A New Communication Framework for Wide-band Cognitive Radio Networks

Ji Li and Jiang Xie
Department of Electrical and Computer Engineering
The University of North Carolina at Charlotte
Email: {jili50, Linda.Xie}@uncc.edu

Abstract—Cognitive radio is a promising technology to solve the spectrum scarcity problem. In a cognitive radio network, a secondary user (SU) first senses a specific range of the spectrum to get its available channels and two SUs need to rendezvous on a common available channel to establish a physical link for communications. The spectrum band allocated for existing wireless services is very wide. However, when a SU works in wide-band spectrum, the increasing delay and energy consumption in spectrum sensing and the time for two SUs to rendezvous will be a serious problem. All the existing rendezvous papers only focus on designing efficient algorithms for guaranteed rendezvous without considering practical communication issues caused by the wide-band spectrum. In this paper, we propose a novel communication framework for wide-band cognitive radio networks without a common control channel. Our proposed framework can efficiently solve the problems coming from the wide-band spectrum and newly joined SUs. Simulation results show that our proposed scheme can achieve fast rendezvous and a high network throughput. To the best of our knowledge, this is the first work with an effective solution to improve the communications in a wide-band cognitive radio network.

I. INTRODUCTION

According to the Federal Communications Commission (FCC), up to 85% of the spectrum allocated to existing wireless services is under utilized [1]. Cognitive radio is a promising technology which can realize the dynamic spectrum access to improve the utilization of the allocated spectrum [2]. In a cognitive radio network (CRN), unlicensed users, called secondary users (SUs), coexist with licensed users, called primary users (PUs). SUs are equipped with one or multiple cognitive radios which can opportunistically access the currently available channels, but have to vacate the channels for the returning PUs [3] [4]. In order to avoid generating interference to PUs, a SU should perform spectrum sensing to get its available channels before accessing any channel.

In a CRN, the available channel sets of a communicating SU pair are the important resources for establishing a physical link [5]. According to [6], there are two models based on the available channel sets of a SU pair, the symmetrical model: the available channel sets of any SU pair are the same, and the asymmetrical model: the available channel sets of any SU pair are different. In order to establish a link between a SU pair, the two SUs should first meet on a common available channel. This process is called rendezvous.

The initialization of a CRN is critical, because each SU may not know any information about other SUs in the same network before information exchange. However, due to the dynamic characteristics of CRNs, obtaining the basic but time-varying control information, such as the available channel sets and locations, of SUs is very challenging. For simplicity, many existing papers address this challenge by assuming the existence of a common control channel (CCC) in the network and using the CCC for all control information exchange [7]–[9]. However, a CCC may suffer the following problems: First, a CCC may not be available simultaneously for all the SUs in a whole network. Second, even though a CCC exists, due to the dynamics of CRNs, its availability over the time is subject to the PU traffic. Third, a single CCC in a network may have the congestion problem and is fragile under attack. In order to overcome these drawbacks, some schemes are proposed to form several SU clusters in a network and choose a common available channel as the CCC in each cluster [10], [11]. However, in order to form a cluster, each SU has to know some information about its neighboring SUs, which is not practical for a CRN without any control information exchange before the initialization.

Recently, blind rendezvous schemes without using a CCC are proposed by implementing a time-slotted system and channel hopping. During a blind rendezvous, a SU hops to a different channel at the beginning of each time slot by following a channel hopping sequence and sends out a Request-to-Send (RTS) message on each channel. If the SU receives a Clear-to-Send (CTS) message from another SU during the same time slot, it indicates that the two SUs meet on the same channel and the rendezvous is successful. Since no information exchange is needed initially, blind rendezvous is practical for CRNs. However, it is very challenging to design an efficient rendezvous algorithm that can guarantee rendezvous without knowing any information of other SUs. In order to evaluate a rendezvous scheme, three performance metrics are defined [12]: \( TTR \), \( MTTR \), and \( ETTR \), where \( TTR \) is the time to rendezvous from the moment the channel hopping starts until the moment two SUs meet on a common available channel, \( MTTR \) is the maximum possible time required to have a successful rendezvous, and \( ETTR \) is the expectation of the time to a successful rendezvous.

The original purpose of cognitive radio is to improve the utilization of the spectrum which has been allocated. However, according to the FCC, the allocated spectrum, ranging from 3KHz to 300GHz, is very wide. Though we cannot expect cognitive radio to work in that wide spectrum, the number of channels a cognitive radio can access may be hundreds or thousands. For instance, according to [7], the TV band which can be utilized by SUs is from 54MHz to 862MHz and the bandwidth of each channel is 6MHz. This corresponds to a total of 134 channels. With the increase of the number of channels in a CRN, the time for a SU to sense the whole band and the energy consumed in the sensing process will increase. In [13], the importance of considering the wide-band scenario is explored and a scheme for mobile access of wide-band networks is proposed. However, to the best of our knowledge, there are no existing papers which can efficiently solve the communication problems in CRNs coming from the wide-band spectrum without a CCC. Therefore, addressing the
problems stemming from the wide-band spectrum is necessary.

In [6], [14]–[19], several rendezvous schemes for CRNs are proposed. However, these schemes only focus on the rendezvous problem itself while ignoring the following practical communication scenarios in wide-band CRNs. First, they do not consider the wide-band scenario. All of them assume that there are only a small number of channels (at most tens of channels) in the network and SUs will utilize all the channels to design the channel hopping sequence and perform channel hopping, while there may be hundreds of channels for a CRN. As the number of channels increases, the time to sense all the channels to get available channels and to rendezvous will increase significantly, according to the results shown in [12]. Second, they do not consider the potential collisions when multiple pairs of SUs rendezvous on the same channel. Thus, a practical communication framework for wide-band CRNs without a CCC that can address the above problems is needed.

In this paper, we propose a novel and practical communication framework in wide-band CRNs. The basic idea of our proposed framework is to divide the whole spectrum into several virtual spectrum segments. Then, each SU is intelligently mapped to a specific spectrum segment. Each spectrum segment has a particular channel hopping sequence for all the SUs belonging to this segment. If a SU wants to communicate with another SU, it first finds the spectrum segment of the SU receiver and then hops to that segment to execute our designed blind rendezvous algorithms which can guarantee successful rendezvous. All the algorithms we designed are distributed algorithms without relying on a centralized entity or CCC. Our proposed framework is practical and can be easily applied to a wide-band CRN to guarantee rendezvous and efficient communications between SUs. To the best of our knowledge, this is the first work that can address both the initialization and communication problems in a wide-band CRN.

The contributions of this paper are summarized as follows.

1) We propose a framework that can address the issue of the increasing delay of communications and spectrum sensing coming from the wide-band spectrum in a CRN through intelligently splitting the wide-band spectrum into several virtual spectrum segments and mapping each SU to a specific spectrum segment.

2) We design efficient distributed blind rendezvous algorithms which can guarantee rendezvous and have a low MTTR and ETRR for both the symmetrical and asymmetrical models.

3) We consider practical communication scenarios in our proposed framework between SUs and in a dynamic CRN with newly joined SUs.

4) We evaluate our designed blind rendezvous algorithms both theoretically and using simulations and demonstrate the advantages of our proposed framework in practical communication scenarios.

The rest of this paper is organized as follows. In Section II, we explain the system model considered in this paper. In Section III, we introduce the basic idea and the flow charts of our proposed framework. In Section IV, we present the details of our proposed blind rendezvous schemes. The proposed spectrum splitting scheme is introduced in Section V. Simulation results are shown in Section VI, followed by the conclusions in Section VII.

II. SYSTEM MODEL

We consider a CRN, where SUs can dynamically access a wide-band spectrum which has a total of \( M \) channels indexed from 1 to \( M \). We assume that \( M \) is a large number which could be hundreds. Each SU has a unique ID which can be represented as a unique positive integer. Furthermore, we assume that each SU can obtain its available channel set using a spectrum sensing algorithm [3]. Initially, a SU does not know any information about other SUs, even their unique IDs. A time-slotted system is adopted in this CRN. In each time slot, a SU either hops onto a channel according to its hopping sequence or stays on its current channel to continue the current communication with another SU. Time slots of different SUs are not necessarily synchronized, which is practical and easy to implement. SUs in the CRN attempt to get information about other SUs within its transmission range, or send to other SUs and receive packets from other SUs.

In the rest of the paper, we consider the following two scenarios concerning the available channel sets of a SU pair.

Symmetrical model: Any SU pair has the same available channel set. According to [20], the similarity of the available channel sets between two SUs within one hop is over 85%. Thus, this model is reasonable. Moreover, studying this scenario can help us to get solutions for more complicated scenarios.

Asymmetrical model: Any SU pair has different available channel sets. This scenario is more practical because of the dynamics caused by PU and SU traffic and mobility.

III. THE PROPOSED FRAMEWORK

The goal of our proposed framework is to design efficient distributed schemes for each SU to guarantee that it can communicate with other SUs successfully in a wide-band CRN. The framework mainly contains three parts: the initialization process, the process to get control information from neighboring SUs, and the process to send packets to a specific SU. The flow charts of these three parts are shown in Fig. 1.

![Fig. 1. The flow charts of our proposed framework.](image-url)
splitting list (SSL). Using the SSL, a SU can obtain the start and end channel in each SS (shown in Algorithm 1). Then, each SU uses its ID (based on Algorithm 1) to determine its home spectrum segment (HS). A SU hops from a channel to another according to the designed channel hopping sequence (based on Algorithm 2 which is explained in Section IV) within its HS. Since SUs are not synchronized, they may be at different positions of the same hopping sequence. This is the initialization process of a SU who just starts in a CRN, as shown in the left of Fig. 1.

Algorithm 1 The algorithm to determine a SU’s home spectrum’s index

Input:

The SU’s ID $x$.

Output:

The SU’s home spectrum’s index;

1. Use Algorithm 5 to get the SSL $l$;
2. $y = (x - 1)\%M + 1$;
3. $sum = 0$;
4. For $i = 0$ to $l$.length $- 1$ do
5. If $y > sum$ and $y \leq sum + l[i]$ then
6. return $i$;
7. end if
8. $sum += l[i]$;
9. end for

After the initialization process, a SU may attempt to get the control information of all its neighboring SUs which are hopping in different spectrum segments (SSs). We assume that when a SU is not transmitting data packets, it performs Algorithm 3 or Algorithm 4 (which is explained in Section IV) to rendezvous with a SU in its home spectrum segment (HS) and exchange their control information. Through this process, each SU already in the CRN can get all the other SUs’ information within the same SS. Therefore, after rendezvous and exchange control information with one existing SU in a SS, a newly joined SU can get the control information of all the SUs in that SS, and all the existing SUs can get the information of the newly joined SU. Then, the newly joined SU can sequentially hop to each SS, rendezvous with one SU in that SS, and exchange all the known control information about other SUs by executing Algorithm 3 or Algorithm 4. Finally, the new SU will obtain and update the control information of all other SUs in the network. A SU can perform the above process periodically to update the control information about other SUs due to the dynamic characteristics of CRNs. The process of getting the control information of all the neighboring SUs is shown in the middle of Fig. 1.

After the above process of control information exchange, a SU will get some information about other SUs such as their IDs. When a SU wants to send packets to a specific SU, it first uses the receiver’s ID to determine the receiver’s home spectrum segment (HS) (based on Algorithm 1). Then, the SU sender executes Algorithm 3 or Algorithm 4 to realize rendezvous with the SU receiver in the SU receiver’s HS for communications. The flow charts of communicating with a specific SU is shown in the right of Fig. 1. Because of the spectrum splitting design, the time to obtain control information and to set up communications can be significantly reduced since a SU pair only needs to rendezvous within a specific spectrum segment (SS). In addition, the probability that multiple pairs of SUs rendezvous on the same channel is also reduced since different pairs of SUs may rendezvous in different spectrum segments (SSs). In the following sections, we will give the details of the proposed algorithms in our framework.

IV. PROPOSED RENDEZVOUS SCHEMES

A. Under the Symmetrical Model

A SU sender and receiver have the same available channel set under the symmetrical model. This may not be very practical. However, the algorithms developed under this model are the basis for a more practical model.

Under this scenario, each SU has the same available channels in each SS. Each SU hops within its HS according to a channel hopping sequence generated based on its available channel set. Assume that the number of the available channels of both the SU sender and receiver in the SU receiver’s HS is $n$.

Since a SU sender and receiver have the same available channel set, we use the indexes of all the available channels in a SS, assuming to be from 1 to $n$, to design the channel hopping sequence. We desire to get a target sequence $s = s_1 s_2 \ldots s_{2n}$ whose length is $2n$ and each channel index from 1 to $n$ appears exactly twice in $s$. When a SU sender wants to rendezvous with a SU receiver, it first hops to the HS of the SU receiver, generates the same channel hopping sequence as the SU receiver’s, and then starts to hop on channel $s_1$. Both the SU sender and receiver hop on each channel according to the sequence sequentially and circularly, which means that when a SU reaches the end of the hopping sequence, it will continue to hop onto the first channel of the sequence in the next time slot. Assume that currently the SU receiver is on channel $s_{cur}$, $1 \leq cur \leq 2n$. We define the rendezvous sequence (RS) as follows:

Rendezvous Sequence (RS): The sequence $s$ is that no matter what $s_{cur}$ is, within $2n - 1$ time slots, the SU sender and receiver will hop on a same channel if both of them hop according to the same hopping sequence sequentially and circularly.

We use $s$ to represent our desired rendezvous sequence for the rest of the paper. We induce the following lemma to help us design our rendezvous sequence.

Lemma 1. Assume that $s_i = s_j = k$, $i < j$, $0 < k \leq n$. If $j - i = k$, $s$ could be a RS.

Proof: The Sequential Distance of the element $k$ in the sequence $s$ is defined as $d_k = j - i - k$. The Circular Distance of the element $k$ in the sequence $s$ is defined as $D_k = 2n + i - j = 2n - k$, which is the number of time slots taken to hop from $s_j$ to $s_i$ circularly in $s$. We define the distance pair of the channel index $s_{p_k}$ $= (d_k, D_k)$. Since each $d_k$ is different and bounded in $[1, n]$ and each $D_k = 2n - d_k$ is different and bounded in $[n, 2n - 1]$, each $p_k$ is different. Assume that initially the SU sender is on channel $s_1$ and the SU receiver is on channel $s_{cur}$. Let $\alpha = cur - 1$ be the distance from $s_1$ to $s_{cur}$. Therefore, $0 \leq \alpha \leq 2n - 1$. We ignore the case when $\alpha = 0$, because under this case the two SUs are already on the same channel. Thus, we only need to prove our lemma when $1 \leq \alpha \leq 2n - 1$. Since each $p_i$ is different, we can definitely find a $p_s$ such that $s_{p_s} = d_i$ or $D_i$. Suppose that $s_{i1} = s_{j1} = t$, $i1 < j1$. Thus, when the SU sender hops on channel $s_{i1}$, it will rendezvous with the SU receiver on channel $s_{j1}$, or when the SU sender hops on channel $s_{j1}$, it will rendezvous with the SU receiver on channel $s_{i1}$. Therefore, sequence $s$ can guarantee rendezvous no matter which channel the SU receiver dwells on at the beginning of the rendezvous process.
In order to efficiently generate \( s \), we notice that \( s \) has the following properties.

**Lemma 2.** When \( n = 4t + 2 \) or \( 4t + 3 \), where \( t \) is an integer and \( t \geq 1 \), the rendezvous sequence does not exist.

**Proof:** Designing a rendezvous sequence \( s \) as described in Lemma 1 is equivalent to dividing \( 1, 2, \ldots, 2n \), these \( 2n \) numbers into two lists \( l_l \) and \( l_r \), each of which has exactly \( n \) different numbers, such that for \( k = 1, 2, \ldots, n \), \( l_r[k] - l_l[k] = k \). Then, we have

\[
\sum_{k=1}^{n} l_r[k] = \sum_{k=1}^{n} l_l[k] = \frac{n(n+1)}{2}, \tag{1}
\]

\[
\sum_{k=1}^{n} l_r[k] + \sum_{k=1}^{n} l_l[k] = 2n = n(2n+1). \tag{2}
\]

Combining (1) and (2), we can get

\[
\sum_{k=1}^{n} l_r[k] = \frac{5n^2 + 3n}{4}. \tag{3}
\]

When \( n = 4t + 2 \) or \( 4t + 3 \), where \( t \) is an integer and \( t \geq 1 \), the right-hand side of (3) is not an integer which contradicts the fact that it is a sum of \( n \) integers.

**Lemma 3.** When \( n = 4t \) or \( 4t + 1 \), where \( t \) is an integer and \( t \geq 1 \), the rendezvous sequence \( s \) always exists.

**Proof:** When \( n = 4 \), sequence “1, 1, 4, 2, 3, 2, 4, 3” satisfies the requirements of our desired rendezvous sequence. When \( n = 5 \), sequence “1, 1, 5, 2, 4, 2, 3, 5, 4, 3” satisfies the requirements of our desired rendezvous sequence. When \( n > 5 \), the proof can be found in the second page of [21].

Based on the proof of Lemma 2 and Lemma 3, we propose Algorithm 2 to generate our desired rendezvous sequence. Using the channel hopping sequence generated based on Algorithm 2 and Lemma 1, we design the rendezvous algorithm under the symmetrical model as shown in Algorithm 3.

We define a round of the channel hopping as the length of the \( 2n \) time slots a SU hops according to the channel hopping sequence sequentially and circularly. An example of rendezvous in one channel hopping round under the symmetrical model is shown in Fig. 2. We assume that the available channel set of a SU pair is \{1, 2, 3, 4\}. Based on Algorithm 2, their channel hopping sequence is “1, 1, 4, 2, 3, 2, 4, 3”. Assume that the SU sender starts hopping from channel \( s_1 = 1 \) and the SU receiver currently stays on channel \( s_3 = 3 \). Then, the SU sender and receiver will rendezvous on channel \( s_4 = 2 \) after 3 time slots.

From Lemma 1 and Algorithm 3, we can induce Theorem 1.

**Theorem 1.** Under the symmetrical model, Algorithm 3 can guarantee rendezvous in one round of the channel hopping. The \( MTR \) of guaranteed rendezvous is \( 2m - 1 \) and the \( ETTR \) is \( m - 1 \), where \( m = n \) if \( n \% 4 = 0 \) or \( n \% 4 = 1 \) and \( m = 4(\lfloor n/4 \rfloor + 1) \) if \( n \% 4 = 2 \) or \( n \% 4 = 3 \).

**Proof:** From Lemma 1 and Algorithm 3, the length of our channel hopping sequence is \( 2m \) and our rendezvous algorithm can guarantee rendezvous. Thus, according to Algorithm 3, the \( MTR \) is \( 2m - 1 \). When the SU sender is on channel \( s_1 \), assume that the SU receiver has equal probability, which is \( 1/2m \), to be on any channel from \( s_1 \) to \( s_{2m} \). According to the proof of Lemma 1, the total number of time slots needed to rendezvous of all possible situations is \( \sum_{i=0}^{2m-1} i = m(2m-1) \). Thus, the \( ETTR \) is \( m(2m-1)/2m \approx m - 1 \).
Algorithm 3 The rendezvous algorithm for a SU sender under the symmetrical model

Input:
- The available channel set $C$ of the SU receiver within the SU sender’s HS.
1. $n$ equals the size of $C$;
2. if $n \% 4 = 2$ or $n \% 4 = 3$ then
3. $m = 4([n/4] + 1)$; /* To guarantee the existence of the RS */
4. else
5. $m = n$;
6. end if
7. Generate the channel hopping sequence $s$ based on $m$ and Algorithm 2;
8. for $k = 1$ to $2m$ do
9. if $s_k > n$ then
10. $s_k = s_k \% n$; /* To guarantee $s_k$ to be within $n$ */
11. end if
12. end for
13. $i = 1$;
14. for $j = 1$ to $2m - 1$ do
15. if the sender receives a CTS packet on channel $C_{s_i}$ of the receiver’s HS then
16. Perform communications;
17. else
18. $i++$;
19. end if
20. end for

B. Under the Asymmetrical Model

Under the asymmetrical model, the SU sender and receiver have different available channel sets. The challenge here is that before rendezvous happens, the SU sender does not know the available channel set of the SU receiver. Here, we assume that they have at least one common available channel. Otherwise, they can never rendezvous. As a SU pair may have different available channel sets, each SU’s channel hopping sequence can only include its available channels within the SU receiver’s HS. We assume that there are $n$ channels in the SU receiver’s HS which are indexed from 1 to $n$. We still use Algorithm 2 to design the base channel hopping sequence for both the SU sender and receiver. Additionally, we will replace the unavailable channels in the base channel hopping sequence with the ones randomly chosen from their own available channels to get their own channel hopping sequences. According to Lemma 1, the replacement will not change the property of our channel hopping sequence. Both the SU sender and SU receiver hop according to their own channel hopping sequence. The SU sender will send the RTS packet on each channel it hops on and the SU receiver listens to each channel it hops on to wait for a SU sender.

Our goal is that we try to design a rendezvous algorithm which can guarantee rendezvous on different channels during several rounds of the channel hopping until rendezvous on the channel which is commonly available for a SU pair. The spectrum splitting scheme in Section V can guarantee the existence of the channel hopping sequences. According to the property of the rendezvous sequence, we can simply change the starting position of a SU sender’s channel hopping during each channel hopping round. During a new channel hopping round, after changing the distance between the SU pair’s positions in the channel hopping sequence, the SU sender and receiver can rendezvous on a new channel. The details of the rendezvous algorithm under the asymmetrical model is shown in Algorithm 4.

Algorithm 4 The rendezvous algorithm for a SU sender under the asymmetrical model

Input:
- The SU receiver’s ID $x$.
- The available channel set $C$ of the SU sender in the SU sender’s HS.
1. Use Algorithm 5 to get the SSL of the whole spectrum band: $l$;
2. Use Algorithm 1 to get the index of the HS of the SU receiver: $r$;
3. $n = l[r]$; /* The index starts from 0 */
4. Generate the channel hopping sequence $s$ based on $n$ and Algorithm 2;
5. for $i = 1$ to $2n$ do
6. $j = i$; /* The starting hopping position of the current round */
7. for $k = 0$ to $2n - 1$ do
8. $\nu = s_j$;
9. if $\nu \notin C$ then
10. Let $\nu$ be a randomly chosen channel in $C$;
11. end if
12. if the rendezvous on $\nu$ succeeds then
13. Perform communications;
14. else
15. $j++$;
16. if $j > 2n$ then
17. $j = 1$/* Circularly hopping */
18. end if
19. end if
20. end for
21. end for

An example of rendezvous under the asymmetric model is shown in Fig. 3. Assume that the available channels of the SU sender are $\{1, 2\}$ and the available channels of the SU receiver are $\{1, 3\}$. Assume that $n = 4$, so the base channel hopping sequence is "1, 1, 4, 2, 3, 2, 4, 3". During the first round of channel hopping, the SU sender starts from channel $s_1 = 1$ and we assume that the SU receiver is on channel $s_3 = 4$. During the first round, after the random replacement process, the channel hopping sequences for the SU sender and SU receiver are "1, 1, 2, 2, 1, 2, 2, 2" and "3, 3, 3, 1, 3, 3, 1, 1", respectively. According to the channel hopping sequences, the SU sender and receiver do not rendezvous on the same channel for the first round. Then, the SU sender continues the second round of rendezvous by starting from channel $s_2 = 1$ and the base channel hopping sequence will be "1, 4, 2, 3, 2, 4, 3, 1". However, the SU receiver will keep using the base rendezvous sequence "1, 1, 4, 2, 3, 2, 4, 3" during the second round. Then, during the second round, after the random replacement process, the channel hopping sequences for the SU sender and SU receiver are "1, 1, 2, 2, 1, 2, 1, 1" and "3, 3, 3, 3, 1, 3, 1, 1", respectively. Finally, the SU sender and receiver rendezvous on channel 1 which is available for both of them at the end of the second round after 15 time slots.

According to the conclusions under the symmetrical model and Algorithm 4, we can induce Theorem 2.

Theorem 2. Under the asymmetrical model, Algorithm 4 can guarantee rendezvous. The MITR of the guaranteed rendezvous is $4n^2$, and the upper bound of the ETTR is
approximately \((2n - 1)/p\), where \(p\) is the probability that a channel is available for both the SU sender and receiver.

**Proof:** The length of the designed sequence \(s\) is \(2n\). According to Algorithm 4, during \(2n\) channel hopping rounds, the initial distance between the SU sender and receiver on the sequence \(s\) will have \(2n\) different values from 0 to \(2n - 1\). Thus, according to the proof of Lemma 1, during \(2n\) rounds, the SU sender and receiver could rendezvous on each channel from 1 to \(n\) at least once, which guarantees rendezvous as long as they have at least one common available channel.

Since the probability that a channel is available for both the SU sender and receiver is \(p\), the probability that a successful rendezvous does not happen until the \(k\)th round is \((1 - p)^{k-1}p\). Therefore,

\[
ETTR \leq \sum_{k=1}^{2n} (1 - p)^{k-1} pk(2n - 1) = \frac{1 - (1 - p)^{2n}}{p} - 2np^2n (2n - 1) \approx \frac{2n - 1}{p}.
\]

Note that the \(O(n)\) \(ETTR\) of our proposed rendezvous algorithms is much better than other existing rendezvous algorithms under the asymmetrical model whose \(ETTR\) is equal to or over \(O(n^2)\) such as the ones in [6], [19], [22].

V. PROPOSED SPECTRUM SPLITTING SCHEME

First, our proposed spectrum splitting algorithm is based on the following mathematical truths.

**Lemma 5.** For any integer \(a = 4t + 2, t > 2\), \(a\) can be represented as \(a = x + y\), where \(x = 4u + 1, u \geq 1\) and \(y = 4v + 1, v \geq 1\), \(a, t, u, v\) are integers.

**Proof:** When \(a = 4t + 2, t > 2\), if \(t = 2, q\) is an integer and \(q \geq 1\), then \(a = 8q + 2\), thus \(a = (4q + 1) + (4q + 1)\). If \(t = 2q + 1, q \geq 1\), then \(a = 8q + 6\), thus \(a = 4q + 1\).

**Lemma 6.** For any integer \(a = 4t + 3, t > 2\), \(a\) can be represented as \(a = x + y + z\), where \(x = 4u + 1, u \geq 1, y = 4v + 1, v \geq 1\), \(a, t, u, v\) and \(w\) are integers.

**Proof:** When \(a = 4t + 3, t > 2\), if \(t = 3, q\) is an integer and \(q \geq 1\), then \(a = 12q + 3\), thus \(a = (4q + 1) + (4q + 1) + (4q + 1)\). If \(t = 3q + 1, q \geq 1\), then \(a = 12q + 7\), thus \(a = 4q + 1\). If \(t = 3q + 2, q \geq 1\), then \(a = 12q + 11\), thus \(a = 4q + 1\).

**Lemma 7.** For any integer \(a = 4t, t \geq 2\), \(a\) can be represented as \(a = x + y\), where \(x = 4u, u \geq 1\) and \(y = 4v, v \geq 1\), \(a, t, u, v\) are integers.

**Proof:** When \(a = 4t, t \geq 2\), if \(t = 2q, q\) is an integer and \(q \geq 1\), then \(a = 8q\), thus \(a = 4q + 4q\). If \(t = 2q + 1, q \geq 1\), then \(a = 8q + 4\), thus \(a = 4q + 4q + 1\).

**Algorithm 5** The algorithm to determine the number of channels in all the spectrum segments

**Input:**
- The total number of channels in the whole spectrum band: \(M\); the threshold of the minimum number of channels in each spectrum segment: \(\theta\)

**Output:**
- The SSL: \(l\)

1. Initialize \(l\) as an empty list:
2. Initialize \(Q\) as an empty queue;
3. \(Q.push(M)\);
4. While \(Q\) is not empty do
5. Let \(\lambda\) be the element in \(Q\)’s head;
6. Pop \(\lambda\) out of \(Q\);
7. Use Lemma 5 to Lemma 8 to represent \(\lambda\) using a set of numbers so that \(\lambda\) is the sum of these numbers; Let the set be \(A\);
8. If \(\forall \beta \in A, \beta \geq \theta\) then
9. Push all the elements in \(A\) into \(Q\);
10. Else
11. Add \(\lambda\) to \(l\);
12. End if
13. End while
14. Return \(l\);

Based on Lemma 5 to Lemma 8 and Algorithm 5, each SU can use the same scheme to split the whole spectrum which guarantees the existence of our designed channel hopping sequence in each spectrum segment. An example of our spectrum splitting scheme is that when \(M = 100\) and \(\theta = 20\), the SSL is \([28, 24, 24, 24]\) and each element in this SSL can guarantee the existence of our rendezvous sequence according to Lemma 3.

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed framework in terms of \(ETTR\) of rendezvous, the average normalized throughput, and the effect of \(\theta\) under the symmetrical and asymmetrical models.

A. The ETTR of Rendezvous under the Symmetrical Model

In order to evaluate our proposed rendezvous algorithm under the symmetrical model, we compare our proposed schemes with an existing CRN rendezvous algorithm called jump-stay [6] with guaranteed rendezvous. We consider a scenario that there are two SUs: one is the SU sender and the other is the SU receiver. The SU sender tries to rendezvous with the SU receiver by using our proposed scheme and the jump-stay scheme.

Fig. 4(a) shows the \(ETTR\) of our proposed scheme and the jump-stay scheme when the SU sender tries to rendezvous with a SU receiver whose HS has 10 to 50 channels. In this simulation, we assume that the ratio of the number of available
channels for the SU pair to the total number of channels in the SU receiver’s HS is 0.8. From the figure we can see that the average time to rendezvous under both schemes increases with the total number of channels. This is consistent with our conclusion in Theorem 1 and the results in [6]. The figure shows that the ETTR of our proposed rendezvous algorithm is always lower than that of the jump-stay scheme when a single SS of a wide-band spectrum is considered.

Fig. 4(b) shows the ETTR of our proposed scheme and the jump-stay scheme in a wide-band scenario with the total number of channels changing from 50 to 300. In this simulation, we assume that the ratio of the number of available channels for the SU pair to the total number of channels in the wide-band spectrum is 0.8 and \( \theta = 20 \) in the spectrum splitting scheme. The figure shows that our proposed rendezvous algorithm can always achieve a much lower ETTR as compared to the jump-stay scheme in a wide-band scenario. This is because that the channel hopping of the jump-stay scheme includes all the channels of the whole spectrum, while our scheme can perform rendezvous in a smaller SS because of the proposed spectrum splitting scheme. Furthermore, due to the spectrum splitting without changing \( \theta \), the performance of our scheme almost keeps the same even when the total number of channels increases.

![Graph showing ETTR comparison](image)

(a) The ETTR within a SS under the symmetrical model  
(b) The ETTR of the whole spectrum under the symmetrical model

Fig. 4. The ETTR under the symmetrical model.

B. The ETTR of Rendezvous under the Asymmetrical Model

We compare our proposed rendezvous algorithm with the jump-stay scheme [6] and enhanced jump-stay scheme [22] under the asymmetrical model which is more practical (the enhanced jump-stay scheme only considers the asymmetrical model). Fig. 5(a) shows the ETTR of our proposed scheme, the jump-stay scheme, and the enhanced jump-stay scheme, when the SU sender tries to rendezvous with the SU receiver whose HS has 10 to 50 channels. In this simulation, we assume that the ratio of the number of available channels for the SU pair to the total number of channels in the SU receiver’s HS is 0.5, the SU pair has the same number of available channels, and 50% of their available channels are the same. The figure shows that the performance of our proposed rendezvous algorithm is better than the jump-stay scheme and enhanced jump-stay scheme under the asymmetrical model within a single SS.

Fig. 5(b) shows the ETTR of our proposed scheme, the jump-stay scheme, and the enhanced jump-stay scheme in a wide-band scenario with the total number of channels changing from 50 to 300. In this simulation, we assume that \( \theta = 20 \) in the spectrum splitting scheme, the ratio of the number of available channels for the SU pair to the total number of channels in the whole-band spectrum is 0.5, the SU pair has the same number of available channels, and 80% of their available channels are the same. This figure shows that the performance of our proposed rendezvous algorithm is always much better than the jump-stay scheme and the enhanced jump-stay scheme in a wide-band scenario.

![Graph showing ETTR comparison](image)

(a) The ETTR within a SS under the asymmetrical model  
(b) The ETTR of the whole spectrum under the asymmetrical model

Fig. 5. The ETTR under the asymmetrical model.

Fig. 6 shows the ETTR of our proposed scheme, the jump-stay scheme, and the enhanced jump-stay scheme within a SS with a total of 40 channels when the ratio (defined as the common available ratio) of the number of common available channels between two SUs to the number of their individual total available channels (assume each SU has the same number of available channels) changes from 0.1 to 0.9. In this simulation, we assume that the ratio of the number of available channels for the SU pair to the total number of channels in the SS is 0.5. This figure shows that the performance of our proposed rendezvous algorithm is always much better than the jump-stay scheme. Our scheme performs better than the enhanced jump-stay scheme when the common available ratio changes from 0.1 to 0.5 and performs almost the same as the enhanced jump-stay scheme when the common available ratio changes from 0.6 to 0.9, which means that our scheme has an advantage when the available channel sets of the two SUs are more distinct.

![Graph showing ETTR comparison](image)

Fig. 6. The ETTR under different common available ratio.

C. The Average Normalized Throughput

We consider practical communication scenarios in a CRN with multiple SU pairs and traffic flows. We evaluate the average normalized throughput for each user in a CRN which
is the average ratio of the number of data packets successfully transmitted out by a SU sender to the total number of data packets generated in a period of time. Since there are no other existing similar rendezvous schemes designed for wide-band CRNs considering the practical communication scenario we consider here, we apply the jump-stay rendezvous algorithm in a wide-band CRN under both the symmetrical and asymmetrical models. We assume that there are 30 SUs in the network: 15 are the SU senders and 15 are the SU receivers. The average packet arrival rate for each SU sender is 5 packets/sec. The length of each packet in the unit of time slots is a random number in [1, 20]. We assume that $\theta = 20$, the total number of channels in the wide-band CRN changes from 50 to 300, and the ratio of the number of available channels for the SU pair to the total number of channels in the whole wide-band spectrum is 0.8.

Fig. 7(a) shows the average normalized throughput of our proposed scheme and the jump-stay scheme under the symmetrical model when the total number of channels within a wide-band spectrum changes from 50 to 300. The result shows that our proposed scheme can maintain a very high average normalized throughput which is much better than the jump-stay scheme when considering the whole wide-band spectrum. This is because that after splitting the spectrum, the time to rendezvous of each transmission pair is much shorter. Thus, the overall throughput is benefited.

Fig. 7(b) shows the average normalized throughput of our proposed scheme and the jump-stay scheme under the symmetrical model when the SU packet arrival rate changes from 1 to 20. We consider a wide-band spectrum with 200 channels here. The result shows that our proposed scheme can maintain a very high average normalized throughput which is much better than the jump-stay scheme under different SU packet arrival rates. We can also notice that the average normalized throughput of the jump-stay scheme decreases significantly when the SU packet arrival rate increases, while our scheme can still keep a high throughput.

while our scheme can still keep a high normalized throughput because of the proposed rendezvous and spectrum splitting schemes.

Fig. 8(b) shows the average normalized throughput of our proposed scheme, the jump-stay scheme, and the enhanced jump-stay scheme under the asymmetrical model when the SU packet arrival rate changes from 1 to 20. We consider a wide-band spectrum with 200 channels. The result shows that our proposed scheme performs much better than both the jump-stay scheme and enhanced jump-stay scheme when the SU packet arrival rate changes. Therefore, our proposed scheme can adapt to different SU traffic under the asymmetrical model.

D. The Effect of the Value of $\theta$

Our spectrum splitting scheme is based on a predetermined value $\theta$. Therefore, how to determine an appropriate $\theta$ is crucial for our proposed communication framework. In this subsection, we show the effect of $\theta$ on the average normalized throughput and the average collision ratio which can help us to determine $\theta$ under both the symmetrical and asymmetrical models. The average collision ratio is defined as the ratio of the number of successful rendezvous with collisions to the total number of rendezvous. The parameters in the following simulation are the same as the ones in the last subsection, except the total number of channels and $\theta$. We set the total number of channels to be 100 and change the value of $\theta$ from 10 to 50.

Fig. 9(a) shows the effect of $\theta$ under the symmetrical model on the average normalized throughput. When $\theta$ increases, the number of channels in each SS increases. This may result in a longer time to rendezvous in our proposed scheme. Thus, the average normalized throughput decreases. Fig. 9(b) shows the effect of $\theta$ on the average collision ratio under the symmetrical model. When $\theta$ increases, the average collision ratio decreases. This is reasonable since when $\theta$ increases, a SU pair can rendezvous on more channels in an SS. Thus, the probability that multiple SUs rendezvous on the same channel is reduced. By observing Fig. 9(a) and Fig. 9(b) together, we can notice that when the collision ratio increases, the average normalized throughput increases also. This indicates that the rendezvous collisions under the symmetrical model may not affect the throughput.

Fig. 10(a) shows the effect of $\theta$ under the asymmetrical model on the average normalized throughput. When $\theta$ increases, the average normalized throughput first increases and then decreases. The reason is that under the asymmetrical
The effect of \( \theta \) under the symmetrical model.

model, each SU will use all the channels in a SS to design a channel hopping sequence. When the number of channels in a SS is small, the \( ETRR \) decreases, but the probability of a collision between the communicating pairs in the same SS increases, which may decrease the average normalized throughput. On the other hand, when the number of channels in a SS increases, the \( ETRR \) increases, which can also decrease the average normalized throughput. Therefore, there exists an optimal \( \theta \) for the asymmetrical model to maximize the average normalized throughput. Under this particular scenario, we can set \( \theta \) to be 35 or 40 to get the maximum average normalized throughput. Fig. 10(b) shows the effect of \( \theta \) on the average collision ratio under the asymmetrical model. When \( \theta \) increases, the collision ratio decreases. By observing Fig. 10(a) and Fig. 10(b) together, we can notice that when the collision ratio decreases, the throughput fluctuates. This indicates that the rendezvous collision under the asymmetrical model is a factor which can affect the throughput.

Based on the above analysis, when the system performance requirements are given, an appropriate \( \theta \) can be predetermined.

VII. CONCLUSION

In this paper, we proposed a novel communication framework for wide-band spectrum CRNs. To the best of our knowledge, this is the first work which addresses practical communication issues in wide-band spectrum CRNs. Our proposed framework includes novel fast rendezvous schemes, intelligent spectrum splitting algorithms, and two practical rendezvous algorithms to improve SUs’ communications. Simulations results show that our proposed framework can make two SUs rendezvous on a common available channel efficiently and achieve a very high average normalized throughput in a wide-band CRN.

REFERENCES

[1] FCC, “Notice of proposed rule making and order,” ET Docket No 03-