Throughput Oriented Lightweight Near-Optimal Rendezvous Algorithm for Cognitive Radio Networks

ChunSheng Xin[a], Sharif Ullah[a], Min Song[b], Zhao Wu[c], Qiong Gu[c], Huanqing Cui[d]

[a] *Old Dominion University, Department of Electrical and Computer Engineering, Norfolk, VA 23435, USA.*
[b] *Michigan Technological University, Department of Computer Science, Houghton, MI 49931, USA*
[c] *Hubei University of Arts & Science, School of Math & CS, Xiangyang, P.R.China*
[d] *Shandong University of Science & Technology, College of Info. Sci. & Eng., Qingdao, P.R.China*

Abstract

In cognitive radio networks, secondary users have to dynamically search and access spectrum unused by primary users. Due to this dynamic spectrum access nature, the rendezvous between secondary users is a great challenge for cognitive radio networks. In this paper, we propose a *Throughput oriEnted lightweight Near-Optimal Rendezvous* (TENOR) algorithm that does not need a common control channel. TENOR has very lightweight overhead and accomplishes near-optimal performance with regard to both throughput and rendezvous time. With TENOR, secondary users are grouped into node pairs that are spread onto different channels in a decentralized manner. The co-channel interference is minimized and the throughput is near optimal. We develop a mathematical model to analyze the performance of TENOR. Both analytical and simulation results indicate that TENOR achieves near-optimal throughput and rendezvous time, and significantly outperforms the state-of-the-art rendezvous algorithms in the literature.

Keywords: Cognitive radio network; rendezvous; lightweight rendezvous; near-optimal rendezvous.

1. Introduction

The proliferation of wireless devices and wireless services creates a tremendous pressure on spectral resource. Within the current spectrum regulatory framework, the exclusive allocation of frequency bands leads to a sporadic utilization of spectrum. To solve this inefficient approach of spectral allocation, *cognitive radio* emerges to enable dynamic access to vacant frequency bands, called *spectrum holes* [1]. A principal characteristics of the cognitive radio technology is to provide

Email addresses: `cxin@odu.edu` (ChunSheng Xin), `mulla001@odu.edu` (Sharif Ullah), `mins@mtu.edu` (Min Song), `wuzhao73@163.com` (Zhao Wu), `gujone@163.com` (Qiong Gu), `cuihq79@163.com` (Huanqing Cui)

the capability of sharing spectrum bands between *secondary users* (SUs) and *primary users* (PUs) in an opportunistic manner, called *dynamic spectrum access*, where SUs access a licensed channel when the PU is not using it. In this paper, we refer to a spectrum band as a *channel* which is consistent with other studies in the literature.

In *cognitive radio networks* (CRNs), communications between SUs face a critical problem due to dynamic change of operation channels: how a transmitter SU finds and switches to the operation channel of the receiver SU for communications, called the *rendezvous problem* for CRNs. Another observation is that given the abundance of unused licensed channels, SUs should take advantage of them by selecting and operating on diverse channels to reduce co-channel interference. Different MAC protocols have been proposed for CRNs [2, 3, 4]. In these protocols a common control channel is used by SUs for rendezvous so that SUs can negotiate a *data channel* for packet transmission. Nevertheless, using a common control channel for rendezvous has a disadvantage that the control channel might get congested or attacked. Hence, there have been many rendezvous schemes designed for CRNs without using a control channel, such as [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. The common idea of those schemes is to employ a *channel hopping* approach. Each SU selects a fixed number of channels in a sequential order, and repeatedly hops on those channels over time slots. The number of time slots in a hopping cycle is called *channel hopping period*. Within a channel hopping cycle, when two SUs hop onto the same channel and the channel is available to both SUs in that slot, then they rendezvous. The key problem for channel hopping based rendezvous algorithms is how to generate channel sequences so that the channel sequences selected at different SUs have enough overlapping channels. This ensures two SUs can rendezvous even though some of the overlapping channels are not available when the SUs hop to them.

Compared with the control channel based rendezvous schemes, the major benefit of channel hopping based rendezvous schemes is diversification of rendezvous channels. Two SUs can rendezvous on any channel as long as their channel sequences overlap on this channel. This addresses channel dynamics, provides robustness for rendezvous, and reduces congestion. However, while those algorithms aim to solve the rendezvous problem, they are not *throughput oriented* by optimizing the rendezvous among all SUs. Instead, those rendezvous algorithms aim to let two SUs rendezvous, i.e., *pairwise rendezvous*. While the pairwise rendezvous can result in rendezvous among all SUs, the throughput under such rendezvous schemes can be sub-optimal or even poor due to several problems. First, channel sequences need to be pre-computed for a channel hopping period. Hence those schemes cannot utilize channel status when computing channel sequences, since an SU cannot know the channel status in the future time slots within a channel hopping period. Second, *the rendezvous is passive*. That is, even if SU A has urgent traffic to SU B, SU A cannot immediately rendezvous with SU B. It still has to follow the channel hopping sequence, until both SUs hop to the same channel at a certain time slot, which may be far in the future. Hence the urgent traffic may not be delivered in time. Third, the number of channels in a given time slot that have rendezvous activities can be very small, and on those rendezvous channels, SUs may not be evenly distributed, resulting in congestion. Due to the first problem, an SU can frequently hop to unavailable channels, which not only reduces rendezvous probability, but can actually result in no rendezvous at all. One may think that each SU may pre-compute channel sequences based on the currently available channels. Unfortunately, most schemes would fail as they rely on a common set of channels to generate channel sequences in order to ensure rendezvous. Furthermore, the channel hopping period is very large for most schemes. This also precludes to generate channel sequences based on the currently available channels, since the current channel status can become obsolete at distant future time slots, for moderately dynamic cognitive radio networks. While a typical remedy suggested by some papers is to randomly pick a currently available channel as a substitute when an SU hops to an unavailable channel in the channel sequence, the performance is not much improved, since a randomly picked channel rarely results in a rendezvous.

To address those issues, we take an approach different from the channel hopping based schemes to design throughput oriented, load balanced, and proactive rendezvous algorithms. In [20], we designed a base rendezvous algorithm that is for pairwise rendezvous. Different from the channel hopping based schemes, which is passive, the algorithm in [20] is proactive. That is, when an SU has traffic to another SU, it immediately estimates and switches to the channel of the receiver for rendezvous, rather than passively hopping on a channel sequence. This results in small rendezvous time, suitable for delivering urgent traffic. However, the algorithm in [20], like all channel hopping based schemes, does not jointly consider rendezvous among all SUs, to avoid congestion on a specific channel. Thus the throughput is sub-optimal. An enhancement of this algorithm to jointly consider all-SU rendezvous was presented in [21]. The main idea is to let sender SUs re-select receiver SUs if a rendezvous channel is congested. The throughput improves significantly compared with [20]. However, it may take a large time to converge to optimal throughput and it also results in overhead due to extra channel switchings when an SU rendezvous on a congested channel.

In this paper, we propose a *Throughput oriEnted lightweight Near-Optimal Rendezvous* (TENOR) algorithm that does not rely on a common control channel, has very lightweight overhead, and accomplishes near-optimal throughput. Different from the algorithm in [20], TENOR is designed to achieve near-optimal throughput, particularly for rendezvous in CRNs with heavy traffic load. Different from the algorithm in [21], TENOR does not have the convergence issue and does not need to conduct extra channel switchings. The main idea of TENOR is to proactively avoid too many SUs to rendezvous with the same SU, through a 'virtual' pairing of SUs, even before SUs start to switch channels. In TENOR, SUs are perfectly paired, and each pair selects a different channel for rendezvous, achieving near-optimal throughput. Through a smart design, all these actions are accomplished without exchanging any control message between SUs in advance. The only requirement for TENOR is that all SUs need to be synchronized on time, e.g., through GPS, like synchronous channel hopping based rendezvous schemes.

Our contributions are summarized as follows:

- We devise a throughput oriented rendezvous algorithm termed TENOR. Similar to channel hopping based schemes, TENOR does not require SUs exchange any control message before rendezvous, such as for SU pairing and channel selection/estimation. With TENOR, SUs are perfectly paired in each slot, and each pair is very likely on a separate rendezvous channel, achieving near-optimal throughput. TENOR can perform proactive rendezvous to deliver high-priority traffic by enabling the sender SU to immediately estimate and switch to the channel of the receiver SU.

- We develop a mathematical model to analyze TENOR on two important rendezvous performance metrics, throughput and rendezvous time. While the previous works on channel hopping based rendezvous schemes provided theoretical results on rendezvous time, we provide theoretical results on both throughput and rendezvous time.

- We extensively simulate TENOR and compare its performance with the state-of-the-art channel hopping based rendezvous schemes. The results indicate that TENOR significantly outperforms those schemes with regard to throughput and rendezvous time. In most scenarios, its throughput is close to the maximum throughput, while the rendezvous time is also close to the optimal rendezvous time conditioned on maximum throughput.

The remainder of the paper is organized as follows. In the next section, we discuss related work. In Section 3, we present the system model. In section 4 we describe the proposed rendezvous algorithm TENOR. In section 5 we develop an analysis model for TENOR. Section 6 presents performance evaluation and section 7 concludes the paper.

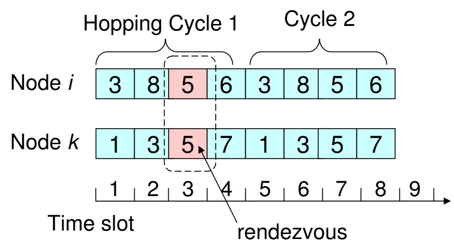

Figure 1: Channel hopping based synchronous rendezvous

2. Related Work and Theoretical Performance Comparison

The widely used approach for rendezvous without relying on a common control channel is through repeated channel hopping over time slots, following a *channel sequence*. Specifically, in channel hopping based rendezvous schemes, SUs use some techniques to pre-compute channel sequences. A channel sequence is a sequential order of channels such as {3, 8, 5, 5, 8, 6}. Note that a specific channel may appear multiple times in a channel sequence. In fact, most channel hopping based rendezvous schemes extensively reuse channels in a channel sequence to better achieve rendezvous. The length of a channel sequence, or the number of time slots in a hopping cycle, is called *channel hopping period*, which is 6 for channel sequence {3, 8, 5, 5, 8, 6}. An SU selects a channel sequence and hops among those channels sequentially in the time unit of a slot, until it hops to the last channel in the sequence. Then it cycles back to the first channel, and the procedure repeats.

Fig. 1 illustrates how two SUs or nodes, i and k, rendezvous under a channel hopping based rendezvous scheme, with a channel hopping period of 4. The channel sequence of node i is {3, 8, 5, 6}, while the channel sequence of node k is {1, 3, 5, 7}. Node i hops to channel 3 on slot 1, channel 8 on slot 2, channel 5 on slot 3, and channel 6 on slot 4. This is one *channel hopping cycle*. Then node i repeats the channel hopping cycle starting from the first channel in the sequence again, i.e., hops to channel 3 on time slot 5, channel 8 on time slot 6, and so on, and so forth. Similarly, node k hops to channel 1 on time slot 1, channel 3 on time slot 2, channel 5 on time slot 3, channel 7 on time slot 4, and then repeats the channel hopping cycle. The two nodes rendezvous at the third time slot of each channel hopping cycle on channel 5. This simple example also illustrates a vulnerability of channel hopping based rendezvous schemes–they cannot adapt to channel dynamics. For instance, if channel 5 becomes unavailable when node i or k hops to it, they cannot achieve rendezvous. This is because channel sequences {3, 8, 5, 6} and {1, 3, 5, 7} have only one overlapping channel, 5, at time slot 3. Hence one of the main design goals for such schemes is to have sufficient channel overlapping between two sequences, so that two nodes

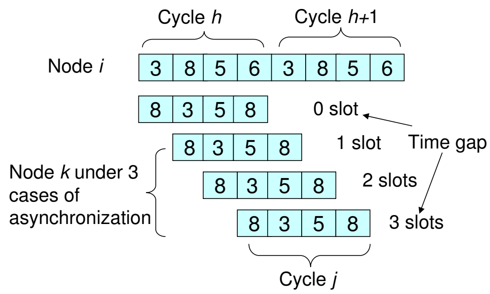

Figure 2: Channel hopping based asynchronous rendezvous

have chance to rendezvous on another overlapped channel when the rendezvous on one overlapping channel fails. This solution can reduce the vulnerability, but does not completely eliminate it, in particular in highly dynamic cognitive radio networks. In Fig. 1, nodes i and k must align their channel hopping cycle boundaries to start at the same time slot in order to achieve rendezvous. This type of channel hopping rendezvous schemes is called *synchronized rendezvous*. An *asynchronous rendezvous scheme* does not require nodes align the boundaries of their channel hopping cycles, as illustrated in Fig. 2, where node k can start its channel hopping cycle one, two, or three slot later than node i. In either case, node k can rendezvous with node i at least once in channel hopping cycle j of node k, with the rendezvous occurring either in hopping cycle h or $h+1$ of node i. For instance, if the time gap between the hopping cycles of the two nodes is 3 slots, then in the second time slot of hopping cycle j, node k hops to channel 3, while at the same time, which is the first time slot of hopping cycle $h+1$ at node i, node i also hops to channel 3. Thus, nodes i and k rendezvous at this time slot.

The key problem for channel hopping based rendezvous schemes is how to generate or compute channel sequences such that there are enough overlapping channels between any two channel sequences. As a matter of fact, this is the major difference between disparate channel hopping based rendezvous schemes. Given channel sequences with such a property, two nodes that independently select two channel sequences can have a good chance on some of the overlapped channels that are available by the time they hop to such channels. For asynchronous rendezvous, it requires an extra property that for any two channel sequences, cyclically rotating one sequence still has enough overlapping channels with the other sequence. The authors in [5] proposed a channel hopping based rendezvous algorithm, which achieves optimal rendezvous time if all channels in the channel sequences are available. A channel hopping based rendezvous algorithm called Jump-Stay was presented in [6], which guarantees rendezvous between two CR users as long as they have some common channels, using a prime number modulus to generate channel sequences. Two quorum based channel hopping based rendezvous algorithms were introduced in [7, 8] to achieve small time to rendezvous, as well as prevent link breakage caused by incumbent PU signals. The authors in [9] also used the quorum system to generate channel sequences for asynchronous rendezvous algorithms. The DRDS algorithm generates channel hopping sequences based through a technique of disjoint relaxed difference sets [13]. The E-AHW algorithm uses an alternating hopping/waiting scheme to construct channel hopping sequences [14]. The MTP algorithm uses a slow-moving channel pointer and a fast-moving channel pointer to search for rendezvous opportunities [15]. The A-HCH algorithm combines fast and slow channel hopping sequences according to SUs' IDs to achieve rendezvous [16]. The CBH algorithm converts SU ID into bit strings, which results in overlap and rendezvous [17]. The DSCR algorithm uses the disjoint set cover technique to generate channel sequences, and is among the best of asynchronous rendezvous schemes with regard to performance [18]. Most of those recent rendezvous schemes were designed for asynchronous rendezvous. A recently proposed synchronous rendezvous scheme was RCCH [19], which uses clockwise and counter-clockwise modulus operation and rotation to generate channel sequences. RCCH is a state-of-the-art synchronous rendezvous scheme. Those two state-of-the-art rendezvous algorithms, RCCH and DSCR, will be compared with the proposed algorithm TENOR, in Section 6. There have been also some studies assuming multiple radios per SU. The authors in [10] extended the Jump-Stay algorithm in [6] for rendezvous with multiple radios per node. The work in [11] also studied the rendezvous with multiple radios per node in an anonymous environment, i.e., the nodes do not know the IDs of each other. The authors in [12] studied how to utilize multiple radios per node to speed up rendezvous. In [22], the authors studied a related problem, multi-channel neighbor discovery in mobile sensing applications, which needs to consider both the node wake-up scheduling for duty cycling and the rendezvous problem between nodes.

Table 1 compares the theoretical performance of various rendezvous algorithms, including the channel hopping period, *expected time to rendezvous* (ETTR), and *maximum time to rendezvous* (MTTR). The OptimalThp is a 'virtual' rendezvous scheme achieving the maximum throughput that assumes an *infinite capacity common control channel* to exchange channel availability information (no congestion) and an ideal algorithm that always schedules SUs in pairs on disparate channels without delay. It is mainly used as a reference point to gauge the performance of practical rendezvous schemes, in particular the proposed scheme TENOR. In the table, $\mathbb{M}$ is the number of global channels (including both available and unavailable channels to SUs), and P is the smallest

Table 1: Performance comparison of rendezvous algorithms

Algorithm	Channel hopping period	ETTR	MTTR
Asynchronous			
JS [6]	$3\mathbb{M}P$	$2\mathbb{M}P(P-G)+P(\mathbb{M}+5-P-\frac{2G-1}{\mathbb{M}})$	$3\mathbb{M}P(P-G)+3P$
DRDS [13]	$3P^2$	$-$	$3P^2+2P$
E-AHW [14]	$147\mathbb{M}P$	$147P(\mathbb{M}-G)+\frac{13P}{6}$	$147P(\mathbb{M}-G+1)$
MTP [15]	$64(M_A^2-M_A)(\lceil\log\log\mathbb{M}\rceil+1)$	$-$	$O((\max\{M_A,M_B\})^2\log\log\mathbb{M}$
A-HCH [16]	$\geq M_AM_B$	$\frac{lM_AM_B}{\triangle G}$	$-$
CBH [17]	$2l_p\hat{P}^2$	$-$	$2l_p\hat{P}^2$
S-QCH [9]	$M_B\mathbb{M}(2\mathbb{M}+1)$	$-$	$(M_B-G+1)\mathbb{M}(2\mathbb{M}+1)$
DSCR [18]	$P(2P+\left\lfloor\frac{P}{2}\right\rfloor)$	$(2P+\left\lfloor\frac{P}{2}\right\rfloor)(P-G+1)-\left\lfloor\frac{P}{2}\right\rfloor$	$(2P+\left\lfloor\frac{P}{2}\right\rfloor)(P-G+1)$
Synchronous			
RCCH [19]	$\frac{\mathbb{M}^2}{2}$	$-$	$\frac{\mathbb{M}^2}{2}$
TENOR	$-$	pairwise: $\frac{2}{\alpha}$, ThpOpt: $\frac{1}{\pi}$	$-$
OptimalThp	$-$	pairwise: 1, ThpOpt: $N-1$	pairwise: 1, ThpOpt: $N-1$

prime number not smaller than $\mathbb{M}$. M_A and M_B are the numbers of available channels of SU A and SU B, respectively. G is the number of common available channels between SU A and SU B. The l is the length of the choice sequence, and Δ is the degree of the symmetrization class in A-HCH. In CBH, $\hat{P}=\max\{P_A,P_B\}$, where P_A and P_B are the smallest prime numbers not smaller than M_A and M_B. The l_p is a constant determined by SUs' IDs in [17]. In TENOR, α is the channel estimation success probability (see Lemma 1), π is the rendezvous probability in a time slot (Theorem 2), and N is the number of SUs or nodes in the cognitive radio network. If the result was not given in the original paper or if it is not applicable, it is denoted as $-$.

Note that in Table 1, the rendezvous time for channel hopping based rendezvous schemes is for pairwise rendezvous. That is, given that two specific SUs A and B want to communicate with each other at a certain time slot, how soon can they rendezvous? For TENOR and OptimalThp, to optimize throughput, they need to jointly consider the rendezvous among all SUs, and hence the pairwise rendezvous time can be sub-optimal. In fact, a better metric to measure their rendezvous performance is *expected time between rendezvous*, rather than expected time to rendezvous. Certainly TENOR and OptimalThp can be adjusted for pairwise rendezvous like other schemes, i.e., minimize the time to rendezvous for two specific SUs. For OptimalThp, it is 1 time slot, while for TENOR, it is $\frac{2}{\alpha}$ time slots, where α is defined in Lemma 1 and usually close to 1. To be fair, we provide these two types of rendezvous time for TENOR and OptimalThp in Table 1, indicated as 'pairwise' and 'ThpOpt', respectively.

The actual rendezvous time of TENOR follows a geometric distribution. While this means the theoretical MTTR is unbounded, it also indicates that the increase of the rendezvous time is in an exponentially decreased probability. In fact, the actual maximum rendezvous time of TENOR in simulations is significantly smaller than the ones of state-of-the-art channel hopping based schemes, RCCH and DSCR, as illustrated in the simulation results in Section 6. Furthermore, while most channel hopping based schemes claim a bounded theoretical MTTR, such MTTR is actually conditional based on an assumption as follows. For many schemes, the MTTR was derived assuming all channels in the channel sequences are available, which is not valid for practical cognitive radio networks. Some papers (e.g., [19]) noted this issue and proposed the concept of conditional MTTR, or MCTTR, which is defined to be the maximum time to rendezvous while the channels in the channel sequences may be unavailable during channel hopping. Nevertheless, even the studies considering MCTTR assumed that the channel status does not change frequently. Specifically, they assumed the channel status does not change before the SUs rendezvous. That is, if a channel is available at the beginning of seeking rendezvous, it is always available until the next rendezvous. For practical cognitive radio networks where the channel status is moderately to highly dynamic, such assumption can become invalid too. As a matter of fact, it can be easily seen that in the worst case, no channel hopping based scheme can guarantee bounded MTTR. For example, assume that whenever an SU hops to an overlapping channel with another channel sequence of a different SU, the channel becomes unavailable in that time slot. Note that this does not need to have highly dynamic channel status change; only one overlapping channel needs to change status. It is clear that this SU would never achieve rendezvous with any other SU in this case.

3. System Model

We assume that each node (an SU) has a cognitive radio for data communications, and a fast wideband spectrum sensor for spectrum sensing. The spectrum sensor is used to detect available channels for SUs in each time slot. However, the spectrum sensor is not a necessary condition for the proposed algorithm to operate. Instead, the cognitive radio for data transmissions can be used to conduct spectrum sensing to find if a channel is available to an SU. The cognitive radio can receive the GPS signal to have a common time reference, so that all nodes are time synchronized. The time is slotted. Hence the nodes in the network operate in a synchronized time-slotted mode. We assume that the network traffic load is high, such that in a time slot, a node likely has new traffic or backlogged traffic from previous time slots to all other nodes. Hence a node does not care which peer node to be selected for rendezvous in a time slot. In fact, with large network traffic load, it is more important to spread nodes onto different channels to send out as many packets as possible. *This avoids co-channel interference and achieves optimal throughput, and hence is a good strategy to address the large network traffic load.* Certainly, TENOR can be adjusted to enable a transmitter to directly rendezvous with a given receiver.

Let N denote the number of nodes in the network. Let $\mathbb{M}$ denote the total number of global channels, including both available and unavailable channels. In practice, the channel availability for different nodes is often heterogeneous. Furthermore, even if the available channels to two nodes are the same, due to false alarm in spectrum sensing, the detected available channels by two nodes are usually not the same, since an available channel may be false alarmed at a node. Therefore, the available channels to different nodes are usually heterogenous. Let M denote the number of channels available to at least one node in the current time slot. Let $\{1, \ldots, M\}$ denote these channels. We use the following model to characterize *channel heterogeneity* among nodes: each channel in the set $\{1, \ldots, M\}$ has probability p to be available and detected by a node. In other words, every node has its own available channel set from which it can choose an operation channel. In reality, a channel detected as 'available' may be actually used by PUs. In this paper, we do not consider the miss-detection in the channel heterogeneity model. In practice, CRNs usually require a very low miss-detection probability to avoid harmful interference to PUs. Hence, the performance analysis model to be presented in Section 5 is not significantly affected by the miss-detection.

Let $\mathcal{C}_i$ and $\mathcal{C}_k$ $(1 \leq i, k \leq N)$ indicate the set of channels that are available and detected by nodes i and k, respectively. Since the channel availability to nodes i and k is independent, the probability that a given channel among the total M channels is commonly available to both nodes is p^2. Therefore, the number of channels that are commonly available to both nodes, i.e., $m = |\mathcal{C}_i \cap \mathcal{C}_k|$, follows a binomial distribution $B(m; M, p^2)$. The mean number of common channels between two nodes is thus Mp^2. Since the mean number of available channels of a node is Mp, we can see that $p = \frac{Mp^2}{Mp}$ *actually represents the average percentage of channels of a node that are also available to a neighbor node*, i.e.,

$$p = E(\frac{|\mathcal{C}_i \cap \mathcal{C}_k|}{|\mathcal{C}_i|}). \tag{1}$$

Thus, we can see that the channel heterogeneity between nodes is indeed controlled by parameter p, with $0 \leq p \leq 1$.

Fig. 3 indicates the channel availability model, to illustrate the set of global channels, set of all available channels, and the set of available channels to a given node. There are total $\mathbb{M} = 12$

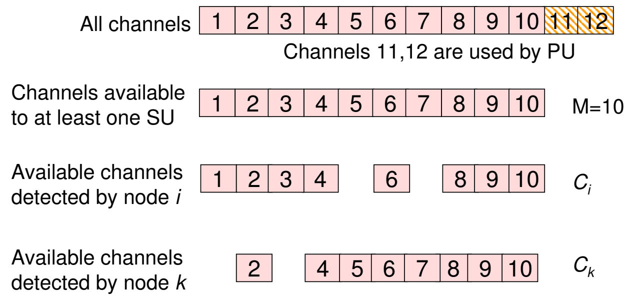

Figure 3: Channel availability illustration with $\mathbb{M} = 12, M = 10$.

channels. At the current time, two channels are used by PUs. This gives $M = 10$ channels available to SUs. Assuming $p = 0.8$, then the number of channels available to a node should be around 8. In Fig. 3, we illustrate the available channels set for nodes i and k, i.e., $\mathcal{C}_i$ and $\mathcal{C}_k$, respectively. Although both have 8 channels in the figure, their channels are different. In fact, there are only 6 common channels between $\mathcal{C}_i$ and $\mathcal{C}_k$, i.e., $|\mathcal{C}_i \cap \mathcal{C}_k| = 6$. Note that on average, the number of common channels between two nodes is around $Mp^2 = 6.4$. Moreover, the percentage of the common channels with regard to the available channels of node i is $\frac{|\mathcal{C}_i \cap \mathcal{C}_k = 6|}{|\mathcal{C}_i|} = 0.75$. On average, this number should be around p as indicated in (1), i.e., 0.8, but does not have to be exactly the same since it is the mean operation in (1).

4. Throughput Oriented Lightweight Near-Optimal Rendezvous (TENOR) Algorithm

The strategy of TENOR is to divide nodes into two equal-size groups in every time slot, called *active nodes*, and *passive nodes*, and then pair each node from the active group to one node in the passive group for rendezvous on a different channel. Thus this strategy spreads nodes onto different channels and enables the active and passive nodes on each channel to successfully rendezvous, which eliminates co-channel interference and achieves optimal throughput, to effectively address the large network traffic load.

We assume that each node uses only one cognitive radio for communications due to the high cost. To facilitate rendezvous, in each time slot, we let the passive nodes choose and stay on their operation channels; the active nodes seek to rendezvous with (different) passive nodes, by estimating and switching to their operation channels. Algorithm 1 implements this scheme in a distributed manner. Specifically, in time slot t, each node uses a random number generator with seed t (the time slot) to generate the states of all nodes, denoted as a *state vector* $[g_1, g_2, \ldots, g_N]$, through a random permutation of an N-element binary vector $[1, \ldots, 1, 0, \ldots, 0]$, where the first $\lceil \frac{N}{2} \rceil$ number of elements are 1 and the rest $\lfloor \frac{N}{2} \rfloor$ number of elements are 0. This is achieved by lines 1 to 2 in Algorithm 1. The use of the same seed (time slot t) by every node ensures the randomly permuted state vector $[g_1, g_2, \ldots, g_N]$ at different nodes is the same. On the other hand, using time slot number t as the seed randomizes the state of a node at different slots so that the nodes in the passive group change from slot to slot.

The pairing between active nodes and passive nodes is implemented in lines 4 to 6 in Algorithm 1. Specifically, let $\tilde{V}_1, \ldots, \tilde{V}_{\tilde{N}}$ denote the $\lceil \frac{N}{2} \rceil$ number of passive nodes, where $\tilde{N} = \lceil \frac{N}{2} \rceil$, and $V_1, \ldots, V_{\lfloor \frac{N}{2} \rfloor}$ denote the $\lfloor \frac{N}{2} \rfloor$ number of active nodes. We get a random permutation of $1, \ldots, \tilde{N}$ and denote them as $j(1), \ldots, j(\tilde{N})$. Then active node V_i is paired with passive node $\tilde{V}_{j(i)}$. Since all nodes generate the same state vector, the set of active nodes determined by the state vector is the same, and the node pairing between active and passive nodes is also the same. In other words, for two different nodes i and k, when each independently runs Algorithm 1 in time slot t, the generated $[g_1, g_2, \ldots, g_N]$ is the same, the node ID set $\mathcal{V}$ is the same, and the permutation $j(1), \ldots, j(\tilde{N})$ is also the same. Note that each node does not need to send any information to each other to achieve this effect.

Next a node checks its state from the state vector. For node i, it is g_i. If g_i is 1, node i is a passive node. A passive node iteratively searches all channels using a random *pointer* until the pointed channel is from its available channel set, depicted in lines 18 to 25. Due to the use of the random pointer, the operation channel is essentially selected in a random manner. Hence passive nodes are likely spread on different channels automatically. Therefore the co-channel interference is minimized as different node pairs are on different channels. A passive node stays on its operation channel, referred to as the *home channel*, within the current time slot, and changes to a different operation channel in the next time slot.

If a node finds that itself is an active node, then it estimates and switches to the operation channel of the paired passive node for rendezvous, described in lines 7 to 16. For active node V_k, its paired passive node is $\tilde{V}_{j(k)}$. Since all nodes generate the same state vector, when node $\tilde{V}_{j(k)}$ runs Algorithm 1 in slot t, it is guaranteed that node $\tilde{V}_{j(k)}$ would be passive. Hence when the active node V_k switches to the operation channel of the paired passive node $\tilde{V}_{j(k)}$, the passive node is guaranteed to be on its home channel, and the rendezvous is thus successful.

Fig. 4 illustrates how nodes 3 and 6 independently run TENOR and achieve rendezvous. Note that each node in the network independently runs TENOR. They do not need to exchange

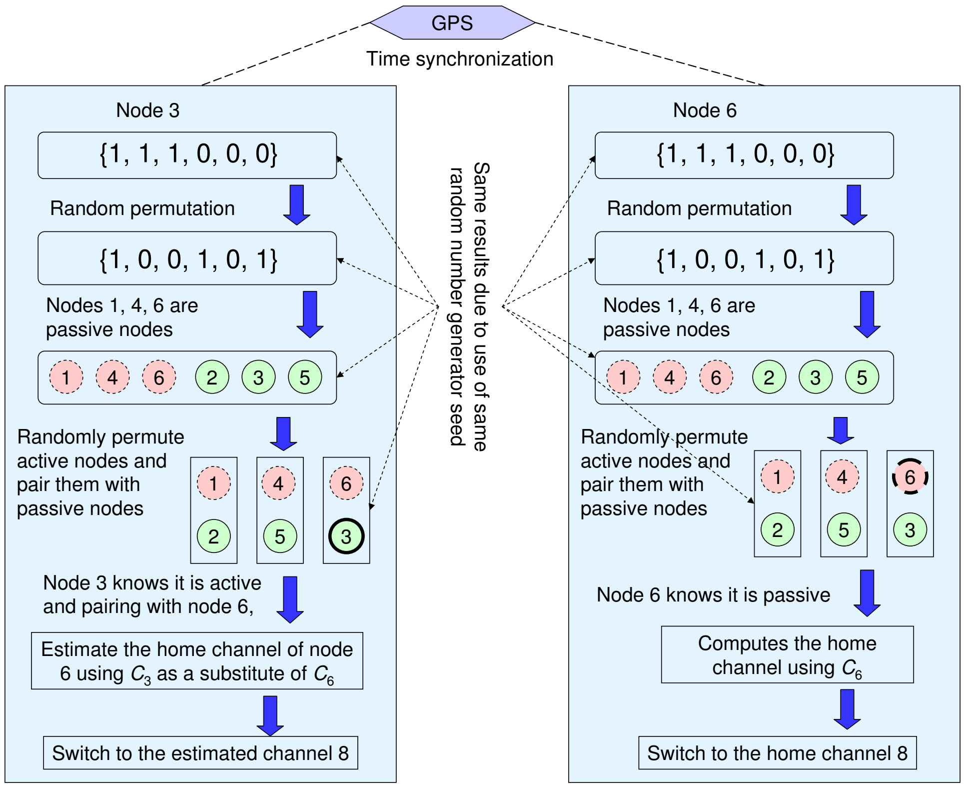

Figure 4: TENOR operation at nodes 3 and 6. Each node operates independently without exchange of control information before rendezvous.

information before rendezvous, except that they need to be synchronized on time through GPS. Each node performs a random permutation to find which nodes are passive. Node 3 does not need to send the node status information to node 6 or any other node. Although it may seem unnecessary for node 3 to compute the states of other nodes since this information is not to be sent to other nodes, this step is actually needed for node 3 to figure out its pairing node later on. Next, node 3 performs another random permutation on the active node IDs, resulting in $\{2, 5, 3\}$. Then node 3 sequentially pairs the permuted active nodes, $\{2, 5, 3\}$, with the passive nodes, $\{1, 4, 6\}$. At this time, node 3 identifies its pairing node, which is node 6. Since node 3 is an active node, node 3 estimates and switches to the home channel of node 6. Similarly, node 6 also performs those steps, totally independently. Since it is a passive node, it simply selects a home channel from its available channels. Note that the node status for all nodes determined by the first permutation are the same since every node uses the current time slot as the seed. In Fig. 4, we illustrate that node 3 correctly estimates the channel of node 6 and hence they successfully rendezvous.

How does an active node j estimate the home channel of a passive node i? The challenge is that

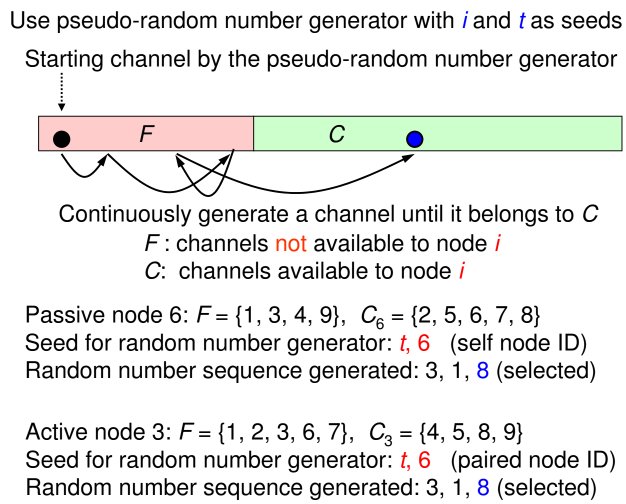

Figure 5: Home channel selection by node 6, and channel estimation by node 3

node j does not know the available channel set of node i, since our rendezvous algorithm, TENOR, was designed without requiring the nodes to exchange any information before they rendezvous. Our approach is to let node j 'simulate' node i to select a home channel for node i. That is, node j iteratively searches all channels using a random *pointer* until the pointed channel is from its available channel set, as depicted in lines 9 to 16. The key difference here is that node j uses its own available channel set $\mathcal{C}_j$ as a substitute of node i's available channel set $\mathcal{C}_i$. As shown in Lemma 1 in the next section, the success probability of such channel estimation is usually high. In the case that the channel estimation fails, then node j is not able to rendezvous with node i in the current time slot. In the next section, we will analyze the expected rendezvous time between two nodes. Note that in the case of channel estimation failure, although node j cannot rendezvous with node i, it may still rendezvous with some other node. If the estimated channel by node j is accidentally the home channel of another passive node k, then node j rendezvous with node k. Furthermore, there may be another active node j' which also incorrectly estimates the home channel of its paired passive node i', but by accident its estimated channel is the same as the estimated channel of node j.

Fig. 5 illustrates how a passive node selects its home channel, and how an active node estimates the home channel of its paired passive node. In the former case, the passive node supplies its available channels set to be set $\mathcal{C}$, e.g., node 6 in Fig. 4 will supply $\mathcal{C}_6$. In the figure, set $\mathcal{F}$ contains the channels unavailable to the node at the current time. TENOR uses a random number generator to continuously generate a list of channels, until a generated channel belongs to set $\mathcal{C}$. Then this last channel is selected as the home channel of the passive node. In the figure, the first generated channel is 3, which does not belong to set $\mathcal{C}$. TENOR continues to generate a new channel, 1, again not in set $\mathcal{C}$. The next generated channel is 8, which belongs to set $\mathcal{C}$. Hence, it is selected as the home channel.

In the case that an active node needs to estimate the home channel of its paired passive node, the active node does not know the available channels set of the paired passive node. With TENOR, the active node uses its own available channels set as a substitute, and then follows the same steps as above to 'select' the home channel of the passive node. For example, in Fig. 4, node 3 needs to estimate the home channel of node 6. Node 3 simply uses $\mathcal{C}_3$ as a substitute of $\mathcal{C}_6$, to become set $\mathcal{C}$ in Fig. 5, and then performs the same procedure to find a channel belonging to set $\mathcal{C}$. As node 3 uses the same seed as node 6, the lists of channels generated by both nodes are the same, $\{3, 1, 8\}$. In the figure, channel 8 is in $\mathcal{C}_3$, while channels 3 and 1 are not. Hence channel 8 is the estimated home channel of node 6. As to be shown in the next section, as long as $\mathcal{C}_3$ and $\mathcal{C}_6$ have a large percentage of common channels, which should be the case for two nearby nodes, node 3 has a very good chance to find the channel of node 6.

It should be noted that if the node ID is public, a jammer can estimate the channel of a node and performs jamming attacks. To address this issue, the node ID can be set as a secret key. When a node joins a network, it has to be authenticated first, and then exchanges its node ID with peer nodes. Without knowing the node IDs, an attacker has difficulty to estimate the node operation channels.

When a new node joins the network, if it knows the node ID of one or more other nodes, it can directly estimate the channel of another node. Note that any node has a probability of 0.5 to be a passive node in a time slot. Hence, if the new node continues to estimate the channel of another node, it will eventually meet the other node. From there, the new node obtains the information of the network.

5. Performance Analysis

5.1. Throughput

We first introduce a lemma from [20] for the success probability of channel estimation, and the channel selection probability by a passive node.

Lemma 1. *Let α denote the probability that active node j successfully estimates the home channel of passive node i. Let β denote the probability that passive node i selects channel m as its home channel. Then α and β are given as*

$$\alpha = \frac{p}{2-p}\left(1-(1-p)^{2M}\right), \tag{2}$$

$$\beta = \tfrac{1}{M}\left(1-(1-p)^{M}\right). \tag{3}$$

where p is the channel heterogeneity probability, and M is the total number of available channels.

Remarks: For large M, $\alpha \approx \frac{p}{2-p} = \frac{1}{2\frac{1}{p}-1}$. Hence, α increases toward 1 when p increases toward 1. If $p = 1$, then $\alpha = 1$. In other words, the success probability of channel estimation increases for a larger p, which is the percentage of common channels between two nodes (see Eq. (1)). Hence, when the channel availability of two nodes is highly relevant, i.e., if most of their available channels are the same, the success probability of channel estimation increases, which results in a higher rendezvous probability. For large M, we have $\beta \approx \frac{1}{M}$. Let $X_k = 1$ if channel k is a home channel of at least one passive node and $X_k = 0$ otherwise. The probability that channel k is not selected as home channel by any passive node is $(1-\beta)^{\tilde{N}}$. For $\tilde{N} \ll M$, the mean number of (distinct) home channels, $\sum_{k=1}^{M} X_k$, is given as follows.

$$\begin{aligned}\sum_{k=1}^{M} \mathrm{E}(X_k) &= \textstyle\sum_{k=1}^{M}\left(1-(1-\beta)^{\tilde{N}}\right)\\ &\approx M\left(1-\left[1-\tfrac{1}{M}\right]^{\tilde{N}}\right) \approx M(1-e^{-\frac{\tilde{N}}{M}})\\ &\approx \tilde{N}\frac{(1-e^{-\frac{\tilde{N}}{M}})}{\frac{\tilde{N}}{M}} \approx \tilde{N}\end{aligned} \tag{4}$$

Thus, although there is no coordination among passive nodes for home channel selection, the passive nodes are likely spread onto channels.

As described in the preceding section, in each slot, an active node is paired with one passive node, estimates the home channel of this passive node, and then switches to the estimated home channel for packet transmission. Next we analyze the throughput, given M, N, $\tilde{N}$, and p. We assume that a traditional single-channel MAC protocol is used by the nodes on the same channel, e.g., the IEEE 802.11 protocol. Let $T(n)$ denote the saturation throughput on a channel given that n nodes are on this channel. Let $B(n : N, q)$ denote the Binomial distribution with parameters N and q.

As a general practice in throughput analysis, we analyze the *saturation throughput*, which is the network capacity under stable conditions [23]. As discussed in Section 3, we assume that the network traffic load is high and a node always has either new or backlogged traffic to another node. The throughput obtained under this assumption is thus the saturation throughput for the network. In practice, the throughput will be smaller when a node has no traffic to another node when they rendezvous. However, even under moderate traffic load, among the node pairs that rendezvous at a time slot, at least some of them would have traffic to each other. Hence the throughput can be maintained as good in TENOR.

Theorem 1. *The total saturation throughput S for a CRN with N nodes and total M available channels is given as follows.*

$$S = M \sum_{\tilde{n}=0}^{\tilde{N}} \sum_{\bar{n}=0}^{\tilde{n}} \sum_{k=0}^{\bar{N}-\tilde{n}} B(\tilde{n} : \tilde{N}, \beta) B(\bar{n} : \tilde{n}, \alpha) \times B(k : \bar{N} - \tilde{n}, \frac{1-\alpha}{M-1}) T(\bar{n} + \tilde{n} + k),$$

where $\tilde{N} = \lceil \frac{N}{2} \rceil$, $\bar{N} = \lfloor \frac{N}{2} \rfloor$, α and β are the channel estimation success probability and channel selection probability, respectively, given in (2) and (3).

Proof 1. *Each passive node independently selects its home channel. For a specific channel m, the probability that a passive node selects it as the home channel is β. Thus the number of passive nodes on a channel, denoted as $\tilde{n}$, follows the Binomial distribution $B(\tilde{n} : \tilde{N}, \beta)$. Each active node is paired with one passive node and switches to the estimated home channel of the passive node. Given that channel m has $\tilde{n}$ passive nodes, there are $\tilde{n}$ corresponding active nodes that are paired with them. Among the $\tilde{n}$ active nodes, let $\bar{n}$ be the number of nodes that successfully estimate the channels of the corresponding passive nodes and hence successfully switch to channel m. Since the channel estimation is independent and the success probability of channel estimation is α, $\bar{n}$ follows a Binomial distribution $B(\bar{n}:\tilde{n},\alpha)$.*

Furthermore, the remaining $\bar{N} - \tilde{n}$ active nodes are paired with passive nodes that do not select channel m as the home channel. Each of the $\bar{N} - \bar{n}$ active nodes can incorrectly estimates the channel of the paired passive node with probability $1 - \alpha$. Now let us consider a specific such active node, say node A. Since the channel selection is essentially random, provided that node A incorrectly estimates channel of the paired passive node, the selected channel is uniformly distributed among the remaining $M - 1$ channels. Hence it has the probability $1/(M - 1)$ to be channel m. In conclusion, *each of the remaining $\bar{N} - \bar{n}$ active nodes has a probability of $\frac{1-\alpha}{M-1}$ to fall into channel m, since the channel estimation error probability is $1-\alpha$. Again, since the channel selection is independent, among the $\bar{N} - \bar{n}$ active nodes that are paired with passive nodes not on channel m, the number of nodes that switch to channel m follows a Binomial distribution $B(k : \bar{N} - \tilde{n}, \frac{1-\alpha}{M-1})$.*

The throughput in channel m, denoted as S_m, is then given as

$$S_m = \sum_{\tilde{n}=0}^{\tilde{N}} \sum_{\bar{n}=0}^{\tilde{n}} \sum_{k=0}^{\bar{N}-\tilde{n}} B(\tilde{n} : \tilde{N}, \beta) B(\bar{n} : \tilde{n}, \alpha) \times B(k : \bar{N} - \bar{n}, \frac{1-\alpha}{M-1}) T(\bar{n} + \tilde{n} + k). \quad (5)$$

The total throughput S is simply given as

$$S = \sum_{m=1}^{M} S_m = M \times S_m. \quad (6)$$

5.2. Expected Time to Rendezvous for Pairwise Rendezvous

As noted in the preceding section, TENOR can be adjusted for pairwise rendezvous, i.e., to enable a transmitter to target a given receiver for rendezvous, e.g., in order to deliver some urgent traffic. In this case, the transmitter node treats itself as an active node and seeks to rendezvous with the receiver node. Certainly, the receiver node has no idea that the transmitter node wants to rendezvous with it; hence it behaves like other nodes and follows the normal procedure of TENOR. The transmitter node takes on average $\frac{1}{\alpha}$ time slots to successfully estimate the home channel of the receiver conditioned on that the receiver is a passive node. The receiver has a probability of 0.5 to be a passive node. Hence, a transmitter takes on average $\frac{2}{\alpha}$ time slots to successfully rendezvous with the receiver.

5.3. Expected Time between Rendezvous for Throughput Oriented Rendezvous

Now we discuss the expected time between two rendezvous for two nodes, in the general operation of TENOR, with the objective to maximize the total network throughput, rather than minimize the time to rendezvous for a pair of nodes.

Theorem 2. *The probability that two nodes rendezvous in a slot, denoted as π, is given as*

$$\pi \approx 0.25M\beta^2 + 0.5\left(\frac{1}{\tilde{N}}\alpha + (1-\frac{1}{\tilde{N}})(1-\alpha)\frac{1}{M}\right)$$
$$+ 0.25\left(\alpha\beta^2[(M+2)\alpha - 2] + \frac{1}{M}(1-\alpha^2)\right).$$

The expected time between rendezvous for two nodes is $1/\pi$ time slots.

Proof 2. *Let us consider the rendezvous between two nodes, say nodes i and j. At each time slot, nodes i and j can be in four joint states: PP, PA, AP, and AA, with the same probability, where 'P' indicates passive and 'A' indicates active. Let R_y denote the probability that nodes i and j rendezvous, i.e., they are on the same channel in the current time slot, when they are in joint state y. First of all, two passive nodes are on a given channel, say channel k, with probability β^2. Therefore, $R_{\mathrm{PP}} = M\beta^2$, since there are M channels.*

If two nodes are in state AA, then the rendezvous scenario is more complicated. First of all, the corresponding paired passive nodes of nodes i and j may be on the same channel by accident, which has probability $M\beta^2$ based on the discussion above. In this case, if both nodes correctly estimate the home channel of their paired passive nodes, which has probability α^2, then they are on the same channel and rendezvous. Furthermore, they also rendezvous if both nodes i and j incorrectly estimate the channel of their passive nodes, but by accident they fall on the same channel. Since in Algorithm 1 the channel is essentially picked randomly, when nodes i and j both incorrectly estimate the home channel of the pair nodes, they have an approximate probability of $\frac{1}{M}$ to be on the same channel. In summary, the probability of rendezvous when the pair passive nodes are on the same channel is $\alpha^2 + (1-\alpha)^2\frac{1}{M}$. Second, if the corresponding paired passive nodes of nodes i and j are on different channels, nodes i and j can be on the same channel only if at least one of them incorrectly estimates the home channel of the paired passive node. The probability of rendezvous in this scenario is approximately $(1-\alpha^2)\frac{1}{M}$. Hence we have

$$\begin{aligned} R_{\mathrm{AA}} \approx\ & M\beta^2[\alpha^2 + (1-\alpha)^2\frac{1}{M}] + \\ & (1-M\beta^2)[(1-\alpha^2)\frac{1}{M}] \\ =\ & \alpha\beta^2[(M+2)\alpha - 2] + \frac{1}{M}(1-\alpha^2). \end{aligned}$$

Next we consider nodes i and j in state PA. Since the active and passive nodes are paired with a random permutation, node j has probability of $\frac{1}{\tilde{N}}$ to be paired with node i. In the case that nodes i and j are paired with each other, then node j has probability α to correctly estimate the channel *of node i. In the case that node j is not paired with node i, node j may still rendezvous with node i if node j incorrectly estimates the channel of its paired passive node and falls into the channel of node i. This has an approximate probability of $(1-\alpha)\frac{1}{M}$. Thus we have*

$$R_{\text{PA}} \approx \frac{1}{\tilde{N}}\alpha + (1 - \frac{1}{\tilde{N}})(1-\alpha)\frac{1}{M} = R_{\text{AP}}.$$

The probability that two nodes rendezvous in a slot, denoted as π, is then given as

$$\begin{aligned} \pi =& \Pr(\text{PA})R_{\text{PP}} + 2\Pr(\text{PA})R_{\text{PA}} + \Pr(\text{AA})R_{\text{AA}} \\ \approx& 0.25M\beta^2 + 0.5\left(\frac{1}{\tilde{N}}\alpha + (1-\frac{1}{\tilde{N}})(1-\alpha)\frac{1}{M}\right) \\ &+ 0.25\left(\alpha\beta^2[(M+2)\alpha - 2] + \frac{1}{M}(1-\alpha^2)\right). \end{aligned} \tag{7}$$

Note that π does not depend on traffic loads or slot number. The mean rendezvous time is $1/\pi$ time slots. From the above discussions, the estimation of π in (7) is conservative. Hence $1/\pi$ can be seen as an upper bound for the mean rendezvous time.

6. Performance Evaluation

We evaluate the performance of the proposed rendezvous algorithm TENOR using both the analytical model and simulations. We compare TENOR with three rendezvous algorithms: 1) RCCH [19], which is a recent state-of-the-art synchronous rendezvous algorithm, 2) Jump-Stay (JS) [6], which is an asynchronous rendezvous algorithm widely cited in the literature, and 3) DSCR [18], which is a recent state-of-the-art asynchronous rendezvous algorithm. In simulations for RCCH, DSCR, and JS, when a node hops to an unavailable channel following the channel sequence, we let it randomly pick a currently available channel as a substitute. This increases their rendezvous probability.

We simulate two single hop CRNs, with 16 and 30 nodes, respectively. The simulation time is 20000 milliseconds and the time slot is 10 millisecond. The total number of global channels, $\mathbb{M}$, is assumed 150. We have used various values for other simulation parameters, N, M and p, which are the number of nodes, number of available channels, and the channel heterogeneity parameter, respectively. The IEEE 802.11b MAC protocol is assumed on each channel for multiple access among nodes on this channel. The saturation throughput $T(n)$ for IEEE 802.11b is obtained by the model in [23], and is listed in Table 2.

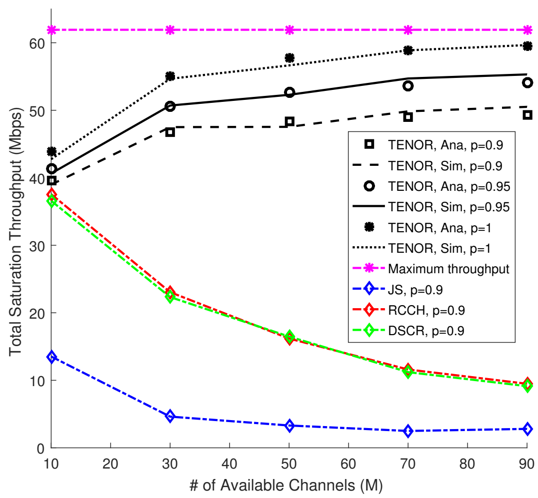

Figure 6: Total saturation throughput with $N = 16$

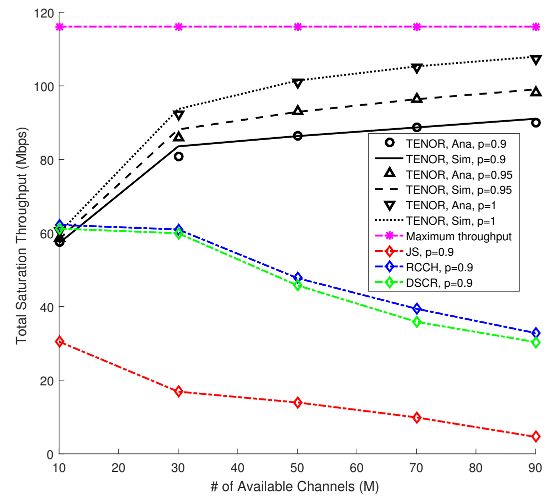

Figure 7: Total saturation throughput with $N = 30$

Figs. 6 and 7 plot the total saturation throughput with respect to the number of available channels (M). Here 'Ana' means analysis results, and 'Sim' means simulation results averaged over all time slots in the simulation period. Fig. 6 exhibits the results for $N = 16$ and Fig. 7 presents the results for $N = 30$. We also plot the throughput of the OptimalThp scheme to measure how close the throughput of other schemes is to the maximum. It is denoted as 'Maximum throughput' in the figures, which is simply $\frac{N}{2}T(2)$, i.e., assuming that the $\frac{N}{2}$ node pairs perfectly rendezvous on $\frac{N}{2}$ distinct channels. We can see that the throughputs of RCCH, DSCR, and JS are all significantly smaller than the one of TENOR, when there are sufficient available channels. Only when the number of available channels is small (10 channels), the throughputs of DSCR and RCCH can be comparable with the one of TENOR. Moreover, when increasing the number of available channels, the throughputs of RCCH, DSCR, and JS all decrease while the one of TENOR increases. This is because with more available channels, the expected time to rendezvous of RCCH, DSCR, and JS increases quickly (see Fig. 11); hence the throughput decreases. On the other hand, with more available channels, the expected time between rendezvous of TENOR largely does not change, while the rendezvous node pairs are more likely spread onto different channels. This results in more channels utilized. Thus the throughput increases, until it reaches a limit that is close to the maximum throughput. From the figures, it is clear that with sufficient available channels, the throughput of TENOR is rather close to the maximum throughput, compared with DSCR and RCCH. Figs. 6 and 7 also indicate that the results from the analysis model match the simulation results well. One may observe that the throughput in Fig. 7 is larger than the one in Fig. 6. This is because the throughput depends on how many channels are used for node rendezvous, which are then utilized for data communications. With more nodes, more number of node pairs are formed for rendezvous on more channels. Thus the throughput is also larger.

From Figs. 6 and 7, one can observe that the channel hopping based schemes perform relatively better when the number of available channels is small. This is because nodes can hop to only a small number of channels. This results in more overlapping channels between channel sequences. Hence the rendezvous probability increases. On the other hand, the small number of available channels also raises a limitation to the throughput performance of those schemes. This is because the small number of available channels translates to less number of channels can be utilized for data communications. More overlapping channels between channel sequences translates to increased rendezvous congestion. Thus, there is a dilemma for channel hopping based rendezvous schemes: 1) with more available channels, the rendezvous probability decreases as the channel overlapping between channel sequences decreases, resulting in throughput decrease; 2) with less available channels, the rendezvous probability increases, but the channels that can be utilized for communications also decreases, resulting in throughput decrease. Therefore, the throughput of channel hopping based schemes is sub-optimal.

Note that the result for 'Maximum throughput' in Figs. 6 and 7 can be seen as the upper bound throughput for an 'ideal' rendezvous scheme using a common control channel when the nodes rendezvous and channel selection are properly coordinated, and there is no congestion. Specifically, if the passive-active pairing approach in TENOR is adopted, and a highly efficient distributed algorithm is used to coordinate all node pairs to select disparate channels for data communications, there will be no co-channel interference. Without considering congestion and overhead, this achieves the maximum throughput illustrated in Figs. 6 and 7. In practice, since nodes need to use multiple access on the common control channel, the actual throughput decreases depending on the congestion level and the efficiency of the distributed channel selection algorithm. Note that to achieve the maximum throughput, nodes still need to be synchronized and paired at the same time, similarly as with TENOR. Otherwise, if the rendezvous is not synchronized, it is difficult to ensure that nodes are perfectly paired, and each pair uses a different channel, which is the foundation to achieve the maximum throughput.

The channel heterogeneity parameter p, which indicates the percentage of channels of a node that are common to a neighbor node as discussed in Section 4, also affects the throughput. The number of common available channels between two nodes is plotted in Fig. 8 as a function of p.

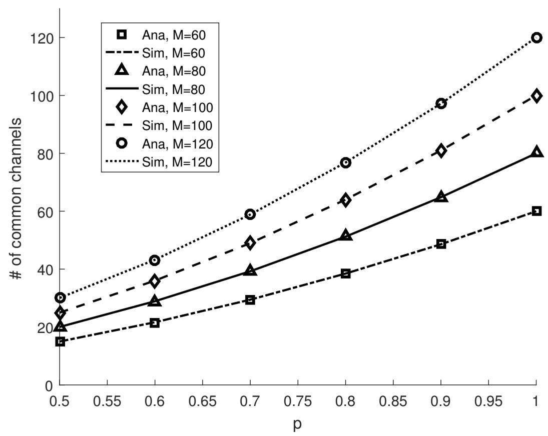

Figure 8: Number of common channels with different p

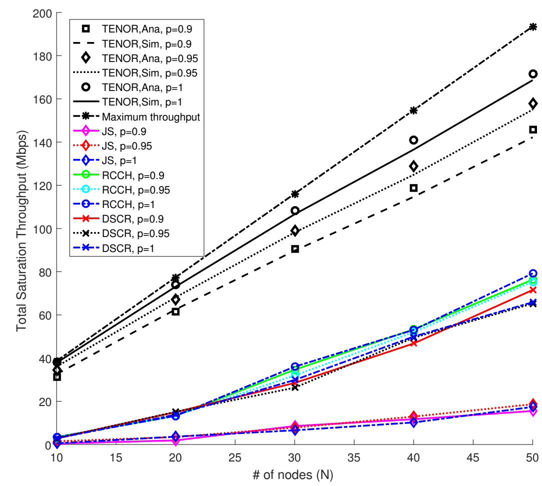

Figure 9: Total saturation throughput with $M = 100$

As discussed in Section 3, the number of common channels is equal to Mp^2 while the number of available channels at a node is Mp. From the figure, the analysis results match the simulation results very well.

With a higher p, the probability that an active node successfully estimates the channel of a passive node is higher. Hence more node pairs successfully rendezvous on their channels, which increases throughput. The effect of the number of nodes on throughput is depicted in Fig. 9, with $M = 100$. The throughput linearly increases as N goes up. As discussed earlier, this is because the throughput depends on the number of channels where nodes rendezvous, while the number of rendezvous channels is approximately the number of node pairs with TENOR. A similar observation is also true for RCCH, DSCR, and JS. In TENOR, not all node pairs perfectly rendezvous and not all rendezvous pairs are on distinct channels. Therefore, the throughput of TENOR is slightly lower than the maximum throughput.

Fig. 10 plots the expected time to rendezvous for RCCH, DSCR, and JS, versus the expected time between rendezvous for TENOR, with $p = 0.9$ and $M = 100$. It is clear from the figure that the expected time between rendezvous of TENOR goes up with an increasing N. This is normal because TENOR aims to let one pair of nodes rendezvous on a separate channel to minimize co-channel interference. Hence, in one time slot, a node can rendezvous with approximately one other node. With a larger N, it takes more time slot for a node to rendezvous with all nodes. Thus, the expected time between rendezvous increases. The analysis results are about 10% larger than the simulation results. This is because the analysis result is actually an upper bound for the expected

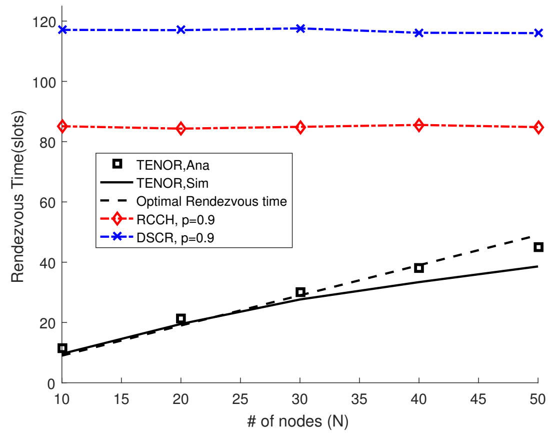

Figure 10: Expected time to rendezvous as a function of the number of nodes, $M = 100, p = 0.9$

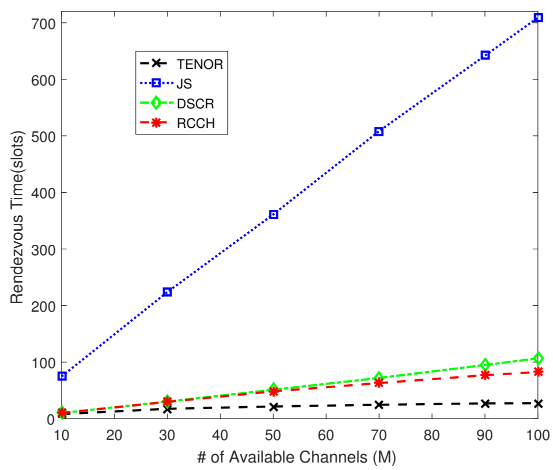

Figure 11: Expected time to rendezvous as a function of the number of available channels, $N = 30, p = 0.9$

time between rendezvous, as discussed in Section 5. We also draw the expected time to rendezvous for RCCH and DSCR as a comparison. The one for JS was not drawn as it is too large compared with other schemes. We can see that the expected time to rendezvous for RCCH and DSCR does not depend on the number of nodes, but is still significantly larger than the one of TENOR. Fig. 10 also depicts the expected time between rendezvous for OptimalThp. Assuming each node has traffic for all other nodes, the minimum number of rendezvous is $N(N-1)/2$. Therefore, the expected time between rendezvous is $(N(N-1)/2)/(N/2) = N-1$, which is depicted 'Optimal rendezvous time' in the figure.

Fig. 11 illustrates the rendezvous time as a function of the number of available channels, in the CRN with 30 nodes. In contrast with Fig. 10, the rendezvous time of RCCH, DSCR, and JS linearly

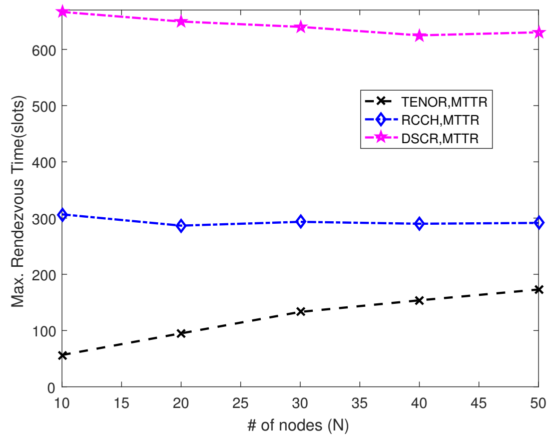

Figure 12: Maximum time to rendezvous (MTTR) as a function of the number of nodes, $M = 100, p = 0.9$

increases, while the one of TENOR changes little. This clearly indicates that the rendezvous time of RCCH, DSCR, and JS primarily depends on the number of channels while the one of TENOR primarily depends on the number of nodes. The maximum time to rendezvous (MTTR) is plotted as a function of the number of nodes, in Fig. 12. We can see that while DSCR and RCCH provide bounded theoretical MTTR, the actual MTTR of DSCR and RCCH from simulations is significantly larger than the actual MTTR of TENOR. The MTTR of RCCH and DSCR does not depend on the number of nodes, while the MTTR of TENOR increases with more number of nodes. Similar to the earlier discussions, this is because the former mainly depends on the number of channels.

7. Conclusion

We have proposed a novel rendezvous scheme termed TENOR for CRNs, and analyzed its performance. The simulation results indicate that TENOR can achieve close to optimal performance in terms of throughput and rendezvous time. Moreover, the analytical results match the simulation results very well. Compared with the recent state-of-the-art channel hopping based rendezvous algorithms, TENOR offers significantly better throughput as well as smaller rendezvous time.

8. Acknowledgments

The research of C. Xin was supported in part by NSF under grants CNS-1418012, CNS-1745632, and CNS-1659795. The research of M. Song was supported in part by NSF under grant CNS-1526152. The research of Q. Gu was supported in part by the Humanities and Social Science Foundation of the Ministry of Education of China (15YJAZH015).

References

[1] K. B. Letaief, W. Zhang, Cooperative Communications for Cognitive Radio Networks, in: Proceedings of the IEEE, Vol. 97, 2009, pp. 878–893.

[2] H. Su, X. Zhang, Cross-layer based opportunistic MAC protocols for QoS provisionings over cognitive radio wireless networks, IEEE J.Sel. Areas Commun. 26 (1) (2008) 118–129.

[3] J. Jia, Q. Zhang, X. S. Shen, HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management, IEEE J. Sel. Areas Commun. 26 (1) (2008) 106–117.

[4] B. Hamdaoui, K. G. Shin, OS-MAC: An efficient MAC protocol for spectrum-agile wireless networks, IEEE Trans. Mobile Comput. 7 (8) (2008) 915–930.

[5] Y. Zhang, Q. Li, G. Yu, B. Wang, ETCH: Efficient channel hopping for communication rendezvous in dynamic spectrum access networks, in: Proc. IEEE Infocom, 2011.

[6] Z. Lin, X. Chu, Y. W. Leung, H. Liu, Jump-stay based channel hopping algorithm with guaranteed rendezvous for cognitive radio networks, in: Proc. IEEE INFOCOM, 2011, pp. 2444–2452.

[7] J. M. Park, K. Bian, R. Chen, A quorum-based framework for establishing control channels in dynamic spectrum access networks, in: Proc. ACM MOBICOM, 2009, pp. 25–36.

[8] K. Bian, J. M. Park, Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks, IEEE Trans. Mobile Comput. 12 (7) (2013) 1294–1307.

[9] J. P. Sheu, C. W. Su, G. Y. Chang, Asynchronous quorum-based blind rendezvous schemes for cognitive radio networks, IEEE Trans. Commun. 64 (3) (2016) 918–930. doi:10.1109/TCOMM.2015.2508803.

[10] R. Paul, Y. J. Choi, Adaptive rendezvous for heterogeneous channel environments in cognitive radio networks, IEEE Trans. Wireless Commun. 15 (11) (2016) 7753–7765. doi:10.1109/TWC.2016.2607170.

[11] B. Yang, W. Liang, M. Zheng, Y. C. Liang, Fully distributed channel-hopping algorithms for rendezvous setup in cognitive multiradio networks, IEEE Trans. Veh. Technol. 65 (10) (2016) 8629–8643. doi:10.1109/TVT.2015.2510359.

[12] L. Yu, H. Liu, Y. W. Leung, X. Chu, Z. Lin, Multiple radios for fast rendezvous in cognitive radio networks, IEEE Trans. Mobile Comput. 14 (9) (2015) 1917–1931. doi:10.1109/TMC.2014.2366746.

[13] Z. Gu, Q.-S. Hua, Y. Wang, F. C. M. Lau, Nearly optimal asynchronous blind rendezvous algorithm for cognitive radio networks, in: 2013 IEEE International Conference on Sensing, Communications and Networking (SECON), 2013, pp. 371–379. doi:10.1109/SAHCN.2013.6645007.

[14] I. H. Chuang, H. Y. Wu, Y. H. Kuo, A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks, IEEE Transactions on Mobile Computing 13 (10) (2014) 2171–2184. doi:10.1109/TMC.2013.2297313.

[15] Z. Gu, H. Pu, Q. S. Hua, F. C. M. Lau, Improved rendezvous algorithms for heterogeneous cognitive radio networks, in: 2015 IEEE Conference on Computer Communications (INFOCOM), 2015, pp. 154–162. doi:10.1109/INFOCOM.2015.7218378.

[16] L. Chen, K. Bian, L. Chen, C. Liu, J.-M. J. Park, X. Li, A group-theoretic framework for rendezvous in heterogeneous cognitive radio networks, in: Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '14, ACM, New York, NY, USA, 2014, pp. 165–174. doi:10.1145/2632951.2632988.
URL `http://doi.acm.org/10.1145/2632951.2632988`

[17] Z. Gu, Q.-S. Hua, W. Dai, Fully distributed algorithms for blind rendezvous in cognitive radio networks, in: Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '14, ACM, New York, NY, USA, 2014, pp. 155–164. doi:10.1145/2632951.2632981.
URL `http://doi.acm.org/10.1145/2632951.2632981`

[18] B. Yang, M. Zheng, W. Liang, A time-efficient rendezvous algorithm with a full rendezvous degree for heterogeneous cognitive radio networks, in: Proc. IEEE INFOCOM, 2016.

[19] G. Y. Chang, W. H. Teng, H. Y. Chen, J. P. Sheu, Novel channel-hopping schemes for cognitive radio networks, IEEE Transactions on Mobile Computing 13 (2) (2014) 407–421. doi:10.1109/TMC.2012.260.

[20] C. Xin, L. Ma, M. Song, C. C. Shen, Performance analysis of a control-free dynamic spectrum access scheme, IEEE Trans. Wireless Commun. 10 (12) (2011) 4316–4323.

[21] C. Xin, M. Song, L. Ma, C. C. Shen, ROP: Near-optimal rendezvous for dynamic spectrum access networks, IEEE Trans. Veh. Technol. 62 (7) (2013) 3383–3391.

[22] L. Chen, K. Bian, M. Zheng, Never live without neighbors: From single- to multi-channel neighbor discovery for mobile sensing applications, IEEE/ACM Transactions on Networking 24 (5) (2016) 3148–3161.

[23] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE J. Sel. Areas Commun. 18 (3) (2000) 535–547.

Algorithm 1: Throughput oriented lightweight near-optimal rendezvous (TENOR)

Input: Node ID i, random number generators Z_s, Z_c

`// Generate the nodes status`

Set t as the seed for the pseudo-random number generator Z_s

Use Z_s to generate a random permutation $\{g_1, \ldots, g_N\}$ from an N-element binary vector $\{1, \ldots, 1, 0, \ldots, 0\}$, where the first $\lceil \frac{N}{2} \rceil$ number of elements are 1

g_i is the status of node i, active or passive, in the current slot

Let $\mathcal{V} = \{V_1, \ldots, V_{\lfloor \frac{N}{2} \rfloor}\}$ denote the IDs of active nodes

Let $\tilde{\mathcal{V}} = \{\tilde{V}_1, \ldots, \tilde{V}_{\lceil \frac{N}{2} \rceil}\}$ denote the IDs of passive nodes

Use Z_s to generate a random permutation of the vector $\{1, \ldots, \lceil \frac{N}{2} \rceil\}$, denoted as $\{j(1), \ldots, j(\lceil \frac{N}{2} \rceil)\}$ `// Node` V_k `pairs with node` $\tilde{V}_{j(k)}$

if $g_i = 0$ **then**

  Let V_k in $\mathcal{V}$ denote node i `// Node` i `is an active node`

  Use the ID of the passive node paired by node i, i.e., $\tilde{V}_{j(k)}$, as the seed for the pseudo-random number generator Z_c

  `// Estimate home channel of node` $\tilde{V}_{j(k)}$

  Let $\mathcal{H} = \{1, \ldots, \mathbb{M}\}$

  repeat

    $k = Z_c(|\mathcal{H}|)$

    $h = \mathcal{H}(k)$

    $\mathcal{H} = \mathcal{H} \backslash \{h\}$

  until $h \in \mathcal{C}_i$

  Node i switches to the estimated channel h

else

  Set i as the seed for the pseudo-random number generator Z_c

  Let $\mathcal{H} = \{1, \ldots, \mathbb{M}\}$

  repeat

    $k = Z_c(|\mathcal{H}|)$ `// Generate a random number` k

    $h = \mathcal{H}(k)$ `// The` k`th channel in` $\mathcal{H}$

    $\mathcal{H} = \mathcal{H} \backslash \{h\}$

  until $h \in \mathcal{C}_i$

  Node i switches to home channel h

Table 2: Throughput of IEEE 802.11b (Mbps)

n	2	3	4	5	6	7	8
$T(n)$	7.74	7.72	7.64	7.55	7.45	7.36	7.28