Fast and Secure Rendezvous Protocols for Mitigating Control Channel DoS Attacks

Mohammad J. Abdel-Rahman*, Hanif Rahbari*, Marwan Krunz*, and Philippe Nain**
*Dept. of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85719, USA
**INRIA, Sophia Antipolis, France
Technical Report
TR-UA-ECE-2012-4
Last update: January 8, 2013

Abstract

The operation of a wireless network relies extensively on exchanging messages over a universally known channel, referred to as the *control channel*. The network performance can be severely degraded if a jammer launches a denial-of-service (DoS) attack on such a channel. In this paper, we design frequency hopping (FH) algorithms that mitigate DoS attacks on the control channel of an asynchronous ad hoc network. More specifically, three FH algorithms (called NUDoS, KMDoS, and NCMDoS) are developed for establishing unicast (NUDoS) and multicast (KMDoS and NCMDoS) communications in the presence of multiple jammers. KMDoS and NCMDoS provide different tradeoffs between speed and robustness to node compromise. Our algorithms are fully distributed, do not incur any additional message exchange overhead, and can work in the absence of node synchronization. Furthermore, KMDoS and NCMDoS have the attractive feature of maintaining the multicast group consistency. NUDoS exploits the *grid quorum system*, whereas KMDoS and NCMDoS use the *uniform k-arbiter* and the *Chinese remainder theorem (CRT)* quorum systems, respectively. Extensive simulations are used to evaluate our algorithms.

Index Terms

Control channel, frequency hopping design, jamming attacks, quorum systems.

I. Introduction

Wireless communications are vulnerable to intentional interference, typically referred to as jamming. The performance of a wireless network can be severely degraded if a jammer launches a denial-of-service (DoS) attack on the control channel, given the significance of this channel in supporting various network functions. Conventional anti-jamming techniques often rely on spread spectrum communications, including frequency hopping (FH). FH has been used in the literature for establishing unicast communications (a.k.a. pairwise rendezvous) in dynamic spectrum access (DSA) networks (e.g., [5]). However, most existing FH designs (e.g., [3]) are based on ad hoc approaches that do not provide any performance guarantees. One way to construct FH sequences in a systematic manner is to use quorum systems [6]. Quorums have been widely used in distributed systems to solve the mutual exclusion problem, the agreement problem, and the

This technical report in for [2].

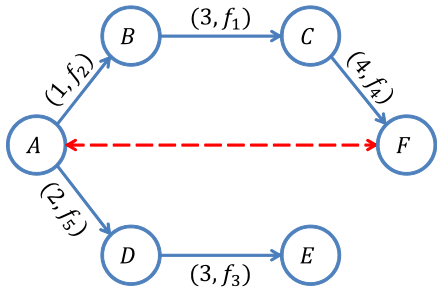

Fig. 1: Multicast as a series of unicasts. Node F receives A's message 3 time slots after node B receives it.

replica control problem. Systematic quorum-based FH approaches for establishing unicast communications in DSA networks have been proposed in [4], [5]. One important advantage of quorum-based FH designs is their robustness to synchronization errors [7]. As will be explained later, some quorum systems, such as *grid*, *uniform k-arbiter*, and *Chinese remainder theorem (CRT)* quorum systems, enjoy certain properties that allow them to be used for asynchronous communications. The approaches in [4], [5] do not intrinsically support *multicast rendezvous*, where all the nodes in a multicast group are required to rendezvous in the same time slot. Furthermore, these protocols are intended for a homogeneous spectrum environment, where all nodes are assumed to perceive the same set of available channels.

Group-based schemes have been proposed to facilitate multicast rendezvous in DSA networks [13]. One drawback of these schemes is the need for the initial step of neighbor discovery. Also, these schemes incur high overhead to maintain a group-based control channel. Even though these solutions establish intra-group communications, the problem of inter-group communications is yet another challenge that remains to be addressed [13].

In [12], the authors proposed an FH-based jamming-resistant broadcast communication scheme, called TDBS. TDBS operates in one of two modes, TDBS-SU and TDBS-AB. In both modes, the broadcast operation is implemented as a series of unicast transmissions, which can lead to multicast inconsistency. For example, a group of nodes may share a *group key* that is used to encode/decode common secure communication messages. For security purposes, this key may have to be updated periodically [15].

However, the change in the group key has to be consistent among all nodes in the multicast group. Such consistency cannot be guaranteed if changes in the group key are conveyed sequentially. Figure 1 shows a network of 6 nodes, where node A needs to send an update message about the group key to nodes B, C, D, E and F. If A's message is conveyed sequentially to B, then to D, then to C and E, and finally to F, using, for example, TDBS-AB, then, nodes B and F will have inconsistent information during the time slots 1, 2, and 3.

Instead of designing different FH sequences that overlap at common slots, the multicast rendezvous in [11] is established after a series of pairwise rendezvous operations that result in tuning all nodes in the multicast group to a common FH sequence. The effectiveness of this approach cannot be maintained under node compromise (if one node is compromised, then the FH sequences of all nodes are exposed).

Our Contributions–In this paper, we propose three FH algorithms to maintain control communications under a DoS attack on the control channel. More specifically, we propose a novel nested grid quorum-based FH algorithm, called NUDoS, for establishing unicast communications in a hostile environment with multiple jammers. NUDoS is faster than previously proposed pairwise rendezvous algorithms, robust to node compromise, and can function in the absence of node synchronization. To establish multicast communications while guaranteeing multicast consistency, we propose two asynchronous quorum-based FH algorithms, called KMDoS and NCMDoS, which provide different tradeoffs between speed and robustness to node compromise. Our algorithms are distributed and do not incur any additional message overhead.

Paper Organization–The rest of this paper is organized as follows. Section II introduces our system and jamming models, defines our evaluation metrics, and states our problem. The NUDoS algorithm is presented in Section III. Section IV presents two algorithms, called KMDoS and CMDoS, for establishing multicast communications in the presence of a control channel DoS attack. Establishing asynchronous communications is discussed in Section V. Section VI compares KMDoS with CMDoS. Based on the results in Section VI, an enhanced version of CMDoS, called NCMDoS, is proposed in Section VII. We

evaluate our algorithms in Section VIII, and conclude the paper in Section IX.

II. Models, Metrics, and Problem Statement

A. System Model

We consider a wireless ad hoc network with k nodes and L channels, denoted by $f_1, f_2, \ldots, f_L$. Without loss of generality, we assume that FH occurs on a per-slot basis, where the slot duration is T seconds. A packet can be exchanged between two or more nodes if they hop onto the same channel in the same time slot. We assume that one time slot is sufficient to exchange one message. If multiple groups happen to meet on the same channel in the same time slot, they use a CSMA/CA-style procedure to resolve contention.

For $j = 1, \ldots, k$, each node j has its unique FH sequence $\boldsymbol{w}^{(j)}$. The channel used in the ith slot of FH sequence $\boldsymbol{w}^{(j)}$ is denoted by $w_i^{(j)}, w_i^{(j)} \in \{f_1, \ldots, f_L\}$. Channel f_j is called a *rendezvous frequency* for the FH sequences $\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(k)}$ if there exists a *rendezvous slot* i such that $w_i^{(m)} = f_j, \forall m \in \{1, \ldots, k\}$. In our setup, each FH sequence consists of several frames. Each frame consists of a block of time-frequency hops.

B. Jamming Model

The jammer behavior is approximated by a two-state discrete-time Markov process, as shown in Figure 2. When the jammer is in state 0 it does not transmit; otherwise, it transmits a jamming signal.

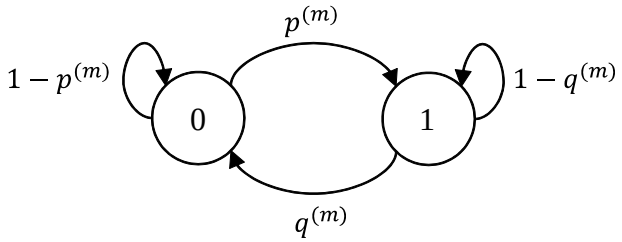

Fig. 2: State transition diagram of channel m under jamming.

Let $\rho^{(m)}$ be the probability that channel m is in state 1, and let $\mathcal{T}_1^{(m)}$ be the expected time (in slots) that channel m spends in state 1 before returning to state 0. Then, the transition probabilities $p^{(m)}$ and

$q^{(m)}$ in Figure 2 can be expressed as:

$$p^{(m)} = \frac{\rho^{(m)}}{1-\rho^{(m)}} \frac{1}{\mathcal{T}_1^{(m)}}, \quad q^{(m)} = \frac{1}{\mathcal{T}_1^{(m)}}. \tag{1}$$

C. Evaluation Metrics

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

Expected Evasion Delay (ED): ED is defined as the time between the successful jamming of the control channel and the re-establishment of a new one [10]. The expected ED is considered because of the existence of a randomly assigned part in our FH sequences.

Expected Hamming Distance (HD): The expected HD for two FH sequences $\boldsymbol{x} = (x_1 \dots x_n)$ 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 $\mathbf{1}_{\{.\}}$ is the indicator function and n is the frame length. In addition to robustness to node compromise, FH sequences with higher HD will have a lower collision probability. A collision occurs when two or more neighboring groups meet on the same channel in the same time slot.

D. Problem Statement

In this section, we state the FH construction problem, formulate it, and propose a centralized solution for it.

Problem Statement: *Given $f_1, \dots, f_L$, k, and n, determine the values of $w_i^{(j)}, 1 \leq i \leq n, 1 \leq j \leq k$, that result in the minimum ED, and achieve a minimum HD of d.*

Problem Formulation: The FH construction problem stated above can be formulated as follows:

Problem 1:

$$\underset{\{w_i^{(j)}: 1 \leq i \leq n, 1 \leq j \leq k\}}{\text{minimize}} \quad l$$

$$\text{Subject to.} \quad w_l^{(i)} = w_l^{(j)}, \forall i, j \in \{1, \dots, k\}, i \neq j \tag{2}$$

$$s_i^{(l)}[w_l^{(i)}] = 0, \forall i \in \{1, \dots, k\} \tag{3}$$

$$\sum_{r=1}^{n} \mathbf{1}_{\{w_r^{(i)} \neq w_r^{(j)}\}} \geq nd, \forall i, j \in \{1, \ldots, k\}, i \neq j \tag{4}$$

where $s_i^{(r)}[f_x] \in \{0, 1\}$ is the state of frequency $f_x \in \{f_1, \ldots, f_L\}$ in the rth time slot, as seen by node i.

Assume that $s_i^{(r)}[f_x], i \in \{1, \ldots, k\}, r \in \{1, \ldots, n\}, x \in \{1, \ldots, L\}$ are given. Then, the solution to Problem 1 gives the minimum ED, denoted by ED^*, and the rendezvous frequency, denoted by f_{u^*}. ED^* and u^* are given by:

$$ED^* = \min_{1 \leq u \leq L} \left\{ \min_{1 \leq r \leq n} \left(\bigvee_{i=1}^{k} s_i^{(r)}[f_u] == 0 \right) \right\} \tag{5}$$

$$u^* = \operatorname*{argmin}_{1 \leq u \leq L} \left\{ \min_{1 \leq r \leq n} \left(\bigvee_{i=1}^{k} s_i^{(r)}[f_u] == 0 \right) \right\} \tag{6}$$

where $\bigvee$ denotes the logical OR operation.

Relaxing the assumption of knowing the future states of the channels, next we propose a centralized algorithm that solves Problem 1 in $\mathcal{O}(knL)$ time, assuming that only the channel's transition probabilities are known.

Centralized Algorithm: Our algorithm, which relies on predicting the future states of the channels given the current states, can be summarized by the following steps:

1) For each slot $j = 1, \ldots, n$, compute p_j^* and l_j^* as follow:

$$p_j^* = \max_{1 \leq l \leq L} \left\{ \prod_{i=1}^{k} p_{n-e+j}^{(l)}(s_i^{(e)}[f_l], 0) \right\} \tag{7}$$

$$l_j^* = \operatorname*{argmax}_{1 \leq l \leq L} \left\{ \prod_{i=1}^{k} p_{n-e+j}^{(l)}(s_i^{(e)}[f_l], 0) \right\} \tag{8}$$

where e is the index of the slot in the current frame when channel f_l has been recently sensed and $p_{n-e+j}^{(l)}(s_i^{(e)}[f_l], 0)$ is the $(n-e+j)$-step transition probability of f_l form state $s_i^{(e)}[f_l]$ to state 0. In general, the v-step transition probability of channel m from state s to state 0 can be expressed as

[9]:

$$p_v^{(m)}(s,0) = \frac{q^{(m)} + p^{(m)}\Big[-\frac{q^{(m)}}{p^{(m)}}\Big]^s \Big[1 - p^{(m)} - q^{(m)}\Big]^v}{p^{(m)} + q^{(m)}}. \quad (9)$$

2) Sort slots ascendingly according to their p_j^* values.

3) Select the top $n - \lceil nd \rceil = \lfloor n(1-d) \rfloor$ slots in the list, and assign frequency $f_{l_j^*}$ to slot j in all FH sequences.

4) For the remaining $\lceil nd \rceil$ slots, assign different frequencies for different FH sequences.

Next, we exploit some properties of quorum systems in designing distributed FH rendezvous algorithms, which will be compared with the centralized algorithm in Section VIII.

III. Unicast Communications

Before describing NUDoS for unicast communications, we first give a few basic definitions that will facilitate further understanding of subsequent sections.

A. Preliminaries

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

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

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

Definition 3 (Rotation k-Closure Property): *A quorum system Q under Z_n is said to satisfy 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 above rotation k-closure property can be exploited to achieve asynchronous unicast as well as multicast communications, as will be explained later. One such quorum system that satisfies the rotation 2-closure property is the grid quorum system [7].

Definition 4 (Grid Quorum System): *A grid quorum system* [7] *arranges the elements of the set* Z_n *as a* $\sqrt{n} \times \sqrt{n}$ *array. In this case, a quorum is formed from the elements of any column plus any row of the grid.*

Figure 3 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 3). 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 quorums of Q and all cyclically rotated quorums of Q by at least two elements. In Figure 3, $G' = rotate(G, 1)$ and $H' = rotate(H, 2)$ intersect at the two elements labeled as I'.

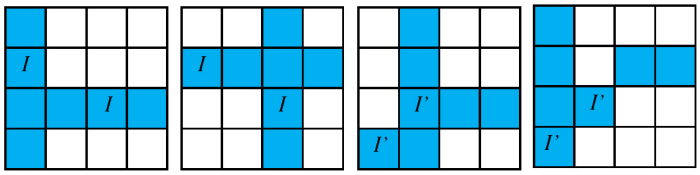

Fig. 3: Rotation 2-closure property of grid quorum systems.

B. NUDoS Algorithm

In NUDoS, every frame of every FH sequence uses $\sqrt{n} - 1$ rendezvous frequencies, where n is the frame length in slots. The following example explains the operation of the NUDoS algorithm for $n = 16$ (hence, each frame of every FH sequence contains $\sqrt{n} - 1 = 3$ rendezvous frequencies).

1) Construct a grid quorum system Q under Z_{16}. Q consists of 16 different quorums, each of $2\sqrt{16}-1 = 7$ elements.
2) Construct an FH sequence $\boldsymbol{w}$ as follows:
   - Select *the outer-most quorum* $G_1^{(1)}$ from the quorum system Q (e.g., $G_1^{(1)} = \{0, 1, 2, 3, 4, 8, 12\}$, where each entry represents the index of a time slot in a 16-slot frame).

- Assign a rendezvous frequency $h_1^{(1)} \in \{f_1, \ldots, f_L\}$ to the FH slots that correspond to $G_1^{(1)}$.
- Delete quorum $G_1^{(1)}$ from the original 4×4 grid and select the *next outer-most quorum* $G_2^{(1)}$ from the resulting 3×3 grid (e.g., $G_2^{(1)} = \{6, 9, 10, 11, 14\}$). Then, assign another rendezvous frequency $h_2^{(1)}$ to the FH slots that correspond to $G_2^{(1)}$.
- Delete quorum $G_2^{(1)}$ from the 3×3 grid, and select the next outer-most quorum $G_3^{(1)}$ from the resulting 2×2 grid (e.g., $G_3^{(1)} = \{7, 13, 15\}$). Then, assign a third rendezvous frequency $h_3^{(1)}$ to the FH slots that correspond to $G_3^{(1)}$.
- Assign a random frequency $h_x^{(1)} \in \{f_1, \ldots, f_L\} \setminus \{h_1^{(1)}, h_2^{(1)}, h_3^{(1)}\}$ to each of the unassigned slots.
- Repeat the above steps for the other frames in $\boldsymbol{w}$.

3) Repeat step 2 for other FH sequences.

Throughout this paper, $h_i^{(j)}$ and $G_i^{(j)}$, $i \in \{1, \ldots, \sqrt{n} - 1\}$, denote the ith quorum-assigned frequency that is assigned to the $(\sqrt{n} - i + 1) \times (\sqrt{n} - i + 1)$ quorum $G_i^{(j)}$ in the jth frame. A pseudo-code of the NUDoS algorithm is shown in Algorithm 1. Figure 4 shows the resulting frames of two FH sequences $\boldsymbol{w}$ and $\boldsymbol{x}$, constructed according to Algorithm 1.

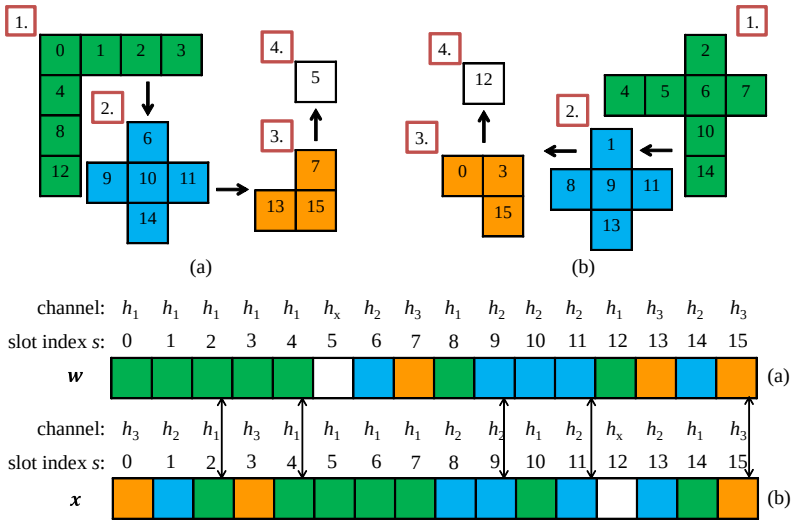

Fig. 4: Generation of the NUDoS nested quorums.

Algorithm 1 NUDoS Algorithm

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

C. Features of the NUDoS Algorithm

NUDoS has two main features. First, because of the nested generation of quorums, the *overlap ratio* between two FH sequences (number of rendezvous slots in a frame divided by the frame length) is significantly higher than the overlap ratio for a non-nested grid quorum-based FH algorithm (herein denoted by UDoS). In UDoS, an FH sequence consists of only one rendezvous frequency, assigned to a $\sqrt{n}\times\sqrt{n}$ quorum. FH systems with a higher overlap ratio work more efficiently in hostile environments, where a jammer may suddenly appear on a rendezvous channel. Besides having a higher overlap ratio, NUDoS involves multiple rendezvous frequencies per frame, which increases the likelihood of rendezvousing.

The advantage of a nested grid quorum with multiple rendezvous frequencies can be formalized by deriving the expected overlap ratio for UDoS and NUDoS, denoted by $\mathcal{V}_{UDoS}$ and $\mathcal{V}_{NUDoS}$, respectively. $\mathcal{V}_{UDoS}$ is composed of the sum of two parts; the expected overlap ratio between the quorum-based assigned parts of the FH sequences, denoted by $\mathcal{V}_{UDoS}^{Q}$, and the expected overlap ratio between the randomly assigned parts, denoted by $\mathcal{V}_{UDoS}^{R}$. Similarly, $\mathcal{V}_{NUDoS}$ is composed of $\mathcal{V}_{NUDoS}^{Q}$ and $\mathcal{V}_{NUDoS}^{R}$. For a given n, $\mathcal{V}_{UDoS}^{Q}$ and $\mathcal{V}_{NUDoS}^{Q}$ can be determined numerically. After some manipulations, $\mathcal{V}_{UDoS}^{R}$ and $\mathcal{V}_{NUDoS}^{R}$ can be expressed as in the following result.

Result 1: $\mathcal{V}^R_{UDoS}$ and $\mathcal{V}^R_{NUDoS}$ can be expressed as functions of L and n as follow:

$$\mathcal{V}^R_{UDoS}(L,n) = \frac{(\sqrt{n}-1)^2}{L}\left\{2-\frac{(\sqrt{n}-1)^2}{n}\right\} \tag{10}$$

$$\mathcal{V}^R_{NUDoS}(L,n) = \frac{1}{L}\left\{2-\frac{1}{n^2}\right\}. \tag{11}$$

Figure 5 depicts $\mathcal{V}_{UDoS}$ and $\mathcal{V}_{NUDoS}$ vs. n for different values of L. It can be observed that $\mathcal{V}_{NUDoS}$ is larger than $\mathcal{V}_{UDoS}$, and both decrease with n.

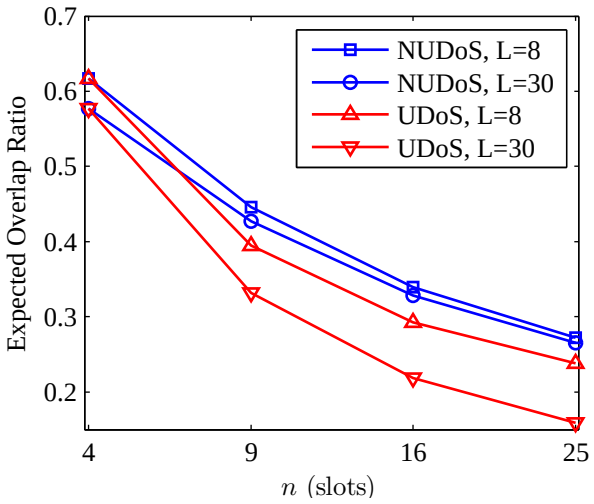

Fig. 5: Expected overlap ratio vs. the frame length.

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

$$\mathcal{K}_n = \prod_{j=0}^{\sqrt{n}-2}(\sqrt{n}-j)^2. \tag{12}$$

Note that $\mathcal{K}_n$ increases with n. A higher $\mathcal{K}_n$ value means more robustness to node compromise.

IV. MULTICAST COMMUNICATIONS

The multicast rendezvous algorithms, AMQFH and CMQFH, proposed in [1] are customized for maintaining multicast communications under a DoS attack on the control channel. The resulted algorithms are called KMDoS and CMDoS, respectively. These algorithms have two main attractive features. First, they allow a node to construct its sequence by only knowing the number of nodes in its multicast group. Hence, these algorithms can be executed in a fully distributed way. Second, these algorithms can still function in the absence of node synchronization.

A. KMDoS Algorithm

The KMDoS algorithm uses the uniform k-arbiter quorum system, which exhibits the rotation $(k+1)$-closure property. Before explaining the KMDoS algorithm, we first define the k-arbiter and uniform k-arbiter quorum systems.

Definition 5 (k-Arbiter Quorum System) [14]: *A k-arbiter quorum system Q under Z_n is a collection of quorums such that $\bigcap_{i=1}^{k+1} G_i \neq \emptyset, \forall G_1, G_2, \ldots, G_{k+1} \in Q$.*

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.

One specific type of k-arbiter quorum systems that is of interest to us is the so-called uniform k-arbiter quorum system. Such a system Q satisfies [8]:

$$Q = \left\{ G \subseteq Z_n \quad : \quad |G| = \left(\left\lfloor \frac{kn}{k+1} \right\rfloor + 1 \right) \right\}. \tag{13}$$

The above 2-arbiter quorum system is a uniform 2-arbiter because each quorum in Q contains $\lfloor 2 \times 4/(2+1) \rfloor + 1 = 3$ elements of Z_4. It is known [8] that the uniform k-arbiter quorum system has the rotation (k + 1)-closure property. In uniform k-arbiter quorum systems, the rotation of one quorum results in another quorum, which makes such quorum systems satisfy the rotation $(k+1)$-closure property.

To create FH sequences that satisfy the rotation $(k+1)$-closure property using a uniform k-arbiter

quorum system, n needs to be selected such that the number of different quorums of length $\lfloor kn/(k+1) \rfloor + 1$ that can be derived from Z_n, denoted by φ, is greater than or equal to $k+1$, i.e.,

$$\varphi \stackrel{\text{def}}{=} \binom{n}{\lfloor \frac{kn}{k+1} \rfloor + 1} \geq k+1. \tag{14}$$

To satisfy (14), $\lfloor \frac{kn}{k+1} \rfloor$ should be less than $n-1$, which requires n to be greater than $k+1$ (k and n are positive integers, and $\frac{k}{k+1}$ is monotonically increasing in k).

We now explain KMDoS through an example. Consider a multicast group of 3 nodes. In KMDoS, each FH sequence consists of several time frames, each containing several slots. Because the uniform 2-arbiter quorum system satisfies the rotation 3-closure property (i.e., any three cyclically rotated quorums overlap in at least one slot), each frame is constructed using one quorum. Thus, the frame length will be n. We set n to the smallest value that satisfies (14), i.e., $n = k+2 = 4$. The following steps are used to obtain the various FH sequences:

1) Construct a universal set $Z_4 = \{0, 1, 2, 3\}$.
2) Construct a uniform 2-arbiter system Q under Z_4.
3) Construct an FH sequence $\boldsymbol{w}$ as follows:
   - Select a quorum from Q and assign it to $G_1^{(1)}$ (e.g., $G_1^{(1)} = \{0, 1, 2\}$).
   - Assign a frequency $h_1^{(1)}$ to the FH slots in the given frame that correspond to $G_1^{(1)}$, and assign a random frequency $h_x^{(1)}$ to the other slots, where $h_1^{(1)}$ and $h_x^{(1)} \in \{f_1, \ldots, f_L\}$.
   - Repeat the above procedure for the other frames using quorum $G_1^{(k)}$ and channel f_k for the kth frame.
4) Repeat step 3 to construct the other FH sequences.

Figure 6 shows three frames of FH sequences $\boldsymbol{w}$, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, constructed according to the KMDoS algorithm.

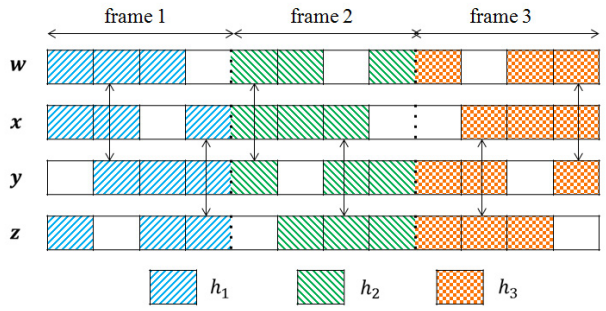

Fig. 6: KMDoS FH construction algorithm.

B. CMDoS Algorithm

The CMDoS algorithm uses the CRT quorum system, which also exhibits the rotation k-closure property. The CRT is formally described as follows [16]:

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

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

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

Using Theorem 1, we can construct quorum systems that satisfy the rotation k-closure property, as in Theorem 2.

Theorem 2: *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 $G_i = \{p_i c_i, c_i = 0, \ldots, y/p_i - 1\}$, satisfies the rotation k-closure property.*

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

$\forall z_1, z_2, z_3 \in Z_{29}$, the three quorums G_1, G_2, and G_3 have an intersection. Thus, the CRT quorum system $Q = \{G_1, G_2, G_3\}$ satisfies the rotation 3-closure property.

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

V. Asynchronous Communications

Result 2: *FH sequences constructed according to NUDoS, KMDoS, and CMDoS can establish asynchronous communications if each FH sequence continues to use the same outer-most quorum and channel in all frames of the FH sequence, i.e., $G_1^{(i)}$ and $h_1^{(i)}$ have the same value for all integer values of i, where $h_1^{(i)}$ and $G_1^{(i)}$ are as defined in Section III-B.*

Proof: Result 2 is a direct consequence of the intersection and rotation closure properties of the grid, uniform k-arbiter, and CRT quorum systems, and the fact that each frame in an FH sequence is constructed using one quorum (the outer-most).

The condition in Result 2 is sufficient but not necessary. Thus, FH sequences can still rendezvous even if the outer-most quorum is changed in some frames, provided that this change does not occur very frequently. This is illustrated in Figure 7, where the outer-most quorum in sequence $\boldsymbol{w}$ changes between quorums H_1 and H_2, and in sequence $\boldsymbol{x}$ changes between quorums H_3 and H_4. The left shaded part of sequence $\boldsymbol{x}$ in Figure 7 represents a cyclic rotation of H_3, and hence, by the rotation closure property, this part overlaps with quorum H_1 of sequence $\boldsymbol{w}$. The right shaded part in sequence $\boldsymbol{x}$ does not generally overlap with H_1 in $\boldsymbol{w}$.

Because the condition in Result 2 is sufficient but not necessary, we require channel $h_1^{(1)}$ to be available for a certain number of slots in the current $G_1^{(1)}$ quorum in order to keep assigning $h_1^{(1)}$ to this quorum in the next frame (i.e., selecting $h_1^{(2)}$ to be $h_1^{(1)}$ and $G_1^{(2)}$ to be $G_1^{(1)}$). Otherwise, $h_1^{(1)}$ is assigned to the quorum for which $h_1^{(1)}$ is maximally available (i.e., the quorum that has the maximum number of available slots during which $h_1^{(1)}$ is predicted to be unjammed).

Remark: In our implementations in Section VIII, channels and quorums are selected based on the forecasted availability of the channels at different quorums, as derived from the jamming model described in Section II-B. A channel is considered available at a future slot if it is predicted to be available at that slot with probability greater than p_{th}. p_{th} is an important parameter that will be studied in Section VIII. The details of the channel and quorum selection procedures are omitted in this paper due to space limitation.

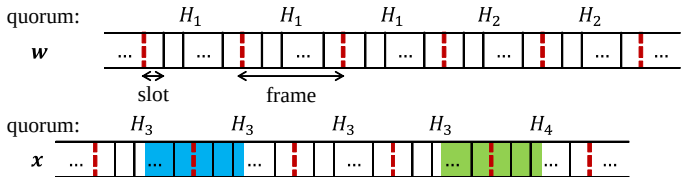

Fig. 7: Example of asynchronous communications.

VI. KMDoS vs. CMDoS: Speed vs. Security

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

A. Expected ED

By examining the structures of the uniform k-arbiter and CRT quorum systems, the expected ED of KMDoS and CMDoS, denoted by $\mathcal{E}_k$ and $\mathcal{E}_c$, respectively, can be expressed as follows:

Result 3: *$\mathcal{E}_k$ is given by:*

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

where $\Gamma(\gamma_i)$ is the probability that slot i is a rendezvous slot and γ_i is 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. After some manipulations, it can be easily shown that $\Gamma(\gamma_i)$ and

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

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

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

Result 4: $\mathcal{E}_c$ *is given by:*

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

where Θ is the probability that a given slot is a rendezvous slot. Θ is given by (k is the multicast group size for CMDoS):

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

Figure 8 compares an upper-bound on the expected ED (i.e., when nodes cannot rendezvous during the randomly assigned slots with probability 1) for KMDoS with the expected ED of CMDoS for $L = 2, 3$. For $L > 3$, the expected ED of CMDoS is much higher than KMDoS. For both algorithms, the expected ED increases with the multicast group size.

B. Expected HD

In KMDoS, the expected HD is the same for all pairs of FH sequences, whereas in CMDoS they are different for different pairs. Thus, for CMDoS, the expected value over all pairs of FH sequences is computed.

Result 5: *Let $\phi \stackrel{def}{=} n - \left\{ \left\lfloor \frac{kn}{k+1} \right\rfloor + 1 \right\}$. Then, the expected HD of KMDoS, denoted by $\mathcal{H}_k$, and its upper*

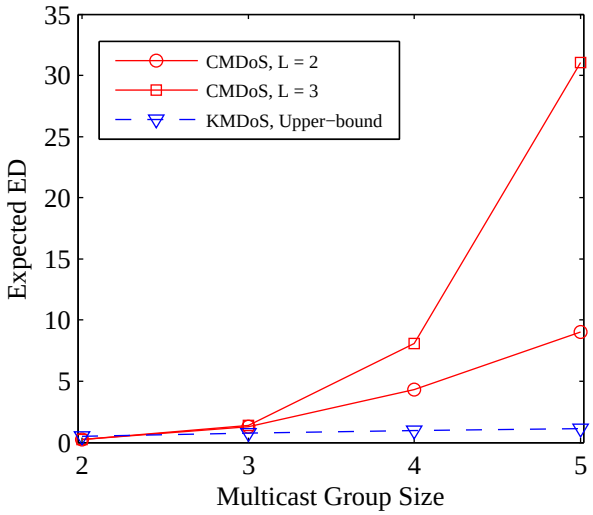

Fig. 8: Expected ED vs. multicast group size.

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

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

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

where $\mathcal{H}_{k,best}$ corresponds to the case when different nodes select different FH sequences, and nodes cannot rendezvous during the randomly assigned slots. $\mathcal{H}_k$ represents the general case when 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).

Result 6: *The expected HD of CMDoS, denoted by $\mathcal{H}_c$, and its upper bound value, denoted by $\mathcal{H}_{c,best}$, are given by:*

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

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

where $\mathcal{H}_{c,best}$ is defined similar to $\mathcal{H}_{k,best}$. This result 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* $\frac{n}{p_i p_j}$.

Figure 9 depicts the expected HD vs. the size of the multicast group for KMDoS and CMDoS. As the multicast group size increases, $\mathcal{H}_c$ increases but $\mathcal{H}_k$ decreases, and hence the gap between $\mathcal{H}_c$ and $\mathcal{H}_k$ increases with the increase in the size of the multicast group.

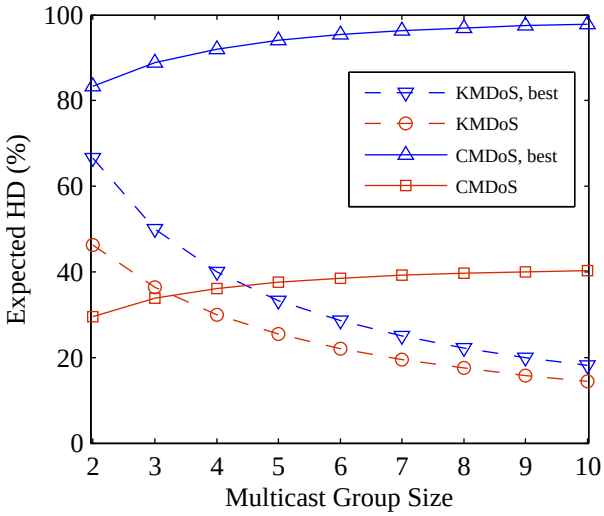

Fig. 9: Expected HD vs. multicast group size ($L = 6$).

VII. NCMDoS Algorithm

As shown in the previous section, the ED of CMDoS is much larger than that of KMDoS, but its average HD is also much higher. To provide a tradeoff between speed of rendezvous and robustness against node compromise, in this section we propose a third multicast rendezvous algorithm, called NCMDoS. NCMDoS is faster than CMDoS, but not as fast as KMDoS. At the same time, the HD of NCMDoS is larger than that of KMDoS, but not as large as CMDoS. We explain the NCMDoS algorithm through an example.

Suppose that the number of nodes in the multicast group is 3. Then, according to the CMDoS algorithm, $p_1 = 2, p_2 = 3, p_3 = 5$, and the frame length $y = p_1 p_2 p_3 = 30$. The difference between CMDoS and NCMDoS is that instead of having one quorum in each frame of an FH sequence, each FH sequence will have a certain number of nested quorums in each frame, depending on the prime number that is used in constructing this FH sequence. The number of quorums in a frame for a given FH sequence is called the

nesting degree of this FH sequence. In contrast to KMDoS and CMDoS, knowing the number of nodes in the multicast group is not enough to construct the multicast FH sequences in NCMDoS. In addition to the multicast group size, a node needs to know its nesting degree. The nesting degree constitutes a tradeoff between ED and HD. Large values of the nesting degree result in a small ED, but also a small HD. In our design, the FH sequence that uses a prime number p_i will have a nesting degree of $\lceil \frac{p_i}{2} \rceil$, so the nesting degree in NCMDoS is different for different FH sequences in the multicast group.

Figure 10 illustrates the NCMDoS design when the multicast group size is 3. 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 3, 2, and 1, respectively. Hence, sequence $\boldsymbol{x}$ will have three nested quorums, each of 5 slots, and each quorum is assigned a different channel (the same treatment is done for sequences $\boldsymbol{y}$ and $\boldsymbol{z}$). Note that sequence $\boldsymbol{x}$ can have five different quorums, each of 5 slots. The selection of 3-out-of-5 quorums and also channel selection are not discussed in this paper because of the space limitation. The performance of NCMDoS will be examined in Section VIII.

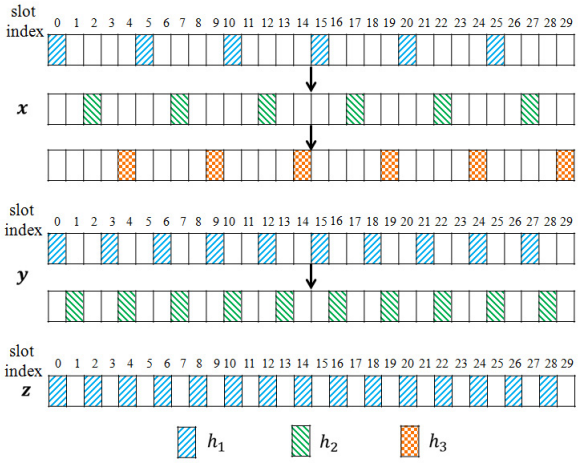

Fig. 10: NCMDoS FH construction algorithm.

VIII. Performance Evaluation

We now present simulation results for the NUDoS, KMDoS, and NCMDoS algorithms, and compare them with the centralized algorithm proposed in Section II-D. The proposed algorithms are studied under different frame lengths, thresholds (p_{th}), jamming probabilities ($\rho^{(m)}$), and values of $\mathcal{T}_1^{(m)}$. Our evaluation

metrics are the ED and the HD. Our algorithms are simulated under a realistic setting of spectrum heterogeneity (each channel could be in different states at different nodes) and no synchronization (the misalignment between FH sequences is randomly selected in each experiment). The 95% confidence intervals are indicated. When they are very tight, they are not drawn to prevent cluttering the graph. Because $\rho^{(m)}$ and $\mathcal{T}_1^{(m)}$ are dependent ($0 \leq \rho^{(m)} \leq \mathcal{T}_1^{(m)}/(\mathcal{T}_1^{(m)}+1)$), different curves of fixed values of $\mathcal{T}_1^{(m)}$ have different ranges of $\rho^{(m)}$. For a given value of $\rho^{(m)}$, increasing $\mathcal{T}_1^{(m)}$ reduces the fluctuation level of channel m. In the centralized algorithm, because of the minimum required HD, nodes may not rendezvous if n is sufficiently small because all the slots in the frame will be assigned differently for different nodes, and hence some ED points are missing. As mentioned in step 4 in the centralized algorithm, the achieved HD is $\lfloor nd \rfloor / n$. Hence, for a fixed d, the HD is different for different frame lengths.

A. Unicast Communications (NUDoS)

1) Evasion Delay (ED): Figures 11 and 12 depict the ED of NUDoS, and compare it with the centralized algorithm (denoted by C). The ED for both NUDoS and the centralized algorithm increases with n because of the reduction in the overlap ratio, as shown in Figure 5. The ED also increases with $\rho^{(m)}$. NUDoS achieves less ED for less fluctuating channels, under medium to high values of $\rho^{(m)}$ (recall that quorums and channels are selected in NUDoS based on the predicted channel's states, and less fluctuating channels are more predictable). While achieving a close HD to the centralized algorithm, the speed of NUDoS is comparable to the centralized algorithm for small to moderate values of $\rho^{(m)}$. The ED of NUDoS increases with p_{th}. Increasing d in (4) increases the ED of the centralized algorithm.

2) Hamming Distance (HD): The HD for NUDoS and C is plotted in Figures 13 and 14. The HD of NUDoS increases with n because of the reduction in the overlap ratio. It also increases with both p_{th} and $\rho^{(m)}$ because of the increase in the number of unassigned slots (in our simulations, each unassigned slot increments the HD by $1/n$). For small values of $\rho^{(m)}$, less fluctuating channels result in larger HD, and the opposite for large values of $\rho^{(m)}$.

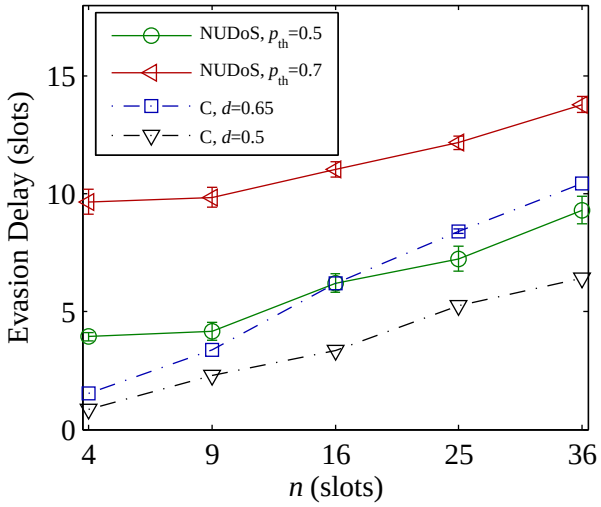

Fig. 11: ED vs. n for NUDoS ($\rho^{(m)} = 0.3, \mathcal{T}_1^{(m)} = 4$).

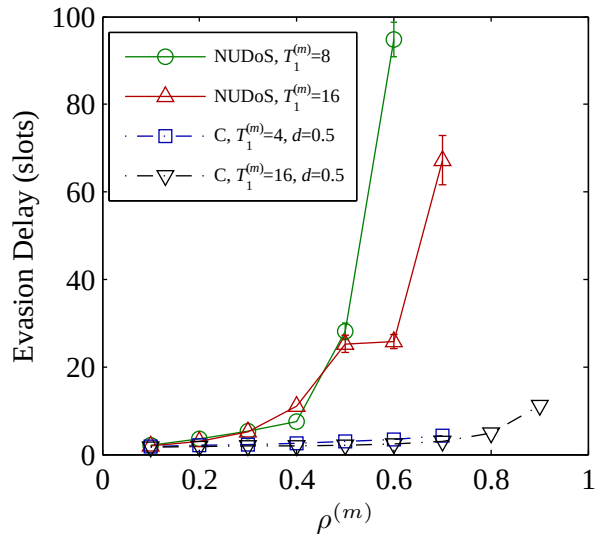

Fig. 12: ED vs. $\rho^{(m)}$ for NUDoS ($n = 9, p_{th} = 0.5$).

B. Multicast Communications (KMDoS and NCMDoS)

1) Evasion Delay (ED): Figures 15 and 16 show the ED of KMDoS and the centralized algorithm. For small values of p_{th} (e.g., $p_{th} = 0.5$), KMDoS maintains a fixed ED value close to the centralized algorithm (with a similar HD) as the group size increases, whereas the ED of KMDoS increases with n for large values of p_{th} (e.g., $p_{th} = 0.6$). The ED of KMDoS increases with $\rho^{(m)}$. KMDoS provides less ED for more fluctuating channels under small values of $\rho^{(m)}$, and the opposite for medium to high values of $\rho^{(m)}$. Figure 17 depicts the ED of NCMDoS vs. $\rho^{(m)}$ for different values of $\mathcal{T}_1^{(m)}$. NCMDoS is much

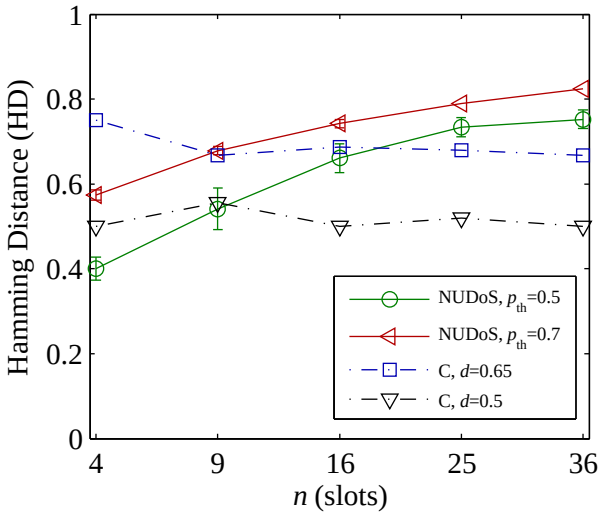

Fig. 13: HD vs. n for NUDoS ($\rho^{(m)} = 0.3, \mathcal{T}_1^{(m)} = 4$).

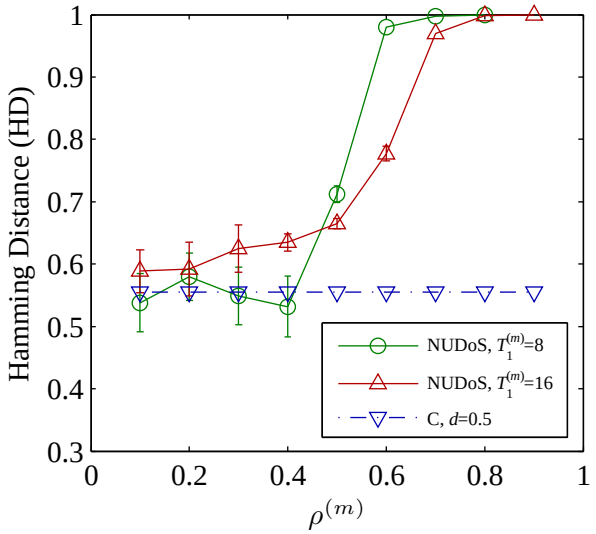

Fig. 14: HD vs. $\rho^{(m)}$ for NUDoS ($n = 9, p_{th} = 0.5$).

slower than both KMDoS and the centralized algorithm.

2) Hamming Distance (HD): As shown in Figure 18, the HD of NCMDoS is larger than that of KMDoS, and the gap increases with the increase in the group size. Figure 19 shows the improvement in HD achieved under NCMDoS (with group size 3) compared to KMDoS (with group size 6). The HD increases with both p_{th} and $\rho^{(m)}$ because of the increase in the number of unassigned slots. $\mathcal{T}_1^{(m)}$ affects the HD of KMDoS and NCMDoS in the same way as NUDoS.

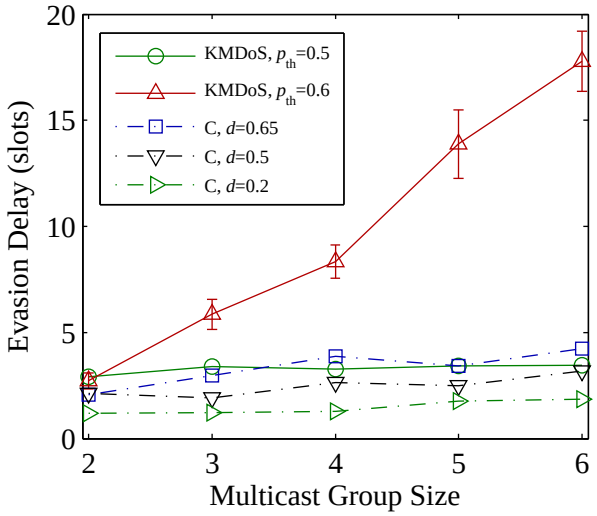

Fig. 15: ED vs. group size for KMDoS ($\rho^{(m)} = 0.4, \mathcal{T}_1^{(m)} = 4$).

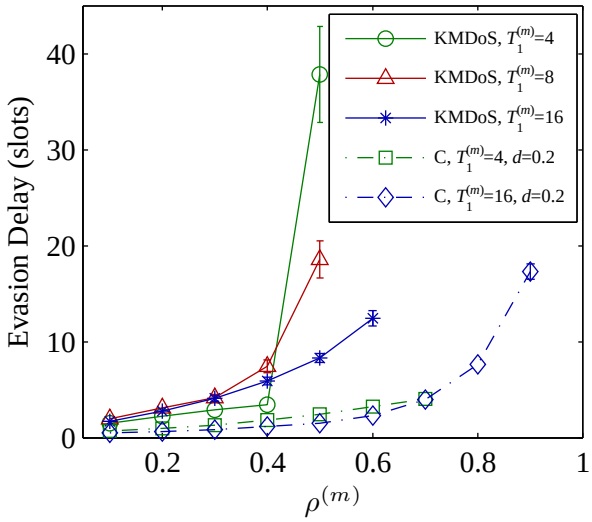

Fig. 16: ED vs. $\rho^{(m)}$ for KMDoS (group size = 6, $p_{th} = 0.5$).

IX. Conclusions

In this paper, we designed three FH algorithms for establishing unicast (NUDoS) as well as multicast (KMDoS and NCMDoS) communications in the presence of a control channel DoS attack. KMDoS and NCMDoS maintain the multicast consistency, and provide different tradeoffs between speed and robustness to node compromise. Our algorithms are distributed, do not incur additional message exchange overhead, and can work in the absence of node synchronization. We simulated our algorithms under a realistic setting of spectrum heterogeneity and lack of synchronization. The effects of different system parameters

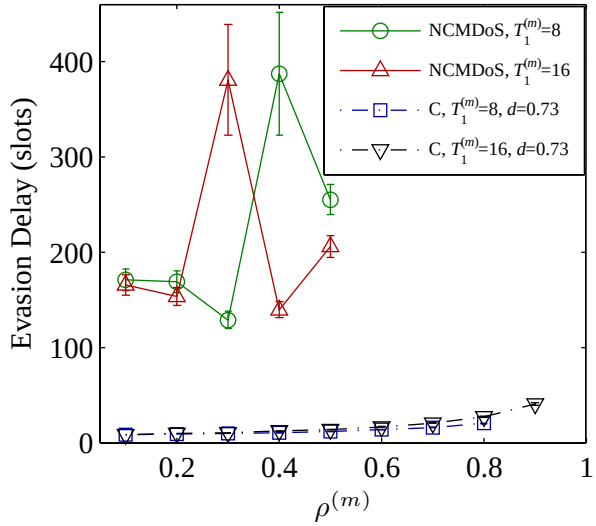

Fig. 17: ED vs. $\rho^{(m)}$ for NCMDoS (group size = 3, $p_{th} = 0.5$).

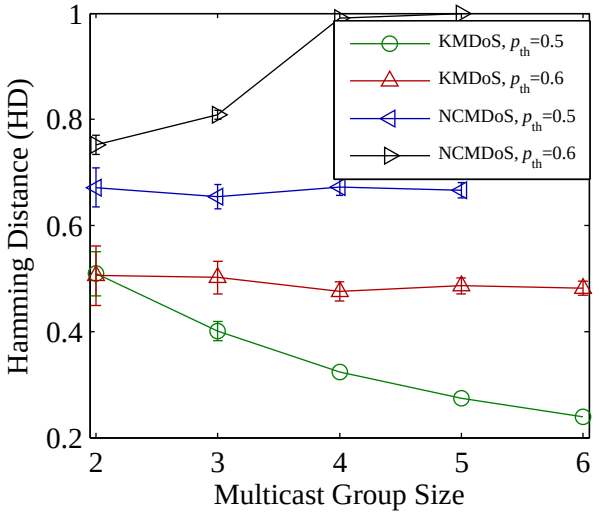

Fig. 18: HD vs. group size for KMDoS and NCMDoS ($\rho^{(m)} = 0.4, \mathcal{T}_1^{(m)} = 4$).

were studied. If these parameters are selected appropriately, our algorithms were found to perform close to the centralized algorithm.

REFERENCES

[1] M. J. Abdel-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. 436–447.

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

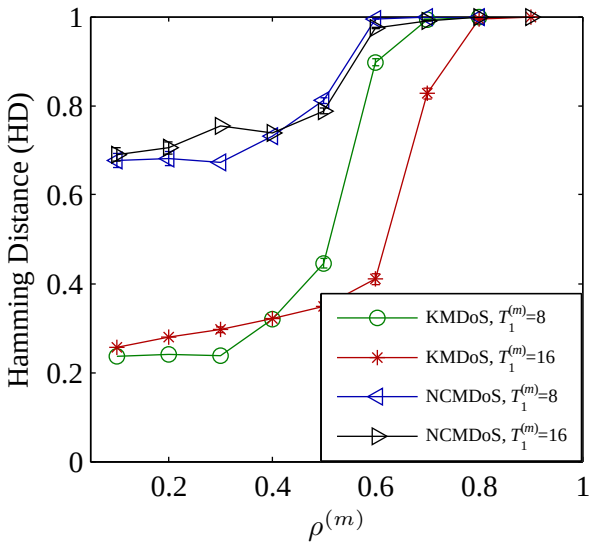

Fig. 19: HD vs. $\rho^{(m)}$ for KMDoS and NCMDoS ($p_{th} = 0.5$).

[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 and J. M. Park, "Asynchronous channel hopping for establishing rendezvous in cognitive radio networks," in *Proc. of the IEEE INFOCOM Mini-Conf.*, 2011.

[5] 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.

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

[7] 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.

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

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

[10] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *Proc. of the ACM WiSec Conf.*, 2009, pp. 169–180.

[11] 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, pp. 2444–2452.

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

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

[14] 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.

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

[16] 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.