We obtain the optimal value of the vector \( q = (q_1, q_2, \ldots, q_L) \) that maximizes a convex combination of the average channel availabilities, as a primary objective function, and maximizes the channel state prediction accuracy as a secondary objective function, subject to the PUs/SUs collisions constraints. Prediction accuracy is related to the time that a channel spends in one state, averaged over all possible states. In general, the prediction accuracy improves with the increase in this average time.

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 \( m \) in state \( i \) and the first time that channel \( m \) returns to state \( i \) after leaving it, respectively. Let \( T_i(m) = \mathbb{E}[T_i(m)] \) and \( R_i(m) = \mathbb{E}[R_i(m)] \). Following standard Markov analysis, the fraction of time that channel \( m \) spends in state \( i \) (i.e., \( T_i(m)/(T_i(m) + R_i(m)) \)) is \( \pi_i(m) \), which was given in Section II-B. \( T_i(m), i \in \{1, 2, 3\} \) can be expressed as follows:

\[
T_1(m) = \frac{1}{\lambda_p(m) + \lambda_s(m)}, \quad T_2(m) = \frac{1}{\mu_p(m)}, \quad \text{and} \quad T_3(m) = \frac{1}{\lambda_p(m) + \mu_s(m)}.
\]

Our multi-objective channel sorting problem is formulated as a two-stage sequential optimization problem. Problem 1 is the first stage and Problem 2 is the second stage.
Problem 1.

\[
\text{maximize } q = (q_1; q_2; \ldots; q_L) \left\{ \mathcal{U}(q) \text{ def } = \sum_{m=1}^{L} \pi_1^{(m)} q_m \right\}
\]

Subject to.

\[
\begin{bmatrix}
1 - \prod_{u=0}^{\sqrt{n} - 1} (1 - p_{(n+i+\sqrt{n}u)}^{(m)}(1, s)) \prod_{v=0, v\neq i}^{\sqrt{n} - 1} (1 - p_{(n+j+\sqrt{n}v)}^{(m)}(1, s))
\end{bmatrix} q_m < \lambda_{2,\text{Col}}^{(m)}(n), \quad (8)
\]

\[
\forall s \in \{2, 3\}, \forall m \in \{1, \ldots, L\}, \forall i, j \in \{0, \ldots, \sqrt{n} - 1\}
\]

\[
\sum_{m=1}^{L} q_m = 1
\]

\[
0 \leq q_m \leq 1, \forall m \in \{1, \ldots, L\}
\]

where \(\lambda_{2,\text{Col}}^{(m)}(n)\) and \(\lambda_{3,\text{Col}}^{(m)}(n)\) are prespecified thresholds on the probabilities of collisions with PUs and other SUs, respectively. Note that both sets of thresholds are functions of the frame length \(n\) and channel \(m\). The objective function in Problem 1 represents a convex combination of the average channel availabilities \(\pi_1^{(m)}, m = 1, \ldots, L\). Constraint (8) restricts the collision probabilities with PUs and SUs, while considering the specific structure of grid quorum systems. Let \(q^*_I\) be an optimal solution to Problem 1, and let \(\mathcal{U}^*_I = \mathcal{U}(q^*_I)\). The goal of the second optimization stage is to give more priority to less fluctuating channels (i.e., channels that, on average, spend longer time in a given state).

Problem 2.

\[
\text{maximize } q = (q_1; q_2; \ldots; q_L) \left\{ \mathcal{F}(q) = \sum_{m=1}^{L} \mathcal{F}_m q_m \text{ def } = \sum_{m=1}^{L} \sum_{i=1}^{3} T_i^{(m)} q_m \right\}
\]

Subject to.

\[
\mathcal{U}^*_I (1 - \epsilon) < \mathcal{U}(q). \quad (11)
\]

Problem 2 aims at maximizing a convex combination of the average times that individual channels spend in various states (\(\mathcal{F}(q)\)) subject to constraints (8) – (10), in addition to the new constraint in (11). Let \(\mathcal{F}^*\) be the optimal value of \(\mathcal{F}(q)\) in Problem 2, and let \(\mathcal{U}^*\) be the corresponding value of \(\mathcal{U}(q)\). In (11), \(\epsilon \in [0, 1]\) restricts the reduction in the first objective function optimal value (i.e., \(\mathcal{U}^*_I - \mathcal{U}^*\)). Increasing \(\epsilon\) increases the effect of the second objective function on channel ordering. Figures 13 and 14 depict \(\mathcal{U}^*\) and \(\mathcal{F}^*\), respectively vs. \(\epsilon\) for different values of \(\lambda = \lambda_{2,\text{Col}}^{(m)}(n) = \lambda_{3,\text{Col}}^{(m)}(n)\) and \(n = 16\). Up to a certain value, increasing \(\epsilon\) decreases \(\mathcal{U}^*\) and increases \(\mathcal{F}^*\).
As will be explained in detail in Section VI, after channel ordering, channel assignment is performed according to the predicted channel availability. Figures 15 and 16 depict, respectively, the effect of $\epsilon$ and $\lambda$ on the average number of available slots (i.e., useful slots), averaged over all possible channel assignments.

**Remark 4.** Other formulations can be used to optimize channel ordering for AMQFH, CMQFH, and nested-CMQFH using the same ordering criterion of NGQFH. To achieve this, constraint (8) needs to be modified according to the structures of the uniform $k$-arbiter and CRT quorum systems. Exact formulations for channel ordering in AMQFH, CMQFH, and nested-CMQFH can be found in [1].

**Remark 5.** 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 (similar to NGQFH), where the rendezvous does not depend only on a single quorum channel, but on several quorum channels.

**Remark 6.** The nesting design of NGQFH and nested-CMQFH improves the robustness of these
algorithms against an intelligent jammer, who may order the channels in a similar way and keep jamming
the best channel continuously. Recall that even if two nodes sort the channels following the same criterion,
they may end up having different orders because of heterogeneity.

VI. ADAPTIVE FH AND QUORUM SELECTION ALGORITHM

We now explain how quorums are selected in the NGQFH, AMQFH, CMQFH, and nested-CMQFH
algorithms. As mentioned before, our quorum selection procedure relies on forecasting the states of various
channels in the next frame, driven by proactive out-of-band sensing of their states in the current frame.
Because this procedure results in online adaptation of quorum (for AMQFH and CMQFH)/quorums (for
NGQFH and nested-CMQFH) selection, and hence online adaptation of the FH sequences, it is an adaptive
FH algorithm. We now explain the adaptive FH algorithm for NGQFH.

Consider the $j$th frame of a given FH sequence. The node that follows this FH sequence starts sensing
channels according to the order obtained in Section V. Let $\{d_1, \ldots, d_{\sqrt{n} - 1}\}$ be the best $\sqrt{n} - 1$ available
channels, ordered decreasingly according to their quality. According to Algorithm 1, we assign $d_1$ to a
$\sqrt{n} \times \sqrt{n}$ quorum of $2\sqrt{n} - 1$ slots, $d_2$ to a $(\sqrt{n} - 1) \times (\sqrt{n} - 1)$ quorum of $2(\sqrt{n} - 1) - 1$ slots, and so
on. In general, the $k$th outer-most $\sqrt{n} - k + 1 \times \sqrt{n} - k + 1$ quorum $G_k^{(j)}$, $k \in \{1, \ldots, \sqrt{n} - 1\}$, of
$2(\sqrt{n} - k + 1) - 1$ slots is selected from all possible quorums so as to maximize the number of quorum
slots for which $d_k$ is idle with probability greater than a threshold $\gamma$. If more than one quorum results
in the same maximum number of slots, we break the tie based on the average idle probability of $d_k$,
averaged over all slots that belong to $G_k^{(j)}$. Formally, the problem of selecting quorum $G_k^{(j)}$ is formulated
as follows:

$$\max_{G_k^{(j)}} \left\{ A(k, n) = \sum_{i=0}^{n-1} 1_{\{p_{(n-k+\sqrt{n}+i)}^{(k')}T(1,1) \geq \gamma\}} + \frac{1}{2(\sqrt{n} - k + 1) - 1} \sum_{i=0}^{n-1} p_{(n-k+\frac{k'}{2}+i)}^{(k')}T(1,1) \right\}$$

(12)

where $G_k$ is the set of all possible $\sqrt{n} - k + 1 \times \sqrt{n} - k + 1$ quorums, $1_{\{\cdot\}}$ is the indicator function,
and $\tau_s$ is the sensing time for one channel. Given that $d_k = f_{k'}$ where $k' \in \{1, \ldots, L\}$, $p_{(n-k+\frac{k'}{2}+i)}^{(k')}T(1,1)$
is the probability that $d_k$ will remain available in the $i$th slot of the next frame, given that it is currently available. Note that $p_i^{(k')}(n-k\frac{x}{s}+i)T(1,1) = 0, \forall i \notin H_k$ when $A(k, n)$ is evaluated at $H_k$. The computation of $p_i(x, y)$ was explained in Section II-B. The second term in (12) is always $< 1$. Hence, for two different quorums $H_k(1)$ and $H_k(2)$, if $H_k(1)$ has more probabilistically available slots, then $G_k^{(j)}$ is set to $H_k(1)$.

In AMQFH and CMQFH algorithms, each frame consists of only one quorum. This quorum is selected using a procedure similar to the outer-most quorum selection procedure in NGQFH. In contrast to NGQFH, nested-CMQFH has the unique feature that all nested quorums of a given frame of an FH sequence have the same size. Because of this feature, channel-quorum assignment in nested-CMQFH needs to be carried out for all nested quorums of a given frame jointly, unlike in NGQFH where each quorum is selected independently according to (12). Formally, the problem of selecting the nested quorums in the $j$th frame of the FH sequence that uses prime number $p_i$ is formulated as follows:

$$\begin{align*}
\text{maximize} & \quad \mathcal{Q}_i \left\{ B(g, p_i) = \sum_{j=1}^{p_i-1} \sum_{l=0}^{g-1} 1 \left\{ p_i^{(j')} \left( \frac{g-j}{p_i} \cdot s + l \right) T(1,1) \geq \gamma \right\} + \frac{1}{(p_i - 1)} \sum_{j=1}^{p_i-1} \sum_{l=0}^{g-1} p_i^{(j')} \left( \frac{g-j}{p_i} \cdot s + l \right) T(1,1) \right\} \right.
\end{align*}$$

where $\mathcal{Q}_i$ is the set of CRT quorums that correspond to prime number $p_i$. 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 larger numbers of quorum slots (recall that the number of quorum slots for prime number $p_i$ is $\frac{g}{p_i} (p_i - 1)$). 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 quorum channels used in the quorum slots.

VII. PERFORMANCE EVALUATION

This section evaluates the performance of our unicast and multicast rendezvous algorithms. NGQFH is studied under different values of $\gamma$ in (12), and frame lengths. NGQFH is compared with M-QCH,
A-QCH, JS_SM, JS_AM, SYNC-ETCH, and ASYNC-ETCH. AMQFH and nested-CMQFH are studied under different values of $\gamma$ in (13), and group sizes. Both unicast and multicast algorithms are studied under different heterogeneity levels $\kappa$. 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 ($\kappa L$ channels), the parameters of a given channel are different at different nodes. We evaluate the unicast algorithms based on the TTR and the prediction accuracy, indicated by the collision rates with PUs/SUs and by missed opportunities (i.e., number of actually available slots that were considered unavailable). The multicast algorithms are evaluated using the same metrics, in addition to the average percentage HD. Our algorithms are simulated under a realistic setting where nodes start rendezvous at different points in time, and in the absence of node synchronization. Specifically, the misalignment between FH sequences is randomly selected in each experiment. The 95% confidence intervals are indicated unless they are very tight.

In our simulations, we use ten licensed channels, each is characterized by specific values of $\lambda_p^{(m)}$, $\lambda_s^{(m)}$, $\mu_p^{(m)}$, and $\mu_s^{(m)}$, which result in specific levels of availability and fluctuation as shown in Figure 17. 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.

A. Unicast (NGQFH)

This section evaluates NGQFH and compares it with A-QCH, M-QCH, JS_AM, JS_SM, ASYNC-ETCH, and SYNC-ETCH in various setups of node synchronization and spectrum heterogeneity. A-QCH is an asynchronous algorithm whereas M-QCH is a synchronous algorithm. Similarly, ASYNC-ETCH is an asynchronous algorithm whereas SYNC-ETCH is a synchronous algorithm. JS_AM was designed for heterogeneous spectrum environments whereas JS_SM was designed for homogeneous environments.

For A-QCH, we use the results in [16] to generate minimal and majority cyclic quorums for different
frame lengths. According to [16], the smallest frame length is 7. A-QCH and M-QCH are simulated assuming that nodes select a common channel in each frame, as mentioned in [4]. It is not explained in [4] how this can be accomplished in a distributed way. A-QCH, M-QCH, ASYNC-ETCH, and SYNC-ETCH are implemented with a per-slot sensing capability; if the channel is unavailable, the node refrains from transmitting leaving no collisions with PUs. In contrast to the per-slot sensing performed in A-QCH, M-QCH, ASYNC-ETCH, and SYNC-ETCH, JS_AM and JS_SM try to avoid unavailable channels by replacing them with available channels after constructing the frame and before start hopping. To compare JS_AM and JS_SM with NGQFH, we assume sensing is performed in JS_AM and JS_SM on a per-frame basis.

1) Prediction Accuracy: Figures 18 and 19 depict the collision rates (for NGQFH, JS_SM, and JS_AM) and missed opportunity rates (for NGQFH, A-QCH, M-QCH, ASYNC-ETCH and SYNC-ETCH) vs. the frame length for different values of $\gamma$. Note that the missed opportunity rate of A-QCH is equal to the average channel occupancy in our setup ($\sim 40\%$), and is independent of the frame length. Similarly for ASYNC-ETCH, except that when the frame length is sufficiently small missed opportunity rate decreases. This is because A-QCH and ASYNC-ETCH equally access all the channels, and they refrain from transmitting over occupied channels. The missed opportunity rate of JS_AM equals zero, but the collision rate is high. This is because JS_AM assumes that channel availability does not change during the frame, and it does not use any channel prediction mechanism. If NGQFH follows a conservative approach by selecting $\gamma$ to be very large (e.g., $\gamma > 0.89$), then the missed opportunity rate increases with the frame length. This is because of the reduction in the prediction accuracy due to increasing the forecasting period. Moreover, the number of quorum channel increases with the frame length, which may result in using low quality channels as rendezvous channels. On the other hand, if NGQFH accesses the slots aggressively by selecting a small $\gamma$ (e.g., $\gamma = 0.75$), then the collision rate increases with the frame length. As shown in Figure 20, the best $\gamma$ that results in the smallest TTR is a function of $\kappa$. 
Fig. 17: Example of 10 channels with different $\pi_1^{(m)}$ and $\gamma_1^{(m)}$.

Fig. 18: Prediction accuracy for NGQFH, A-QCH, JS_SM (JS_AM is similar), and ASYNC-ETCH.

Fig. 19: Missed opportunity rates for M-QCH and SYNC-ETCH.

2) TTR: Figures 21 and 22 depict the average TTR of NGQFH, A-QCH, JS_SM, JS_AM, and ASYNC-ETCH vs. the frame length for homogeneous and heterogeneous environments. Based on the previous discussion, we consider two values of $\gamma$ for NGQFH, 0.875 (for $\kappa = 0$) and 0.75 (for $\kappa = 0.4$). Even with the strong assumption made in A-QCH that nodes select a common channel in each frame, NGQFH has significantly smaller TTR than A-QCH irrespective of the frame length. It also has smaller TTR than JS_AM and JS_SM when the number of channels equals 10 (i.e., frame length of JS_AM and JS_SM is 33 slots). The improvement in TTR achieved by NGQFH increases with the number of channels, since the TTR of JS_AM and JS_SM increase with the number of channels as shown in [13] while NGQFH is not affected. For small frame lengths, JS_AM and JS_SM has small TTR but at the cost of having high collisions, as shown in Figure 18. Note that the TTR of NGQFH remains almost the same as the frame length increases even though the collision rate increases with the frame length. This is because of the increase in the number of rendezvous channels with the frame length. In Figure 21(b), fully distributed A-QCH represents a variant of A-QCH where nodes select their quorum channels independently. Fully distributed A-QCH is simulated starting from a frame length of 10. Frames with lengths of 4 or 9 slots will not have any randomly assigned slot. Since quorum channels might be different at the rendezvousing nodes, nodes may not rendezvous if they rely only on quorum channels. Assuming the same setup as
NGQFH where nodes select their quorum channels independently, fully distributed A-QCH has much larger TTR than NGQFH. This corroborates our claim that a rendezvous protocol requires a distributed mechanism for channel ordering, and proves the efficiency of our proposed ordering mechanism.

Figures 23 and 24 depict the average TTR of NGQFH, M-QCH, JS_SM, JS_AM, and SYNC-ETCH vs. the frame length for homogeneous and heterogeneous environments, assuming nodes are synchronized. As shown in the figures, NGQFH achieves smaller TTR than M-QCH, JS_SM, JS_AM, and SYNC-ETCH.

B. Multicast

1) Prediction Accuracy: Figure 25 depicts the collision and missed opportunity rates of AMQFH and nested-CMQFH vs. $\gamma$ for two group sizes. The collision rate decreases whereas the missed opportunity rate increases with $\gamma$. Both collision and missed opportunity rates increase with the group size.
2) TTR: Figure 26 shows the effect of $\gamma$ on TTR for different values of $\kappa$. The best $\gamma$ that results in the smallest TTR depends on $\kappa$. Figure 27 shows the average TTR of AMQFH and nested-CMQFH vs. group size for different values of $\kappa$. TTR is averaged over all runs that result in reasonable TTR ($< 400$ slots). Average TTR increases with both group size and $\kappa$. AMQFH has smaller average TTR than nested-CMQFH. The percentage number of runs where TTR exceeds 400 slots is shown in Figure 28. This number increases with $\kappa$ and it is smaller for nested-CMQFH than AMQFH, especially when $\kappa$ exceeds 0.2. This is because AMQFH does not support a nested design.

3) HD: Figure 29 plots the HD of AMQFH and nested-CMQFH vs. group size for different values of $\kappa$ and $\gamma$. The HD increases with $\kappa$, and decreases with $\gamma$. The HD is non-increasing with the group size.
Fig. 25: Prediction accuracy for AMQFH and nested-CMQFH.

Fig. 26: TTR vs. $\gamma$ for AMQFH and nested-CMQFH.

Fig. 27: TTR vs. group size for AMQFH and nested-CMQFH.

Fig. 28: TTR vs. $\kappa$ for AMQFH and nested-CMQFH.

Fig. 29: HD for AMQFH and nested-CMQFH.
VIII. CONCLUSIONS

In this paper, we developed asynchronous algorithms for pairwise and multicast rendezvous in spectrum-heterogeneous DSA networks. To account for PU dynamics, we developed an algorithm for adapting the proposed FH designs on the fly. This adaptation was achieved through an optimal mechanism for channel sensing and assignment, and a quorum selection mechanism.

Simulation results were obtained under different settings. If $\gamma$ is selected appropriately, NGQFH achieves a significant improvement in TTR and detection accuracy compared to previous unicast rendezvous algorithms. The best $\gamma$ is a function of $\kappa$. AMQFH can provide smaller TTR than nested-CMQFH, especially for $\kappa = 0$. However, the percentage number of runs where TTR exceeds 400 slots is higher for AMQFH when $\kappa > 0.2$. Nested-CMQFH achieves better HD than AMQFH.

REFERENCES

A. AMQFH vs. CMQFH: Speed vs. Security

1) Expected TTR: By examining the structures of the uniform k-arbiter and CRT quorum systems, the expected TTR of AMQFH and CMQFH, denoted by $T_a$ and $T_c$, respectively, can be expressed as follows:

**Result 4:** $T_a$ is given by:

$$T_a = \sum_{i=1}^{n-1} \left[ i \Gamma(\gamma_{i+1}) \prod_{j=1}^{i} (1 - \Gamma(\gamma_j)) \right]$$

where $\Gamma(\gamma_i)$ and $\gamma_i, i = 1, \ldots, n-1$, are given by ($k$ is the multicast group size minus one for AMQFH):

$$\Gamma(\gamma_j) = \sum_{i=0}^{k} \left[ \left( \frac{k+1}{i} \right) \gamma_j^{k+1-i} \left( \frac{1 - \gamma_j}{L} \right)^i \right] + \left( \frac{1}{L} \right)^k (1 - \gamma_j)^{k+1}$$

$$\gamma_i = \frac{\lfloor kn/k+1 \rfloor - i + 2}{n} + \frac{i - 1}{n} \times \frac{\lfloor kn/k+1 \rfloor - i + 3}{n - i + 1}. \tag{16}$$

**Proof.** Result 4 can be easily obtained, knowing that $\Gamma(\gamma_i)$ represents the probability that slot $i$ is a rendezvous slot and $\gamma_i$ represents the probability that slot $i$ is a quorum slot (i.e., assigned a rendezvous frequency). Recall that nodes can rendezvous during a quorum slot or during a randomly-assigned slot. Hence, equation (15) considers rendezvous under all possible combinations of $i$ randomly assigned slots and $k+1-i$ quorum slots. Equation (14) is the discrete expectation formula of TTR, which takes values in $\{0, 1, \ldots, n-1\}$. Note that the probability that slot $i$ is a quorum slot ($\gamma_i$) depends on $i$. In (16), $\gamma_i$ is computed by conditioning on the states (quorum/non-quorum) of the slots $j < i$. Because the quorum slots in the uniform $k$-arbiter quorum system are consecutive (see e.g., Figure 8), we have only two cases; all slots $j < i$ were quorum slots (which occurs with probability $(n - i + 1)/n$), or all slots except one were quorum slots (which occurs with probability $(i - 1)/n$). Hence, the two terms in (16).

**Result 5:** $T_c$ is given by:

$$T_c = \Theta \sum_{i=1}^{n-1} i(1 - \Theta)^i$$

\[\]
where \( \Theta \) is given by \((k \text{ is the multicast group size for CMQFH})\):

\[
\Theta = \sum_{i=0}^{k-1} \left[ \left( \frac{1}{L} \right)^i \sum_{\forall \{e_1, \ldots, e_{k-i}\} \in \{p_1, \ldots, p_k\}} \prod_{j=k-i+1}^{k} \frac{(e_j - 1)/e_j}{e_1 \cdots e_{k-i}} + \left( \frac{1}{L} \right)^{k-1} \prod_{l=0}^{k-1} \frac{e_l - 1}{e_l}. \right] \tag{18}
\]

**Proof.** Similar to (14), equation (17) represents the discrete expectation formula of TTR, which takes values in \( \{0, 1, \ldots, n-1\} \). \( \Theta \) represents the probability that a given slot is a rendezvous slot in CMQFH, similar to \( \Gamma \) in AMQFH. Result 5 can be easily obtained after considering the following:

- In CMQFH, the probability that a given slot is a quorum slot in the FH sequence that uses prime number \( p_i \) is \( 1/p_i \), and the probability that it is a randomly-assigned slot is \( 1 - 1/p_i = (p_i - 1)/p_i \) (see e.g., Figure 12). Note that, in contrast to AMQFH, this probability is independent of the slot index. This comes from the specific structure of the CRT quorum system used in CMQFH, where quorum slots are equally-spaced in the FH sequence.

- There is only one *multicast rendezvous slot* in a CMQFH frame of length \( p_1p_2 \cdots p_k \). A multicast rendezvous slot is a slot where all the \( k \) nodes are at a quorum slot. Therefore, the probability that a given slot is a multicast rendezvous slot is \( 1/(p_1p_2 \cdots p_k) \). \( \blacksquare \)

**2) Expected HD:** In AMQFH, the expected HD is the same for all pairs of FH sequences, whereas in CMQFH they are different for different pairs. Thus, for CMQFH, the expected value over all pairs of FH sequences is computed.

**Result 6:** Let \( \phi \overset{\text{def}}{=} n - \left\{ \left\lfloor \frac{kn}{k+1} \right\rfloor + 1 \right\} \). Then, the expected HD of AMQFH, denoted by \( H_a \), and its upper bound value, denoted by \( H_{a,\text{best}} \), are given by:

\[
H_a = \frac{L - 1}{nL} \left\{ \frac{\phi - 1}{\phi + 1} + \frac{\phi}{\varphi} \right\} \tag{19}
\]

\[
H_{a,\text{best}} = \frac{\phi + 1}{n} \tag{20}
\]

where \( \varphi \) is defined in (7).

**Proof.** Result 6 can be obtained by noticing that \( H_{a,\text{best}} \) corresponds to the case where different nodes
select different FH sequences, and nodes cannot rendezvous during the randomly assigned slots. \( \mathcal{H}_a \) represents the general case where nodes can select different FH sequences (occurs with probability \( (\varphi - 1)/\varphi \)) or the same FH sequence (occurs with probability \( 1/\varphi \)), hence the two separate terms in (19). \( \phi \) represents the number of randomly assigned slots in each frame of the FH sequence. ■

**Result 7:** The expected HD of CMQFH, denoted by \( \mathcal{H}_c \), and its upper bound value, denoted by \( \mathcal{H}_{c,\text{best}} \), are given by:

\[
\mathcal{H}_c = \frac{L - 1}{2Lk^2} \sum_{i=1}^{k} \sum_{j=1}^{k} \left( 1 - \frac{1}{p_ip_j} \right)
\]

\[
\mathcal{H}_{c,\text{best}} = \frac{1}{2\binom{k}{2}} \sum_{i=1}^{k} \sum_{j=1}^{k} \left( 1 - \frac{1}{p_ip_j} \right)
\]

**Proof.** \( \mathcal{H}_{c,\text{best}} \) is defined similar to \( \mathcal{H}_{k,\text{best}} \). Result 7 can be easily obtained if we consider the fact that the number of similar quorum slots between two CMQFH-based FH sequences that use prime numbers \( p_i \) and \( p_j \) is \( \frac{y}{p_ip_j} \), where \( y \) is the frame length. ■