Rendezvous in Dynamic Spectrum Wireless Networks

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-2013-2
Last update: September 27, 2013

Abstract

Establishing communications in a dynamic spectrum access (DSA) network requires the communicating parties to "rendezvous" before transmitting their data packets. Frequency hopping (FH) provides an effective method for rendezvousing without relying on a predetermined control channel. Previous FH-based rendezvous designs do not account for *fast* primary user (PU) dynamics, which results in extremely long rendezvous delay. Furthermore, these designs mainly target pairwise rendezvous, and do not intrinsically support *multicast rendezvous*. Intrinsic multicast rendezvous is required to consistently update the secret information in a multicast group. In this paper, we first design a *grid-quorum*-based FH algorithm, called NGQFH, for pairwise rendezvous. NGQFH can achieve efficient rendezvous under fast PU dynamics. It is also robust against insider attacks, such as node compromise. Using the *uniform k-arbiter* and *Chinese Remainder Theorem* quorum systems, we then propose three multicast rendezvous algorithms, which provide different tradeoffs between the time-to-rendezvous and robustness to node compromise. Our rendezvous algorithms are tailored for asynchronous and 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 efficient mechanisms for channel sensing, assignment, and quorum selection. Our simulations validate the fast rendezvous capability of the proposed algorithms, their PU detection accuracy, and their robustness to insider attacks.

Index Terms

Dynamic frequency hopping, dynamic spectrum access, quorum systems, rendezvous.

I. Introduction

Motivated by the need for more efficient utilization of the licensed spectrum, and supported by recent regulatory policies (e.g., [9]), 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 link between two or more communicating parties requires them to rendezvous (i.e., meet on a common channel at some point in time) and exchange control messages needed for connection establishment (e.g., negotiating the transmission parameters). In the absence of centralized control, the rendezvous problem is quite challenging in multi-channel DSA networks, 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 CR networks rely on a dedicated control channel (e.g., [7]). While presuming a common control channel (CCC) surely 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 [17]. Second, PU dynamics and spectrum heterogeneity make it extremely difficult to always maintain a single dedicated CCC [18].

Preliminary results in this paper were presented at the IEEE DySPAN Conference, Oct. 2012.

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 [10]. Quorums have been widely used in distributed systems to solve the mutual exclusion problem, the agreement problem, and the replica control problem. In this paper, we exploit quorum-based approaches in the design and analysis of FH protocols for control channel establishment in DSA systems. The consideration of quorum systems for FH-based rendezvous was pioneered by Bian et. al. in [4], [5]. Other FH approaches were proposed in [3], [8], [16], [24]. By requiring every pair of nodes to utilize all rendezvous channels for control, the FH designs in [24] improve the capacity of DSA networks during the rendezvous phase. Similar to [4], [5], the approaches in [16], [24] do not account for fast channel variations, where channel availability can vary during the rendezvous process. Furthermore, these approaches do not support multicast rendezvous, as explained in this section.

One key advantage of quorum-based FH designs is their robustness to synchronization errors [12]. Specifically, some types of quorum systems (e.g., *grid*, *uniform* k*-arbiter* [13], and *Chinese Remainder Theorem* [23]) enjoy certain properties that allow them to be used for asynchronous operation.

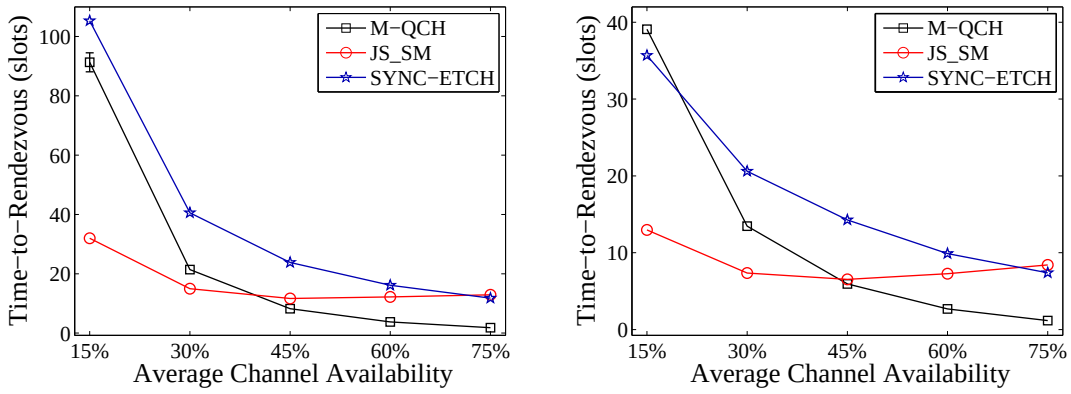
Multicast rendezvous–In multicast rendezvous, a subset of nodes forms a multicast group. Group members need to rendezvous simultaneously in the same time slot. This capability is not intrinsically supported in the quorum-based FH approaches in [4], [5]. The authors in [16] designed an algorithm for establishing multicast communications. Instead of designing different FH sequences that overlap at common slots, multicast is established after 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 [16], if a node is compromised, then the FH sequences of all nodes are exposed. In contrast, in our approach different nodes follow different FH sequences.

Group-based schemes have been proposed in [18] to facilitate multicast rendezvous. These schemes can be divided into two categories: (i) neighbor coordination schemes (e.g., [6]), where neighboring nodes broadcast their channel parameters to make a group-wide decision, and (ii) cluster-based schemes (e.g., [15]), 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 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 [18].

In [17], 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 that occur during the rendezvous process. Moreover, 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 common messages. For security purposes, this key may have to be updated periodically [22]. 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.

PU dynamics–In DSA networks, different channels experience different patterns of PU activity, resulting in different average availability (a.k.a. percentage occupancy). The mean duration of the duty cycle[2] for partially occupied channels can, in general, take values from tens of seconds to several hours [11], [19]. 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 2 hours, whereas this value can exceed 10 hours for DCS 1800 DL channel number 70 [19]. 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 channel variations, the average channel availability time

[2]Duty cycle is the time between two successive idle-to-busy PU transitions.



(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 for SYNC-ETCH is $2 \times 6 + 1 = 13$ and its period is $6 \times 13 = 78$. The frame length for JS_SM is $3 \times 7 = 21$.

and its fluctuation level (i.e., rate of transitions between idle and busy states) need to be considered when constructing the FH sequences.

Another limitation of previous FH designs is that channel availability is often modeled as a binary variable. This simplistic approach does not capture differences in the time-averaged channel availability, which in reality can be mapped to a continuous variable in the range $(0, 1]$ (where '1' means that the PU is active all the time). By incorporating the time-averaged channel availability, our modeling approach provides an effective tool for designing FH rendezvous protocols that are robust against fast PU dynamics.

To illustrate the effect of channel dynamics on the TTR, we simulate three previously proposed FH algorithms, M-QCH [5], JS_SM [16], and SYNC-ETCH [24], 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 is 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 [5] to minimize the TTR in a synchronous environment. However, as shown in Figure 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, 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 sever, as will be shown in Section VIII.

Our Contributions–The main contributions of this paper are 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 several rendezvous channels are used within several nested quorums. When integrated with optimal channel ordering and adaptive quorum selection schemes, 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 for multicast rendezvous: AMQFH, CMQFH, and nested-CMQFH, which provide different tradeoffs between the TTR and robustness to node compromise. These algorithms are tailored for asynchronous and heterogeneous DSA networks.

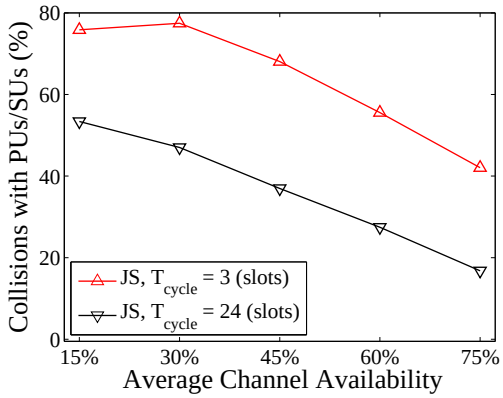


Fig. 2: Collision rate vs. average channel availability for JS_SM.

- We develop an algorithm for adapting the channel hopping 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–The remainder of this paper is organized as follows. In Section II, we present the system and channel models and the evaluation metrics. 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. Section V discusses the ability of our algorithms to operate in asynchronous and heterogeneous setups. We introduce our optimal channel ordering algorithm in Section VI, followed by our adaptive FH and quorum selection algorithm in Section VII. We evaluate the protocol in Section VIII. Finally, Section IX concludes the paper. Due to space limitation, the proofs of all the results have been omitted. They can be found online at [2].

II. Models and Metrics

A. System Model

We consider a single-hop 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 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. 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. The slot duration is assumed to be long enough to fit retransmissions. We consider a slot duration in the order of 10s of milliseconds.

Each SU j, $j = 1, \ldots, K$, 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 $1, 2, \ldots, K$ if there exists a *rendezvous slot* i such that $w_i^{(m)} = f_j, \forall m \in \{1, \ldots, K\}$. As in previous quorum-based FH designs, in our setup each FH sequence is divided into several *time frames*. Each frame corresponds to a block of time-frequency pairs.

B. Channel Activity Model

We assume that each channel $f_m, m \in \mathcal{L}$ 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 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., $\rho_i^{(m)} = \sum_{j \neq i} \alpha_{ij}^{(m)}$.

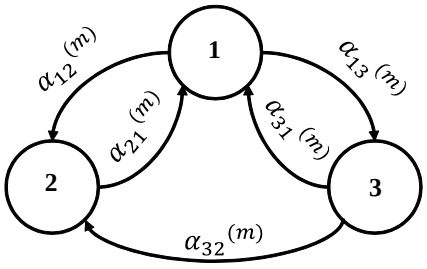


Fig. 3: State transition diagram for channel f_m.

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 must leave that channel immediately, so $\alpha_{32}^{(m)} \neq 0$ in general. Let $\boldsymbol{A}^{(m)}$ be the infinitesimal generator matrix for channel m. The (i, j) entry of $\boldsymbol{A}^{(m)}$ equals to $\alpha_{ij}^{(m)}$ if $i \neq j$ and equals to $-\rho_i^{(m)}$ if $i = j$. 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 rate $\lambda_s^{(m)}$ and depart with rate $\mu_s^{(m)}$, both according to Poisson processes. Therefore, $\boldsymbol{A}^{(m)}$ is given by:

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

Let $\boldsymbol{P}_t^{(m)}$ 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 [14]:

$$\boldsymbol{P}_t^{(m)} = e^{t\boldsymbol{A}^{(m)}}, \quad t \geq 0. \tag{1}$$

Let $\boldsymbol{\pi}^{(m)} = \left(\pi_1^{(m)}, \pi_2^{(m)}, \pi_3^{(m)}\right)$ be the steady-state distribution for channel m. Then, $\boldsymbol{\pi}^{(m)}$ can be written as:

$$\pi_1^{(m)} = \frac{\mu_p^{(m)}\left(\lambda_p^{(m)}+\mu_s^{(m)}\right)}{\left(\lambda_p^{(m)}+\mu_p^{(m)}\right)\left(\lambda_s^{(m)}+\lambda_p^{(m)}+\mu_s^{(m)}\right)}, \; \pi_2^{(m)} = \frac{\lambda_p^{(m)}}{\left(\lambda_p^{(m)}+\mu_p^{(m)}\right)}, \; \pi_3^{(m)} = \frac{\mu_p^{(m)}\lambda_s^{(m)}}{\left(\lambda_p^{(m)}+\mu_p^{(m)}\right)\left(\lambda_s^{(m)}+\lambda_p^{(m)}+\mu_s^{(m)}\right)}.$$

C. Metrics

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

1) Expected Time-to-Rendezvous (TTR): The TTR is defined as the time until two (or more for multicast) nodes to rendezvous. The expectation is considered because of two reasons. First, the existence of a randomly assigned part in our FH sequences, as discussed later. Second, due to the randomness in PU dynamics.

2) Expected Hamming Distance (HD): The expected HD for two FH sequences $\boldsymbol{x} = (x_1 \dots x_n)$ and $\boldsymbol{y} = (y_1 \dots y_n)$ is defined as $E[\left(\sum_{i=1}^{n} \mathbf{1}_{\{x_i \neq y_i\}}\right)/n]$, where $\mathbf{1}_{\{\cdot\}}$ is the indicator function and n is the frame length. 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).

III. Nested Quorum-Based FH Algorithm for Pairwise Rendezvous

A. Preliminaries

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

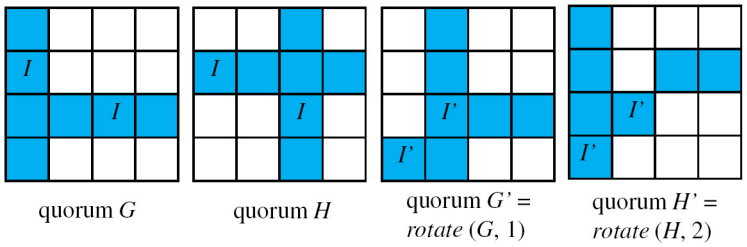


Fig. 4: Rotation closure property of grid quorum systems.

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$(G, i) = \{(x + i) \mod n, x \in G\}$ 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$(G_j, i_j) \neq \emptyset$.

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 [12].

Definition 4. A grid quorum system arranges the elements of Z_n as a $\sqrt{n} \times \sqrt{n}$ array, where n must be the square of a positive integer. 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 $G' = rotate(G, i)$ 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, $G' = rotate(G, 1)$ and $H' = rotate(H, 2)$ intersect at the two elements labeled as I'.

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

Before describing NGQFH, we first discuss a non-nested quorum-based FH design (herein referred to as GQFH). This design is the basis for the FH schemes in [4], [5]. In GQFH, a grid quorum system is defined on Z_n. The slotted time is divided into frames, each containing n slots. The slots of each frame form the $\sqrt{n} \times \sqrt{n}$ grid, from which the quorums are derived. For each FH sequence, a grid quorum (a column and a row) is randomly selected. One common rendezvous channel is assigned to all quorums.

To make the FH sequences more resilient to PU dynamics and node compromise, we propose a nested design, whereby every frame of every FH sequence uses $\sqrt{n} - 1$ rendezvous channels. We call the number of rendezvous channels the *nesting degree* of the FH sequence. As in GQFH, in NGQFH a $\sqrt{n} \times \sqrt{n}$ quorum is selected for each FH sequence, and a first rendezvous channel is assigned to the slots that correspond to the selected quorum. We call this $\sqrt{n} \times \sqrt{n}$ quorum *the outer-most quorum*. The column and row that correspond to the outer-most quorum are then deleted from the grid, and another quorum is selected from the resulted $(\sqrt{n} - 1) \times (\sqrt{n} - 1)$ grid. A second rendezvous channel is assigned to this smaller quorum. This quorum elimination procedure continues for $\sqrt{n} - 1$ iterations.

We explain the operation of the NGQFH algorithm via the following example. Let $n = 9$ (hence, each frame contains $\sqrt{n} - 1 = 2$ rendezvous frequencies). Consider the jth frame of one FH sequence $\boldsymbol{w}$.

1. Construct a grid quorum system Q under Z_9. Q has 9 different quorums, each containing $2\sqrt{9} - 1 = 5$ elements that comprise one row and one column of the 3×3 grid.
2. Construct the FH sequence $\boldsymbol{w}$ using the following procedure:

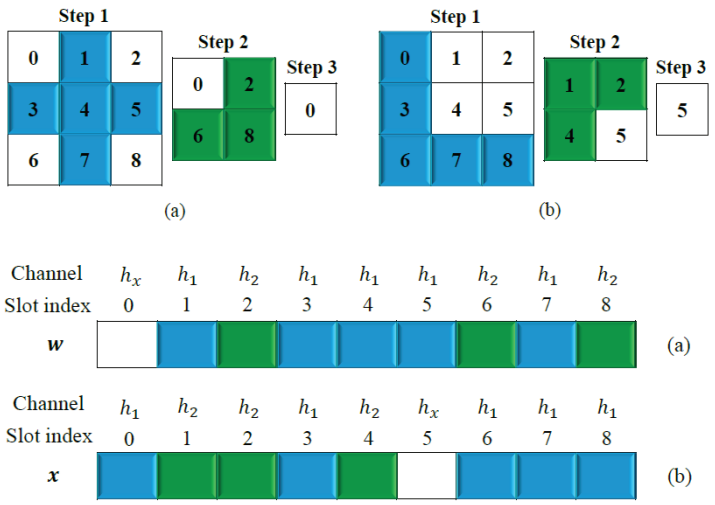


Fig. 5: Generation of the nested quorums ($n = 9$).

- Select *the outer-most quorum* $G_1^{(j)}$ of frame j from the quorum system Q (e.g., $G_1^{(j)} = \{1, 3, 4, 5, 7\}$, where each entry represents the index of a time slot in a 9-slot frame). The criteria for selecting $G_1^{(j)}$ will be explained in Section VII.
- Assign the first rendezvous channel $h_1^{(j)} \in \mathcal{L}$ to the slots that correspond to $G_1^{(j)}$. The selection of rendezvous channels $h_i^{(j)}, i \in \{1, \ldots, \sqrt{n} - 1\}$ and $j = 1, 2, \ldots$, will be discussed in Section VI.
- Delete quorum $G_1^{(j)}$ from the original 3×3 grid and select the *next outer-most quorum* $G_2^{(j)}$ from the resulting 2×2 grid (e.g., $G_2^{(j)} = \{2, 6, 8\}$). Then, assign another rendezvous frequency $h_2^{(j)}$ to the slots that correspond to $G_2^{(j)}$.
- Assign a random frequency $h_x^{(j)} \in \mathcal{L} \setminus \{h_1^{(j)}, h_2^{(j)}\}$ to each of the remaining unassigned slots in the frame, according to the procedure in Section VI.
- Repeat the above procedure for all frames j in the sequence $\boldsymbol{w}$.

3. Repeat Step 2 for other FH sequences.

Throughout this paper, $h_i^{(j)}$, $i \in \{1, \ldots, \sqrt{n} - 1\}$ and $j = 1, 2, \ldots$, denotes the ith quorum channel that is assigned to the $(\sqrt{n} - i + 1) \times (\sqrt{n} - i + 1)$ quorum $G_i^{(j)}$ in the jth frame. $h_1^{(j)}$ and $G_1^{(j)}$ are called the outer-most channel and the outer-most quorum of frame j, respectively. To simplify the notation, when the nesting degree is 1, we use $h^{(j)}$ instead of $h_i^{(j)}$. A pseudo-code of the NGQFH algorithm for constructing one frame of one FH sequence is shown in Algorithm 1. Figure 5 shows the resulting frames of two sequences $\boldsymbol{w}$ and $\boldsymbol{x}$, constructed according to NGQFH. Since only one frame is considered in Figure 5, the superscript in $h_i^{(j)}$ is dropped.

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 (defined as the fraction of rendezvous slots in a frame) is significantly higher than the overlap ratio for a non-nested design, herein referred to as grid-quorum FH (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 several rendezvous channels per frame, which increases the likelihood of a successful rendezvous.

The merits of a nested grid quorum can be formalized by deriving the expected overlap ratio for GQFH and NGQFH, denoted by $\mathcal{O}_{GQFH}$ and $\mathcal{O}_{NGQFH}$, respectively. $\mathcal{O}_{GQFH}$ is composed of the sum of two parts; the expected overlap ratio between the quorum-based assigned parts of the FH sequences, denoted

Algorithm 1 NGQFH Algorithm

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

by $\mathcal{O}_{GQFH}^{Q}$, and the expected overlap ratio between the randomly assigned parts, denoted by $\mathcal{O}_{GQFH}^{R}$. Similarly, $\mathcal{O}_{NGQFH}$ is composed of $\mathcal{O}_{NGQFH}^{Q}$ and $\mathcal{O}_{NGQFH}^{R}$. For a given n, $\mathcal{O}_{GQFH}^{Q}$ and $\mathcal{O}_{NGQFH}^{Q}$ can be determined numerically, by computing the average overlap ratio over all possible quorums selections ($(\sqrt{n}\ !)^4$ possibilities in NGQFH and $(\sqrt{n})^4$ in GQFH). $\mathcal{O}_{GQFH}^{R}$ and $\mathcal{O}_{NGQFH}^{R}$ can be expressed as functions of L and n:

$$\mathcal{O}_{GQFH}^{R} = \frac{(\sqrt{n}-1)^2}{L}\left\{2 - \frac{(\sqrt{n}-1)^2}{n}\right\} \tag{2}$$

$$\mathcal{O}_{NGQFH}^{R} = \frac{1}{L}\left\{2 - \frac{1}{n^2}\right\}. \tag{3}$$

These expressions are obtained as follows. Consider two FH sequences $\boldsymbol{x}$ and $\boldsymbol{y}$, each with frame length n. Let H_1 and H_2 be the grid quorums used in constructing $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, following the GQFH algorithm (i.e., H_1 is used in every frame of $\boldsymbol{x}$ and H_2 is used in every frame of $\boldsymbol{y}$). 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 $\frac{(\sqrt{n}-1)^2}{n}$.

The number of randomly assigned slots in $\boldsymbol{x}$ and $\boldsymbol{y}$ is $n - (2\sqrt{n} - 1) = (\sqrt{n} - 1)^2$. The numbers of randomly assigned slots that are common to $\boldsymbol{x}$ and $\boldsymbol{y}$ 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 $\mathcal{O}_{GQFH}^{R}$, we initially assume that the randomly assigned portions of $\boldsymbol{x}$ and $\boldsymbol{y}$ are nonoverlapping. Then, we subtract the overlapped randomly assigned slots which are counted twice. After some straightforward manipulations, the expression for $\mathcal{O}_{GQFH}^{R}$ in (2) can be obtained. The randomly assigned portions of $\boldsymbol{x}$ and $\boldsymbol{y}$ in the NGQFH algorithm consist of one slot only, which can be common to $\boldsymbol{x}$ and $\boldsymbol{y}$ with probability $\frac{1}{n^2}$. The $\mathcal{O}_{NGQFH}^{R}$ expression in (3) can be obtained by following the same approach

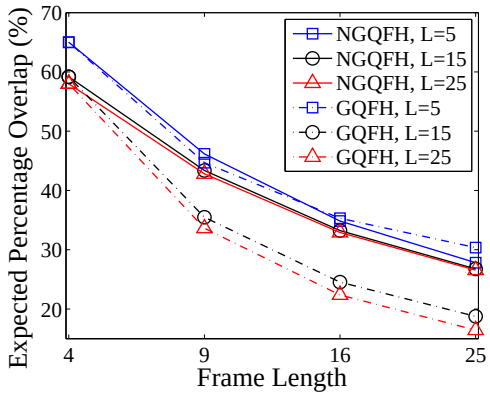


Fig. 6: Expected overlap ratio of GQFH and NGQFH.

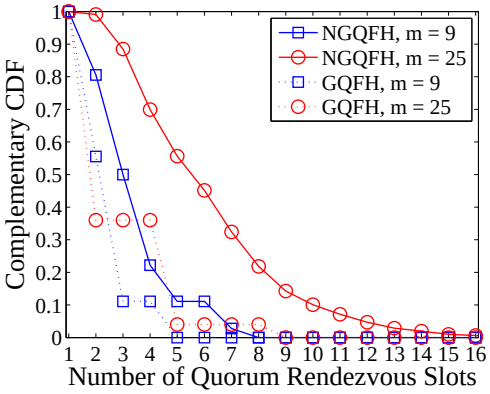


Fig. 7: Probabilistic guarantees of GQFH and NGQFH.

used in deriving $\mathcal{O}^R_{GQFH}$.

Figure 6 depicts $\mathcal{O}_{GQFH}$ and $\mathcal{O}_{NGQFH}$ vs. n for different values of L. As expected, $\mathcal{O}_{NGQFH}$ is larger than $\mathcal{O}_{GQFH}$, and both decrease with n.

The second attractive feature of NGQFH is that it is more robust against intelligent adversaries, which use quorum-based FH sequences for jamming, than GQFH. Assume, for example, that a set of adversaries had launched an insider attack, in which a subset of nodes were compromised and the quorum structure of their sequences was revealed. Then, these adversaries will try to infer the quorums used by the other nodes. NGQFH is more robust against such attacks because its sequences are composed of $\sqrt{n}-1$ nested quorums that are generally different for different frames of the same FH sequence, and also different for different FH sequences. Hence, if a node is compromised and the quorum structure of its sequence is exposed, it is much more difficult to infer the sequences of other nodes, compared with the case when GQFH is used. The robustness of NGQFH to the quorum-based jamming attack can be characterized by the number of different sequences that can be generated using a given frame length n, denoted by $\mathcal{K}_n = \prod_{j=0}^{\sqrt{n}-2}(\sqrt{n}-j)^2$. Note that $\mathcal{K}_n$ increases with n. A higher $\mathcal{K}_n$ implies more robustness to the above quorum-based jamming attack.

IV. Quorum-Based FH Algorithms for Multicast Rendezvous

In this section, we present two algorithms for constructing a set of FH sequences for multicast rendezvous. These algorithms have two main attractive features. First, they allow nodes to construct their sequences independently by knowing only 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 is based on the uniform k-arbiter quorum system, which exhibits the rotation $(k+1)$-closure property.

Definition 5. A quorum system Q under Z_n is called k-arbiter if every set of $k+1$ quorums $\mathcal{V}_{k+1} = \{G_1, G_2, \ldots, G_{k+1}\} \subset Q$ satisfies the following (k + 1)-intersection property [21]:

$$\bigcap_{i=1}^{k+1} G_i \neq \emptyset. \tag{4}$$

One specific type of k-arbiter quorum systems that is of interest to us is the so-called uniform k-arbiter quorum system [13]. Such a system satisfies:

$$Q = \left\{ G \subseteq Z_n \quad : \quad |G| = \left\lfloor \frac{kn}{k+1} \right\rfloor + 1 \right\}. \tag{5}$$

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 has been shown [13] 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.,

$$\varphi \stackrel{\text{def}}{=} \binom{n}{\lfloor \frac{kn}{k+1} \rfloor + 1} \geq k+1. \tag{6}$$

To satisfy (6), one can easily show that n needs to be strictly greater than $k+1$.

We now explain AMQFH through an example. Consider a multicast group of 3 nodes. 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 FH sequence is constructed using one quorum. Thus, the frame length will be n. We set n to the smallest value that satisfies (6), i.e., $n = k+2 = 4$. The following steps are used by each node to obtain the various FH sequences:

1. Construct a universal set Z_n (in this example, $Z_4 = \{0,1,2,3\}$).
2. Construct a uniform 2-arbiter system Q under Z_n.
3. Construct an FH sequence $\boldsymbol{w}$ as follows:
   - Select a quorum from Q and assign it to $G^{(1)}$ (e.g., $G^{(1)} = \{0,1,2\}$).
   - Assign a frequency $h^{(1)} \in \mathcal{L}$ to the FH slots in the given frame that correspond to $G^{(1)}$, and assign a random frequency h_x to the remaining slots. (As will be explained later, in homogenous opportunistic spectrum environments, all nodes often assign the same $h^{(1)}$ frequency.)
   - Repeat the above procedure for the other frames in $\boldsymbol{w}$.

The channel and quorum selection procedures will be explained in Sections VI and VII, respectively. Figure 8 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 3-out-of-4 sequences from Figure 8.

B. CRT Multicast FH Algorithm (CMQFH)

Our second algorithm (CMQFH) uses the Chinese Remainder Theorem (CRT) quorum system, which also exhibits the rotation k-closure property. CMQFH is more resilient to insider attacks than AMQFH, but it is slower than AMQFH. The CRT is formally stated in [23]. Using CRT, we can construct quorum systems that satisfy the rotation k-closure property, as in Theorem 1 [13].

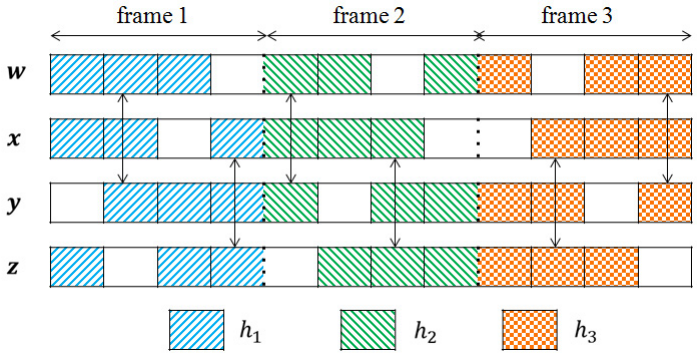


Fig. 8: AMQFH FH construction algorithm.

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 of the CRT quorum system, consider quorums $G_1 = \{0,\ 2, 4,\ \ldots,\ 28\}$, $G_2 = \{0,\ 3,\ 6,\ldots,27\}$, and $G_3 = \{0,\ 5,\ 10,\ \ldots,\ 25\}$ constructed using prime numbers 2, 3, and 5, respectively. It is easy to verify that the quorum system $\{G_1,\ G_2,\ G_3\}$ satisfies the rotation 3-closure property.

The CMQFH algorithm for generating k asynchronous FH sequences is similar to the AMQFH algorithm, but with two main differences. First, the frame length, denoted by y, is equal to $y = \prod_{i=1}^{k} p_i$. Second, CMQFH uses the CRT quorum system instead of the uniform $(k-1)$-arbiter quorum system.

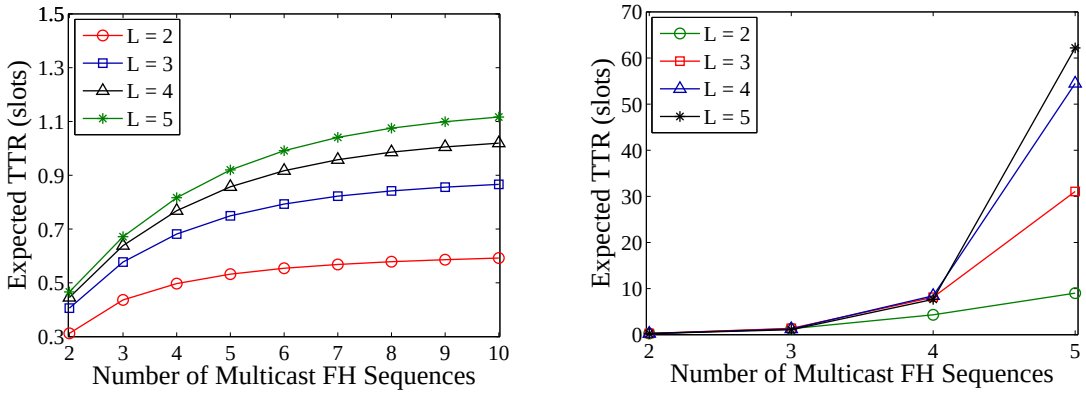
C. AMQFH vs. CMQFH (Speed vs. Security)

This section compares AMQFH and CMQFH. Both algorithms are implemented in a distributed way as follows. First, the source of a multicast transmission uses a series of pairwise rendezvous to communicate the size of the multicast group to the target multicast receivers (this step may have already been done as part of establishing a multicast session). 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 FH sequences.

In Appendix A, we provide analytical results for the expected TTR and expected HD of AMQFH and CMQFH algorithms. These results are used to obtain the plots in Figures 9 and 10. 'AMQFH, best' and 'CMQFH, best' in Figure 10 refer to the case when different nodes select different quorums, and their randomly assigned parts are nonoverlapping. 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. Figure 10 depicts the expected HD vs. the multicast group size for AMQFH and CMQFH. 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.

As shown in Figures 9 and 10, 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 and jamming attacks, we propose a modified version of CMQFH that borrows the nesting concept of NGQFH, proposed in Section III-B. We call this modified CMQFH algorithm *the nested-CMQFH*. As will be shown in Section VIII, 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 each FH sequence contains a number of quorums, called the *nesting degree*. In our design, the FH sequence that uses prime number p_i will have a nesting



(a) AMQFH (b) CMQFH

Fig. 9: Expected TTR vs. multicast group size.

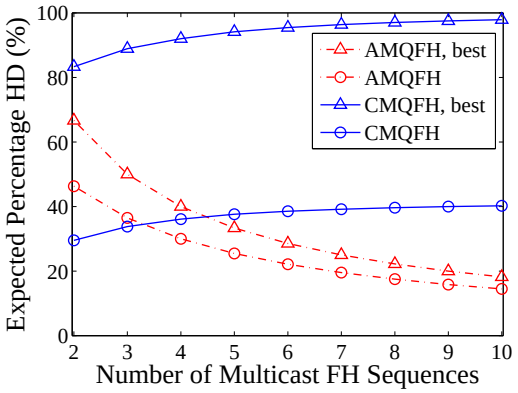


Fig. 10: Expected HD vs. multicast group size ($L = 6$).

degree of $p_i - 1$. In contrast to NGQFH, the nesting degree in nested-CMQFH can be different for different FH sequences, depending on whether or not the prime numbers used in constructing the FH sequence are the same. The nesting degree provides a tradeoff between TTR and HD. A large nesting degree results in a small TTR, but also a small HD. The selection criterion of the prime number for nested-CMQFH will be explained later in Section VII.

Figure 11 illustrates the idea behind nested-CMQFH for a multicast group of 3 nodes. The prime numbers used in constructing FH sequences $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are 5, 3, and 2, respectively, and the corresponding nesting degrees are 4, 2, and 1, respectively. Hence, sequence $\boldsymbol{x}$ will have four nested quorums, each of 5 slots, and each quorum is assigned a different channel (the same treatment is done for $\boldsymbol{y}$ and $\boldsymbol{z}$).

Remark 1. As will be shown in Section VIII, in a homogeneous spectrum environment AMQFH is faster than nested-CMQFH but it is less robust than nested-CMQFH. However, when spectrum opportunities are highly heterogeneous (i.e., set of available channels varies in space), nested-CMQFH is faster and also more robust to node compromise than AMQFH.

V. Asynchronous and Heterogeneous Rendezvous

A. Asynchronous Rendezvous

FH sequences constructed according to NGQFH, AMQFH, and nested-CMQFH can support asynchronous rendezvous if each FH sequence continues to use the same (outer-most) quorum and the same (outer-most) frequency in all frames, i.e., for all FH sequences, $h_1^{(j)}$ and $G_1^{(j)}$ are the same for all j. This result is a direct consequence of the intersection and rotation closure properties of the grid, uniform

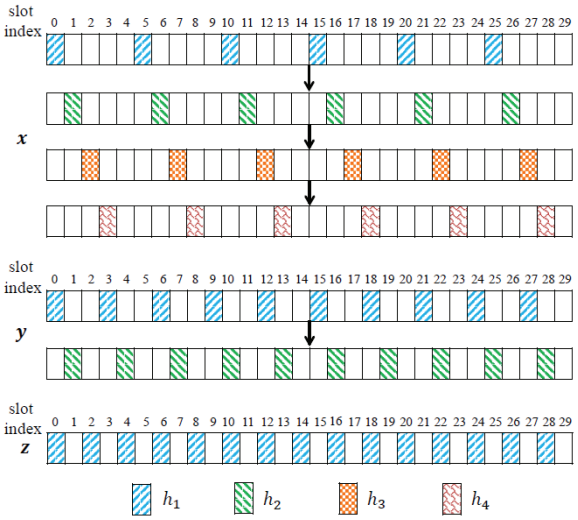


Fig. 11: Nested-CMQFH FH construction algorithm.

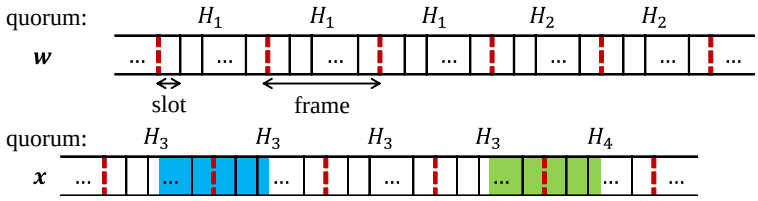


Fig. 12: Example of asynchronous rendezvous with time-varying (per frame) quorums.

k-arbiter, and CRT quorum system, and the fact that each frame in an FH sequence is constructed using one quorum (the outer-most).

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 (they may still rendezvous if the change in the outer-most quorum does not occur very frequently). We try to avoid such changes by pushing nodes to use more available and less fluctuating channels, as will be explained in Section VI. The scenario of asynchronous rendezvous under time-varying quorums is illustrated in Figure 12, where the outer-most quorum of sequence $\boldsymbol{w}$ changes from quorum H_1 to H_2, and the outer-most quorum of sequence $\boldsymbol{x}$ changes from quorum H_3 to H_4. The left shaded part of sequence $\boldsymbol{x}$ in Figure 12 represents a cyclic rotation of H_3, and hence, by the rotation k-closure property, it is guaranteed that this part overlaps with quorum H_1 of $\boldsymbol{w}$. The right shaded part of sequence $\boldsymbol{x}$ does not generally overlap with H_1 in $\boldsymbol{w}$ because it is composed of two different quorums.

To increase the likelihood of rendezvous, we keep using the outer-most rendezvous channel as long as it is available for use in at least a minimum number of slots in the current outer-most quorum. This way, we avoid unnecessary quorum/channel changes, and continue assigning the same channel to the same outer-most quorum in the next frame. A suitable channel selection criteria (Section VI) is another parameter that further reduces these changes. 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 VII.

B. Heterogeneous Rendezvous

In Figures 5, 8, and 11, FH sequences were constructed using the same rendezvous channels, 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 these algorithms whereby each node assigns channels to quorum slots mainly based on the forecasted 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. By adopting a nesting design, we increase the chances of having common quorum channels. Hence, when neighboring nodes construct their FH sequences independently, they will likely end up having a few quorum channels in common.

We later evaluate the unicast (NGQFH) and multicast (AMQFH and nested-CMQFH) rendezvous algorithms in a heterogeneous environment with different heterogeneity levels. We define the heterogeneity level κ for a multicast group (unicast is a special case) as the fraction of channels whose parameters differ between any two links 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 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.

VI. Optimal Channel Ordering

In the previous sections, we presented unicast and multicast rendezvous algorithms without explaining how the rendezvous channels are selected in each frame. In NGQFH, the ith quorum channel in a frame ($h_i^{(j)}$ for frame j) is assigned to $2(\sqrt{n}-i+1)-1$ slots. Thus, the outer-most channel is assigned to more slots than the next outer-most channel, which in turn is assigned to more slots than the next outer-most channel, and so on. Accordingly, we select the outer-most channel to be the "best" available channel in $\mathcal{L}$, the next outer-most as the next best available channel, and so on. With this approach, better channels are assigned to more quorum slots. In here, the best available channel is selected according to several factors, as will be explained in this section. In contrast to NGQFH, all quorum channels in a nested-CMQFH frame are assigned to the same number of slots, however we still need to identify the best p_i-1 available channels in $\mathcal{L}$ (for the sequence that uses prime number p_i). Similarly, the best available channel needs to be found in AMQFH (recall that the nesting degree of AMQFH sequences is 1). Channel sorting is also exploited during assigning channels to the randomly assigned slots, as will be explained later. Therefore, each node is required to *independently* sort the available channels (no message exchange is assumed between the nodes).

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 expected to be available). However, the time complexity of this exhaustive search is given by:

$$\begin{aligned}
&\mathcal{O}\left(\binom{\bar{L}}{\sqrt{n}-1}\left(n^{\frac{1}{2}}!\right)^2\right), && \text{NGQFH}\\
&\mathcal{O}\left(\binom{n}{\lfloor\frac{kn}{k+1}\rfloor+1}\left(\lfloor\tfrac{kn}{k+1}\rfloor+1\right)\bar{L}\right), && \text{AMQFH}\\
&\mathcal{O}\left(\sum_{i=1}^{k}\binom{\bar{L}}{\lceil\frac{p_i}{2}\rceil}\frac{p_i!\,y^{\lceil\frac{p_i}{2}\rceil}}{\left(p_i-\lceil\frac{p_i}{2}\rceil\right)!\,p_i}\right), && \text{nested-CMQFH}
\end{aligned}$$

where $\bar{L}$ is the number of available channels, n is the frame length of NGQFH and 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 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 channel ordering mechanism, and in Section VII 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. 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 other SUs below certain thresholds. The weights will be used for two 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 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 m in state i and the first time that channel 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 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 II-B. $\mathcal{T}_i^{(m)}, i \in \{1,2,3\}$ can be expressed as follows:

$$\mathcal{T}_1^{(m)} = \tfrac{1}{\lambda_p^{(m)}+\lambda_s^{(m)}},\ \mathcal{T}_2^{(m)} = \tfrac{1}{\mu_p^{(m)}},\ \text{and}\ \mathcal{T}_3^{(m)} = \tfrac{1}{\lambda_p^{(m)}+\mu_s^{(m)}}.$$

To sort channels based on the above criteria, we propose the following optimization problem for NGQFH. This ordering mechanism starts over when the estimate of at least one of the channel parameters changes.

Problem 1.

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

$$\text{Subject to.} \quad \left[1 - \prod_{u=0}^{\sqrt{n}-1} (1 - p^{(m)}_{(n+i+\sqrt{n}u)T}(1,s)) \prod_{\substack{v=0 \\ v\neq i}}^{\sqrt{n}-1} (1 - p^{(m)}_{(n+j\sqrt{n}+v)T}(1,s)) \right] q_m < \lambda^{(m)}_{s,Col}(n), \tag{7}$$

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

$$\sum_{m=1}^{L} q_m = 1 \tag{8}$$

$$0 \leq q_m \leq 1, \forall m \in \{1,\ldots,L\} \tag{9}$$

where $\lambda^{(m)}_{2,Col}(n)$ and $\lambda^{(m)}_{3,Col}(n)$ are prespecified thresholds on the probabilities of collisions with PUs and SUs, respectively. The objective function in Problem 1 represents a convex combination of the average channel availabilities. Constraint (7) restricts the collision probabilities with PUs and SUs, while considering the specific structure of the grid quorum system. The term in the square brackets represents the probability that at least one quorum slot is in collision. Each selection of i and j in (7) corresponds to one quorum. 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 AMQFH and nested-CMQFH, after

replacing (7) by (10) and (11) for AMQFH and nested-CMQFH, respectively.

$$\left[1-\prod_{\substack{u=0\\u\neq i}}^{n-1}(1-p^{(m)}_{(n+u)T}(1,s))\right]q_m<\lambda^{(m)}_{s,Col}(n),\quad \forall s\in\{2,3\},\forall m\in\{1,\ldots,L\},\forall i\in\{1,\ldots,\varphi\} \tag{10}$$

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

where φ is as in (6), $\rho \stackrel{\text{def}}{=} \max_{1\leq i\leq k} p_i$, and $\psi_i \stackrel{\text{def}}{=} \left\lceil \frac{\rho}{p_i} \right\rceil$, $i = 1, \ldots, k$. In contrast to grid and uniform k-arbiter, different quorums in a CRT quorum system have different sizes. Because of this collision probabilities in (11) are computed in a slightly different way than (7) and (10). 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 (7), (10), and (11) are used to restrict the fluctuation level of the selected channels, such that highly fluctuating channels are excluded from the ordered list. The fluctuation level of a channel affects its prediction accuracy. Less fluctuating channels are more predictable, and consequently result in smaller TTR, as will be shown in Section VIII.

Next, we provide an example that explains our sorting criteria. Consider the six channels in Table I, and assume the same collision probability thresholds for PUs and SUs (denoted by λ). There are three pairs of channels, each have the same $\pi_1^{(m)}$ but different values of $\mathcal{T}_1^{(m)}$. Some of the channels with smaller $\pi_1^{(m)}$ have larger values of $\mathcal{T}_1^{(m)}$. Table I also includes the values of $\pi_2^{(m)}/\pi_3^{(m)}$, which show how the busy time is distributed between PUs and SUs. We present in Table II the sorted list of channels for NGQFH with frame length of 16 slots (Note that the other algorithms exhibit a similar behavior). For this example, the minimum value of λ that keeps the problem feasible is 0.15. Because SUs and PUs are treated the same in this example, the collision constraint enforces channels that spend longer time in state 2 to receive lower weights. Therefore, f_2 is preferred over f_1, and f_5 is preferred over f_3 and f_4. By increasing λ and thus the feasibility region, a higher priority is given to the objective function than the constraints. In this case, channels with higher availability receive better ranks.

In contrast to the CRT quorum system, grid and uniform k-arbiter quorum systems have the unique feature that each quorum consists of several consecutive elements of the universal set, therefore a big portion of the quorum slots in an NGQFH or AMQFH frame are consecutive. More specifically, a $\frac{\sqrt{n}}{2\sqrt{n}-1}\times 100\%$ of quorum slots are consecutive in NGQFH, and at least a $\frac{n-\phi}{\phi+1}\times 100\%$ of quorum slots are consecutive in AMQFH where $\phi \stackrel{\text{def}}{=} n-\left\{\left\lceil\frac{kn}{k+1}\right\rceil+1\right\}$ (note that $(n-\phi)\gg\phi$). This feature needs to be considered in the sorting mechanisms of NGQFH and 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 NGQFH and 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 NGQFH and AMQFH is formulated as a two-stage sequential optimization problem. Problem 1 above is the first stage (after replacing (7) by (10) for AMQFH) and Problem 2 is the second stage. Let $\boldsymbol{q}_I^*$ be an optimal solution to Problem 1, and let $\mathcal{U}_I^* = \mathcal{U}(\boldsymbol{q}_I^*)$.

TABLE I: Channel parameters.

Ch. (m)	$\pi_1^{(m)}$	$\mathcal{T}_1^{(m)}$	$\frac{\pi_2^{(m)}}{\pi_3^{(m)}}$	Ch. (m)	$\pi_1^{(m)}$	$\mathcal{T}_1^{(m)}$	$\frac{\pi_2^{(m)}}{\pi_3^{(m)}}$
f_1	0.777	35.7143	4.68	f_4	0.5767	27.7778	188.36
f_2	0.777	22.0751	2.96	f_5	0.3716	25	1.07
f_3	0.5767	37.037	188.36	f_6	0.3716	18.5185	1.07

TABLE II: Channel order for NGQFH ($\epsilon = 0$, best is leftmost).

λ	$\epsilon = 0.0$					
0.15	f_2	f_1	f_5	f_3	f_4	f_6
0.18	f_2	f_1	f_3	f_4	f_5	f_6

Problem 2.

$$\underset{\boldsymbol{q}=(q_1,q_2,\ldots,q_L)}{\text{maximize}} \left\{ \mathcal{F}(\boldsymbol{q}) = \sum_{m=1}^{L} \mathcal{F}_m q_m \stackrel{\text{def}}{=} \sum_{m=1}^{L} \mathcal{T}_1^{(m)} q_m \right\}$$
$$\text{Subject to. } \mathcal{U}_I^*(1-\epsilon) < \mathcal{U}(\boldsymbol{q}). \tag{12}$$

Problem 2 aims at maximizing a convex combination of the average sojourn times of state 1 subject to constraints (7) – (9) for NGQFH (and (8) – (10) for AMQFH), 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 $\mathcal{U}(\boldsymbol{q})$. 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.

Reconsider the example in Table I. With $\lambda = 0.15$, the feasibility region is very small and even with some perturbation from $\mathcal{U}_I^*$ (i.e., $\epsilon > 0$), the second objective function does not play any role in ordering. Next, consider expanding the feasibility region by increasing λ to 0.18. With $\epsilon = 0$, f_2 is preferred over f_1 because it has a smaller probability of collision with PUs. However, with $\epsilon = 0.03$, f_1 is preferred over f_2 because this improves $\mathcal{F}^*$ without affecting $\mathcal{U}_I^*$. As mentioned earlier, Problem 2 does not differentiate between states 2 and 3. With $\epsilon = 0.07$, we give even more importance to the channels which are available in consecutive slots, and f_2 falls after f_3 and f_4. Further violation (e.g., $\epsilon = 0.1$) will benefit the channels with high sojourn time (captured by Problem 2) and balanced distribution of states (captured by the constraints). Thus, f_5 is promoted to the second position in the rank. If we relax the threshold further, we can get a result similar to the case when $\epsilon = 0.07$. We conclude that the longer the run-length of consecutive quorum slots, the higher the value of ϵ is appropriate.

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. Moreover, the nesting design of NGQFH and nested-CMQFH improves their robustness against an intelligent jammer, who may order the channels in a similar way and keep jamming the best channel continuously.

VII. Adaptive FH and Quorum Selection

We now explain how quorums are selected in NGQFH, AMQFH, and nested-CMQFH. 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(s) selection, and hence online adaptation of the sequences, it is an adaptive FH algorithm. We now explain this algorithm for NGQFH.

Consider the jth frame of a given FH sequence. The node that follows this sequence starts sensing channels according to the order obtained in Section VI (the sensing starts $\lceil L\tau_s \rceil$ slots before the beginning

TABLE III: Channel order for NGQFH ($\epsilon > 0$).

λ		ϵ	$=$	0.03		
0.15	f_2	f_1	f_5	f_3	f_4	f_6
0.18	f_1	f_2	f_3	f_4	f_5	
λ		ϵ	$=$	0.07		
0.15	f_2	f_1	f_5	f_3	f_4	f_6
0.18	f_1	f_3	f_4	f_2	f_5	
λ		ϵ	$=$	0.1		
0.15	f_2	f_1	f_5	f_3	f_4	f_6
0.18	f_1	f_5	f_3	f_4	f_2	f_6

of the frame). Let $\{d_1, \ldots, d_{\sqrt{n}-1}\}$ be the best $\sqrt{n}-1$ available channels, ordered decreasingly according to their quality. According to NGQFH, we assign d_1 to a $\sqrt{n} \times \sqrt{n}$ quorum, d_2 to a $(\sqrt{n}-1) \times (\sqrt{n}-1)$ quorum, and so on. In general, the kth outer-most $(\sqrt{n}-k+1) \times (\sqrt{n}-k+1)$ quorum $G_k^{(j)}, k \in \{1, \ldots, \sqrt{n}-1\}$ 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 γ. 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:

$$\underset{\mathcal{G}_k}{\text{maximize}} \Bigg\{ \mathcal{A}(k,n) = \sum_{i=0}^{n-1} \mathbf{1}_{\{p^{(k')}_{(n-k\frac{\tau_s}{T}+i)T}(1,1) \geq \gamma\}} + \frac{1}{2(\sqrt{n}-k+1)-1} \sum_{i=0}^{n-1} p^{(k')}_{(n-k\frac{\tau_s}{T}+i)T}(1,1) \Bigg\} \tag{13}$$

where $\mathcal{G}_k$ is the set of all $(\sqrt{n}-k+1) \times (\sqrt{n}-k+1)$ quorums and $\mathbf{1}_{\{.\}}$ is the indicator function. Given that $d_k = f_{k'}$ where $k' \in \mathcal{L}$, $p^{(k')}_{(n-k\frac{\tau_s}{T}+i)T}(1,1)$ is the probability that d_k will remain available in the ith slot of the next frame, given that it is currently available. $p^{(k')}_{(n-k\frac{\tau_s}{T}+i)T}(1,1) = 0, \forall i \notin H_k$ when $\mathcal{A}(k,n)$ is evaluated at H_k. The computation of $p_t^{(m)}(i,j)$ was explained in Section II-B. The second term in (13) is < 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 $H_k^{(1)}$.

The quorum selection procedure of AMQFH is similar to that of NGQFH, as in (13). In contrast to NGQFH, nested-CMQFH has the unique feature that all nested quorums in a frame have the same size. Because of this, channel-quorum assignment in nested-CMQFH is performed jointly for all nested quorums of a frame, unlike in NGQFH where each quorum is selected independently. Formally, the problem of selecting the nested quorums in the jth frame of the FH sequence that uses prime number p_i is formulated as follows:

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

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 a larger fraction of quorum slots. The fraction of quorum slots in a sequence with

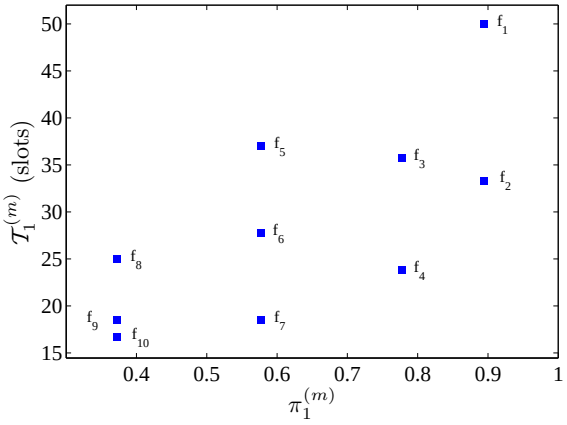


Fig. 13: Example of 10 channels with different $\pi_1^{(m)}$ and $\mathcal{T}_1^{(m)}$.

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 quorum channels used in the quorum slots.

VIII. Performance Evaluation

This section evaluates the performance of our unicast and multicast rendezvous algorithms. NGQFH is studied under different values of γ in (13), 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 γ in (14), and group sizes. Both unicast and 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 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 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 Figure 13. 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

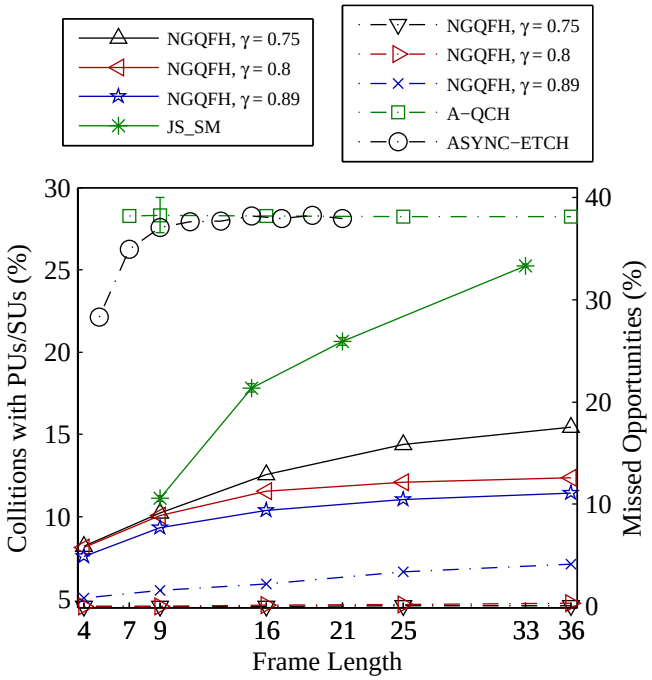


Fig. 14: Prediction accuracy vs. frame length.

for heterogeneous environments whereas JS_SM was designed for homogeneous environments. Because different algorithms were designed targeting different environments, our simulation figures only compare the algorithms that share a common setup.

Similar to GQFH (explained in Section III-C), A-QCH is a non-nested quorum-based FH algorithm (in contrast to NGQFH). However, each frame in an A-QCH sequence contains two subsequences, one is derived using a minimal cyclic quorum and the other uses a majority cyclic quorum. To implement A-QCH, we use the results in [20] to generate minimal and majority cyclic quorums for different frame lengths. It turns out that the frame length cannot be smaller than 7 [20]. A-QCH and M-QCH are simulated assuming that nodes select a common channel in each frame, as mentioned in [5]. Even though it is not explained in [5] how this can be accomplished in a distributed way, we assume that nodes negotiate a priori to agree on a common channel. 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 this per-slot sensing, JS_AM and JS_SM try to avoid unavailable channels by replacing them with available channels after constructing the frame and before start hopping. In order to compare JS_AM and JS_SM with NGQFH, we assume that JS_AM and JS_SM have an out-of-band sensing capability performed on a per-frame basis to identify the list of available channels at the beginning of each frame. Note that for some of the simulated algorithms (specifically, JS_AM, JS_SM, SYNC-ETCH, and ASYNC-ETCH) the frame length depends on the number of channels, whereas the frame length of the other algorithms (i.e., NGQFH, A-QCH, and M-QCH) is independent of the number of channels. Therefore, in order to have a valid comparison, we present the simulation results of these two sets of algorithms in separate figures.

1) Prediction Accuracy: Figures 14 and 15 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 γ. 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. It is also the case for ASYNC-ETCH, with an exception for sufficiently small frame lengths where missed opportunity rate is less. This is because A-QCH and ASYNC-ETCH access channels equally without preferring one

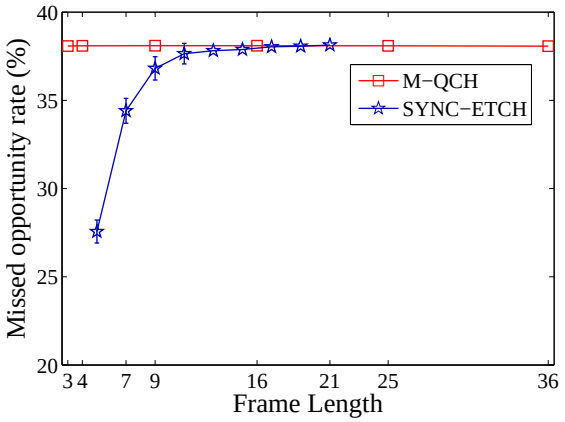


Fig. 15: Collision rate for M-QCH and SYNC-ETCH.

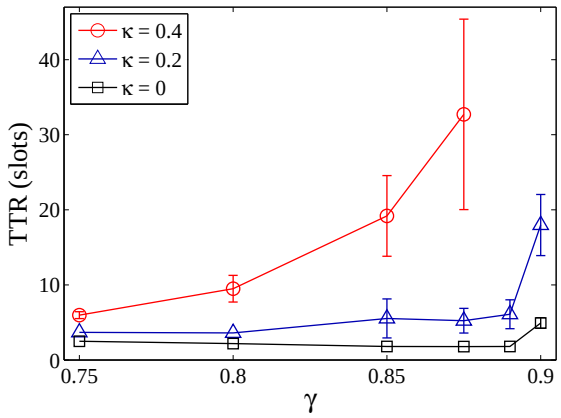
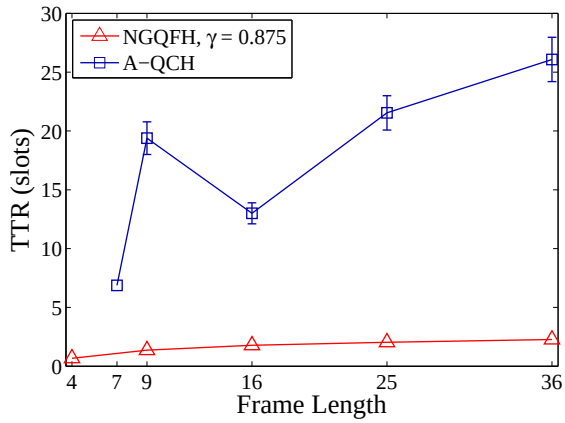
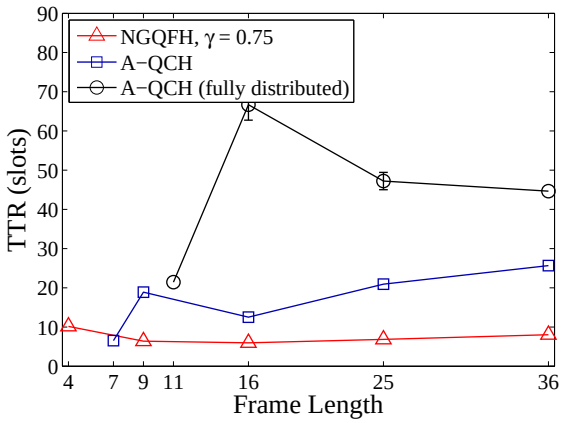


Fig. 16: TTR vs. γ for NGQFH (frame length = 16).



(a) $\kappa = 0$

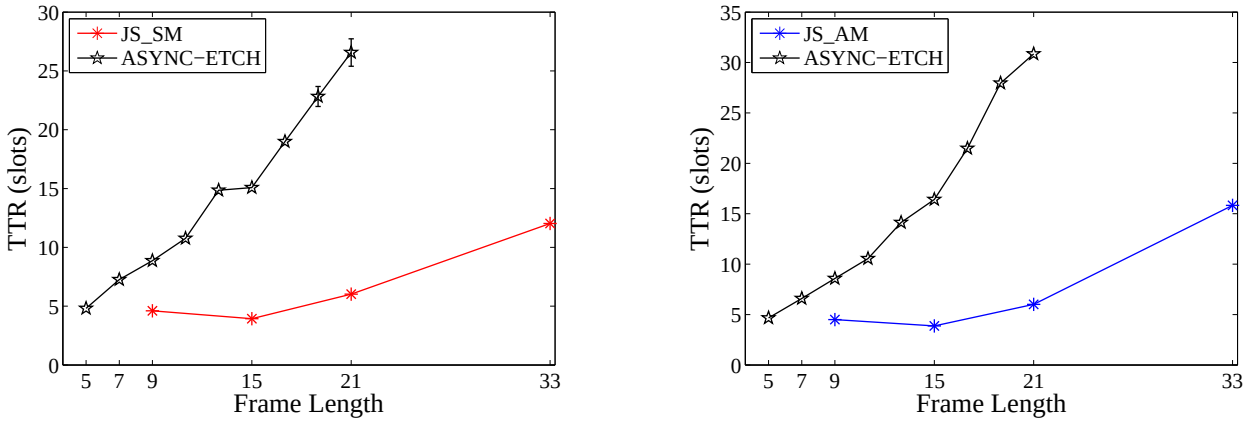


(b) $\kappa = 0.4$

Fig. 17: TTR for NGQFH and A-QCH.

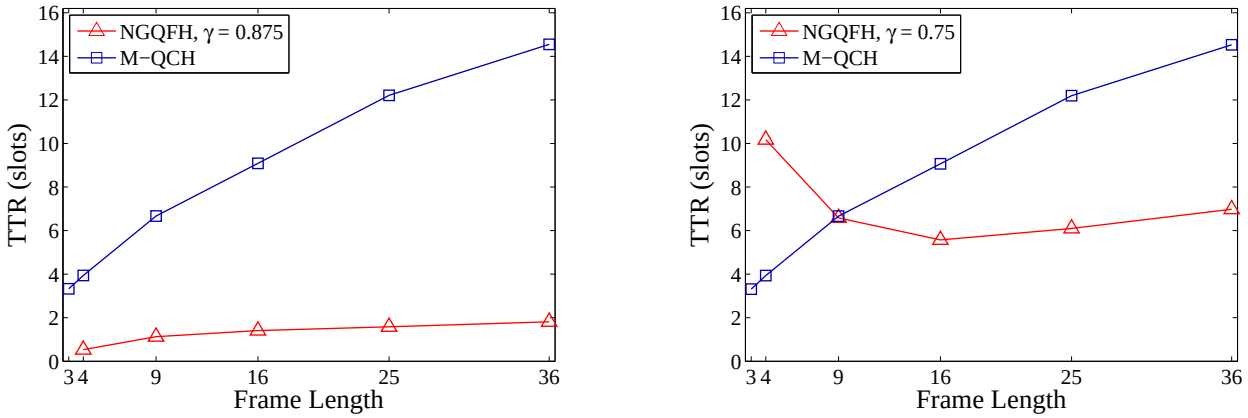
channel over another, and they refrain from transmission when a channel becomes occupied. The missed opportunity rate of JS_AM equals zero, but the collision rate is high. This is because JS_AM assumes that channel availability does not change during the frame, and it does not use any channel prediction mechanism. If NGQFH follows a conservative approach by selecting γ to be very large (e.g., $\gamma > 0.89$), then the missed opportunity rate increases with the frame length. This is a consequence of the reduction in the prediction accuracy due to increasing the forecasting period. Moreover, the number of quorum channels 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 γ (e.g., $\gamma = 0.75$), then the collision rate increases with the frame length. As shown in Figure 16, the best γ that results in the smallest TTR is a function of κ. The larger the value of κ, the smaller γ is required.

2) TTR: Figures 17 and 18 depict the average TTR vs. the frame length for homogeneous and heterogeneous environments in an asynchronous setup. Based on the previous discussion, we consider two values of γ 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 achieved improvement in TTR by NGQFH intensifies with the number of channels, since the TTR of JS_AM and JS_SM increase with the number of channels as shown in [16] 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



(a) $\kappa = 0$ (b) $\kappa = 0.4$

Fig. 18: TTR for JS_AM, JS_SM, and ASYNC-ETCH.

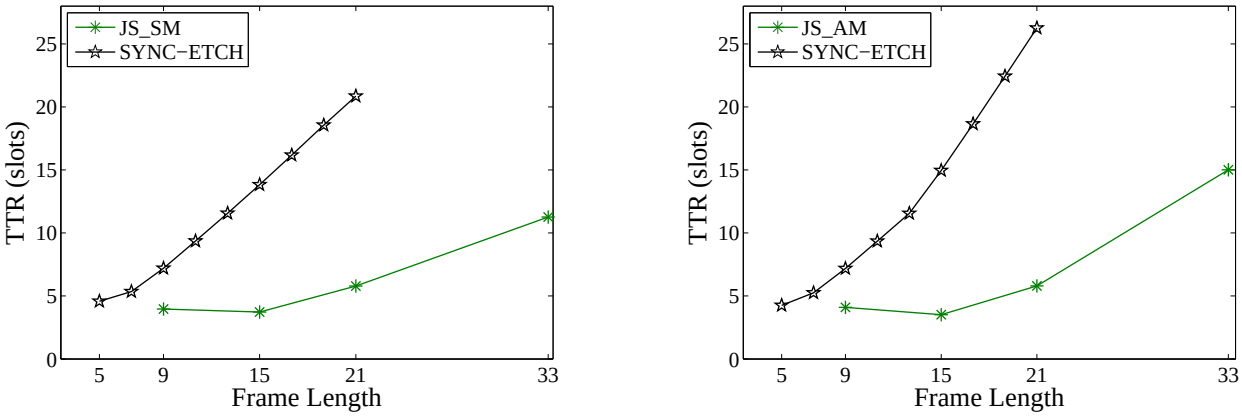


(a) $\kappa = 0$ (b) $\kappa = 0.4$

Fig. 19: TTR for NGQFH and M-QCH in the presence of node synchronization.

Figure 14. 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 (therefore, rendezvous opportunities) as a consequence of increasing the frame length. Another important point is that, in contrast to A-QCH whose smallest frame length is 7, NGQFH can accommodate frame lengths as small as 4 and achieve TTR as small as 1, which is well below the theoretical lower-bound of A-QCH under no PU dynamics. In Figure 17(b), fully distributed A-QCH represents a variant of A-QCH where the pre-negotiation assumption is relaxed and nodes select their quorum channels independently. Fully distributed A-QCH is simulated starting from a frame length of 11. This is because frames with lengths of 4 or 9 slots will not have any randomly assigned slot, and 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.

Similarly, the TTR is plotted in Figures 19 and 20 when all nodes are synced. One can easily verify that the TTR of NGQFH is several orders of magnitude less than that of the other algorithms, except for JS_AM, which has a similar performance to NGQFH. However, JS_AM is simulated using a smaller number of channels than NGQFH for the same frame length, which increases the possibility of two nodes



(a) $\kappa = 0$ (b) $\kappa = 0.4$

Fig. 20: TTR for JS_AM, JS_SM, and SYNC-ETCH in the presence of node synchronization.

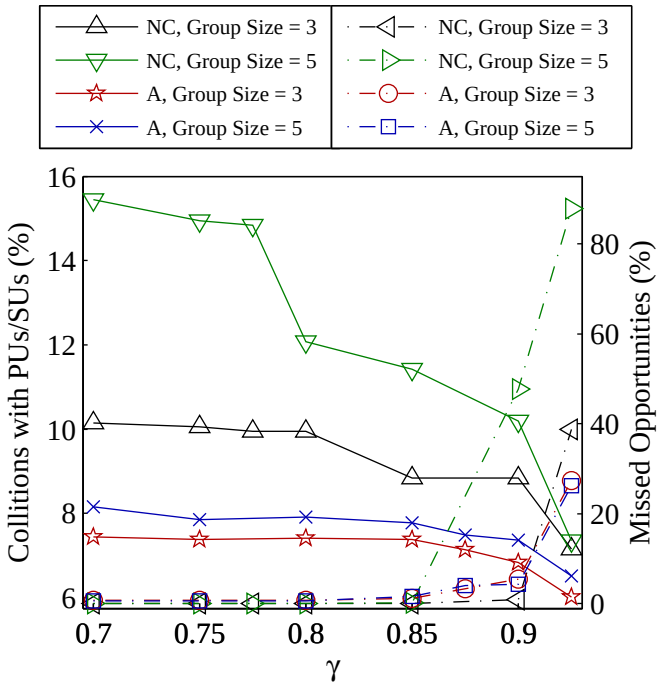


Fig. 21: Prediction accuracy for AMQFH and nested-CMQFH.

selecting the same channel for JS_AM compared to NGQFH.

B. Multicast

Now, let us consider the performance of our multicast algorithms. To the best of our knowledge, there is no other non-sequential multicast algorithm in the literature. Hence, we only study and compare our proposed algorithms.

1) Prediction Accuracy: Figure 21 depicts the collision and missed opportunity rates with respect to γ for two group sizes. As expected, a conservative prediction (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 prediction 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.

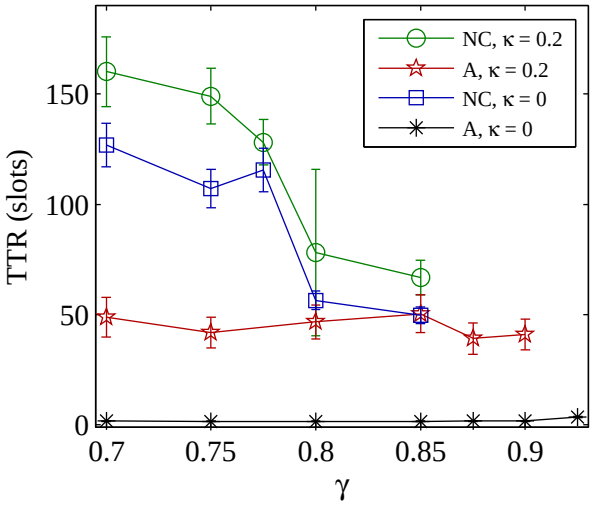


Fig. 22: TTR vs. γ (multicast).

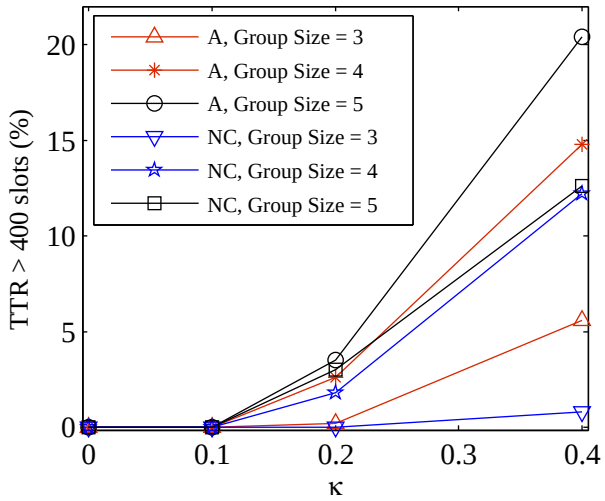


Fig. 23: TTR vs. κ (multicast).

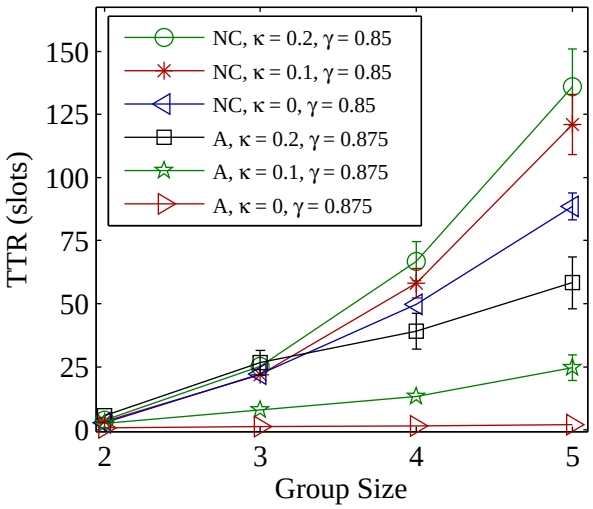


Fig. 24: TTR vs. group size (multicast).

2) TTR: Figure 22 shows the effect of γ on TTR for different values of κ. As shown in the figure, AMQFH is faster than nested-CMQFH. Note that 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 also often depends on κ.

AMQFH and nested-CMQFH do not provide any guarantee for a multicast group to always rendezvous within a reasonable time in heterogeneous environments. However, we evaluate these algorithms with respect to their ability to promise a probabilistic guarantee. Specifically, we select 400 slots as a nominal value for a reasonable TTR, and then we take the percentage of runs with TTR > 400, as shown in Figure 23. This figure shows that this percentage increases with κ. Although counterintuitive, nested-CMQFH proves to be more reliable (i.e., more likely to achieve a TTR smaller than 400 slots) than AMQFH in heterogeneous environments. The reason is that, in AMQFH nodes either rendezvous quickly or do not rendezvous. If nodes cannot meet quickly in AMQFH, this means that they have totally different sets of "best channels", and so they fail to rendezvous. In contrast to AMQFH, in nested-CMQFH nodes eventually manage to rendezvous because of the nested nature of the algorithm. The curves are also increasing with κ, and the nested design is more capable in coping with heterogeneous environment.

However, the percentage of runs with TTR exceeds 400 slots does not completely characterize the performance of the multicast algorithms. In addition, Figure 24 shows the 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).

3) HD: The ability of the proposed algorithms to provide a high HD is considered in Figure 25. 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 prediction 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 decreases when γ is increased.

IX. Conclusions

In this paper, we developed asynchronous algorithms for pairwise and multicast rendezvous in 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 γ is selected appropriately, NGQFH achieves a significant improvement in TTR and detection accuracy compared to previous algorithms. The best γ depends on the heterogeneity level. AMQFH can provide smaller TTR than nested-CMQFH, but the latter can provide a better probabilistic guarantee for rendezvous. Also, nested-CMQFH achieves better HD than AMQFH and is more robust against jamming.

References

[1] M. J. Abde-Rahman, H. Rahbari, and M. Krunz, "Adaptive frequency hopping algorithms for multicast rendezvous in DSA networks," in *Proc. of the IEEE DySPAN Conf.*, Oct. 2012, pp. 494–505.

[2] M. J. Abdel-Rahman, H. Rahbari, and M. Krunz, "Rendezvous in dynamic spectrum wireless networks," University of Arizona, Tech. Rep. TR-UA-ECE-2013-2, May 2013. [Online]. Available: http://www2.engr.arizona.edu/~mjabdelrahman.

[3] 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. of the ACM MobiCom Conf.*, 2004, pp. 216–230.

[4] K. Bian, J. M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proc. of the ACM MobiCom Conf.*, 2009, pp. 25–36.

[5] K. Bian, J.-M. Park, and R. Chen, "Control channel establishment in cognitive radio networks using channel hopping," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 689–703, 2011.

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

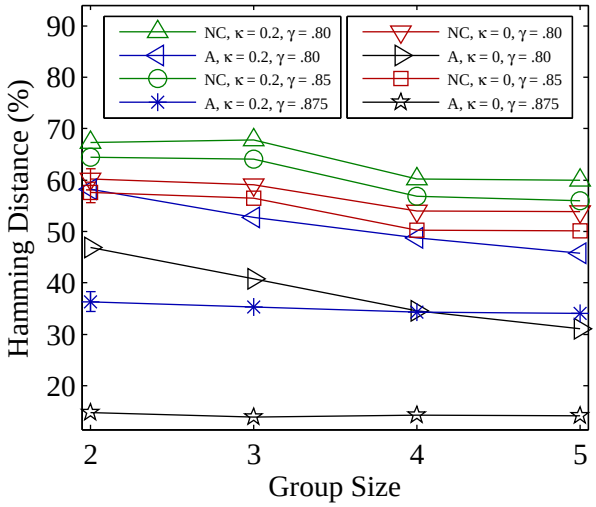


Fig. 25: HD for AMQFH and nested-CMQFH.

[7] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "IEEE 802.22: The first worldwide wireless standard based on cognitive radios," in *Proc. of the IEEE DySPAN Conf.*, Nov. 2005, pp. 328–337.

[8] L. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *Proc. of the IEEE DySPAN Conf.*, 2008.

[9] 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.

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

[11] T. Harrold, R. Cepeda, and M. Beach, "Long-term measurements of spectrum occupancy characteristics," in *Proceedings of the IEEE DySPAN Conference*, 2011, pp. 83–89.

[12] 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 Networks and Applications*, vol. 10, pp. 169–181, Feb. 2005.

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

[14] G. F. Lawler, *Introduction to Stochastic Processes*. Chapman and Hall/CRC, Taylor and Francis Group, 2006.

[15] L. Lazos, S. Liu, and M. Krunz, "Spectrum opportunity-based control channel assignment in cognitive radio networks," in *Proc. of the IEEE SECON Conf.*, June 2009, pp. 1–9.

[16] 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. of the IEEE INFOCOM Conf.*, 2011.

[17] S. Liu, L. Lazos, and M. Krunz, "Thwarting inside jamming attacks on wireless broadcast communications," in *Proc. of the ACM WiSec Conf.*, 2011.

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

[19] 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 Transactions on Vehicular Technology*, vol. 60, no. 6, pp. 2519–2533, 2011.

[20] W.-S. Luk and T.-T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *IEEE International Conference on Distributed Computing Systems*, 1997, pp. 100–106.

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

[22] D. R. Stinson, *Cryptography: Theory and Practice*. Chapman and Hall/CRC, Taylor and Francis Group, 2006.

[23] C.-H. Wu, J.-H. Hong, and C.-W. Wu, "RSA cryptosystem design based on the Chinese remainder theorem," in *Proc. of Asia and South Pacific Design Automation Conf.*, 2001, pp. 391–395.

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

APPENDIX

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 $\mathcal{T}_a$ and $\mathcal{T}_c$, respectively, can be expressed as follows:

Result 1. $\mathcal{T}_a$ is given by:

$$\mathcal{T}_a = \sum_{i=1}^{n-1} \left[i\Gamma(\gamma_{i+1}) \prod_{j=1}^{i} (1 - \Gamma(\gamma_j)) \right] \tag{15}$$

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):

$$\Gamma(\gamma_j) = \sum_{i=0}^{k} \left[\binom{k+1}{i} \gamma_j^{k+1-i} \left(\frac{1-\gamma_j}{L} \right)^i \right] + \left(\frac{1}{L} \right)^k (1-\gamma_j)^{k+1} \tag{16}$$

$$\gamma_i = \frac{\left\lfloor \frac{kn}{k+1} \right\rfloor - i + 2}{n} + \frac{i-1}{n} \times \frac{\left\lfloor \frac{kn}{k+1} \right\rfloor - i + 3}{n-i+1}. \tag{17}$$

Proof. Result 1 can be easily obtained, knowing that $\Gamma(\gamma_i)$ represents the probability that slot i is a rendezvous slot and γ_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 (16) considers rendezvous under all possible combinations of i randomly assigned slots and $k+1-i$ quorum slots. Equation (15) 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 (γ_i) depends on i. In (17), γ_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 (17). ■

Result 2. $\mathcal{T}_c$ is given by:

$$\mathcal{T}_c = \Theta \sum_{i=1}^{n-1} i(1-\Theta)^i \tag{18}$$

where Θ is given by (k is the multicast group size for CMQFH):

$$\Theta = \sum_{i=0}^{k-1} \left[\left(\frac{1}{L} \right)^i \sum_{\substack{\forall \{e_1, \ldots, e_{k-i}\} \\ \in \{p_1, \ldots, p_k\}}} \frac{\prod_{j=k-i+1}^{k} (e_j - 1)/e_j}{e_1 \ldots e_{k-i}} \right] + \left(\frac{1}{L} \right)^{k-1} \prod_{l=0}^{k-1} \frac{e_l - 1}{e_l}. \tag{19}$$

Proof. Similar to (15), equation (18) represents the discrete expectation formula of TTR, which takes values in $\{0, 1, \ldots, n-1\}$. Θ represents the probability that a given slot is a rendezvous slot in CMQFH, similar to Γ in AMQFH. Result 2 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 11). 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 $p_1 p_2 \ldots p_k$. 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 $1/(p_1 p_2 \ldots p_k)$. ■

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 3. Let $\phi \stackrel{\text{def}}{=} n - \left\{ \left\lfloor \frac{kn}{k+1} \right\rfloor + 1 \right\}$. Then, the expected HD of AMQFH, denoted by $\mathcal{H}_a$, and its upper

bound value, denoted by $\mathcal{H}_{a,\text{best}}$, are given by:

$$\mathcal{H}_a = \frac{L-1}{nL}\left\{\frac{(\varphi-1)(\phi+1)}{\varphi}+\frac{\phi}{\varphi}\right\} \tag{20}$$

$$\mathcal{H}_{a,\text{best}} = \frac{\phi+1}{n} \tag{21}$$

where φ is defined in (6).

Proof. Result 3 can be obtained by noticing that $\mathcal{H}_{a,\text{best}}$ corresponds to the case when different nodes select different quorums, and their randomly assigned parts are nonoverlapping. $\mathcal{H}_a$ 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 (20). ϕ represents the number of randomly assigned slots in each frame of the FH sequence. ■

Result 4. 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:

$$\mathcal{H}_c = \frac{L-1}{2Lk^2}\sum_{i=1}^{k}\sum_{j=1}^{k}\left(1-\frac{1}{p_ip_j}\right) \tag{22}$$

$$\mathcal{H}_{c,\text{best}} = \frac{1}{2\binom{k}{2}}\sum_{i=1}^{k}\sum_{\substack{j=1\\ j\neq i}}^{k}\left(1-\frac{1}{p_ip_j}\right). \tag{23}$$

Proof. $\mathcal{H}_{c,\text{best}}$ is defined similar to $\mathcal{H}_{k,\text{best}}$. Result 4 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. ■