Fast Rendezvous for Spectrum-Agile IoT Devices with Limited Channel Hopping Capability

Sigit Aryo Pambudi
North Carolina State University
890 Oval Drive
Raleigh, NC 27606
Email: sapambud@ncsu.edu

Wenye Wang
North Carolina State University
890 Oval Drive
Raleigh, NC 27606
Email: wwang@ncsu.edu

Cliff Wang
Army Research Office
4300 S Miami Blvd
Durham, NC 27703
Email: cliff.wang@us.army.mil

***Abstract*—The explosive number of IoT nodes and adoption of software-defined radio have enabled an efficient method of exploiting idle frequency spectrums called *dynamic spectrum access* (DSA). The foremost problem in DSA is for a pair of nodes to rendezvous and form a control channel prior to communication. Existing schemes require a channel hopping (CH) pattern with length $O(N^2)$, which is overly complex especially when the number of channels N is large. Moreover, the CH patterns are designed assuming DSA nodes have unlimited CH capability, which is hardly satisfied by nodes with long frequency switching time and limited sensing capacity. In this paper, *we design a low-complexity rendezvous scheme that account for CH capability limits*. The CH capability is captured using *spectrum slice graphs* that describe the possible channels for the next hop, given the currently-visited channel. By viewing the CH patterns as random walks over the spectrum graphs, we assign the walks with optimal transition probabilities that achieve the smallest rendezvous delay. The resulting symmetric random CH (S-RCH) scheme, which is suitable for IoT nodes without predetermined roles, achieves a lower rendezvous delay than existing Modular Modified Clock (MMC) scheme and offers more than 80% successful rendezvous in mobile networks.**

I. Introduction

Internet-of-Things (IoT) is an emerging paradigm in which smart objects (e.g., smart phones, wearables, and RFID tags) are interconnected via wireless communications to form an extension to the Internet [1]. By 2020, it is expected that there are about 5 IoT devices near each person, with up to an explosive number of 35 billion devices worldwide [2]. Such a dense network of IoT devices will require the wireless spectrum—that is limited in bandwidth—to be utilized efficiently. To this end, another new paradigm called software-defined radio (SDR) enables wireless devices to adjust their transmit power, carrier frequency, and modulation scheme intelligently. By employing SDR, IoT nodes may act as unlicensed users (i.e., secondary users, SUs) to search for spectrum "holes" and communicate over unused frequency bands, without interfering with licensed users (i.e., primary users, PUs). Such an opportunistic way of efficiently accessing wireless channels is referred as *dynamic spectrum access* (DSA) [3].

In networks with DSA, one of the most challenging aspect, especially in fully distributed settings, is to establish a common channel prior to communication. A common method of providing control channel is to assign one or more dedicated channels for all nodes [3]. However, the IoT nodes may occupy their own locations and have different sets of available channels, due to the distinct channel occupation pattern of nearby PUs. Moreover, a PU can initiate a wireless transmission over the control channel(s). As these channels are blocked, IoT nodes cannot perform any data transmission, although other channels may still be unoccupied. Finally, allowing multiple nodes to use one or few control channels may introduce bottleneck, which is intensified in high-density settings like IoT.

To relax the diverse availability, channel blocking, and bottleneck problems, control channel establishment in DSA networks typically relies on channel hopping (CH) [4], [5]. A CH scheme, programmed into every IoT node's SDR, evenly divides the time into slots and then generates a *CH pattern*, that is, the sequence of channels that the associated node has to hop into at each time slot. It is the CH scheme's duty to ensure that any pair of nodes can *rendezvous* [4], i.e., to hop to a common channel for control channel establishment.

The majority of CH schemes in the literature are *orthogonal sequence*-based, where each node employs a fixed-length CH pattern that is repeated indefinitely, or at least until a rendezvous is achieved [5], [6]. Early schemes had considered the *asymmetric* case, in which one SU identified as a sender and another SU acting as a receiver employ two different CH patterns, respectively [7]. These CH schemes have also been applied to *heterogeneous* networks where the SUs have different ranges of observable channels [4], [8]. More recent studies, on the other hand, considered the *symmetric* case where all SUs employ identical hopping patterns irrespective of their roles [9]. Unfortunately, the aforementioned CH schemes impose a strong assumption in designing their CH pattern: SUs must be able to hop to any arbitrary channel, despite the vast bandwidth that their SDRs have to cover.

In DSA networks, the large spectrum bandwidth imposes practical CH limits to SU nodes. Specifically, modern standards—such as LTE over unlicensed band (LTE-U) [10]—provides up to 700MHz of spectrum bandwidth, while state-of-art SDR like Ettus E310 only has a digital bandwidth of $BW_{SDR} = 56$MHz. Hopping to an arbitrary channel spaced larger than BW_{SDR} apart from the current channel

This work is supported in part by Army Research Office under grant W911NF-15-2-0102 and NSF award CNS-1527696.

requires more than 30ms of switching time [11], which is way beyond the 10ms CH interval specification.[1] Moreover, the IoT nodes should also sense the channels before every new hop, to avoid colliding with a PU transmission in the newly-selected channel. To this end, Nyquist-based sensing [12] can be applied, but limits the range of senseable and candidate channels for the next-hop to $\frac{BW_{SDR}}{2}$ Hz away from the current channel. As a result, the CH pattern becomes restricted by the frequency switching time and channel sensing limits, and we refer to IoT nodes with these restrictions as having a *limited CH capability*. To account for the CH limits, the CH patterns of existing schemes [4]–[9] must be re-designed completely.

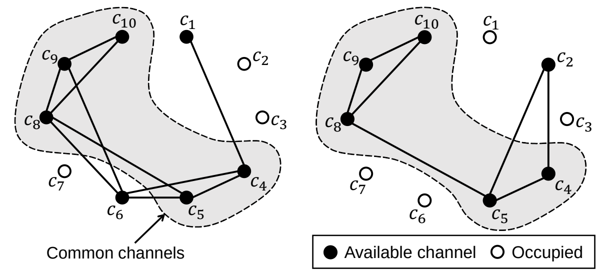
More importantly, existing schemes are not suitable for IoT-based DSA nodes due to the following reason. Existing DSA standards like LTE-U divides the spectrum into up to $N = 7000$ channels.[2] Under large N, the existing orthogonal sequence-based schemes [4]–[9], which are known to require designing and storing CH pattern with $O(N^2)$ length, become overly complex, especially for low-complexity IoT nodes.

Motivated by the lack of applicable schemes for IoT-based DSA networks, we design graph-based CH schemes for rendezvous between IoT nodes. Specifically, we build *spectrum slice graphs* to capture the limited CH capability of IoT nodes, and then treat the proposed CH schemes as random walks over the graphs. Then, the random walks are configured so that the corresponding CH schemes achieve quick rendezvous irrespective of the choice of the initial channel hop, by minimizing a *maximum expected TTR* (MTTR) metric. Our major contributions are outlined as follows.

1) We propose spectrum slice graphs to capture the possible CH sequences of a pair of IoT nodes, SU_a and SU_b, with limited CH capability. As exemplified by the spectrum slice graph of SU_a in Fig. 1(a), the vertices (nodes) represent the set of available channels while their neighbors are the possible channels for the next hop.
2) To account for IoT nodes that do not have distinct predetermined roles, we propose a symmetric random CH (S-RCH) scheme where SU_a and SU_b symmetrically walk over their respective spectrum slice graph. Unlike existing sequence-based schemes that require long CH patterns [4]–[9], S-RCH provides a low-complexity, memoryless way of generating CH sequences on-the-fly. Then, we improve the performance of S-RCH by allowing the SU nodes to detect rendezvous on different but neighboring channels (vertices) of the spectrum slice graph.
3) We perform numerical evaluations and show that the proposed S-RCH scheme has a lower TTR than the R-RCH scheme, both of which outperform the state-of-art scheme [13]. Finally, we show that the proposed schemes provide at least 80% successful rendezvous under realistic node mobility based on traces [14].

[1]Each radio frame has a length of $\delta = 10$ms, according to LTE-U [10]. Similarly, IEEE 802.22 specifies a frame length of $\delta = 10$ms.

[2]Existing LTE-U standard divides the U-NII bands into 20Mhz channels, and reserves the 100kHz channel raster for future uses. We envision the smaller raster size will enable a fine-grained and efficient channel allocation for dense IoT-based DSA networks, which could be a norm in the future.



(a) Graph $\mathcal{G}_a$ for node SU_a. (b) Graph $\mathcal{G}_b$ for node SU_b.

Fig. 1. Illustration of spectrum slice graphs.

The rest of this paper is organized as follows. Section II describes the network model and problem formulation. The S-RCH scheme is presented in Section III and analyzed in Section IV. Finally, Section V concludes this paper.

II. Models and Problem Formulation

In this section, we explain the major challenges in achieving rendezvous between nodes with limited channel hopping capability. Then, we outline a graph-based approach to accomplish rendezvous. Finally, we state our main research problem.

A. Network Model

We consider a DSA network operating over an $(N \times B_0)$ Hz spectrum that is divided into a set $\mathcal{C} = \{c_0, \ldots, c_{N-1}\}$ of non-overlapping channels, where c_i denotes the i-th channel, B_0 is the bandwidth of each channel, and N is the total number of channels. The DSA network consists of PUs as the high-priority users and IoT nodes as the lower-priority SUs. Each SU is equipped with half-duplex SDR that can be tuned to any of the N channels. The SUs are also capable of performing spectrum sensing, to obtain their sets of available channels that are temporally unoccupied by co-located PUs. The channel occupation status is assumed to be slowly-dynamic so that the DSA network is mildly time-varying.

B. Channel Hopping for Rendezvous

We focus on the pairwise *rendezvous* problem, in which two IoT nodes—respectively denoted as SU_a and SU_b—want to establish a mutual control channel before data transmission. However, this work can be extended to a multi-user setting by employing message passing scheme [15]. The time is assumed to be slotted and each slot—denoted as a non-negative integer t—has a uniform span of δ.[1] Let $\mathcal{C}_a \subset \mathcal{C}$ and $\mathcal{C}_b \subset \mathcal{C}$ be the sets of channels available to nodes SU_a and SU_b, respectively. The sets $\mathcal{C}_a$ and $\mathcal{C}_b$ are not necessarily identical to each other (i.e., *heterogeneous* [4], [8]) since SU_a and SU_b may be placed at different locations and surrounded by distinct sets of PU neighbors and channel occupation status. However, we assume that $\mathcal{C}_a \cap \mathcal{C}_b \neq \emptyset$ to ensure that nodes SU_a and SU_b can still find a common channel for rendezvous.

In DSA networks, rendezvous is commonly achieved via *channel hopping* (CH), which is a process describable as follows. Let $X_a(t) \in \mathcal{C}_a$ be the channel that the IoT node SU_a hops into at time slot t, while the collection $\{X_a(t) : t \in \mathbb{N}_0\}$ denotes the *CH pattern* of SU_a. Similar denotations exist for the other IoT node SU_b by exchanging $(\cdot)_a$ with subscript $(\cdot)_b$. Then, two SU nodes achieve rendezvous when they both hop to the same channel at the same time slot (i.e., $X_a(t) = X_b(t)$).

C. Limits on Channel Hopping Capability

IoT-based DSA networks will operate over a very wide frequency spectrum. For example, TV white space in Kansas, USA provides up to 204MHz of bandwidth [16], while LTE-U provides a 5.15 to 5.85 GHz unlicensed spectrum [10]. In contrast to such a wide spectrum, IoT nodes have limited hardware specifications, which leads to the following challenges.

1) Large frequency tuning time: Consider the CH pattern of node SU_a, which has an SDR with an associated bandwidth of BW_{SDR} Hz. When SU_a wants to hop from channel $X_a(t) = c_i$ that has a center frequency f_i to a new channel $X_a(t+1) = c_j$ with frequency f_j, it performs one of the following steps.

- The node can do a *one-stage tuning* [17], by digitally shifting the current center frequency f_i to a new one f_j, as long as these two frequencies are within the SDR bandwidth. Specifically, the indices of channels c_i and c_j must be within a *maximum CH range* of

$$R := \lfloor (BW_{SDR} - B_0)/2B_0 \rfloor . \tag{1}$$

  For example, see Fig. 2(a) for a one-stage tuning from f_i to $f_j = f_{j_1}$. In state-of-art SDRs, one-stage tuning takes up to 3ms [17], while low-power radios for mobile can take longer. The division by 2 ensures that SU_a can hop to both the left and right sides of the current frequency.
- Otherwise, the new channel c_j is outside of the maximum CH range (i.e., see Fig. 2(a) when $f_j = f_{j_2}$) and SU_a needs to perform a *two-stage tuning* [17]. Specifically, SU_a' local oscillator (LO) frequency must be tuned to near the target f_j and digital shifting is applied to bring the carrier frequency to f_j. As a result, two-stage tuning involves re-setting and waiting for the LO to stabilize. This can take up to 31ms [11], which is way beyond the $\delta = 10$ms time slot period.[1]

To keep the frequency tuning within $\delta = 10$ms, it is desirable to do a one-stage tuning.

2) Limited channel sensing capability: Before hopping to a new channel c_j, an IoT node must ensure that c_j is unoccupied to avoid interference with an ongoing PU transmission. Under the lack of spectrum database [16], especially when the DSA network operates over unlicensed bands [10], channel occupancy status must be checked via sensing. To ensure a quick sensing, IoT nodes with limited SDR bandwidth may only perform a one-time partial-band Nyquist sampling (PBNS) [12], which senses a part of the spectrum but ignores the remaining parts.

To capture how PBNS limits CH pattern, we denote

$$L := \lfloor BW_{SDR}/2B_0 \rfloor \tag{2}$$

as the number of *senseable* channels. The factor 2 in the denominator is because a perfect reconstruction of a signal with bandwidth B_0 needs to be sampled at a Nyquist rate of at least $2B_0$. An example of the set of channels available to node SU_a after a PBNS with $L = 7$ is depicted in Fig. 2(b). The available channels (i.e., $\{c_4, c_5, c_6, c_8\}$) then serve as the candidates for the next hop.

3) Combined Channel Hopping Capability Limits: The frequency tuning delay and channel sensing capacity not only co-exist, but also impose a *combined CH range* of

$$R_{ch} := \min (R, \lfloor (L-1)/2 \rfloor) . \tag{3}$$

To be specific, an IoT node that currently visits channel c_i may hop to another channel c_j if and only if the CH range restriction $|i-j| \le R_{ch}$ is satisfied. In this paper, all IoT nodes are assumed to have identical CH ranges, and we refer the nodes with such a restriction to have a *limited CH capability*.

D. Graph-Based Rendezvous Under Channel Hopping Limits

Limited CH capability reduces the set of available channels for the next hop, a key ingredient that determines when and where rendezvous between SU_a and SU_b occurs. To describe the reduced set of next-hop channels, we define the notion of *spectrum slice graphs* as follows.

Definition 1 (Spectrum slice graphs)**.** *Let $x \in \{a, b\}$ be an index and vertices $\mathcal{C}_x$ represent the set of channels available to node SU_x. Let $\mathcal{E}_x$ be the set of edges, where edge $e(c_i, c_j)$ is in $\mathcal{E}_x$ if $c_i, c_j \in \mathcal{C}_x$ and $|i - j| \le R_{ch}$. Then, the spectrum slice graph of node SU_x is defined as $\mathcal{G}_x := (\mathcal{C}_x, \mathcal{E}_x)$.*

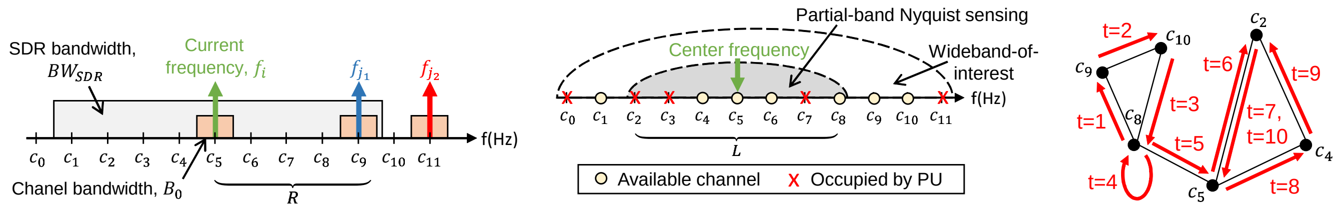
An example of a spectrum slice graph $\mathcal{G}_a$ for node SU_a is depicted in Fig. 1(a). The set of vertices and edges in Fig. 1(a) capture the set of possible next-hop channels in relation to the currently-visited channels, which is described by the combined CH range in (3). For instance, SU_a that is currently at channel c_5 may only choose the neighbors c_4, c_5, c_6, or c_8 for the next time slot.[3] In addition to $\mathcal{G}_a$, node SU_b also has a corresponding spectrum slice graph $\mathcal{G}_b$ (see Fig. 1(b) for an example), which generally have different vertices and edges due to the *heterogeneous* channel assumption [4], [8].

The spectrum slice graphs are then used by the SUs to determine their respective CH patterns.

Definition 2 (CH pattern)**.** *Let $c_{i_t} := X_x(t) \in \mathcal{C}_x$ be the channel visited by node SU_x ($x \in \{a, b\}$) at time t and $\mathcal{N}_x(c_{i_t}) := \{c : e(c_{i_t}, c) \in \mathcal{E}_x\}$ be the neighbors of vertex c_t in the spectrum slice graph $\mathcal{G}_x$. Given c_{i_t} and the CH capability limit of SU_x, the next-hop channel $c_{i_{t+1}}$ must be selected among $\mathcal{N}_x(c_{i_t}) \cup \{c_{i_t}\}$. Then, CH pattern of SU_x is defined as a collection $W_x := \{X_x(t) : t \in \mathbb{N}_0\}$.*

In Fig. 2(c), we illustrate a CH pattern of node SU_b that starts by visiting channel $X_b(0) = c_8$. Channel (vertex) c_8 is connected via edges to vertices $\mathcal{N}_b(c_8) = \{c_5, c_9, c_{10}\}$ and the

[3]Without loss of generality, we allow the node to re-choose the existing channel c_5 for the next hop.



(a) Frequency shifting limit. (b) Channel sensing limit. (c) CH pattern of SU_b over $\mathcal{G}_b$.

Fig. 2. Illustration of limited channel chopping capability in DSA networks.

next-hop channel must be selected among $\mathcal{N}_b(c_8) \cup \{c_8\}$. In this example, SU_b decides to hop to $X_b(1) = c_9$ at the next time slot, $t = 1$. By repeating the aforementioned process, we obtain a CH sequence of $W_b = \{c_8, c_9, c_{10}, c_8, c_8, \dots\}$, as depicted by the red arrows in Fig. 2(c).

The path formed by the CH sequence in Fig. 2(c) is essentially a walk on graph $\mathcal{G}_b$. We assume that the next-hop channel is selected randomly according to a pre-determined probability distribution, such that the CH sequence becomes a *random walk* [18] over a spectrum slice graph.

With Definition 2, the rendezvous problem between nodes with limited CH capability can be formulated as follows.

Definition 3. *A rendezvous is achieved if there exists* $c^* \in \mathcal{C}_a \cap \mathcal{C}_b$ *and* $t^* \in \mathbb{N}_0$ *such that* $X_a(t) = X_b(t) = c^*$.

In other words, rendezvous occurs when the random walks performed by SU_a and SU_b over their respective spectrum slice graphs both visit the same channel (vertex).

E. Problem Formulation

The primary goal behind rendezvous is to quickly find a common control channel between two DSA nodes in a distributive manner. Thus, delay performance is of critical importance, and we aim to minimize the delay until rendezvous. To achieve our minimization goal, we employ the notion of *time-to-rendezvous*

$$TTR := \min_{t \geq 0} \{t : X_a(t) = X_b(t) = c^*, c^* \in \mathcal{C}_a \cap \mathcal{C}_b\} \quad (4)$$

from [4], and then use TTR to define the following metric.

Definition 4. *The maximum expected time-to-rendezvous (MTTR) with respect to all possible initial channel hops,* $X_a(0)$ *and* $X_b(0)$*, is defined as*

$$TTR_{max} := \sup_{X_a(0), X_b(0)} \mathbb{E}(TTR). \quad (5)$$

When MTTR is minimized, nodes can establish a control channel and start data transmission immediately, which leads to short medium access delay and high user satisfaction. Unlike existing CH schemes for nodes with ideal capabilities [4]–[9], minimizing MTTR between IoT nodes with limited CH ability remains a wide-open problem. Hence, we ask: *How to design the CH patterns of IoT nodes with limited CH capability for achieving a fast MTTR?*

III. Symmetric Random Channel Hopping (S-RCH)

In this section, we first outline a graph-based symmetric random channel hopping (S-RCH) scheme and then show how to configure S-RCH for quick rendezvous. Further, we leverage channel sensing to achieve a quicker rendezvous.

A. Preliminaries on Graph-Based S-RCH Scheme

IoT-based DSA networks are envisioned to support many types of applications, including wireless sensor networking, vehicle-to-vehicle communications, and mesh-based online messaging [19]. In these applications, IoT nodes communicate with each other in a peer-to-peer manner, without pre-determined hierarchical (i.e., master-slave) roles. Thus, it is desirable that the nodes have totally symmetric schemes, including for rendezvous [4]–[9]. Unfortunately, the existing symmetric rendezvous schemes ideally assume that nodes are able to hop to any arbitrary channel, instead of taking into account their limited CH capability.

To facilitate rendezvous between a pair of nodes SU_a and SU_b with limited CH capability, we apply a graph-based approach. Specifically, we let SU_a and SU_b apply the CH pattern in Definition 2, which result in walks W_a and W_b, respectively. Given the current channel $X_a(t) = c_i$, node SU_a randomly selects the next-hop channel $X_a(t+1) = c_j$ according to a matrix $\mathbf{P}_a$, whose (i,j)-th element is equal to the *transition probability*

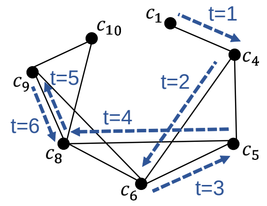
$$P_a(c_i, c_j) := \Pr\{X_a(t+1) = c_j | X_a(t) = c_i\}. \quad (6)$$

Similar denotations also exist for SU_b by exchanging $()_a$ with subscript $()_b$. Although the elements of $\mathbf{P}_a$ and $\mathbf{P}_b$ are not necessarily equal, in Section III-B we will apply an identical (symmetric) policy for calculating their values. As a result, matrices $\mathbf{P}_a$ and $\mathbf{P}_b$ and their corresponding CH patterns are statistically symmetric. The resulting scheme is referred as a *symmetric random channel hopping* (S-RCH).

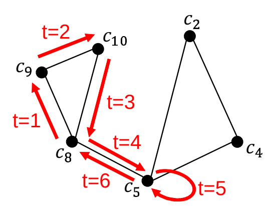
Fig. 3 exemplifies an S-RCH corresponding to the spectrum slice graphs in Figs. 1(a) and 1(b). The dotted lines in Fig. 3(a) illustrates walk W_a, the solid lines in Fig. 3(b) represents W_b, while their resulting CH patterns are depicted in Fig. 3(c). As indicated by the figure, W_a and W_b are continued until rendezvous is achieved on channel c_5 at $t = 6$.

B. Achieving Quick Rendezvous

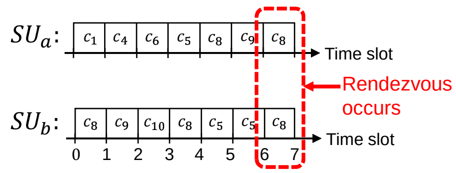
An immediate question is how to achieve our main goal of minimizing MTTR. According to the illustration in Fig. 3,



(a) Walk W_a on graph $\mathcal{G}_a$.



(b) Walk W_b on graph $\mathcal{G}_b$.



(c) CH pattern versus time.

Fig. 3. Illustration of symmetric random channel hopping (S-RCH) scheme.

the TTR (equivalently, MTTR) of S-RCH scheme is fully determined by walks W_a and W_b. Moreover, the vertices (channels) are selected at every hop by the walks according to their transition probability matrices. As a result, the walks are random and their characteristics, including the resulting MTTR, are controllable by setting the transition probabilities.

In the following, we assign the transition probabilities of random walks W_a and W_b to minimize MTTR. Our approach is to first show that MTTR is an increasing function of the transition probability matrices' eigenvalues. Then, we use the eigenvalues and formulate a convex optimization problem to obtain the optimal transition probabilities. The main result of our approach is outlined as follows.

Theorem 1. *The transition probabilities of walks W_a and W_b that achieve the smallest MTTR satisfy*

$$P_a(c_i, c_j) = \begin{cases} \frac{1}{d_a(c_i)+1}, & \text{if } e(c_i, c_j) \in \mathcal{E}_a \text{ or } c_i = c_j, \\ 0, & \text{otherwise}, \end{cases} \quad (7)$$

$$P_b(c_i', c_j') = \begin{cases} \frac{1}{d_b(c_i')+1}, & \text{if } e(c_i', c_j') \in \mathcal{E}_a \text{ or } c_i' = c_j', \\ 0, & \text{otherwise}, \end{cases} \quad (8)$$

$\forall c_i, c_j \in \mathcal{C}_a$ and $\forall c_i', c_j' \in \mathcal{C}_b$, where $d_a(c_i) := |\mathcal{N}_a(c_i)|$ and $d_b(c_i') := |\mathcal{N}_b(c_i')|$ are the degrees of vertices c_i and c_i'.

Before proving Theorem 1, we present several denotations. Let $\boldsymbol{\pi}_a := \{\pi_a(c_i)\}_{c_i \in \mathcal{C}_a}$ be a *stationary distribution* vector that satisfies $\boldsymbol{\pi}_a \mathbf{P}_a = \boldsymbol{\pi}_a$ and $\boldsymbol{\pi}_a \mathbf{1} = \mathbf{1}$. To this end, $\pi_a(c_i) \in (0, 1)$ is the unconditional likelihood that W_a will be at channel c_i after a sufficiently large number of hops. We assume that similar denotations exist for random walk W_b.

Moreover, let $P^t(c_i, c_j)$ be the (i, j)-th element of matrix $\mathbf{P}^t$, where $\mathbf{P}^t = \mathbf{P}^{t-1} \cdot \mathbf{P}$ is the t^{th} power of matrix $\mathbf{P}$. Then, we have some general properties of random walks as follows.

Definition 5. *A random walk W over graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$ with transition probability matrix $\mathbf{P}$ and stationary distribution vector $\boldsymbol{\pi}$ is*

1) *reversible if $\pi_{c_i} P(c_i, c_j) = \pi_{c_j} P(c_j, c_i), \forall c_i, c_j \in \mathcal{C}$,*
2) *irreducible if $\exists t < \infty$ such that $P^t(c_i, c_j) > 0$ for any $c_i, c_j \in \mathcal{C}$, and*
3) *aperiodic if $\exists N < \infty$ such that $\forall c_i \in \mathcal{C}, P^n(c_i, c_i) > 0$ for all $n \geq N$.*

Definition 5 can be applied to our walks-of-interest, W_a and W_b, by adding subscripts $()_a$ and $()_b$, respectively. To this end, the second and third conditions in Definition 5 are satisfied by W_a and W_b since their respective spectrum slice graphs are connected and non-bipartite.[4] On the other hand, many variants of random walk over graph can be designed to satisfy the first condition. In fact, we can show that W_a and W_b employing the transition probabilities in (7)-(8) have the stationary distributions of

$$\pi_a(c_i) = \frac{d_a(c_i) + 1}{2|\mathcal{E}_a| + |\mathcal{C}_a|} \text{ and } \pi_b(c_i') = \frac{d_b(c_i') + 1}{2|\mathcal{E}_b| + |\mathcal{C}_b|}. \quad (9)$$

Thus, it is reasonable to assume that W_a and W_b satisfy all the three conditions in Definition 5.

To proceed, we consider W_a and W_b as two parallel but independent walks and state their current locations as a two-dimensional state $(c_i, c_i') := (X_a(t) = c_i, X_b(t) = c_i')$. Furthermore, by denoting (c_j, c_j') as the state at the next time slot, the transition probabilities of W_a and W_b can be combined into a two-variate transition probability

$$P_{ab}((c_i, c_i'), (c_j, c_j')) = P_a(c_i, c_j) \times P_b(c_i', c_j'). \quad (10)$$

Subscript $()_{ab}$ indicates that P_{ab} corresponds to the parallel walks. Let $n_a := |\mathcal{C}_a|$ and $n_b := |\mathcal{C}_b|$ be the number of available channels at nodes SU_a and SU_b, respectively. Then, we can state the $(n_a n_b) \times (n_a n_b)$ transition probability matrix of the parallel walks as $\mathbf{P}_{ab} = \mathbf{P}_a \otimes \mathbf{P}_b$, where $\otimes$ denotes the Kronecker matrix product operator.

Further, we study several properties regarding the parallel walks, which for simplicity is denoted as W_{ab}. Denote $\boldsymbol{\pi}_{ab}$ as the $(n_a n_b) \times 1$ stationary distribution vector corresponding to the transition probability matrix $\mathbf{P}_{ab}$. Let the mixing time of the parallel walks W_{ab} be denoted as

$$\tau_{mix}^{ab}(\epsilon) := \inf \left\{ t : \max_{c_i \in \mathcal{C}_a, c_i' \in \mathcal{C}_b} \left| (P^t((c_i, c_i'), (\cdot)) - \pi_{(\cdot)}) \right| \leq \epsilon \right\}.$$

The mixing time is the amount of steps required until the transition probability of the walk W_{ab} is close enough (within a factor ϵ) to its stationary distribution. Then, we have the following lemma.

Lemma 1. *The S-RCH scheme has an MTTR that satisfies*

$$TTR_{max} \geq 1/2 \times \times \tau_{mix}^{ab}(1/4) - 1/2. \quad (11)$$

Proof: We start by finding the upper bound of the right-hand side. Specifically, we adopt the *maximum hitting time* $t_{Hit} := \max_{c_i, c_j \in \mathcal{C}_a, c_i', c_j' \in \mathcal{C}_b} \mathbb{E}_{(c_i, c_i')}[\tau_{(c_j, c_j')}]$ from [18], where the *first hitting time*

$$\tau_{(c_j, c_j')} := \min\{t \geq 0 : X_a(t) = c_j, X_b(t) = c_j'\} \quad (12)$$

[4] A graph is *bipartite* if the vertices can be divided into two disjoint sets and no two vertices in the same set are adjacent.

denotes the first time state $(X_a(t), X_b(t)) = (c_j, c'_j)$ is visited by the walk. We know from [18, Theorem 10.14] that the maximum hitting time is bounded by

$$t_{Hit} \geq 1/2 \times \tau_{mix}^{ab}(1/4) - 1/2. \tag{13}$$

With (13) obtained, the remaining task is to relate t_{Hit} with MTTR. Let the *largest first hitting time* be denoted as $\tau_{(c_j,c'_j)}^{max} := \max_{c_j\in\mathcal{C}_a, c'_j\in\mathcal{C}_b} \tau_{\{c_j,c'_j\}}$. Then, we observe that MTTR is lower-bounded by

$$TTR_{max} \geq \tau_{(c_j,c'_j)}^{max} \geq t_{Hit}. \tag{14}$$

The first inequality on the left side holds because MTTR requires an additional condition $X_a(t) = X_b(t) = c^*$ to be satisfied (see (4)-(5)) and is more restricted that the largest first hitting time. The second inequality in (14), on the other hand, holds by the relation between the maximum and expected values. Finally, combining (14) with (13) completes the proof of Lemma 1. ■

Lemma 1 reveals that MTTR is lower-bounded by a factor proportional to the mixing time $\tau_{mix}^{ab}(\epsilon)$, while minimizing the best-case MTTR is equivalent to minimizing $\tau_{mix}^{ab}(\epsilon)$. Thus, we will optimize $\tau_{mix}^{ab}(\epsilon)$ to design the transition probability matrix $\mathbf{P}_{ab}$. Before proceeding, we outline several useful properties as follows (proofs omitted due to page limit).

Lemma 2. *Parallel walks W_{ab} is reversible, irreducible, and aperiodic. Moreover, W_{ab} has a unique stationary distribution vector of $\boldsymbol{\pi}_{ab} = \boldsymbol{\pi}_a \otimes \boldsymbol{\pi}_b$.*

Now we are ready to derive the mixing time upper and lower bounds that will be used for minimizing the MTTR in Lemma 1. Let

$$\lambda_a^* := \max\{\lambda_2^a, -\lambda_n^a\} \tag{15}$$

be the *second largest eigenvalue magnitude* (SLEM) [20] associated with walk W_a, where λ_i^a is the i-th largest eigenvalues of $\mathbf{P}_a$. Moreover, let $\pi_a^* := \min_{c_i\in\mathcal{C}_a} \pi_{c_i}^a$ be the smallest element of the stationary distribution vector $\boldsymbol{\pi}_a$. Similar notations also exist for walk W_b by substituting a with symbol b.

Proposition 1. *The mixing time of parallel walks W_{ab} is lower and upper bounded by*

$$\left(\frac{1}{1-\lambda_a^*\lambda_b^*} - 1\right)\log\left(\frac{1}{2\epsilon}\right) \leq \tau_{mix}^{ab}(\epsilon) \leq \frac{1}{1-\lambda_a^*\lambda_b^*}\log\left(\frac{1}{\epsilon\times\pi_a^*\pi_b^*}\right). \tag{16}$$

Proof: From Lemma 2, we know that parallel walks W_{ab} is reversible, irreducible, and periodic. Thus, according to [18], W_{ab} has the mixing time bounded by

$$\left(\frac{1}{1-\lambda^*(\mathbf{P}_{ab})} - 1\right)\times\log(\frac{1}{2\epsilon}) \leq \tau_{mix}^{ab}(\epsilon) \leq \frac{1}{1-\lambda^*(\mathbf{P}_{ab})}\times\log\left(\frac{1}{\epsilon\times\pi_{min}(\mathbf{P}_{ab})}\right), \tag{17}$$

where $\lambda^*(\mathbf{P}_{ab})$ and $\pi_{min}(\mathbf{P}_{ab})$ are respectively the SLEM and the smallest entry of $\boldsymbol{\pi}_{ab}$. First, we focus on the common term, $\frac{1}{1-\lambda^*(\mathbf{P}_{ab})}$. According to [21, Theorem 4.2.12], we know that $\lambda_i^a\times\lambda_j^b$ for any $i\leq n_a$ and $j\leq n_b$ is also an eigenvalue of $\mathbf{P}_{ab}$, such that the denominator can be re-stated as $1-\lambda^*(\mathbf{P}_{ab}) = \inf_{2\leq i\leq n_a, 2\leq j\leq n_b} 1 - |\lambda_i^a||\lambda_j^b|$.

Next, we consider the $\frac{1}{\epsilon\times\pi_a\pi_b}$ term in the upper bound. According to Lemma 2, $\pi_{c_i}^a \times \pi_{c'_i}^b$ for any $c_i\in\mathcal{C}_a$ and $c'_i\in\mathcal{C}_b$ is also a stationary distribution corresponding to $\mathbf{P}_{ab}$, and $\boldsymbol{\pi}_{ab} = \boldsymbol{\pi}_a\otimes\boldsymbol{\pi}_b$ is unique. Thus, the logarithm term in the upper bound can be obtained by taking the minimum over all possible $c_i\in\mathcal{C}_a$ and $c'_i\in\mathcal{C}_b$, completing the proof. ■

With Proposition 1 and Lemma 1 at hand, we are now ready to prove Theorem 1 as follows.

Proof of Theorem 1: Lemma 1 reveals that minimizing MTTR is equivalent to minimizing the mixing time $\tau_{mix}^{ab}(1/4)$, which according to Proposition 1 is proportional to $\lambda_a^*\times\lambda_b^*$. Note that λ_a^* and λ_b^* are the SLEMs contributed by walks W_a and W_b, respectively, while the walks are operated independently to each other. As a result, minimizing $\lambda_a^*\times\lambda_b^*$ can be done by minimizing λ_a^* and λ_b^* separately.

We start by minimizing the SLEM λ_a^* of random walk W_a, which has a transition probability matrix of $\mathbf{P}_a$. Since $\mathbf{P}_a$ is a stochastic matrix, it has a largest eigenvalue of $\lambda_1 = 1$ [18, Sec. 12.2]. This implies the other eigenvalues are smaller or equal to one, so that the SLEM becomes $\lambda_a^*\leq 1$. Let $\mathbf{I}$ be an identity matrix. When $\mathbf{P}_a$ is projected to the null space of $\mathbf{1}$, that is, by using a projection function $(\mathbf{I}-\frac{1}{n_a}\mathbf{1}\mathbf{1}^T)\mathbf{P}_a(\mathbf{I}-\frac{1}{n_a}\mathbf{1}\mathbf{1}^T)$, the largest eigenvalue magnitude is equal to the SLEM. Moreover, since the *spectral norm* $||\mathbf{A}||_2 := \sup_{||\mathbf{x}||\geq 0}\frac{||\mathbf{A}\mathbf{x}||_2}{||\mathbf{x}||_2}$ is equal to the largest eigenvalue magnitude of $\mathbf{A}$, then the projection $||(\mathbf{I}-\frac{1}{n_a}\mathbf{1}\mathbf{1}^T)\mathbf{P}_a(\mathbf{I}-\frac{1}{n_a}\mathbf{1}\mathbf{1}^T)||_2 = ||\mathbf{P}_a - \frac{1}{n_a}\mathbf{1}\mathbf{1}^T||_2$ becomes equal to the SLEM. As a result, the problem of minimizing the mixing time of W_a, which is equivalent to minimizing λ_a^*, can be re-stated as

$$\min_{\mathbf{P}_a}\left|\left|\mathbf{P}_a - \frac{1}{n_a}\mathbf{1}\mathbf{1}^T\right|\right|_2 \text{ s.t. } \mathbf{P}_a\mathbf{1} = \mathbf{1},\ \mathbf{P}_a\geq\mathbf{0}, \quad \text{and } P_a(c_i,c_j) = 0, \forall c_i,c_j : e(c_i,c_j)\notin\mathcal{E}_a \text{ and } c_i\neq c_j. \tag{18}$$

The inequality $\mathbf{P}_a\geq\mathbf{0}$ indicates that the transition probability matrix has non-negative elements, while the last constraint ensures that the random walk is only performed over the edges of the spectrum slice graph $\mathcal{G}_a$.

Solving (18) requires the full knowledge regarding the set of edges $\mathcal{E}_a$ for assigning the elements of $\mathbf{P}_a$ all at once [20], which is not available due to the limited sensing capacity of the employed PBNS algorithm. When SU_a is at channel c_i, however, it knows the immediate neighbors of c_i, which can be used for determining the i^{th} row of $\mathbf{P}_a$. Let $p_j := P_a(c_i,c_j)$ and $\mathbf{p}_a(c_i) := \{p_j\}_{c_j\in\mathcal{C}_a}$ for notation simplicity. Notice that the spectral norm is upper-bounded by the *Frobenius norm* $||\mathbf{A}||_F := (\sum_{i,j}A_{i,j}^2)^{\frac{1}{2}}$, where $A_{i,j}$ denotes the (i,j)-th element of matrix $\mathbf{A}$. As a result, (18) can be restated into the problem of assigning the u^{th} row of $\mathbf{P}_a$ as follows.

$$\min\left|\left|\mathbf{p}_a(c_i) - \frac{1}{n_a}\mathbf{1}\right|\right|_F \text{ s.t. } \mathbf{p}_a(c_i)\mathbf{1} = 1, \tag{19}$$

$$\text{and } P_a(c_i, c_j) \begin{cases} = 0, & \text{if } e(c_i, c_j) \notin \mathcal{E}_a \text{ and } c_i \neq c_j, \\ \geq 0, & \text{otherwise.} \end{cases}$$

The optimal solution to (19) can be found as follows. Let γ be the Lagrange multiplier [22] and $\hat{\mathcal{N}}_a(c_i) = \mathcal{N}_a(c_i) \cup \{c_i\}$. Due to constraint $p_j = 0$ for all $c_j \in \mathcal{C}_b \setminus \hat{\mathcal{N}}_a(c_i)$, the Lagrangian of (19) becomes $L(\mathbf{p}_a, \gamma) = \sum_{c_j \in \hat{\mathcal{N}}_a(c_i)} (p_j - \frac{1}{n_a})^2 + \sum_{c_j \in \mathcal{C} \setminus \hat{\mathcal{N}}_a(c_i)} (0 - \frac{1}{n_a})^2 - \gamma (\sum_{c_j \in \hat{\mathcal{N}}_a(c_i)} p_j - 1)$. To proceed, we take the partial derivatives of the Lagrangian with respect to γ and p_j for all $c_j \in \mathcal{N}_a(c_i)$, to obtain

$$L_\gamma(\mathbf{p}_a, \gamma) = \sum_{c_j \in \hat{\mathcal{N}}_a(c_i)} p_j - 1 = 0, \text{ and} \tag{20}$$

$$L_{p_j}(\mathbf{p}_a, \gamma) = 2\left(p_j - \frac{1}{n_a}\right) - \gamma = 0, \quad \forall c_j \in \hat{\mathcal{N}}_a(c_i). \tag{21}$$

By plugging (21) into (20), we get $\frac{\gamma}{2} + \frac{1}{n_a} = \frac{1}{d_a(c_i)+1}$. Substituting this back to (21) obtains the result in (7).

Finally, by repeating the steps above to walk W_b, we can also get (8), which completes the proof. ■

C. PBNS-Assisted S-RCH for Quicker Rendezvous

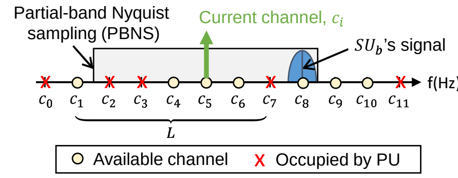
According to Definition 3, rendezvous occurs if walks W_a and W_b both visit the same vertex (channel). In this subsection, we show that the nodes can still achieve rendezvous even if they hop to two different, but neighboring vertices.

Consider an S-RCH scheme whose CH patterns in Fig. 3 is re-drawn into Fig. 4(b). We focus on time slot $t = 3$ when SU_a visits channel c_5 in its spectrum slice graph. Instead of checking SU_b's arrival on vertex c_5 only, SU_a may "query" the neighboring vertices $\mathcal{N}_a(c_5)$ (see the shaded area with dotted boundary in Fig. 4(b)). Vertex c_8 that senses the presence of SU_b "tells" SU_a; then, SU_a directly switch to c_5 for rendezvous. With SU nodes' ability to "query" neighboring channels, rendezvous can be achieved at time t^* if

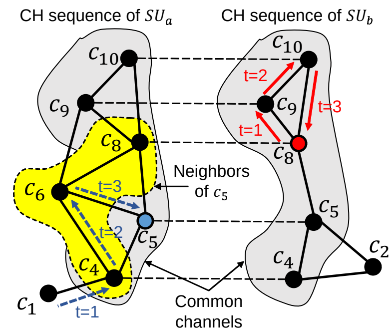
$$X_a(t^*) = c_{i_a}, X_b(t^*) = c_{i_b}, \text{ and } c_{i_a} \in \mathcal{N}_a(c_{i_b}). \tag{22}$$

The question is, how to "query" the neighboring vertices? Consider node SU_a. As depicted in Fig. 4(b), our approach is to apply PBNS with range L centered at SU_a's current channel, $X_a(t) = c_5$. PBNS is able to listen to the occupation of nearby channels, including a rendezvous pilot signal from SU_b at channel c_8, which is equivalent to querying and obtaining feedback from the neighboring vertices. The aforementioned approach can be implemented if SU_b's time slot begins earlier than that of SU_a; otherwise, SU_a must transmit pilot signal while SU_b listens. The SU nodes, however, cannot tell whether their respective slot is earlier or later. In face of such an uncertain beginning of time slot, we apply the following *listen before transmit* (LBT) strategy.

1) **Listen phase:** Each SU (i.e., SU_a) applies PBNS. Suppose there is a pilot signal detected at the current or neighboring channels. Then the SU immediately switches to that channel, decodes the signal, waits until the signal ends, and transmits an acknowledgment (ACK) to indicate a successful rendezvous



(a) PBNS for querying neighboring channels.



(b) CH patterns of SU_a and SU_b.

Fig. 4. An illustration of PBNS-assisted S-RCH scheme.

2) **Transmit phase:** Otherwise, the SU hops to a new channel (i.e., $X_a(t+1)$) selected according to Theorem 1, transmits a pilot signal, and waits for an ACK from SU_b. If an ACK is received, then the SU stops because rendezvous had occurred. If not, increment time t and go to the listen phase to attempt another rendezvous.

In this paper, the aforementioned approach is simply referred as a *PBNS-assisted S-RCH* scheme.

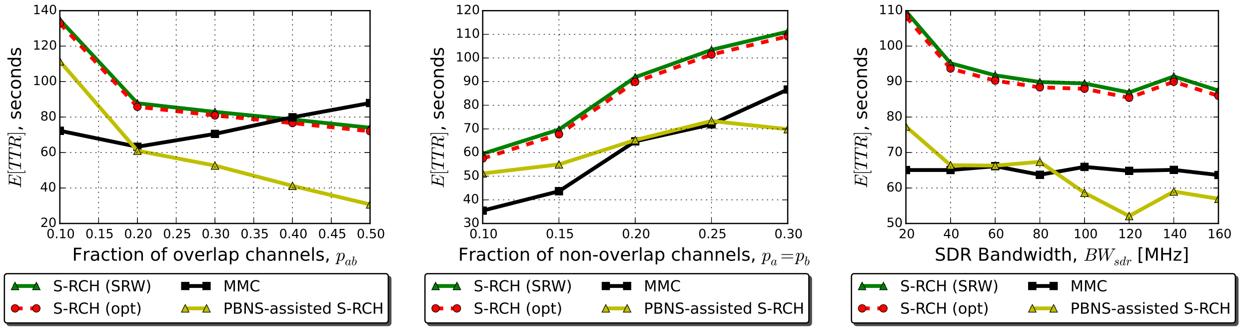
IV. Performance Evaluation

After outlining and discussing how to achieve quick rendezvous using S-RCH scheme, we evaluate its performance via numerical and trace-based evaluations.

A. Parameter Setup

We consider an IoT-based DSA network using LTE-U over the 5GHz spectrum, with a total bandwidth of 500MHz [10]. To facilitate future high-density applications, we assume an SU (IoT) bandwidth of $B_0 = 100kHz$,[2] which results in $N = 5000$ channels. In accordance to LTE-U frame length, each time slot has length $\delta = 10$ms.[1] Each IoT node uses an embedded SDR—such as Ettus E310 USRP—with a digital bandwidth of $BW_{SDR} = 56MHz$, which corresponds to a combined CH range of $R_{ch} = 139$.

In the 5GHz unlicensed spectrum, PUs mainly consist of IEEE 802.11 WiFi devices with 20MHz channel bandwidth. In other words, each PU occupies 200 SU channels. We assume moderate PU activities, which results in the following proportions of available channels. *Overlap ratio* $p_{ab} := \frac{|\mathcal{C}_a \cap \mathcal{C}_b|}{N}$ is the fraction of channels available to IoT nodes SU_a and SU_b. Unless specified otherwise, p_{ab} is set to 0.2. Moreover, $p_a := \frac{|\mathcal{C}_a \setminus (\mathcal{C}_a \cap \mathcal{C}_b)|}{N}$ and $p_b := \frac{|\mathcal{C}_b \setminus (\mathcal{C}_a \cap \mathcal{C}_b)|}{N}$ denote the *fractions of non-overlap* channels available to SU_a and SU_b, respectively. These fractions have a default value of $p_a = p_b = 0.2$.



(a) Versus fraction of overlap channels. (b) Versus fraction of non-overlap channels. (c) Versus SDR's bandwidth.

Fig. 5. Expected TTR performance with respect to various network parameters.

B. Time-to-Rendezvous Evaluation

We compare the proposed S-RCH scheme applying the optimal probabilities in Theorem 1, denoted as `S-RCH (opt)`, to a similar scheme applying the transition probabilities of simple random walk (`S-RCH (SRW)`). We also compare the proposed S-RCH scheme to the existing Modular Modified Clock (`MMC`) [13] scheme. Finally, we evaluate the performance improvement provided by the PBNS-assisted S-RCH scheme in Section III-C. Our evaluation is implemented in Python and all results are averaged over 10^4 network realizations.

1) Effect of the fraction of overlap channels (p_{ab}): We consider the effect of increasing p_{ab} to the expected TTR in Fig. 5(a). As p_{ab} becomes higher, rendezvous is more likely to occur early since there are more commonly-available SU channels. Our hypothesis is verified by the decreasing expected TTR of the `S-RCH (SRW)` scheme. Compared to `S-RCH (SRW)`, the proposed `S-RCH (opt)` scheme provides an improved expected TTR since it attempts to minimize the MTTR bound in Lemma 1, by applying Theorem 1.

According to the triangle-marked plot in Fig. 5(a), the PBNS-assisted S-RCH scheme can further reduce the TTR since it allows SU nodes to detect rendezvous signals outside their current channels, as long as (22) is satisfied. Thus, despite causing the CH range limit R_{ch} in (2)-(3), PBNS sensing can actually be leveraged to improve rendezvous performance.

When SU nodes employ `MMC`, their SDR must be able to hop to any arbitrary frequency, by applying a two-stage tuning that takes up to 31ms at each hop [11]. To provide additional time for transmitting and decoding rendezvous signals, we set the `MMC` scheme's time slot to $4\delta = 40$ms. Fig. 5(a) indicates that `MMC` has an expected TTR that outperforms the proposed schemes when p_{ab} is low. Otherwise, S-RCH with and without PBNS assistance outperform MMC, by up to 70.2%.

2) Effect of the fraction of non-overlap channels (p_a and p_b): We consider the effect of the fraction of non-overlap channels by setting $p_a = p_b$ and increasing their values from 0.1 to 0.3 in Fig. 5(b). As p_a and p_b increase, the random walks performed in S-RCH effectively spend more time in the non-overlap channels than in the common channels, $\mathcal{C}_a \cap \mathcal{C}_b$. Since rendezvous in these schemes only happen in the common channels, rendezvous is less likely to occur and the expected TTR is increased. Then, the expected TTR becomes an increasing function of p_a and p_b, as indicated in Fig. 5(b).

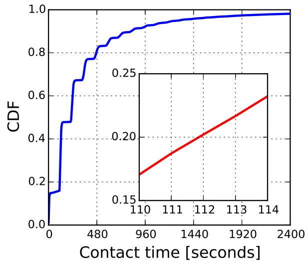
3) Effect of SDR's bandwidth (BW_{SDR}): We increase BW_{SDR} from 20MHz up to 160MHz and plot the resulting expected TTR in Fig. 5(c). According to (1)-(3), BW_{SDR} is proportional to the maximum CH range R_{ch}. Moreover, higher R_{ch} is helpful when the SU nodes start in channels with vastly different indexes (e.g., $X_a(0) = c_0$ and $X_b(0) = c_{N-1}$). Specifically, under the *best-case* policy, SU_a and SU_b may respectively increment and decrease their channel indices by R_{ch}, which results in a TTR lower bounded by $\lceil \frac{N}{2R_{ch}} \rceil$. In Fig. 5(c), we observe that the inversely-proportional scaling of TTR versus R_{ch} (equivalently, BW_{SDR}) can also be observed by the proposed schemes. The expected delay of MMC, on the other hand, does not depend on R_{ch} and remains constant.

C. Time-to-Rendezvous in Mobile Traces

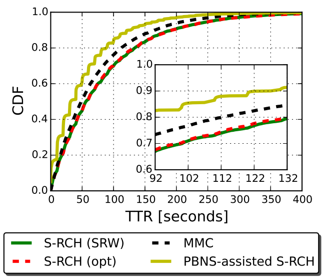
Among the use cases of IoT-based networks is for efficient information exchange among co-located nodes via mesh-based mobile application. Modern mesh-based app [19], consists of mobile nodes with momentary inter-node contacts as their SUs. One major concern is whether the delay for achieving pairwise rendezvous is sufficiently small so that the opportunities provided by momentary contacts can be exploited.

To examine the proposed CH schemes' capability to facilitate rendezvous, we employ the `cambridge/haggle` data set [14] that records the pairwise Bluetooth sightings by groups of nodes carrying small devices (iMotes) in indoor environments. Specifically, we consider the `Exp6` trace that collected the time and duration of contacts between iMotes distributed to 78 students attending the Infocom'06 conference between April 23 to 26, 2006. The `Exp6` trace is employed to portray future conference settings, in which there are many international participants carrying 5G handsets with SDR capability but not equipped with data roaming access, due to expensive roaming fee. In this case, the participants can use the SDR to access unused spectrum via DSA for communications.

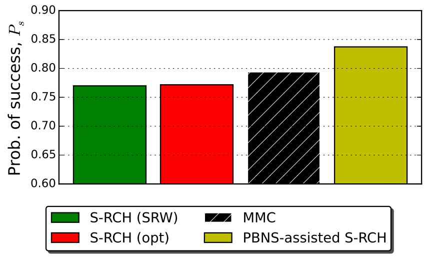
We assume that the Bluetooth sightings provided by the `Exp6` data trace captures all the possible short-range physical contacts between mobile SUs. We collect *contact times* $\{CT\}$, which quantifies the duration from when a pair of



(a) Mobile SUs' contact times.



(b) Distribution of TTR.



(c) Probability of successful rendezvous.

Fig. 6. Rendezvous performance in mobile indoor environment [14].

nodes come into contact until they move out of each others' contacts, and discard the results when the contact time is zero. The corresponding cumulative distribution function (CDF), $P\{CT < x\}$, is plotted in Fig. 6(a).

To proceed, we present Fig. 6(b) that plots the CDF of TTR corresponding to the schemes evaluated in Fig. 5. Then, we relate the TTR plots to the CDF of contact times in Fig. 6(a). To be successful, a rendezvous must occur within a contact period, i.e., when $TTR < CT$. Consider the CDF of the contact times in Fig. 6(a) when most (80%) of the contact times have not ended, that is, $CT > 112$ (see the red plot in the inset of Fig. 6(a)). Then, Fig. 6(b) indicates that the proposed `S-RCH (opt)` and `PBNS-assisted S-RCH` schemes achieve successful rendezvous with probabilities 0.706 and 0.927, respectively.

Although possible, comparing Figs. 6(a) and 6(b) for each possible contact time and TTR—as in the example above—is cumbersome. To provide a more concise way for evaluating a rendezvous scheme's ability to overcome the node mobility, we denote a *successful rendezvous probability* metric as follows.

$$P_s = P\{TTR < CT\} = \sum_{x=0}^{\infty} P\{TTR < x\} P\{CT = x\}.$$

Successful rendezvous probability quantifies the likelihood that a successful rendezvous can be achieved, over all the possible contact times, CT. From the resulting P_s plots in Fig. 6(c), we observe that the optimal transition probabilities applied by the proposed `S-RCH (opt)` scheme provide 0.24% higher success rate than the naive `S-RCH (SRW)` scheme and a 2.81% lower P_s than the existing `MMC` schemes. Moreover, the proposed `PBNS-assisted S-RCH` provides $P_s = 83.8\%$ of successful rendezvous, which is a 5.45% improvement over existing `MMC` scheme.

V. Conclusion

We proposed a symmetric random channel hopping (S-RCH) scheme to achieve quick rendezvous between nodes with limited channel hopping capability. We modeled S-RCH as random walks over spectrum slice graphs, and assign the walks with optimal transition probabilities that minimize time-to-rendezvous. We show that S-RCH outperforms the existing Modular Modified Clock scheme, while achieving 83.8% successful rendezvous in indoor mobile environment. Our findings in this paper not only open a new research avenue towards rendezvous for low-cost, low-complexity IoT nodes, but also provide guidelines for real-world implementation.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] "IoT connected devices installed base worldwide," https://www.statista.com/statistics/471264/, accessed: 06-21-2017.

[3] X. Zhang and H. Su, "CREAM-MAC: Cognitive radio-enabled multi-channel MAC protocol over dynamic spectrum access networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 110–123, 2011.

[4] B. Yang, M. Zheng, and W. Liang, "A time-efficient rendezvous algorithm with a full rendezvous degree for heterogeneous cognitive radio networks," in *Proceedings of IEEE INFOCOM*. IEEE, 2016, pp. 1–9.

[5] C.-S. Chang, W. Liao, and T.-Y. Wu, "Tight lower bounds for channel hopping schemes in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2343–2356, 2016.

[6] D. Yang, J. Shin, and C. Kim, "Deterministic rendezvous scheme in multichannel access networks," *Electronics Letters*, vol. 46, no. 20, pp. 1402–1404, 2010.

[7] K. Bian *et al.*, "Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1294–1307, 2013.

[8] S.-H. Wu, C.-C. Wu, W.-K. Hon, and K. G. Shin, "Rendezvous for heterogeneous spectrum-agile devices," in *Proceedings of IEEE INFOCOM*. IEEE, 2014, pp. 2247–2255.

[9] 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. ACM MobiCom'04*. ACM, 2004, pp. 216–230.

[10] L.-U. Forum, "Coexistence Study for LTE-U SDL V1.0 (2015-02)," *LTE-U Technical Report*, 2015.

[11] R. Bell, "Maximum supported hopping rate measurements using the universal software radio peripheral software defined radio," in *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.

[12] Z. Sun and J. N. Laneman, "Performance metrics, sampling schemes, and detection algorithms for wideband spectrum sensing," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5107–5118, 2014.

[13] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 216–227, 2011.

[14] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.

[15] J. Jia and Q. Zhang, "Rendezvous protocols based on message passing in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5594–5606, 2013.

[16] "Google Spectrum Database," https://www.google.com/get/spectrumdatabase/, accessed: 06-28-2017.

[17] "USRP Hardware Driver and USRP Manual," http://files.ettus.com/manual/page_general.html, accessed: 06-28-2017.

[18] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. AMS, 2009.

[19] "FireChat," http://opengarden.com/firechat/, accessed: 2016-03-02.

[20] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[21] R. A. Hom and C. R. Johnson, "Topics in matrix analysis," *Cambridge UP, New York*, 1991.

[22] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ. Press, 2004.